

**ФЕДУНЕЦЬ А.Д.**

кандидат технічних наук, доцент,
доктор філософії в технічних науках,
завідувач кафедри інформатики, прикладної
математики та фізики Кіровоградського
національного технічного університету

*Світлій пам'яті вчителя, колеги, успішного вченого і просто чудової людини
присвячується...*

ЕКОНОМІЧНА ІНФОРМАТИКА**ПІДРУЧНИК****ДЛЯ СТУДЕНТІВ ЕКОНОМІЧНИХ СПЕЦІАЛЬНОСТЕЙ**

**Кропивницький
ЦНТУ
2021**

УДК: 681.3(004.9)
ББК 32.08
Ф47

Федунець А.Д., Рибаківа Л.В., Пархоменко Ю.М., Кислун О.А.
**Економічна інформатика: Підручник. – 2-е видання, виправлене
та доповнене - Кропивницький : ЦНТУ, 2021. – 210 с.**
ISBN

У підручнику систематизовано викладено основи економічної інформатики, включаючи інформаційні технології, поняття апаратного та програмного забезпечення засобів автоматизованої обробки електронної документації, основи офісного програмування. Висвітлено призначення, основні властивості і особливості побудови сучасних експертних навчальних систем та перспективи розвитку інформаційних технологій.

Для здобувачів вищої освіти очної та заочної форми навчання.

Навчальне електронне видання комбінованого використання.
Можна використовувати в локальному та мережному режимах

УДК:
681.3(004.9)
Ф47

Рецензенти:

Осадчий Сергій Іванович, д-р техн. наук, професор, завідувач кафедри
Автоматизації виробничих процесів ЦНТУ.

Віхрова Лариса Григорівна, канд. техн. наук, професор, декан факультету
Автоматики та енергетики ЦНТУ

*Рекомендовано Вченою радою Центральноукраїнського національного технічного
університету до друку та використанню підручника у навчальному процесі здобувачами
вищої освіти очної та заочної форми навчання, протокол № 8 від 29 березня 2021 року*

Розповсюдження і тиражування без дозволу авторів заборонено

ISBN

© Федунець А.Д., Рибаківа Л.В.,
Пархоменко Ю.М., Кислун О.А., 2021
© Видавець Лисенко В.Ф.

З М І С Т

1. Предмет, мета, завдання економічної інформатики	6
2. Теоретичні основи економічної інформатики	8
Інформатика.....	8
Інформація, економічна інформація.....	9
Інформаційна система.....	17
Організація системної обробки економічної інформації.....	23
Сутність системного підходу до обробки економічної інформації і використання нових інформаційних технологій.....	26
Методичні основи автоматизації інформаційних процесів.....	29
Інформатика та вдосконалення форм і методів управління виробництвом..	32
Інформатика та удосконалення форм і методів обліку.....	37
Інформатика та удосконалення форм і методів контролю і аналізу.....	38
Інформатика та удосконалення методів складання зведеної звітності.....	39
Інформатика та удосконалення форм і методів управління.....	39
Методичні основи автоматизації процесів збирання і обробки економічної інформації.....	40
Методичні основи організації економічного аналізу.....	43
3. Системне забезпечення економічних процесів	46
Технічне забезпечення обробки інформації про економічні процеси.....	45
Поняття базового технічного комплексу комп'ютерної техніки.....	47
Програмне забезпечення обробки інформації про економічні процеси.....	55
Операційна система Windows.....	57
Операційна система MacOS.....	58
Unix-подібні операційні системи.....	58
Прикладне програмне забезпечення.....	60
4. Інформаційні технології обробки економічної інформації, мережні технології.....	65
5. Застосування Інтернет в економіці.....	76
Доступ користувачів до мережі Internet.....	79
Система адрес у мережі Internet	80
Телеконференція (сервіс Usenet news).....	81
Інтерактивне спілкування.....	82
Internet-економіка.....	82
Internet- трейдинг	84
6. Організація комп'ютерної безпеки та захисту інформації.....	86
Види комп'ютерних злочинів, суб'єкти їх вчинення та міри запобігання...	86
Сучасні моделі та методи організації безпеки комп'ютерів у мережах.....	88
Програмний захисту інформації від комп'ютерних вірусів.....	92
Основні заходи для захисту від вірусів.....	94
Віруси й Інтернет– правила роботи у мережах.....	94
7. Основи WEB-дизайну.....	97

Основні поняття	97
Основи програмування Web-документів	97
Приклад створення власної Web-сторінки	103
Створення Web-стрінки з використанням програми FrontPage.....	104
Критерії оцінки ефективності Web-сторінок та Web-сайтів.....	117
Побудова ефективних сайтів для деяких типових класів.....	120
Критична оцінка методик покращання сайту.....	120
Рекомендації.....	121
8. Програмні засоби роботи зі структурованою інформацією.....	123
Основні вимоги до систем електронного документообігу.....	124
Класифікація програмних засобів роботи з електронною документацією ...	125
Загальні поняття про технологію роботи з електронною документацією ...	125
Користування системами управління електронними документами.....	127
Автоматизована система документообігу на основі Lotus Notes.....	129
Головні автоматизовані функції систем автоматизованого документообігу	131
9. Програмні засоби роботи з базами та сховищами даних.....	133
Основні об'єкти СУБД Access , їх властивості, способи формування та обробки	133
Технологія створення бази даних і таблиць.....	136
Заповнення таблиць даними.....	136
Робота з таблицями.....	136
Створення зв'язків між таблицями.....	137
Створення запитів на вибірку даних.....	137
Відкриття вікна нового запиту.....	137
Вибір даних з однієї таблиці.....	137
Встановлення властивостей полів.....	139
Введення умов відбору.....	139
Умови відбору для дат і часу.....	139
Використання параметрів запиту.....	140
Багатотабличні запити.....	141
Створення запиту на основі іншого запиту.....	142
Використання майстра запитів.....	142
Відкриття, копіювання, збереження, перейменування і видалення запитів.	142
Виконання розрахунків в запитах.....	143
Використання побудови виразів.....	143
Підсумкові запити.....	145
Перехресні запити.....	145
Обмеження на використання запитів на вибірку для поновлення даних....	146
Створення перехресних запитів за допомогою майстра.....	146
Настроювання властивостей запиту.....	146
Запити на зміну даних.....	148
Запит на поновлення.....	148
Створення нової таблиці.....	149
Запит на додавання.....	150
Помилки при виконанні запитів на зміну.....	150
Видалення групи записів.....	151

10. Основи офісного програмування.....	152
Поняття про розробку програм.....	152
Поняття про етапи роботи над програмою.....	153
Поняття про відлагодження програми.....	154
Поняття про тестування програм.....	155
Локалізація помилок.....	156
Поради початківцю.....	156
Оператор привласнення.....	157
Виклики підпрограм.....	158
Послідовність операторів.....	158
Складаний оператор.....	158
Умовний оператор.....	159
Розгалуження.....	159
Цикли.....	160
Розділ оголошень та ініціалізації.....	162
Типи даних та ефективність програми.....	162
Стандартні типи даних.....	164
Складні типи даних	167
Записи	169
Об'єкти.....	169
Файли.....	169
Типи файлів.....	171
Adobe Photoshop.....	173
Flash-технології.....	176
GIF-анімація	178
Java.....	181
11. Експертні навчальні системи.....	194
Призначення та основні властивості експертних систем.....	194
Склад і взаємодія учасників побудови й експлуатації експертних систем ..	195
Переваги використання експертних систем.....	196
Особливості побудови й організації експертних систем.....	196
Основні режими роботи експертних систем.....	197
Відмінність експертних систем від традиційних програм.....	198
Технологія розробки експертних систем.....	198
Знання, типи представлення знань в експертних системах.....	200
Переваги та слабкі місця експертних систем.....	201
Сфера застосування та перспективи розвитку.....	202
12. Перспективи розвитку інформаційних технологій.....	203
Перспективи розвитку електронної пам'яті	203
Перспективи розвитку інтелектуального потенціалу.....	206
Розвиток науково-освітнього інформаційного сегменту.....	207
Перспективи розвитку інформаційних технологій в суто економічних застосуваннях.....	208

РОЗДІЛ 1

ПРЕДМЕТ, МЕТА, ЗАВДАННЯ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ

Інформатика (зокрема, економічна інформатика як її органічна складова) охоплює вивчення інформації – її збір, аналіз та обробку, які здійснюються засобами електронних систем. Методи і засоби інформатики надходять до споживача у вигляді інформаційних технологій, під якими розуміють сучасні види інформаційного обслуговування, організовані з використанням спеціалізованої до цього електронної техніки та електронної комунікації (комп'ютерної техніки та комп'ютерних мереж).

Інформатика вивчає те загальне, що властиве всім різновидам конкретних інформаційних технологій, а саме: носіям інформації, каналам зв'язку, інформаційним сигналам, даним, повідомленням, тощо, які описуються такими характеристиками, як надійність, ефективність, інформаційність, надмірність, змістовність.

Всі інформаційні процеси складаються з прийому, кодування, передачі, декодування, збереження, отримання, відображення інформації. Водночас інформатика здійснює переходи від загального до одиничного та навпаки, враховуючи конкретну специфіку загальних закономірностей функціонування інформаційних технологій у тій або іншій сфері людської діяльності, зокрема, у соціально-економічній сфері.

Економічна інформатика вивчає засоби та методи обробки тієї частини інформації, яка відображає суспільні відносини економічного характеру та передбачає використання можливостей пристосованих до такої обробки електронної техніки, зокрема – обчислювальної, комп'ютерної.

Предметом вивчення економічної інформатики як навчальної дисципліни є засоби автоматизації інформаційних процесів з використанням економічних даних.

Мета вивчення економічної інформатики – формування знань про принципи побудови та функціонування комп'ютерної техніки, алгоритмізацію та організацію процесів обробки інформації на персональних комп'ютерах, програмне забезпечення персональних комп'ютерів і комп'ютерних мереж, а також ефективне використання сучасних інформаційно-комунікаційних технологій у професійній діяльності, обраній для фахової підготовки.

Завдання, що передбачають досягнення мети:

- вивчення теоретичних основ інформатики та комп'ютерної техніки,
- вивчення можливостей комп'ютерної техніки та її програмного забезпечення щодо обробки інформації економічного змісту,
- набуття навичок ефективного використання можливостей систем оброблення економічних даних та систем програмування для персональних комп'ютерів і локальних комп'ютерних мереж,
- дослідження соціально-економічних систем та розв'язування завдань фахового спрямування за допомогою засобів автоматизованої обробки економічної інформації.

Зміст дисципліни розкривається в темах:

1. Предмет, методи і завдання дисципліни.
2. Теоретичні основи економічної інформатики.
3. Системне забезпечення інформаційних процесів.
4. Мережні технології.
5. Застосування Інтернету в економіці.

6. Організація комп'ютерної безпеки та захисту інформації.
7. Основи ВЕБ-дизайну.
8. Програмні засоби роботи зі структурованою інформацією.
9. Програмні засоби роботи з базами та сховищами даних.
10. Основи офісного програмування.
11. Експертні і навчальні системи.
12. Перспективи розвитку інформаційних технологій

Світова економічна система вже вступила в епоху формування світового інформаційного суспільства. Це новонароджене суспільство характеризується не тільки якісно новим виробничим апаратом, який базується на комп'ютеризованих засобах праці та інформаційних технологіях, але й новими соціальними відносинами. Впровадженням цих відносин потрібно управляти. Особливою мірою це стосується процесів інформатизації. При проведенні широкої інформатизації насамперед потрібно зосереджуватися на вирішенні низки завдань:

- постійне вивчення та формування сфери застосування інформаційних продуктів та послуг;
- орієнтація основних видів діяльності на застосування передових технологій, що ґрунтуються на впровадженні інформаційних систем;
- підготовка, введення, корегування правових та господарських норм, які забезпечують функціонування інформації як товару з врахуванням загальноприйнятих правил;
- формування та введення базових стандартів, які регламентують форму представлення, способи обробки та пересилання даних інформації (протоколи обміну, інтерфейси, тощо) з врахуванням міжнародних стандартів;
- забезпечення достатнього рівня комп'ютерної грамотності та інформаційної культури серед персоналу та заохочення до цього потенційних користувачів;
- створення основних компонентів інфраструктури інформатизації організації;
- орієнтація потреб всіх сфер соціально-економічного розвитку організації на використання інформаційних технологій;
- досягнення достатньої повної взаємодії із національною інформаційною інфраструктурою, а також вихід в глобальну комп'ютерну мережу для користування нагромадженими базами даних і знань;
- здійснення масштабного застосування інтегрованих систем обробки інформації;
- постійне відслідковування зростаючих можливостей використання систем масового інформаційного обслуговування, зокрема через Інтернет;
- створення конкурентоспроможного виробництва інтелектуалізованих продуктів та інформаційних послуг;
- підтримка фундаментальних досліджень в перспективних галузевих напрямках з орієнтацією на використання найновіших досягнень в галузі інформаційних технологій.

Досягнення мети вивчення економічної інформатики повинно передбачати таку тактику, пристосовану до впровадження інформаційних технологій, яка в жодному разі не повинна бути випадково організована, а опиратися на чітку довготермінову стратегію соціально-економічного розвитку, яка стосується окремої людини, трудових колективів підприємств, організацій, галузей, народів країн світу, світової спільноти в цілому.

РОЗДІЛ 2

ТЕОРЕТИЧНІ ОСНОВИ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ

Інформатика

Людина впродовж свого життя має справу з обробкою інформації. Це стосується управління різноманітними об'єктами, виконання функційних обов'язків, вибору конкретних дій. Якість її життєдіяльності визначається не лише успадкованими генетичними даними, але й тим, наскільки успішним сталося пізнання нею законів і методів обробки інформації та їх ефективного використання. Останнє є предметом науки, яку називають інформатикою.

Інформатика - наука, що досліджує закони і методи переробки та накопичення інформації.

Ускладнення сучасного промислового виробництва, необхідність оперативного зв'язку між різноманітними підприємствами, потреба в опрацюванні за короткий час величезної кількості інформації поставили на порядок денний питання автоматизації (виконання технологічного процесу переважно чи повністю за допомогою техніки) багатьох функцій управління різними галузями виробництва, науки, освіти. Об'єктивна можливість такого управління з'явилось завдяки досягненням математики, кібернетики, теорії інформації, розвитку електронно-обчислювальної техніки. Завдяки останній відбувається інтенсивний процес математизації та інформатизації не лише природних і технічних, але й суспільних наук. Математичні моделі та інформація, на основі якої вони будуються й на основі якої здійснюється математичне моделювання, широко використовуються в хімії, біології, медицині, психології, лінгвістиці. Економіка – одна із прерогативних сфер їх ефективного застосування, де людино-машинні технології обробки інформації знайшли чи не найбільш широке застосування.

Основу людино-машинної технології обробки інформації становить мережа сполучених між собою каналами автоматичного зв'язку обчислювальних центрів різних рівнів (сюди входить розгалужена мережа терміналів - автоматизованих робочих місць персоналу). У таких системах потік паперової документації трансформується в кругообіг машинно-орієнтованої електронної інформації.

У пам'яті обчислювальних машин, включених у мережу, функціонує постійно обновлювана інформаційна база, яка вичерпно характеризує всі рівні економіки. За допомогою системи програм, які керують роботою таких машин, розв'язуються практично всі або більшість задач економічного і соціального розвитку. Завдячуючи такому стану розвитку обчислювальної техніки створено можливість не просто робити моментальні “знімки” стану народного господарства в даний момент, але й регулювати поведінку економіки як “живої” кібернетичної системи.

Задача інформатики – виявити та дослідити закономірності формування, накопичення, обробки інформації за допомогою засобів обчислювальної техніки, удосконалити методи, що сприяють підвищенню ефективності інформаційних процесів, що супроводжують соціально-економічний розвиток суспільства.

Інформація, економічна інформація

Під **інформацією** (від лат. information - роз'яснення, виклад) у свій час розуміли відомості, які передаються людьми усним, письмовим або іншим засобом

за допомогою умовних сигналів, технічних засобів тощо. З середини ХХ сторіччя інформація визнана загальнонауковим поняттям, що включає в себе обмін відомостями між людьми, людиною й автоматом, автоматом і автоматом, обмін сигналами у тваринному і рослинному світі, передачу ознак від клітини до клітини, від організму до організму тощо, залежно від сфери, якої стосується його використання.

Інформація – загальнонаукове поняття, що включає в себе: обмін відомостями між людьми, між людиною й автоматом, між автоматом і автоматом; обмін сигналами у тваринному і рослинному світі; передачу ознак від клітини до клітини, від організму до організму.

У залежності від сфери використання, інформація може бути економічною, технічною, генетичною тощо. Під економічною розуміється інформація, що характеризує виробничі відношення в суспільстві.

Інформація є одним із ресурсів, який може накопичуватися, реалізуватися, поновлюватися, і який є придатним для колективного використання та (на відміну від інших ресурсів) у процесі споживання не втрачає своїх якостей. Передусім, це головний елемент будь-якої з функцій управління. Володіння повною, достовірною, актуальною та оперативною інформацією уможливує отримання ринкових переваг, знижує фінансовий ризик, ефективно підтримує прийняття рішень.

Інформацію розглядають у нерозривній єдності з комп'ютерними системами (обчислювальною технікою, укомплектовано додатковими електронними пристроями), які забезпечують її збирання, реєстрацію, зберігання, передавання й перетворення. За допомогою комп'ютерів усю інформацію можна швидко одержати, "відсортувати" у заздалегідь визначеному порядку, що позбавляє необхідності переглядати стоси паперів у пошуках потрібних відомостей. Комп'ютери не створюють інформацію із нічого, але вони здатні надзвичайно швидко сприймати, сортувати, аналізувати та інтерпретувати її за допомогою програмних засобів, розроблених людиною.

По мірі накопичення інформації та зростання швидкості її кругообігу, все більшого значення набувають інформаційні технології, які базуються на застосуванні комп'ютерів, активній участі користувачів (непрофесіоналів у галузі програмування) в інформаційному процесі, високому рівні "дружнього" інтерфейсу користувача, широкому використанні пакетів прикладних програм загального призначення, доступі до віддалених баз даних і програм завдяки обчислювальним мережам ЕОМ.

Широке використання новітніх інформаційних технологій, необхідність роботи з документами, матеріалами, базами даних конкретної організації або установи вдома, у готелі, у транспортних засобах привела до появи так званих віртуальних офісів. Останнім часом мова вже йде про віртуальні підприємства та Internet-економіку, основу якої складає електронний бізнес, побудований на спільних діях бізнес-процесу в особі бізнесмена та комп'ютера або іншого автоматизованого засобу зв'язку з обміну інформацією.

Використання Internet (поряд з цим Intranet та Extranet) спільно з новітніми інформаційними технологіями роблять віртуальний бізнес привабливим і прибутковим.

Структурно віртуальне підприємство складається з територіально роз'єднаних фірм чи співробітників, що обмінюються продуктами своєї праці і спілкуються винятково електронними засобами при мінімальному або цілком відсутньому особистому контакті. Це вже не просто департаменти, відділи, цехи і групи, а сукупність бізнес-процесів, і робота організовується навколо бізнес-

процесів, де не товари, а процеси їх створення приносять компаніям довгостроковий успіх.

Комп'ютери стали основним "знаряддям праці" всіх спеціалістів (також і аналітиків), тому розглядати особливості формування й використання сучасної інформаційної бази економічного аналізу неможливо без урахування цієї обставини.

Цінність і своєчасність управлінського рішення значною мірою залежить від здатності в потрібний момент зібрати, проаналізувати та проінтерпретувати інформацію. Спеціалісти кажуть, що рецепт для ефективного рішення є тільки один: 90 % інформації і 10% натхнення. Прийняття управлінських рішень пов'язане з постійним перетворенням інформації, а сам процес управління має інформаційний характер. На кожній зі стадій управління використовується конкретна вхідна інформація й одночасно формується результатна вихідна інформація, яка є вхідною на інших стадіях управління. Стадії управління повторюються, утворюючи замкнутий контур. Широкий доступ користувачів до інформації на всіх стадіях управління можливий завдяки сучасним інформаційним технологіям, а також організації баз і банків даних, які забезпечують прямий і зворотний обмін інформацією.

Як було відмічено раніше, один із видів інформації – економічна інформація. Економічну інформацію поділяють на кілька типів, які є суттєвими для розробки інформаційних технологій. Це: факти, оцінки, прогнози, узагальнені зв'язки, конфіденційна інформація, чутки.

Факт - інформація про подію або умову, що їх можна безпосередньо спостерігати та аналізувати (наприклад, виробництво, продаж). Фактична інформація, яка використовується для аналізу, є дуже різноманітною і великою за обсягом, формується з різних джерел, потребує накопичення в базі даних і дальшої обробки, в основній масі циклічно повторюється.

Оцінка – кількісна чи якісна міра щодо значення тієї чи іншої характеристики певного об'єкта у прийнятій шкалі. Оцінка ґрунтується на висновках, зроблених щойно чи колись у минулому. Вона не завжди є достовірною і може мати різні похибки, пов'язані із використанням певних методів вимірювання і розрахунків, похибками у вибірці, глибиною професійних знань спеціаліста, який вибрав методу розрахунку та дав оцінку. За результатами оцінки формується база даних для прогнозування майбутнього, особливо коли використовуються бази знань та експертні системи.

Прогнози – випереджаюча інформація про розвиток певних процесів у майбутньому. Прогнози частково ґрунтуються на аналогії і частково на тому, що заведено називати "здоровим глуздом". Для прогнозування використовуються різні методи й моделі розрахунків (кореляційний і регресивний аналіз, екстраполяція тенденцій), знання експертів і спеціалістів у конкретній сфері. Основою для оцінки та прогнозування є узагальнені зв'язки, які характеризують рівень залежності досліджуваного показника від одного чи кількох інших показників, як, наприклад, обсяг продажу і план витрат ресурсів в підприємства.

Конфіденційна інформація – інформація, що не підлягає розголошенню (як і джерело її отримання). До неї відносять відомості про плани конкурентів, про науково-технічні досягнення, можливі зміни в законодавстві країн стосовно виробництва, торгівлі, податків тощо.

В аналітичних розрахунках використовуються фактичні та гіпотетичні дані, але певне значення надається й чуткам. Хоч чутки є не дуже надійним джерелом інформації, проте коли їх опубліковано в пресі, оприлюднено в публічних виступах

і заявах, вони можуть тимчасово спричинити певні зміни на ринку (навіть якщо зміст чуток не відповідає дійсності).

Аналітична інформація подається в табличній та графічній формах, у вигляді текстів і динамічних числових рядів.

Найбільш поширеною формою подання первинної та результатної інформації є таблична. Таку форму мають первинні документи (рядки в документах і реквізити, які можна розглядати як графи), вихідні документи (звіти, відомості, розрахунки, таблиці), дані на машинних носіях у реляційних базах даних. Обробка табличної інформації забезпечується засобами електронних таблиць (Excel, Quattro Pro, Lotus 1-2-3) та організацією реляційних баз даних з використанням систем управління базами даних (СУБД): FoxPro, Access, Informix, Oracle. У реляційній базі даних будь-який документ чи лінійний файл даних подається у вигляді набору двовимірних плоских таблиць, які складаються зі стовпчиків (граф) і рядків. Необхідні аналітичні задачі розв'язуються за допомогою спеціального програмного забезпечення.

Чільне місце у відображенні результатів аналітичних розрахунків належить графічній формі, яка дає наочне й концентроване уявлення про динаміку процесів, функціональні залежності між двома і більше факторами економічного аналізу, сприяє виявленню тенденцій і закономірностей. Графіки різних видів будуються за допомогою електронних таблиць, окремих текстових процесорів або спеціального програмного забезпечення (StatGraphs, Graphics).

Текстова інформація - найменш формалізована форма відображення інформації. Переважну більшість такої інформації взагалі не фіксовано на машинних носіях, певна частина її перебуває в базах даних, певна – в іншій діловій документації. Для її формування в електронному вигляді використовуються текстові процесори Windows Word, Лексикон та ін.

Використання Web-технологій забезпечує організацію, ведення і даліше використання баз даних у вигляді гіпертексту. Гіпертекстова технологія дає змогу користувачеві переміщуватися від одних об'єктів інформації до інших з урахуванням їхніх змістових і семантичних зв'язків.

Усю сукупність інформації, яка використовується в економічному аналізі й забезпечує управлінську систему та заінтересованих сторонніх користувачів, можна розподілити (з урахуванням головних джерел) на внутрішню та зовнішню групи, які утворюються з планово-облікових та позаоблікових даних.

До зовнішньої інформації відносять: матеріали перевірок податкової служби, пояснювальні й довідкові записки, листування, матеріали цільових обстежень, перевірок, особистих спостережень працівників та аналітиків, спеціальних вибірок, відомості, передані по радіо чи телебаченню, технічну документацію на устаткування, технологію тощо. Зовнішню офіційну бухгалтерську й статистичну звітність уніфіковано відповідними стандартами бухгалтерського, податкового та статистичного обліку і звітності.

Джерела зовнішньої інформації:

- публікації; звіти державних агентств; звіти торговельних асоціацій; наукові публікації; аналітичні журнали; довідники та списки;
- інші підприємства: постачальники; замовники; конкуренти;
- інформаційна індустрія: фірми, які надають інформаційні послуги (консультування), досліджують відповідні проблеми.

Публікації містять різноманітну інформацію з багатьох питань, що стосуються економічного аналізу: економічні умови, виробництво, продаж товарів (послуг). Обсяг і надійність такої інформації є різними в різних галузях і країнах. Офіційною відкритою звітною інформацією вважають бухгалтерський баланс підприємства (річний, квартальний), звіти про фінансові результати (річний, квартальний), звіт про фінансово-майновий стан підприємства, документи податкової звітності.

Корисну інформацію можна одержати і від підприємств, з якими налагоджено економічні стосунки. Наприклад, якщо дізнатися в постачальника упаковки про кількість упаковки, замовленої конкурентом, то можна визначити прогнозний рівень продажу нового продукту конкуруючою фірмою. Корисну інформацію регулярно можна отримувати від рекламних агентств як послуги рекламодавцю та від засобів масової інформації, котрі надають її, бажаючи продати "час і місце".

Надзвичайно корисною інформацією про товар, якість упаковки або обслуговування, а також про можливі дії конкурентів володіють покупці, особливо дилери та оптовики. Хоч як не дивно, але й від персоналу конкуруючих фірм, не зважаючи на суворі обмеження щодо розголошення окремих видів інформації, часом теж можна отримати певні корисні відомості, не наражаючись на судовий процес.

Фірми, які спеціалізуються на збиранні та аналізі інформації, або пропонують стандартизовані інформаційні продукти, або виконують конкретне завдання замовника. Складність використання зовнішньої інформації в аналітичних розрахунках в Україні полягає в тому, що в нас лише формуються такі фірми. Основними споживачами результатів обробки зовнішньої аналітичної інформації є контрольні органи, банківські установи, фондові та товарні біржі, інші ділові партнери підприємства, у тому числі потенційні інвестори, кредитори.

Найбільша частина інформації надходить із внутрішніх джерел: дані оперативного й бухгалтерського обліку, планування тощо. Збирання інформації відбувається регулярно (у строки бухгалтерської та статистичної звітності), або епізодично (у разі потреби). Збирання епізодичної інформації потребує розробки спеціальних процедур проведення обстежень, одержання індивідуальних і групових оцінок.

До джерел внутрішніх облікових даних належать: бухгалтерський облік і звітність; статистичний облік і звітність; оперативний облік і звітність. Дані бухгалтерського обліку дають об'єктивну кількісну характеристику різноманітних господарських операцій, узагальнену характеристику всієї сукупності засобів господарства за складом і розміщенням, за джерелами утворення і цільовим призначенням. Для цього використовуються методи наскрізного і безперервного спостереження, суворе документування, систематизація на рахунках, групування в баланс і інших звітних таблицях.

Крім даних бухгалтерського обліку, для оцінки виконання планів (щомісячних, квартальних, річних) використовуються й статистичні дані, завдяки яким виявляються певні економічні закономірності.

Поточний аналіз використовує відомості оперативного обліку і звітів, забезпечуючи швидке отримання відповідної інформації. Результатна інформація подається у вигляді даних про недоліки в роботі та відповідних винуватців або про невикористані резерви та способи їх мобілізації. На відміну від поточного аналізу, оперативний використовує щоденну інформацію про діяльність підприємства, дані первинних бухгалтерських та інших документів, матеріалів контролю, спостереження, нарядів тощо.

Вибіркові облікові дані необхідні для деталізації показників звітності. Використовуються епізодичні вибірки, спостереження, поглиблені перевірки. Джерелом вибірових даних є поточний бухгалтерський облік і первинна документація.

До позаоблікових даних відносять матеріали зборів трудових колективів, бесід з робітниками і службовцями підприємств тощо. Особливо багато інформації можна одержати з питань організації праці й виробництва, фінансового стану. На практиці керівники різних рівнів виконують власний оперативний аналіз, ведуть реєстрацію найважливіших процесів і операцій, приблизні розрахунки, планують відповідні заходи. У такий спосіб кожна особа, яка відповідає за прийняття рішень, одночасно стає і користувачем (інформантом), і джерелом інформації (інформатором).

У проектуванні інформаційної бази економічного аналізу враховують основні характеристики економічної інформації, які впливають на вибір інформаційних технологій розв'язування задач. Це можна зробити за допомогою її класифікації за певними ознаками.

За стадіями управління розрізняють прогнозу, планову, облікову, нормативну інформацію та інформацію для аналізу господарської діяльності, оперативного управління.

Прогнозну інформацію пов'язано з функцією прогнозування, планову - з плануванням (стратегічним, техніко-економічним, оперативно-виробничим); облікову - з управлінським, фінансовим обліком; інформацію аналізу господарської діяльності - з функцією економічного аналізу; оперативного управління й регулювання - з відповідними функціями.

Нормативна інформація виникає і використовується на стадіях технічної підготовки виробництва, а також в інших випадках, наприклад, для формування цін, тарифікації. Вона містить норми й нормативи, ціни, розцінки, тарифи, а також деякі інші дані, наприклад, заздалегідь обумовлені табличні величини (ставки прибуткового податку з громадян).

Нормативну інформацію можна умовно поділити на нормативно-правову та нормативно-довідкову. До нормативно-правової належать розпорядження органів законодавчої та виконавчої влади (закони, акти), які регулюють економічну діяльність підприємства.

Нормативно-довідкова інформація - це норми та нормативи, які визначаються керівництвом підприємства (наприклад, норми витрат матеріалів). До довідкової відносять інформацію, однакову для всіх функціональних різновидів: назва підприємства, міністерства, відомства, штатний розпис, список постачальників і покупців тощо.

Нормативна й довідкова інформація створюють фонд нормативно-довідкової інформації, призначений для розв'язання різноманітних управлінських завдань.

За місцем утворення інформація для економічного аналізу поділяється на внутрішню й зовнішню. Внутрішня інформація - це сукупність даних, що виникають на самому об'єкті і характеризують його діяльність. Вона формується на стадії конструкторсько-технологічної підготовки виробництва, у поточному виробництві товарів і їх збуті, оперативному, бухгалтерському та статистичному обліку. Частина внутрішньої інформації фіксовано на машинних носіях, частину - тільки на паперових документах. Зовнішня інформація виникає за межами об'єкта і має безпосереднє відношення до досліджуваної предметної області.

За стабільністю аналітичну інформацію поділяють на постійну (сталу), умовно-постійну та змінну. Постійна інформація не змінює своїх значень (наприклад, звітні дані); умовно-постійна зберігає їх протягом тривалого періоду (наприклад, нормативи, норми), а змінна характеризується частою зміною своїх значень (наприклад, відомості про нарахування заробітної плати). Період стабільності має конкретний характер для певних задач, управлінських робіт. Його визначають за певний проміжок часу (наприклад, за місяць).

За стадіями створення економічна інформація поділяється на вхідну та вихідну. До вхідної інформації відносять дані, що необхідні для розв'язання аналітичних задач (наприклад, наявність основних засобів, кількість і склад працівників). Вхідна інформація реєструється в місці її збирання чи виникнення і вводиться в ПЕОМ без попередньої обробки. Вхідна первинна інформація є найбільш детальною і становить основу для наступної логічної та арифметичної обробки даних. До вхідної інформації може належати не лише змінна, а й умовно-постійна та постійна інформація за особливо великої ролі умовно-постійної.

Вихідна інформація є підсумком обробки вхідних даних, але вона містить поряд з результатною інформацією і певні первинні дані.

Спеціального значення набуває проміжна інформація, яка потрібна для розв'язування тих самих задач у наступних періодах. Результатні дані в багатьох випадках архівуються й накопичуються у базі даних для розв'язування інших взаємозв'язаних задач, для вивчення динаміки показників чи процесів.

За насиченістю реквізитами аналітична інформація буває: достатньою, недостатньою і надлишковою. Достатня - це цілком конкретна за змістом мінімальна інформація, яка необхідна для розв'язання певної задачі. Недостатність інформації робить розв'язання задачі неможливим; надлишкова містить зайві дані, які або зовсім не використовуються, або виконують контрольню-дублюючі функції. Для сучасної економічної інформації характерним є поєднання надмірності й недостатності даних. Це пов'язане з традиційними методами обробки даних, що призводять до дублювання інформації, із наявністю застарілих даних, браком окремих показників, необхідних для аналізу, у звітності, статистиці, поточних документах.

За способом фіксації розрізняють усну й документовану інформацію, зафіксовану на паперових і машинних носіях. Запис даних на машинних носіях - необхідна умова наступної автоматизованої обробки інформації. Запис - це найбільш трудомісткий процес, на який припадає до 95 % усіх помилок. Треба брати до уваги також і те, що на сучасних підприємствах значна частина внутрішньої інформації є фіксованою, записаною в базі даних, а другу її частину відображено в традиційних документах, звітах і за необхідності вона додатково формується в базі даних. Зовнішня інформація є переважно нефіксованою і зберігається в друкованому вигляді. З розвитком інформаційних служб і засобів телекомунікації значну частину зовнішньої інформації також можна буде вводити в базу даних підприємства з використанням машинних носіїв.

За елементами структури інформації розрізняють реквізит, показник, масив, потік. Ця класифікація має значення для розробки баз і банків даних. Відомі й інші схеми класифікації економічної інформації: за формою подання даних - алфавітна, цифрова, алфавітно-цифрова; за стабільністю руху відносно системи керування - періодична (регулярна), неперіодична (епізодична), одноразова; за певними додатковими ознаками - наприклад, інформація про фінансовий стан тощо; за предметом господарської

діяльності - інформація про продукцію, інформація про послуги. Цю класифікацію можна деталізувати аж до найменувань специфікаційної номенклатури товарів.

Найважливішими особливостями економічної інформації є: динамічність, взаємозв'язаний характер, нерівномірність виникнення протягом року, дискретність, єдність і взаємообумовленість показників (як передумова системного використання економічної інформації в процесі машинної обробки).

Специфічними ознаками економічної інформації є її залежність від об'єкта управління; переважання у формі подання даних алфавітно-цифрових знаків; необхідна висока точність результатів обчислень та оформлення їх у вигляді, зручному для сприйняття людиною; значне поширення документів як носіїв вхідних даних та результатів обробки; великі обсяги оброблюваної інформації; можливість одержання значної кількості похідних даних у разі обробки тих самих показників за різною методикою, постійне нагромадження й тривале зберігання.

Економічна інформація досить неоднорідна, вона має складну схему взаємозв'язків окремих її видів. Різновидами економічної інформації є облікова інформація та інформація аналізу господарської діяльності, яким притаманні як спільні властивості, так і низка специфічних особливостей.

Обліково-аналітична інформація є основою для прийняття рішень з організації, планування й регулювання господарської діяльності підприємства. Особливості облікової інформації залежать від видів обліку (оперативний, бухгалтерський, статистичний); форми бухгалтерського обліку (таблично-автоматизована, діалогова, без паперова); від джерел та методів формування змінної та умовно-постійної інформації та алгоритму їх машинної обробки.

Облікові дані відображають фактичний стан виробничо-господарської діяльності підприємства і є інформаційною моделлю виробництва. Крім відображення дійсного стану об'єкта, на облікову інформацію покладається й ретроспективна функція, що потребує тривалого зберігання цієї інформації (у вигляді форм бухгалтерської звітності).

Кінцеві дані є результатом розв'язання облікових задач зі знаходження зведено-підсумкових величин за встановленими групувальними ознаками, для чого реалізуються арифметичні та логічні операції, операції сортування інформації. Якість облікової інформації залежить від використаної методології її сприйняття, систематизації та узагальнення, а також від специфічних особливостей облікових задач:

- розв'язності (завжди розрахунковим або логічним способами);
- алгоритмічності;
- документальності вхідних та вихідних даних;
- використання тих самих даних для формування різних показників;
- інформаційного зв'язку облікових задач;
- необхідності накопичення даних;
- регламентації термінів розв'язання задач.

Аналітична інформація відрізняється від облікової. Дані для аналізу беруть з одного або кількох різновидів економічної інформації (планової, облікової, нормативної, прогнозної). Аналітична інформація нагромаджується, зберігається та використовується відповідно до планів та програм аналітичної роботи на підприємстві згідно з визначеною організаційною формою.

Аналітичним розрахункам властива наявність значного обсягу логічних операцій, поєднаних з арифметичними за складними алгоритмами. Можливість формалізованого запису даних сприяє використанню ЕОМ для розв'язування аналітичних задач. До інформації, яка використовується для аналізу господарської діяльності, висуваються такі вимоги: достовірність, своєчасність, повнота, детальність, багатоаспектність.

Результати аналізу подаються у вигляді графіків, діаграм тощо. Для аналітичної роботи найбільшу значущість має процедура математичної обробки та наступне використання результатів аналітичних розрахунків у інтересах управління. На відміну від облікової інформації, коло споживачів результатів аналітичних розрахунків є значно ширшим та різноманітнішим. Бухгалтерська та аналітична інформація мають бути релевантними, тобто відповідати вимогам розв'язання відповідних функціональних управлінських завдань.

Інформаційна сукупність з мінімальним складом реквізитів-ознак і реквізитів-основ, достатня для створення елементарного документа (документо-рядка), утворює економічний показник. Показник характеризує об'єкт управління з якісного й кількісного боку – має назву, яка розкриває його форму, та значення, котре доповнює форму кількісно-якісними характеристиками. Набір взаємозв'язаних даних однієї форми (однієї назви) з усіма її значеннями - це масив даних, що є основною інформаційною сукупністю, котрою оперують в інформаційних процедурах. Сукупність масивів, що стосуються даних тієї самої ділянки управлінської роботи, називають інформаційним потоком.

Під фізичною структурою даних розуміють відповідні структурні одиниці залежно від носія інформації та способу її фіксації. Наприклад, структурною одиницею найвищого рівня інформації на паперовому документі є весь комплект документів, усе документаційне господарство об'єкта управління. Одиницями нижнього рівня є зона документа – рядок, графа, позиція.

Виробничо-господарська діяльність вимірюється не одним, а численними економічними показниками, які утворюють відповідну систему. Від добору цих показників залежить глибина аналізу та обґрунтування висновків і заходів за результатами аналізу. Економічні показники поділяються на: кількісні і якісні; натуральні, трудові, вартісні; абсолютні й відносні; загальні й часткові; планові (нормативні) і фактичні (звітні); основні й допоміжні; вихідні й розрахункові.

Кожний показник системи має певний зміст і значення для аналізу. Для економічного аналізу важливе їх комплексне використання, яке забезпечує всебічне й об'єктивне дослідження господарської діяльності підприємства. Кількість показників може змінюватися через їх диференціацію (інтеграцію) залежно від цілей і змісту аналізу, глибини вивчення відповідних об'єктів та процесів.

Для створення систем обробки аналітичної інформації особливого значення набувають машинні структури даних, що пов'язано з розміщенням масивів даних у пам'яті ЕОМ. Для розв'язування аналітичних (економічних) задач на ЕОМ визначено таку ієрархію даних у порядку зростання їх логічної складності.

За характером взаємозв'язку елементів усі структури даних можна поділити на лінійні та ієрархічні (нелінійні).

Для ідентичного відображення даних у процесах обробки інформації на різних рівнях управління застосовуються системи класифікації та кодування, які формалізовано описують показники, об'єкти, явища, процеси. В економічному

аналізі використовується кілька видів класифікаторів: локальні, що їх призначено для використання на одному об'єкті (підприємстві, фірмі) і котрі не виходять за його межі; територіальні, регіональні або галузеві, призначені для відповідних об'єктів; державні (національні), що розробляються на державному рівні і, як правило, є обов'язковими для використання; міжнародні.

Розв'язування аналітичних задач передбачає їх порівнянність, що можливе за наявності єдиних систем угруповань, побудованих за єдиними класифікаційними ознаками. З 1994 р. в Україні розроблено близько 20-ти національних статистичних класифікацій, які базуються на засадах методології міжнародних статистичних класифікацій (наприклад, "Класифікація видів економічної діяльності", "Класифікація продукції та послуг", "Класифікація валют" тощо). У майбутньому будуть створені системи класифікаторів, сумісних на державному, регіональному, галузевому рівнях та на окремих об'єктах народного господарства. А перспективи розширення виробничих і торговельних зв'язків з іншими країнами зумовлюють необхідність використання й міжнародних класифікаторів.

Автоматизована обробка аналітичної інформації передбачає використання економічних і машинних кодів. Економічні коди застосовуються для подання техніко-економічної інформації, наприклад, код основного засобу, код матеріалу. Машинні коди використовуються для управління ЕОМ, подавання команд і т. д. (наприклад, службові коди ЕОМ).

Коди можуть бути цифровими, алфавітними, алфавітно-цифровими. Останнім часом набули широкого використання штрихові коди, які на міжнародному рівні застосовуються у виробництві, торгівлі, для митного контролю, статистики, обліку. Штрихове кодування сприяє автоматизованій ідентифікації та електронному обміну даними, створенню інформаційної бази для контролю, аналізу та управління товарно-грошовим обігом. Єдиний код товару забезпечує вільний обмін електронною інформацією в процесі його виробництва, складування, транспортування, реалізації.

Взаємозв'язок економічного аналізу й інформації виявляється в тому, що в процесі аналізу здійснюється контроль за самою інформацією, яка, у свою чергу, є основою для проведення аналізу.

Останнім часом у країнах Західної Європи, у США, Канаді, Австралії та інших країнах все більшої популярності набуває так званий *інформаційний аудит*, який передбачає систематичне вивчення використання інформації, ресурсів і потоків, що проводиться як щодо користувачів, так і щодо існуючих документів для моніторингу рівня їх залучення для виконання завдань організації.

Доцільність такого напрямку аудиту пояснюється необхідністю перевірки даних, які надходять до інформаційної системи, на точність і автентичність за допомогою спеціальних програм, тому що завжди існує небезпека підроблення електронних звітів або даних. Особливістю інформаційного аудиту саме і є перевірка не тільки звичайних документів, а й комп'ютерних систем, програм і файлів. Пошук аудиторських доказів теж здійснюється електронним способом, що ставить високі вимоги до кваліфікації аудитора, його знань комп'ютерної техніки і програмних засобів.

Інформаційний аудит відповідає на запитання: яка інформація є на підприємстві і де саме; хто нею користується; якою є міра її корисності; як підтримувати її та доповнювати новою; скільки на це потрібно часу та грошей.

Одним із різновидів інформаційного аудиту є аудит процесів збирання, обробки, зберігання й розподілу інформації. Його результати встановлюють рівень інформаційних потреб системи управління й основних підрозділів підприємства (з урахуванням різниці в потребах спеціалістів і керівників); визначають відповідність потреб поставленим цілям, міру задоволення потреб; особливості використовуваних документів (первинних, вторинних) та джерел їх виникнення; вірогідність даних, їхню автентичність; якість і надійність процесів обробки, зберігання, передавання інформації; адекватність витрат на інформаційні процеси, як порівняти із досягнутими результатами. До інформаційного аудиту включають також аудит інформаційних систем, комп'ютерний аудит, аудит систем комунікації.

На жаль, нині в Україні інформаційний аудит використовується ще недостатньо. Більшість організацій, підприємств, банків покладаються на керівні організації чи провайдерів, мотивуючи це браком коштів. Але збитки від порушень нормального функціонування інформаційної системи, втрати стратегічно важливої комерційної інформації є значно більшими від витрат на забезпечення якісної інформації.

Інформаційна система

У зв'язку з застосуванням нової інформаційної технології, заснованої на використанні засобів електронного зв'язку та комп'ютерної техніки, широко використовується поняття "інформаційна система" (ІС). При цьому термін "система" вживається в основному в двох змістах:

1 – система як деяка властивість, що складається в раціональному сполученні й упорядкованості всіх елементів визначеного обсягу в часі і просторі так, що кожний із них сприяє успіху діяльності всього об'єкта (з таким трактуванням пов'язане розуміння координації і синхронізації дій персоналу управління, об'єднаних із метою досягнення поставлених цілей);

2 – система як об'єкт, що володіє достатньо складною, певним чином упорядкованою внутрішньою структурою (наприклад, виробничий процес).

У загальному виді поняття "система" охоплює комплекс взаємозалежних елементів, що діють як єдине ціле в інтересах досягнення поставлених цілей.

Кожну систему характеризує:

- структура системи - множина елементів системи і взаємозв'язків між ними (приклад: організаційна і виробнича структура фірми);
- функції елементів (приклад: управлінські функції - прийняття рішень структурним підрозділом фірми);
- вхід і вихід кожного елемента і системи в цілому (приклад: матеріальні або інформаційні потоки, що надходять у систему або виводяться нею);
- цілі й обмеження системи та її окремих елементів (приклад: досягнення максимального прибутку; фінансові обмеження).

Кожна система має властивості подільності і цілісності. Подільність означає, що систему можна уявити як підсистему, що складається з самостійних частин, кожна з яких може розглядатися як самостійна підсистема. Можливість виділення підсистем (декомпозиція системи) спрощує її аналіз, розробку, впровадження й експлуатацію. Виділення (декомпозиція) підсистем являє собою достатньо складну задачу. Властивість цілісності вказує на узгодженість цілі функціонування всієї системи з цілями функціонування її підсистем і елементів.

Інформаційна система являє собою систему збору, передачі, переробці інформації про об'єкт, що постачає працівників різноманітного рангу інформацією для реалізації функції управління.

Інформаційна система створюється для конкретного об'єкта. Ефективна інформаційна система бере до уваги розходження між рівнями управління, сферами дії, а також зовнішніми обставинами і дає кожному рівню управління тільки ту інформацію, що йому необхідна для ефективної реалізації функцій управління.

Впровадження інформаційних систем провадиться з метою підвищення ефективності виробничо-господарської діяльності фірми за рахунок не тільки опрацювання і збереження рутинної інформації, автоматизації офісних робіт, але і за рахунок принципово нових методів управління, заснованих на моделюванні дій спеціалістів фірми при прийнятті рішень (методи штучного інтелекту, експертні системи і т.п.), використанні сучасних засобів телекомунікації (електронна пошта, телеконференції), глобальних і локальних обчислюваних мереж тощо.

У залежності від ступеня (рівня) автоматизації виділяють ручні, автоматизовані й автоматичні інформаційні системи. Ручні ІС характеризуються тим, що всі операції по переробці інформації виконуються людиною. Автоматизовані ІС - частина функції (підсистем) управління або опрацювання даних здійснюється автоматично, а частина - людиною. Автоматичні ІС - усі функції управління й опрацювання даних здійснюються технічними засобами без участі людини (наприклад, автоматичне управління технологічними процесами).

По сфері застосування можна виділити такі класи інформаційних систем: наукові дослідження; автоматизоване проектування; організаційне управління; управління технологічними процесами.

Наукові ІС призначені для автоматизації діяльності науковців, аналізу статистичної інформації, управління експериментом. ІС автоматизованого проектування призначені для автоматизації праці інженерів-проектувальників і розроблювачів нової техніки (технології). Такі ІС допомагають здійснювати розробку нових виробів і технологій їхнього виробництва, виконувати різноманітні інженерні розрахунки (визначення технічних параметрів виробів, видаткових норм - трудових, матеріальних і т.д.), створювати графічну документацію (креслення, схеми, плани), здійснювати моделювання проєктованих об'єктів, створювати керуючі програми для верстатів із числовим програмним керуванням.

ІС організаційного управління призначені для автоматизації функцій адміністративного (управлінського) персоналу. До цього класу відносяться ІС управління як промисловими (підприємства), так і непромисловими об'єктами (банки, біржі, страхові компанії, готелі і т.д.) і окремими офісами (офісні системи).

ІС управління технологічними процесами призначені для автоматизації різноманітних технологічних процесів (гнучкі виробничі процеси, металургія, енергетика і т.п.).

Практично всі розглянуті різновиди інформаційних систем незалежно від сфери їхнього застосування включають один і той же набір компонентів:

- функціональні компоненти: функціональні підсистеми (модулі, бізнеси-додатки);
- функціональні задачі, моделі й алгоритми;
- компоненти системи опрацювання даних: інформаційне забезпечення, програмне забезпечення, технічне забезпечення, правове забезпечення, лінгвістичне забезпечення;

– організаційні компоненти (персонал): нова організаційна структура фірми, персонал (штати, посадові інструкції).

Під *функцією управління* розуміється спеціальний постійний обов'язок одного або декількох осіб, виконання якого призводить до досягнення визначеного ділового результату.

Під *функціональними компонентами* розуміється система функцій управління - повний набір (комплекс) взаємозв'язаних у часі і просторі робіт із управління, необхідних для досягнення поставлених перед підприємством цілей.

Будь-яка складна управлінська функція розчленовується на ряд більш дрібних задач і зрештою доводиться до безпосереднього виконавця. Саме від того, як буде виконане те або інше завдання окремим працівником, залежить успіх у рішенні кінцевих задач фірми в цілому. Таким чином, уся сукупність складних управлінських впливів повинна мати своїм кінцевим результатом доведення загальних задач, що стоять перед підприємством, до кожного конкретного виконавця незалежно від його службового положення. Природно, приведені положення підкреслюють не тільки індивідуальний, але і груповий характер функцій управління, а діловий (практичний) результат утворюється не епізодично, а постійно. Весь процес управління фірмою зводиться або до лінійного (наприклад, адміністративного) керівництва підприємством або його структурним підрозділом, або до функціонального керівництва (наприклад, матеріально-технічне забезпечення, бухгалтерський облік і т.п.). Тому декомпозиція інформаційної системи по функціональній ознаці містить у собі виділення її окремих частин, які називають функціональними підсистемами (ПС) (функціональними модулями, бізнес-додатками), що реалізують систему функцій управління. Функціональна ознака визначає призначення підсистеми, тобто те, для якої області діяльності вона призначена і які основні цілі, задачі і функції вона виконує. Функціональні підсистеми в істотному ступені залежать від предметної області (сфери застосування) інформаційних систем.

Незважаючи на різноманітні сфери застосування ІС, ряд функціональних підсистем мають те саме найменування (наприклад, бухгалтерський облік і звітність), проте їх внутрішній зміст для різноманітних об'єктів значно відрізняється. Специфічні особливості кожної функціональної підсистеми містяться в так званих "функціональних задачах" підсистеми. Звичайно управлінський персонал або зв'язує це поняття з досягненням визначених цілей функції управління, або визначає його як роботу, що повинна бути виконана визначеним засобом у визначений період. З появою нових інформаційних технологій поняття "задача" розглядається ширше - як закінчений комплекс опрацювання інформації, що забезпечує або видачу прямих керуючих впливів на хід виробничого процесу, або видачу необхідної інформації для прийняття рішень управлінським персоналом. Таким чином, задача повинна розглядатися як елемент системи управління, а не як елемент системи опрацювання даних.

Вибір складу функціональних задач функціональних підсистем управління здійснюється з урахуванням основних фаз управління: планування; обліку, контролю й аналізу; регулювання (виконання).

Планування - це управлінська функція, що забезпечує формування планів, відповідно до яких буде організоване функціонування об'єкта управління.

Звичайно виділяють перспективне (5-10 років), річне (1 рік) і оперативне (доба, тиждень, декада, місяць) планування.

Облік, контроль і аналіз - це функції, що забезпечують одержання даних про стан керованої системи за визначений проміжок часу; визначення факту і причини відхилень фактичного стану об'єкта управління від його планованого стану, а також перебування розмірів цього відхилення. Облік ведеться по показниках плану в обраному діапазоні (обрїї) планування (оперативний, середньостроковий і т.д.).

Регулювання (виконання) - це функція, що забезпечує порівняння планованих і фактичних показників функціонування об'єкта управління і реалізацію необхідних керуючих впливів при наявності відхилень від запланованих у заданому діапазоні (відрізку).

Прагнення компаній підвищити ефективність ІС стимулює поява більш нових апаратних і програмних засобів, що, у свою чергу, підштовхують користувачів до подальшої модернізації ІС. Зрозуміло, ця "кільцева гонка" не є самоціллю: завдяки їй підприємці можуть більш адекватно реагувати на зміну ринкової кон'юнктури і отримувати максимум прибутку при мінімальному ризику. Логіка розвитку ІС в останні 30 років наочно демонструє ефект маятника: централізована модель опрацювання даних на базі мейнфреймів, що домінувала до середини 80-х років, усього за декілька років уступила свої позиції розподіленій архітектурі однорангових локальних мереж (ЛС) персональних комп'ютерів, але потім почалося поворотне прямування до централізації ресурсів системи. Сьогодні в центрі уваги знаходиться технологія "клієнт-сервер", що ефективно об'єднує переваги своїх попередників.

Перше покоління ІС (1960-1970 р.) будувалося на базі центральних ЕОМ за принципом "одне підприємство - один центр опрацювання", а в якості стандартного середовища виконання додатків (функціональних задач) служила операційна система фірми ІВМ - MVS.

Друге покоління ІС (1970-1980 р.): перші кроки до децентралізації ІС, у процесі якої користувачі стали просувати інформаційні технології в офіси і відділення компаній, використовуючи міні-комп'ютери типу DEC VAX. Паралельно почалося активне упровадження високопродуктивних СКБД типу DB2 і пакетів комерційних прикладних програм. Таким чином, кардинальним нововведенням ІС цього покоління стала двох і трьохрівнева модель організації системи опрацювання даних (центральна ЕОМ - міні-комп'ютери відділень і офісів) з інформаційним фундаментом на основі децентралізованої бази даних і прикладних пакетів.

Третє покоління ІС (1980 - початок 1990-х рр.): бум розподіленого мережного опрацювання, головною рушійною силою якого був масовий перехід на персональні комп'ютери (ПК). Логіка корпоративного бізнесу зажадала об'єднання розрізнених робочих місць у єдину ІС - з'явилися обчислюванні мережі розподіленого опрацювання. Проте вже незабаром в однорангових мережах стали виявлятися перші ознаки ієрархічності - спочатку у виді виділених файлів-серверів, серверів преси і телекомунікаційних серверів, а потім і серверів додатків. На якомусь етапі зростаючу потребу в концентрації ресурсів ІС, відповідальних за адміністрування системи (організацію обчислювального процесу), підтримку корпоративної бази даних і виконання пов'язаних із нею централізованих додатків, удалося задовольнити в так званій моделі "середнього калібру" за рахунок використання UNIX-серверів, що випускаються ІВМ, DEC, Hewlett-Packard, Sun і ін. Ринок серверів став одним із самих динамічних секторів комп'ютерної індустрії. При розвитку ІС третього покоління ідея чистого (однорангового) розподіленого опрацювання помітно потьмяніла і поступилася місцем ієрархічній моделі клієнт-сервер.

Четверте покоління ІС. Відмітні риси сучасних ІС, насамперед ієрархічна організація, у котрій централізоване опрацювання і єдине управління ресурсами ІС на верхньому рівні сполучиться з розподіленим опрацюванням на нижньому, визначаються синтезом рішень, апробованих у системах попередніх поколінь. Інформаційні системи четвертого покоління акумулюють такі основні особливості: повне використання потенціалу настільних комп'ютерів і середовищ розподіленого опрацювання; модульна побудова системи, що передбачає існування багатьох різноманітних типів архітектурних рішень у рамках єдиного комплексу; економія ресурсів системи (у самому широкому розумінні цього терміна) за рахунок централізації збереження й опрацювання даних на верхніх рівнях ієрархії ІС; наявність ефективних централізованих засобів мережного і системного адміністрування (організації обчислювального процесу), що дозволяють здійснювати наскрізний контроль за функціонуванням мережі і управління на всіх рівнях ієрархії, а також забезпечуючих необхідну гнучкість і динамічну зміну конфігурації системи; різке зниження так званих "прихованих витрат" - експлуатаційних витрат на утримання ІС, що включають витрати, що важко виділяються в явному виді, що непросто передбачити в бюджеті організації: підтримка функціонування мережі, резервне копіювання файлів користувачів на віддалених серверах, настроювання конфігурації робочих станцій і підключення їх у мережу, забезпечення захисту даних, відновлення версій програмного забезпечення і т.д.. Передбачається, що розвиток ІС четвертого покоління буде йти по шляху однієї з трьох моделей: великої, середньої або малої.

По логіці згаданих моделей у структурі ІС повинні існувати один або декілька "*інформаційних вузлів концентрації*" (ІВК), кожний із яких об'єднує апаратні і програмні засоби, призначені для ефективної підтримки роботи кінцевих користувачів. З цією ж ціллю в подібних вузлових центрах системи зосереджується спеціалізований персонал, що виконує функції системного адміністрування, управління мережними ресурсами і технічною підтримкою. Кінцеві користувачі працюють у середовищі локальних мереж, і їхні індивідуальні додатки і дані максимально локалізуються на рівні станцій клієнтів. Залучення ресурсів вузла концентрації відбувається тільки в рідкісних випадках, наприклад при зверненні до корпоративної бази даних або резервному копіюванні файлів.

"Неавтоматизоване" підприємство функціонує як система, у якій усі документи (у тому числі внутрішні) існують у паперовому виді. До середини 90-х років хазяїн неавтоматизованого підприємства відчув необхідність хоча б часткової оптимізації діяльності і став автоматизувати елементи виробничого процесу, здійснювати фінансові розрахунки, вести бухгалтерський облік, контролювати систему продажу і закупівель. Це були перші "острівці" в загальному потоці неавтоматизованої економічної діяльності.

В міру росту підприємства здійснювалася автоматизація структур більш високого порядку - усієї системи виробництва, усієї системи роботи з постачальниками, усієї системи роботи з партнерами. Подальше укрупнення діяльності підприємства потребувало автоматизації його організаційної структури. Нарешті, автоматизація всього документообігу підприємства фактично поставила знак рівності між поняттями "підприємство" і "інформаційна система".

Простота приведеної послідовності спроможна ввести в оману. Здається, що впровадження інформаційної системи обмежується установкою комп'ютерів, мережного устаткування і деякого програмного забезпечення. При цьому загальна вартість проекту як би виходить рівною сумі вартостей указаних компонентів, що просто візьмуть на себе основну частину рутинних операцій, нічого не змінюючи в стилі роботи керівництва і персоналу підприємства. На жаль, це не так. Типових рішень по документообігу немає, впровадження будь-яких існуючих пакетів потребує кропіткої попередньої роботи системних аналітиків. Не дивно, що

послуги консультантів і системних аналітиків вартісно дорожче, чим устаткування - така світова практика. Комп'ютерна інформаційна система підприємства не може бути річчю в собі, просто комплексом АРМів, зв'язаних із єдиною базою даних: у такому варіанті вона мертва. На етапі консалтингу повинні бути вироблені регламенти і посадові інструкції, що вдихнуть життя в комп'ютерну інформаційну систему. Таким чином інформаційна система трансформується у корпоративну - це комп'ютерну інформаційну систему й організаційно-інформаційну систему, нероздільно пов'язані між собою.

Можна виділити чотири чинники цілісності згаданого комплексу:

- концептуальна узгодженість бізнес-процесів, для автоматизації яких створюється ІС, що зберігається на всьому протязі її життєвого циклу;
- технологічна цілісність, яка проявляється в застосуванні погодженого набору промислових інформаційних технологій для управління інформаційними ресурсами підприємства;
- відповідність функціональності робочих місць співробітників їхнім посадовим обов'язкам;
- єдиний регламент обслуговування й експлуатації всіх компонентів ІС, розроблювальний при її створенні.

Сполучення означених властивостей принципово відрізняє ІС від суми додатків з тим же набором функцій і дозволяє їй справлятися з комплексом проблем, непереборних при безсистемній автоматизації бізнесу. Корпоративна ІС заснована на моделі бізнесу, звільненому від другорядних деталей схематичному описі діяльності підприємства. Модель формалізує окремі бізнес-функції (наприклад, виписування накладної або прийом замовлення) і регламентує структуру бізнес-процесів, тобто послідовність виконання бізнес-функцій у повсякденній діяльності підприємства (наприклад, пошук співробітника на вакантну посаду супроводжується публікацією опису вакансії, збором анкет і резюме, перевіркою фактів, що знаходяться в них, співбесідами тощо). На різних підприємствах ці бізнес-функції можуть бути пов'язані по-різному, що і відбиває специфіку роботи відділів кадрів. Модель визначає, яка послідовність дій співробітників, що працюють із ІС, припустима, а яка – ні (наприклад, якщо, відповідно до моделі, відвантаження товару можливе тільки після його оплати, ІС просто не дозволить виписати накладну, поки інформація про оплату не буде внесена в систему). Модель фіксує логічні взаємозв'язки даних, відповідно до яких зміна якоїсь інформації викликає каскад узгоджених змін (наприклад, оплата рахунку на виписаний товар викликає бухгалтерські перевірки і робить можливим видачу товару покупцю). Орієнтація на промисловий рівень базових технологій на етапі створення системи збільшує початкові витрати і складність системи. Але корпоративна ІС створюється з розрахунком на розвиток бізнесу, тому слабкі ланки неминуче прийдеється змінювати на працюючій ІС, що сполучено з набагато більшими витратами, чим закладка тривкого фундаменту вже при створенні ІС.

Організація системної обробки економічної інформації

Науково-технічний прогрес глибоко впливає на розвиток економіки. Тому необхідно і далі всіляко забезпечувати прискорення, а також застосовувати його наслідки у виробництві та управлінні. Сучасні електронні обчислювальні машини, периферійна і організаційна техніка, яка є елементом цього наукового прогресу, знаходять широке застосування в організації системної обробки економічної інформації.

Системна обробка економічної інформації – система збирання і обробки, формування та забезпечення у ритмі виробництва (тобто за час, який близький до реального) інформацією користувачів різних рівнів, а також зберігання і розмноження всієї потрібної для управління інформації (документів) з застосуванням нової інформаційної технології.

Об'єктами, яких стосується обробка інформації в умовах ринкової економіки є: галузі виробництва, асоціації, територіальні та інші комплекси, виробничі та науково-виробничі об'єднання і підприємства, державні, акціонерні, оренди і, спільні, малі та інші підприємства, біржі, організації та установи. Найпоширенішими об'єктами управління у промисловості, де ефективно використовуються можливості системної обробки економічної інформації, є підприємства, які знаходяться на повному господарському розрахунку, тобто є самостійними.

Щоб означені підприємства успішно діяли, надзвичайно важлива компетентність, професіональна підготовленість та заповзятість, нарешті, комп'ютерна грамотність як керівників, так і фахівців різних рівнів. Для успішного виконання своїх посадових обов'язків (функцій) зазначеним працівникам у нових умовах економічних відносин потрібно використовувати і нові форми та методи організації управління підприємством. Ці нові форми та методи повинні ґрунтуватися на сучасних засобах обчислювальної, периферійної та організаційної техніки, яка і використовується для організації і функціонування автоматизованих систем збирання і обробки, формування та забезпечення інформацією користувачів, а також для розмноження та зберігання всієї потрібної інформації, яка зв'язана з повним циклом управління виробництвом.

Управління як процес є актом трудової діяльності працівників. Він здійснюється з застосуванням знарядь і предметів праці. Предметом праці у процесі управління є інформація, а знаряддям її на сучасному етапі є технічні засоби вимірювання, сприйняття, реєстрації, передачі, обробки, аналіз у, зберігання і розмноження цієї інформації.

Управління організаційним (соціальним) об'єктом, яким є підприємство, являє собою процес, що складається з взаємозв'язаних та взаємозалежних елементів. Такі елементи – керуюча, керована та інформаційна системи. Основним змістом керуючої системи є переведення керованої системи з одного стану в інший відповідно до заздалегідь поставленої мети за допомогою інформаційної системи.

Загальна логіка організації процесу управління полягає у тому, що перш за все ставиться мета, потім визначаються засоби та розроблюються заходи до її досягнення, нарешті ставиться завдання, вирішення якого і приведе до поставленої мети. Кінцевою стадією управлінської дії є вироблення та виконання рівень, які забезпечать нормальний хід процесів на керованому об'єкті за раніше складеним планом, розробленим відповідно до поставленої мети.

Управління підприємством, по своїй суті та змістом, складається з сукупності взаємозв'язаних функцій. Експлуатаційні можливості сучасного комплексу технічних засобів, що використовується в автоматизованій системі збирання і обробки економічної інформації, дозволяють автоматизовано виконувати ряд процедур у цих функціях. Стан науково-практичних розробок та технічний рівень зазначеного комплексу визначили можливості автоматизованого виконання таких процедур управлінського процесу:

– у *прогнозуванні та плануванні* - багатоваріантні розрахунки при розробці прогнозів, перспективних і поточних економічних та соціальних планів розвитку підприємства, а також оперативно-виробничих планів і планів з технічної підготовки виробництва з метою наступного визначення оптимальних взаємозв'язаних наборів показників;

– в *організації* - моделювання організаційних структур управління та імітація процесів виробництва при різних критеріях і параметрах з метою вибору оптимальних;

– у *координації і регулюванні* - подання команд на робочі місця (поки що на низовому рівні управління виробництвом) відповідно до плану, технологічного процесу чи інструкції, складених на ті чи інші види робіт або операції;

– у *контролі* - спостережінні за станом керованого об'єкта по всіх параметрах, а також за своєчасним і повним виконанням керівних команд;

– в *обліку* - одноразове збирання (у ритмі виробництва) і системна обробка всієї фактичної (разом з довідковою, плановою, нормативною та іншою) достовірної інформації яро наявність та рух ресурсів, а також про процеси та явища, що відбуваються у виробничо-господарській та іншій діяльності підприємства;

– в *аналізі* - зіставлення (поміж собою) нормативних, планових і фактичних показників, які характеризують ті чи інші операції або процеси виробничо-господарське і та іншої діяльності, виявлення відхилень (у кількісних, вартісних, відносних та інших величинах) від заданих параметрів із зазначенням причин і винуватців цих відхилень, оцінка виконання плану у різних аспектах та виявлення факторів, що впливають на ці відхилення;

– у *звітності* - автоматичне формування (на основі первинних даних) зведених показників, що відображаються у тилових формах установленої бухгалтерської, статистичної та іншої звітності за допомогою спеціальних перевідних масивів - довідників, а також одночасне створення машинних носіїв з відповідними зведеними показниками звітності для передання (інколи це робиться по каналах зв'язку) їх до зовнішніх та інших установ (інституція).

Відомо, що в умовах автоматизованої системи збирання і обробки економічної інформації первинну базу наукового прогнозування і планування складають обґрунтовані норми і нормативи, як і застосовуються в усіх процесах та явищах, що відбуваються на об'єкті управління. Процедури на їх розробку, аналіз та всебічне використання також повністю автоматизовані.

Експлуатаційні можливості персональних електронних обчислювальних машин (ПЕОМ), що використовуються як автоматизовані робочі місця (АРМ) фахівців різних рівнів, дозволяють оперативно (у ритмі виробництва) і у системному порядку обіграти достовірну первинну (фактичну) масову інформацію на робочих місцях, де вона виникає; потім всю цю інформацію попередньо обробляти таким чином, щоб у ритмі виробництва повністю забезпечувати вихідною (результатною) інформацією, з одного боку, широкі потреби користувачів - керівників та фахівців даного (низового) рівня управління, а з другого, передавати в стислому чи іншому вигляді по інформацію на більш високий (середній та верхній) рівень управління. На цьому рівні зібрана інформація (після додаткової обробки, а інколи і в такому, первинному, вигляді) використовується не тільки для аналізу і прийняття управлінських рішень, але і для обліку, контролю, комплексного економічного аналізу результатів виробничо-господарської та іншої діяльності структурних ланок і підприємства в цілому, а також для складання установленої зведеної бухгалтерської, статистичної та іншої звітності.

Одноразове збирання фактичної інформації та системна її обробка на ПЕОМ дозволяє забезпечувати в принципі без паперове (тобто на екрані ПЕОМ) повне оперативне інформування користувачів (керівників) різних рівнів про процеси та

явища, що відбуваються на об'єктах управління. Забезпечення таким чином інформацією керівників створює умови для виконання у повному (в інформаційному аспекті) обов'язку посадових функцій відповідно до наданих прав, обов'язків та міри відповідальності за доручену справу. Забезпечення інформацією може здійснюватись як автоматично (за заздалегідь розроблений графіком), так і у діалоговому режимі (за запитом користувача).

Отже, за допомогою сучасних засобів обчислювальної, периферійної і організаційної техніки можна більш якісно і ефективно виконувати функції управління як виробництвом у цілому, так і його структурними ланками. Крім того, за допомогою автоматизованої системи збирання і обробки економічної інформації можна удосконалювати форми та методи управління виробництвом.

Сутність системного підходу до обробки економічної інформації і використання нових інформаційних технологій

Науковою основою для раціонального або ефективного управління соціально-економічними й організаційними системами є системний підхід.

Системний підхід - ця сукупність методологічних принципів і положень, які дозволяють розглядати систему як єдине ціле з погодженою діяльністю всіх її елементів або частин, які називаються інколи підсистемами.

Означені при визначенні системного підходу підсистеми в ряді випадків виступають як “самостійні” системи нижчого рівня. Такий підхід передбачає вивчення кожного елементу або частини (підсистеми) в його взаємозв'язку і взаємодії з іншими елементами або частинами (підсистемами); дозволяє спостерігати заміни, які відбуваються у системі як результат зміни її основних елементів або частин (підсистем); виявляти специфічні системні властивості, робити обґрунтовані припуски відносно закономірності розвитку системи та визначити оптимальний режим її функціонування.

Об'єкт управління, будь-то підприємство або виробниче об'єднання, є організаційно-економічною системою. В цих об'єктах управління взаємозв'язок керуючої і керованої систем визначається через інформаційну систему, в якій інформаційні сукупності, що характеризують нормативні, планові і фактичні явища та процеси, також щільно взаємозв'язані і взаємозалежні.

На підприємстві конструкторська і технологічна інформація характеризує структуру і технологію виготовлення виробів, нормативна і розцінкова інформація є базою при плануванні виробництва, облікова (фактична) інформація характеризує процеси і явища, що здійснюються у виробничо-господарській та іншій діяльності. Через останню інформацію визначається ступінь виконання плану виробництва по виготовленню виробів за всіма параметрами відповідно до конструкторських і технологічних вимог.

Якщо нормативно-планова і конструкторсько-технологічна (тобто умовно-постійна) інформація зазначає, “що”, “в якій кількості”, “як”, “де”, “на чому”, “за допомогою чого”, “в яких умовах” слід робити, то фактична (облікова) відповідає на ті ж питання, але стосується того, що уже зроблено. То ж перша - умовно-постійна інформація - доповнюється другою - фактичною.

Якщо формування і обробка інформаційних сукупностей виконується вручну, то ці загальні умовно-постійні показники в процесі управлінських і інших

операція багаторазово переписуються фахівцями різних рівнів - конструкторами і технологами, плановиками і бухгалтерами, економістами і нормувальниками тощо.

Якщо ці умовно-постійні показники один раз зафіксувати на машинних носіях, їх можна багаторазово використовувати фахівцями в своїй роботі, в тому числі і в обліку. Так, якщо доповнити їх фактичними показниками, що виникають у виробничо-господарській діяльності, то всю по інформаційну сукупність можна обробляти автоматично на ПЕОМ.

Зазначені процеси можна здійснити, використовуючи комплекси засобів обчислювальної і периферійної техніки, за допомогою якої раціонально виконується чотири групи основних операцій: збір і передача інформації в пам'ять ПЕОМ, її обробка за заздалегідь розробленими алгоритмами і програмами, видача обробленої інформації користувачам із пам'яті ПЕОМ, зберігання і пошук інформації. В залежності від експлуатаційних можливостей і ступеня застосування ПЕОМ при виконанні цих операцій, здійснюється часткова чи повна комплексна механізація операцій, часткова чи повна автоматизація операцій. Повна або комплексна автоматизація операцій на практиці і в літературі відома як машинна або автоматизована обробка економічної інформації.

Застосування технічних засобів для автоматизованого виконання зазначених операцій суттєво знижує трудомісткість збору і обробки фактичної (первинної) інформації, зменшує кількість помилок, звільнює багато часу у фахівців для виконання інших обов'язків. Більше того, створюються реальні умови для всебічного одноразового збору і багаторазової обробки інформації на різних ділянках, починаючи з тієї, де інформація виникає вперше, тобто з конструкторської підготовки виробництва. Конструкторська (така, що характеризує склад виробу) інформація, яка зафіксована на машинному носії, на різних етапах управлінсько-виробничих процесів автоматизовано доповнюється технологічною, а потім нормативною, плановою і, нарешті, фактичною інформацією. Все це свідчить про те, що перехід від механізації або автоматизації на окремих ділянках обліку і планування до системної автоматизованої обробки економічної інформації по підприємству в цілому є об'єктивною необхідністю.

Удосконалення технічних засобів і методів по збиранню і обробці економічної інформації, наприклад, поява ПЕОМ, які використовуються як АРМ фахівців, свідчить про те, що є реальна можливість збирати первинну (фактичну) інформацію в ритмі виробництва, тобто близько до реального часу, здійснити перехід від рішення окремих обліково-планових і нормативних задач до системного рішення задач по управлінню підприємством в цілому. Цьому сприяє і впровадження комп'ютерної інформаційної технології, яка одержала назву нової інформаційної (без паперової) технології.

Основу нової інформаційної (без паперової) технології, при якій застосовуються ПЕОМ, складають розподілені системи сучасної обчислювальної техніки, "приятне" програмне забезпечення, розвиток комунікацій, без паперове (тобто екранне) видавання обробленої інформації користувачам. Користувачу, який не є програмістом, надана можливість прямого спілкування з ПЕОМ за допомогою діалогового режиму. Програмно-апаратні засоби (бази даних, експертні системи, бази знань тощо), якими забезпечується ПЕОМ, створюють зручність у роботі, дозволяють не тільки автоматизувати процес зміни форми і місця розміщення інформації, але і змінити її

зміст. Таким чином ПЕОМ надають можливість фахівцю підвищити продуктивність праці шляхом збільшення обсягів робіт, які він виконує індивідуально.

Нова інформаційна (без паперова) технологія є не що інше, як сукупність автоматизованих процесів руху (по каналах зв'язку) і обробки інформації тієї чи іншої предметної області (або їх сукупності), що видається користувачу на екран ПЕОМ.

Для такої технології характерні:

- праця користувача на ПЕОМ у режимі маніпулювання даними;
- наскрізна інформаційна підтримка на всіх етапах проходження інформації на основі інтегрованої бази даних, яка забезпечує єдину уніфіковану форму зображення, зберігання, пошуку, відображення, відновлення і захисту інформації;
- без паперовий процес обробки інформації, при якому на папері фіксується (за необхідністю) лише остаточний варіант розрахунків;
- документація і необхідні дані доводяться до відома користувача через екран ПЕОМ;
- інтерактивний (діалоговий) режим рішення задач (одержання необхідних показників);
- можливість колективного виконання документів на основі об'єднаної каналами зв'язку групи ПЕОМ, які працюють в одному режимі;
- можливість адаптивної перебудови форм і способу подання інформації у процесі рішення задачі або при одержанні необхідних показників.

Нова інформаційна технологія може бути впроваджена на об'єкті управління двома способами: в локальні інформаційні структури які засновані на адаптації нової інформаційної технології до діючої організаційної структури; удосконаленням діючої організаційної структури управління.

Перший спосіб впровадження нової інформаційної технології приводить до локального удосконалення методів обробки інформації, які діють на об'єкті управління і не міняє діючу організаційну структуру управління. Рационалізуються тільки методи збору і обробки інформації на робочих місцях фахівців шляхом організації окремих автоматизованих робочих місць. У цьому випадку здійснюється розподіл обов'язків поміж оператором ПЕОМ (технічним працівником) і фахівцем. При цьому відбувається зливання операцій по збиранню і обробці первинної (інколи з умовно-постійною інформації), а в ряді випадків і з функцією прийняття управлінських рішень.

За другим способом впровадження нової інформаційної технології удосконалюється організаційна структура управління таким чином, щоб ця технологія принесла як найбільшу ефективність управлінню виробництвом. В цьому випадку відбувається розвиток комунікацій і розробляються нові інформаційні і організаційні взаємозв'язки, створюються комплекси взаємозв'язаних АРМ керівників і фахівців різних рівнів, впроваджуються розподілені бази даних. При цьому способом продуктивність організаційної структури управління збільшується за рахунок раціонального розподілення обробки інформації; відбувається зниження обсягів інформаційних потоків по каналах зв'язку, приближення обробленої інформації до користувачів різних рівнів, збирання і обробка первинної інформації в ритмі виробництва, яка передається відповідним користувачам в такому ж ритмі.

Таким чином, перший спосіб впровадження нової інформаційної технології орієнтований на діючу організаційну структуру управління. При цьому ступінь

ризиком від впровадження цієї технології зводиться до мінімуму, адже витрати незначні і організаційна структура не змінювалась. Другий спосіб спрямований на принципову зміну організаційної структури управління. Однак для обох способів принципово змінюються методи використання обчислювальної, периферійної і організаційної техніки: замість централізованого впроваджується децентралізований збір і первинна обробка інформації, ці процеси наближаються у першу чергу до місць, де виникає і використовується первинна (фактична) інформація. На практиці і в літературі ці методи використання одержали назву розподілених обчислювальних систем, в яких поряд з централізованою обробкою інформації на великих і середніх ЕОМ застосовується децентралізована, тобто розосереджена (розподілена) обробка інформації з використанням мікро-ЕОМ (ПЕОМ).

Щоб організувати системну автоматизовану обробку економічної інформації, яка використовується для управління об'єктом, потрібно створити таку систему, в якій достовірна первинна (фактична) інформація, яка характеризує виробничо-господарську і іншу діяльність, один раз в мінімальному складі в ритмі виробництва фіксувалась би і в такому ритмі передавалась і оброблювалась (разом з умовно-постійною) на ЕОМ таким чином, щоб одержана в максимумі результатна інформація була закінчена обробкою і повністю задовольняла би усім вимогам управління об'єктом, у тому числі інформуванню керівників і фахівців різних рівнів, а також процесам формування і планування, обліку і контролю, складання установленої зведеної звітності і проведення комплексного економічного аналізу за різні періоди часу і по різних структурних ланках і об'єкту управління в цілому.

Така системна обробка економічної інформації може здійснюватись на різних рівнях, у масштабі різних організаційно управлінських ланок - цеху або виробничої одиниці, підприємства або виробничого об'єднання, галузі, регіону тощо. Зі збільшенням масштабності управлінської ланки значно зростають трудомісткість і складність організації системної обробки економічної інформації.

Методичні основи автоматизації інформаційних процесів

Оперативність і якість управлінських рішень, які приймаються керівниками різних рівнів, значно залежить від своєчасного забезпечення процесу управління необхідною достовірною інформацією, яка характеризує процеси та явища, що відбуваються на тому чи іншому об'єкті управління.

Автоматизована система збирання і обробки економічної інформації починається з одноразового виміру первинної інформації, яка характеризує процеси та явища, що відбуваються у виробничо-господарській та іншій діяльності певного об'єкта управління чи його складових частин. Ця інформація фіксується в пам'яті ПЕОМ і негайно обробляється по розроблених алгоритмах, а потім використовується користувачами цієї ПЕОМ для виконання своїх посадових функцій. Для централізованої обробки і підведення підсумків зазначену інформацію збирають по каналах зв'язку.

Ефективність автоматизованої системи збирання і обробки економічної інформації в першу чергу залежить від застосованих методів і засобів по вимірюванню, фіксації і передачі інформації на обробку, а також від своєчасної її обробки і передавання користувачам різних рівнів.

На кожному етапі науково-технічного прогресу методично удосконалюється організація обліку (першу чергу – первинного) в напрямку його автоматизації.

Рівню технології обробки інформації повинен відповідати такий же (або близький до нього) рівень технології її збирання і передавання на обробку.

Важливим елементом методичного удосконалення обліку є наближення планових і облікових одиниць вимірювання до показників, що характеризують об'єкти управління. Це означає, що можна збирати як фактичні показники тільки відхилення їх від планових, додаючи при необхідності ряд ознак, що характеризують виконавців, а також причини та винуватців відхиляємих величин. Все це значно скорочує обсяги інформації, що збирається, виключає її дублювання, зменшує трудомісткість процесу її збирання.

Щоб первинна (один раз зібрана) інформація, що характеризує фактичні процеси та явища, які проходять на об'єктах управління, після обробки і на ЕОМ використовувалась персоналом управління не тільки для обліку, аналізу та складання установленної звітності, а також і для вироблення конкретних управлінських рішень в різних режимах (оперативному, поточному тощо), автоматизоване збирання первинної (фактичної) інформації для її наступної обробки на ЕОМ організовується в ритмі виробництва.

Автоматизоване збирання єдиної первинної інформації в ритмі виробництва може бути здійснення за допомогою персональних ЕОМ, які використовуються як АРМ, на основі так званого бездокументного методу з послідуочим підтвердженням переданої до обробки інформації первинними документами, або такими, що їх замінюють. Технологія формування і оформлення в ритмі виробництва первинних показників, що характеризують масові процеси або явища, здійснюється автоматизовано завдяки попередньому створенню бібліотеки форм первинних і звітних (результатних) документів, що видаються на екран, або на паперовій носій, а також масивів з умовно-постійною (довідковою, нормативною, плановою, договірною), а інколи і розрахунковою інформацією.

Відомі два основні способи організації технології автоматизованого збирання єдиної первинної інформації у ритмі виробництва: 1) з попереднім автоматичним формуванням документа – заготовки (форма документа, що заповнена умовно-постійною інформацією) на екрані ПЕОМ, та наступним друкуванням заготовки на паперовому носії; 2) з послідуочим друкуванням первинного документа (або такого, що його замінює) на паперовому носії, форма якого спочатку автоматично формується і заповнюється умовно-постійною інформацією на екрані ПЕОМ.

Перший спосіб характеризується тим, що первісно з бібліотеки на екран ПЕОМ видається форма первинного документу, в яку автоматизовано з бази даних вписується відповідно умовно-постійна або розрахункова інформація. Заповнена таким чином форма перетворюється в документ-заготовку. Ця заготовка за допомогою принтера друкується на папері в потрібній кількості примірників, котрі передаються потім за призначенням. По мірі того, як здійснюється виробничо-господарська чи інша операція, фактична інформація про цю операцію проставляється (записується) від руки на папері у відповідну графу цієї форми. Потім цю інформацію також записують (за допомогою клавіатури дисплея) у відповідну графу цієї форми, що є на екрані дисплея. Одночасно ця інформація (у вигляді форми, первинного документу, тощо) вводиться до пам'яті ПЕОМ і стає здобутком (завдяки каналам зв'язку) відповідних користувачів.

Первинний документ, складений цим способом, створюється так би мовити в два заходи: первісно готується документ-заготовка на папері, а потім остаточно складається первинний документ. Наприклад, складений таким чином на папері у двох примірниках багаторядковий лімітний (лімітно-заборний) документ-заготовка передається до складу матеріальних цінностей і до цехової комори. Там ці документи-заготовки використовуються як основа при видаванні (одержанні) відповідних цінностей. Фактично видана (одержана) кількість матеріалів проставляється вручну завідуючим складом у відповідній графі цієї форми. Водночас з цим проставляються (за допомогою клавіш дисплея) фактичні показники в аналогічну форму цього документу, яка виведена на екран ПЕОМ, що встановлена на складі і використовується як АРМ зав. складом. Введену в пам'ять ПЕОМ інформацію можуть уже використовувати (завдяки каналам зв'язку) будь-які користувачі як в управлінні виробничо-господарською діяльністю, так і в обліку, аналізі тощо. Так само формується і заповнюється маршрутно-технологічна документація (лист або карта), що використовується при обліку виготовлюваної партії деталей або виробів, що збираються тощо.

Другий спосіб характеризується тим, що із бібліотеки форм на екран ПЕОМ, яка використовується також як АРМ, видається форма відповідного документа. Ця форма, завдяки багаторівневому меню, автоматизовано заповнюється відповідно умовно-постійною інформацією, що знаходиться у базі даних. Потім за допомогою клавіш та дисплею у відповідних графах і рядках цієї форми проставляються фактичні (кількісні та інші) реквізити. Після перевірки вірності уведених на екранну форму даних, принтер автоматично друкує на паперовому носії первинний документ. Уведені в пам'ять ПЕОМ дані можуть зараз же використовуватись (завдяки каналам зв'язку) відповідними користувачами. Наприклад, цей спосіб використовується для складання первинних документів: на матеріальні цінності, що надійшли від постачальників на склад підприємства (багаторядковий прибутковий ордер, акт про приймання матеріалів тощо); при видаванні готової продукції зі складів на сторону для реалізації (накази, накази-накладні, платіжні вимоги тощо). Отже, за цим способом формування первинного документа відбувається його побудова спочатку на екрані ПЕОМ, а потім на паперовому носії.

Переваги зазначених способів автоматизованого збирання фактичної інформації і формування первинних документів очевидні; вони не трудомісткі, оперативні, прості за виконанням тощо.

Автоматизація збирання первинної (фактичної) інформації дозволяє організувати її обробку як в оперативному, так і в поточному режимах.

Оперативна обробка первинної (фактичної) інформації на ПЕОМ, зібраної в ритмі виробництва, полягає в перетворенні її по спеціально розробленим для цих цілей алгоритмах зараз же після введення її в пам'ять ПЕОМ. Результати обробки звичайно видаються (відповідно до регламентуючої інформації) на екран ПЕОМ, яка встановлена на робочому місці користувача-фахівця, керівника, тощо, і використовуються для спостереження та контролю за виробництвом, коректування оперативних планів, проведення оперативного аналізу процесів та явищ, які відбуваються у виробничо-господарській діяльності об'єктів з наступним виробленням і прийняттям управлінських рішень тощо. Отже, ця інформація призначена в основному для оперативного управління і економічного аналізу.

Поточна обробка первинної (фактичної) інформації полягає в тому, що вона спочатку збирається, накопичується, а інколи і групується (разом з умовно-постійною інформацією), а потім обробляється по типових алгоритмах і видається на паперові носії у вигляді машинограм (таблиць), а інколи і на екран ПЕОМ, яка установлена на робочому місці користувача. Використовується ця інформація як бухгалтерські документи і довідки, а також для заповнення установлених форм бухгалтерської та статистичної звітності за рівні періоди часу, проведення комплексного економічного аналізу виробничо-господарської діяльності об'єкта управління.

Інформатика та вдосконалення форм і методів управління виробництвом

Всебічне використання сучасних електронних обчислювальних машин, периферійної техніки) засобів (каналів) зв'язку для збирання, обробки і передачі інформації користувачам істотно впливає на методи виконання функцій управління виробництвом, створює умови для більш оперативного і якісного їх здійснення. Все це сприяє удосконаленню управління об'єктом у цілому і його структурними ланками.

Відомо, що основою для прийняття управлінських рішень є інформація. Тому від своєчасного збирання достовірної інформації, що всебічно характеризує всі процеси і явища виробничо-господарської діяльності підприємства та його структурних ланок, а також від оперативної обробки і поставлення її користувачам повністю залежить (при належній компетентності керівника) якість і своєчасність керівництва виробничими процесами та господарською діяльністю підприємства.

Застосовуючи для досягнення цієї мети ПЕОМ там, де виникає первинна інформація (комора, склад, виробнича ділянка тощо), можна значно удосконалити форми і методи збирання цієї інформації за рахунок автоматизації процесу. Суть такого процесу полягає у тому, що завдяки попередньо сформованим умовно-постійним масивам (з нормативною, розцінковою, довідковою, планово-договірною та іншою умовно-постійною інформацією), а також бібліотеці форм вхідних і вихідних документів, можна форми первинних документів заповнювати автоматично тією інформацією, яка є для цього документа постійною (її до 90-95%).

Такий підхід до організації збирання фактичної інформації дозволяє здійснювати без паперове (на екран і в пам'ять ПЕОМ) одноразове оперативне збирання і фіксування достовірної первинної інформації та її попередню обробку у ритмі виробництва, а потім передавати цю інформацію у такому ж ритмі усім користувачам для використання.

Слід відмітити, що частота збирання первинної інформації, яка характеризує стан і параметри виробничих та господарських процесів та явищ, її оперативна обробка визначається частотою і швидкістю змін, що проходять на керованому об'єкті, тобто у ритмі виробництва. Наприклад, коли ті чи інші процеси плануються за годинним графіком, то при наявності негативних в цих процесах факторів необхідно збирати, обробляти і видавати відповідним керівникам оброблену інформацію про відхилення не пізніше, чим за кожні півгодини або три чверті години. Така інформація потрібна керівникові для оперативного втручання в ці процеси так швидко, щоб усунути наявні причини відхилень до їх закінчення. При такому способі інформування керівників можна сподіватись на те, що управління процесами буде своєчасним і ефективним. Зібрана таким чином оперативна інформація є не тільки єдиною для всіх користувачів, але і достовірною, а тому вона може бути використана і для господарського (бухгалтерського) обліку.

Оброблена і узагальнена або зведена по розроблених алгоритмах інформація на рівні підприємства використовується не тільки для вирішення загальновиробничих задач поточного управління, але і для складання установленої бухгалтерської, статистичної та іншої зведеної звітності.

Використовуючи широкі експлуатаційні можливості сучасних засобів обчислювальної техніки, систему прогресивних науково-обґрунтованих норм і нормативів, можна серйозно удосконалити планування виробництва і підняти його на науковий рівень. Можна, наприклад, промодельювати кілька варіантів, а потім вибрати оптимальний з відповідною організаційною структурою управління підприємством; пробалансувати систему відповідних показників та визначити напрямок пропорційного розвитку виробництва; здійснювати взаємозв'язок прогнозування, перспективного, поточного та оперативного планування; широко використовувати нормативні методи на всіх рівнях планування та управління; удосконалити систему нормативно-планових показників у використанні трудових, матеріальних, фінансових та інших ресурсів.

Особливо ефективним є застосування методів і засобів обробки інформації для удосконалення планування в часовому та пооб'єктному аспектах. Для цього розробляється система взаємно ув'язаних та збалансованих соціально-економічних показників для річного, квартальних (місячних), декадних (тижневих), добових (змінних) планів - часовий аспект, а для кожного робочого місяця, бригади, ділянки, так і для цехів і підприємства у цілому - пооб'єктний аспект. Крім того, установлюється єдиний крок для інформаційних сукупностей в обліку і плануванні, за допомогою якого є можливість оперативно і автоматизовано формувати дані для -аналізу.

Все це дозволяє значно удосконалити і підвищити рівень планування, а разом а тим поліпшити управління виробничими процесами шляхом взаємозв'язку сукупних показників в часовому та пооб'єктному аспектах. Крім того, є реальна можливість звільнити працівників, зайнятих розрахунками планових показників і складанням планів, від стомливої і малоефективної ручної праці.

Важливе значення має оперативне збирання і обробка показників, простота і доступність цих показників всім користувачам в часовому та пооб'єктному аспектах. Існуючі форми і методи обліку з його традиційним позадачним підходом при збиранні та обробці економічної інформації в нових умовах не підходять. На основі автоматизованої (діючої у ритмі виробництва) системи збирання і обробки економічної інформації доцільно організувати лише один - господарський облік замість трьох видів обліку (оперативного або оперативно-технічного, бухгалтерського і статистичного), які існують сьогодні на підприємствах. Це можливо тому, що: по-перше, використовується для нього тільки одне (загальне для усіх зазначених видів обліку) джерело фактичної (облікової) інформації - це виробничо-господарська та інша діяльність підприємства і його існуючі ресурси; по-друге, збирається ця фактична інформація одноразово, у ритмі виробництва, при єдиному технологічному процесі; збирається як достовірна, оброблюється (по різних алгоритмах) і записується у базу (бази) даних ПЕОМ у місцях виникнення інформації, а потім передається (також у ритмі виробництва) по каналах зв'язку усім іншим користувачам більш вищого рівня, які використовують цю інформацію (разом з іншою умовно-постійною) для складання (також на ПЕОМ) аналітичних та синтетичних реєстрів різного (у тому числі і бухгалтерського) призначення, прийняття управлінських рішень тощо.

Завдяки безперервному технологічному процесу обробки інформації можна забезпечити повну ув'язку даних оперативного, бухгалтерського і статистичного обліку в аналітичному та синтетичному (зведеному) аспектах, достовірність цих даних, яку завгодно ступінь деталізації або групування, за станом на певну дату або за який завгодно час чи період та інше.

В умовах обробки інформації на ПЕОМ відкриваються широкі перспективи для нової автоматизованої або електронної форми обліку. У методичному відношенні основу цієї форми обліку складає системний регламентний перелік виробничо-господарських та інших операцій, який найбільш повно відображає стан, процеси та явища, що проходять на об'єкті управління в той чи інший період часу. Економічний зв'язок зазначених операцій в аналітичному і синтетичному аспектах у базі (базах) даних вирішується відповідно до вимог системної обробки економічної інформації. Подвійний характер взаємозв'язаних змін на об'єктах обліку знаходить відображення в цільовому одноразовому записі кожної первинної операції. При цьому первинна (фактична) інформація записується у базу даних з мінімальним набором таких реквізитів, за допомогою яких (та інших додаткових постійних даних) можна одержувати максимум потрібних даних. Такий підхід орієнтований на однорівневе або багаторівневе системне збирання і обробку, зберігання і видання необхідних даних користувачам за допомогою сучасного комплексу обчислювальної та периферійної техніки.

Принципова відмінність нової форми обліку від діючих полягає у тому, що початок автоматизованого облікового процесу починається не з обробки первинних документів, а з одноразового (у ритмі виробництва) запису на магнітний носій (пам'ять ПЕОМ) всіх первинних (фактичних) даних, які виникають при виробничо-господарській та іншій діяльності. Ці дані (разом з іншими) обробляються, передаються по каналах зв'язку користувачам різних рівнів, а потім зберігаються певний час. Видача обробленої інформації користувачам усіх рівнів здійснюється у різних режимах, по різних формах і на різних носіях, за будь-який період часу і за станом на будь-яку дату, а також про різні об'єкти управління. При цьому, фактична інформація може видаватись після її обробки не тільки повністю і самостійно, але і в порівнянні із зведеними нормативними, плановими та іншими даними, із записаними відхиленнями і відносними величинами та іншими показниками. Така інформація може використовуватись при економічному аналізі та прийнятті управлінських рішень.

Маючи єдину базу даних, що включає нормативну і планову, довідкову, договірну та іншу постійну інформацію, а також фактичні показники (які збираються у ритмі виробництва) і установлений єдиний крок для інформаційних сукупностей в плануванні і обліку, можна автоматично і оперативно порівнювати планові (інколи і зведені нормативні) дані з фактичними в різних аспектах і отримувати відхилення (кількісні і вартісні) з відносними величинами цих відхилень. Інформація про відхилення потім використовується користувачами різних рівнів. Робляться й інші обчислення та порівняння. Треба указати, що записи (позиції), у яких планові і фактичні показники збігаються за тією чи іншою ознакою, як правило, користувачам можуть не видаватись.

Такий метод автоматичного контролю і економічного аналізу дозволяє не тільки звільнити персонал управління від ручного виконання технічних операцій,

але і підняти управління на більш високий ступінь, прискорити вироблення і прийняття оптимальних рішень.

У цих умовах є можливість робити пофакторний аналіз і таким чином встановлювати фактори, що впливають на виконання тих чи інших процесів і операцій. При розрахунках показників для оперативного і поточного аналізу використовуються також не тільки традиційні прийоми та засоби економічного аналізу (абсолютні, відносні і середні величини; методи порівняння, групування і ланцюгові підстановки; балансовий, індексний та інші методи), але і математичні методи (графічні, кореляційний і регресійний аналіз, лінійне і динамічне програмування, матричні методи імітації та ін.).

Організація економічного аналізу таким чином значно скорочує час не тільки для відбору необхідних для цієї мети даних і оперативного проведення аналізу, але й для прийняття управлінських рішень. При цьому, оброблена інформація про відхилення буде тією, що слугуватиме при організації управління за методом відхилень.

Автоматизована система збирання і обробки, в. також зберігання інформації у базі (базах) даних істотно впливає на методи складання установленної зведеної звітності. Існуючі типові форми бухгалтерської, статистичної та іншої звітності абсолютно не пристосовані для машинного заповнення; в них включається багато дублюючих, постійних і похідних показників. Тому ці форми слід замінити іншими, більш придатними для машинного заповнення, а також значно спрощеними. При цьому треба виходити а того, що складання звітності повинно здійснюватись автоматично не тільки на підприємствах, але і на всіх зовнішніх (вищестоящих, територіальних) рівнях.

Автоматично складати звітність на рівні підприємства можливо тому, що є, з одного боку, база даних, в якій фактична інформація може бути згрупованою в потрібних напрямках і узагальненою необхідними доповненнями; з другого боку, є масиви-довідники з кодами рахунків бухгалтерського обліку, які відповідають рядкам і графам типових форм діючої звітності. Крім того, існує масив бібліотеки типових форм, які можна видавати як на екран ПЕОМ, так і на паперовий носій. Одночасно з складанням звітності на підприємстві формують машинні носії з такою ж інформацією як і в звітності, яка потім передається по каналах зв'язку (або іншим способом) у зовнішні організації, де вона також автоматично зводиться у форми. При її зведенні ряд показників програмно контролюється.

Таким чином, організація системної обробки економічної інформації, застосування нових методів автоматизованого складання звітності на різних рівнях, а також заміна діючих типових форм звітності більш досконалими і придатними для машинного складання, при іншій їх кількості, змісту та зовнішньому вигляді приводять до повного усунення ручної праці фахівців управління.

Діючі форми і методи управління виробництвом на підприємствах для умов ринкової економіки не завжди ефективні. Для цих умов при системній обробці інформації і широкого застосування ПЕОМ треба створювати нові форми та більш раціональні методи управління підприємством. Одним з таких напрямків удосконалення форм управління може бути, наприклад, об'єктно-цільовий підхід із закінченим циклом управління при визначенні організаційної структури, нових прав і обов'язків, а також міри відповідальності за свої посадові функції керівників і фахівців різних рівнів.

Цільова об'єктно-функціональна структура апарату управління дозволить компетентне, комплексно, компактно і своєчасно здійснювати процеси управління, як це і потрібно при ринковій економіці. Цільовими об'єктами управління є ресурси: матеріальні, трудові, фінансові, а також основні засоби виробництва і готова продукція. Увесь процес управління тим чи іншим ресурсом розглядається як одне ціле, в рамках якого діють взаємозалежні інформаційні потоки, що проходять через усі функції управління (планування, облік, контроль тощо). Можуть бути й інші напрямки удосконалення форм управління.

Посадові права, обов'язки і міра відповідальності працівників апарату управління тієї чи іншої структури (ресурсу) розроблюються для нових умов функціонування, а тому вони будуть основою при формуванні регламентуючої інформації. При їх складанні слід враховувати те, що всі процеси по збиранню і обробці, зберіганню та виданню користувачам інформації, а також деякі логічні операції будуть виконуватись у системному порядку і автоматизовано.

Регламентуюча інформація повинна характеризувати у формалізованому вигляді ці нові посадові права, обов'язки керівників і фахівців даної структури, їх міру відповідальності за прийняте (або не прийняте) рішення а також включати відповіді на запитання кому, яка, в якому вигляді і в які строки повинна надходити після обробки необхідна для управління інформація. Програма-диспетчер забезпечить користувачів необхідною інформацією автоматично.

При системній обробці інформації створюються умови і для удосконалення методів управління виробництвом. Так наприклад, можливо буде застосувати метод управління по відхиленнях. Суть його полягає ось у чому. Коли виробничі процеси і господарські операції виконуються за встановленими завданнями, то управлінські впливи на ці процеси та операції звичайно не здійснюються, а тому інформація про них у цьому випадку керівникові даної ланки буде не потрібна. Коли інформація до нього буде все ж таки надходити, то буде створюватись інформаційне перевантаження. У цьому випадку відсутність інформації у керівника буде свідчити про те, вір справи ідуть за заздальгідь розробленим планом або сценарієм.

Інформація буде надходити на екран ПЕОМ даного керівника лише в тих випадках, коли в указаних процесах або операціях будуть відхилення (які відносяться до компетенції даного керівника) фактичних показників від заданих, або іншим чином встановлених. При цьому інформація буде надходити тільки про відхилення від заданих параметрів і то лише по тих керованих об'єктах, в яких є наявні відхилення. Інформацію про відхилення будуть супроводжувати деякі специфічні ознаки: питома вага відхилень, причина і винуватець їх тощо. Маючи елементи порівняльного аналізу, цю інформацію можна використати для вироблення, а потім і прийняття управлінського рішення.

Завдяки організації управління за методом відхилень значно умонтуються обсяги інформації, які надходять керівникові або фахівцю, і прискорюються строки підготовки та вироблення того чи іншого управлінського рішення.

Розглянуті форми і методи удосконалення управління виробництвом на базі комплексів ПЕОМ створюють основу для організації систематизованого інформування (тобто забезпечення всією необхідною цільовою інформацією користувача) керівників і фахівців різних рівнів при без паперовій технології (новій інформаційній технології). Без паперовий спосіб забезпечення користувачів необхідними цільовими даними у різних їх угрупованнях і комбінаціях, за будь-який період часу або на будь-яку

дату і в розрізі різних об'єктів управління ґрунтується, з одного боку, на функціонуванні автоматизованого банку (банків) даних, що поповнюється інформацією у ритмі виробництва, а з другого боку, на телеобробці даних.

Таким чином, організація автоматизованої системи збирання і обробки економічної інформації сприяє побудові раціональної структури апарату управління підприємством і більш ефективному його використанню для виробничих цілей. Функціонування зазначеної системи створює умови для підвищення культури управління, а також звільняє керівників і фахівців різних рівнів від ручної праці по збиранню і обробці інформації, скорочує строку вироблювання і прийняття управлінських рішень, поліпшує їх якість тощо.

Інформатика та удосконалення форм і методів обліку

В умовах ринкової економіки значення обліку на підприємствах важко переоцінити. Тут важливе значення має оперативне збирання і обробка показників, простота і доступність цих показників всім користувачам в часовому та пооб'єктному аспектах. Тому існуючі форми і методи обліку з його традиційним позадачним підходом при збиранні та обробці економічної інформації в нових умовах не підходять. Крім того, на основі автоматизованої (діючої у ритмі виробництва) системи збирання і обробки економічної інформації доцільно організувати лише один - господарський облік замість трьох видів обліку (оперативного або оперативно-технічного, бухгалтерського і статистичного), які існують сьогодні на підприємствах. Це можливо тому, що: по-перше, використовується для нього тільки одне (загальне для усіх зазначених видів обліку) джерело фактичної (облікової) інформації - це виробничо-господарська та інша діяльність підприємства і його існуючі ресурси; по-друге, збирається ця фактична інформація одноразово, у ритмі виробництва, при єдиному технологічному процесі; збирається як достовірна, оброблюється (по різних алгоритмах) і записується у базу (бази) даних ПЕОН (АРМ), поставленої у місцях виникнення інформації, а потім передається (також у ритмі виробництва) по каналах зв'язку усім іншим користувачам більш вищого рівня, які використовують цю інформацію (разом з іншою умовно-постійною) для складання (також на ПЕОН) аналітичних та синтетичних реєстрів різного (у тому числі і бухгалтерського) призначення, прийняття управлінських рішень тощо.

Таким чином, завдяки безперервному технологічному процесу (всій первинній інформації, яка збирається одноразово і записується у базу даних), можна забезпечити повну ув'язку даних оперативного, бухгалтерського і статистичного обліку в аналітичному та синтетичному (зведеному) аспектах, достовірність цих даних, яку завгодно ступінь деталізації або групування, за станом на певну дату або за який завгодно час чи період та інше.

Система обробки економічної інформації не сумісна а формами, які існують у бухгалтерському обліку при ручній або напівручній обробці інформації. Ці форми не можна заповнювати машинним способом при безперервному процесі. Крім того, в них є ряд показників, як і багаторазово повторюються тощо.

В умовах, коли ПЕОМ використовується як АРМ і таким чином забезпечується у ритмі виробництва збирання і обробка (при єдиному процесі) всієї первинної інформації в місцях її виникнення, а потім передача її по каналах зв'язку усім користувачам різних рівнів, відкриваються широкі перспективи для нової

автоматизованої або електронної форми обліку. У методичному відношенні основу цієї форми обліку складає системний регламентний перелік виробничо-господарських та інших операцій, який найбільш повно відображає стан, процеси та явища, що проходять на об'єкті управління в той чи інший період часу. економічний зв'язок зазначених операцій в аналітичному і синтетичному аспектах у базі (базах) даних вирішується відповідно до вимог системної обробки економічної інформації. Подвійний характер взаємозв'язаних змін на об'єктах обліку знаходить відображення в цільовому одноразовому записі кожної первинної операції. При цьому первинна (фактична) інформація записується у базу даних з мінімальним набором таких реквізитів, за допомогою яких (та інших додаткових постійних даних) можна одержувати максимум потрібних даних. Такий підхід орієнтований на однорівневе або багаторівневе системне збирання і обробку, зберігання і видання необхідних даних користувачам за допомогою сучасного комплексу обчислювальної та периферійної техніки.

Слід відзначити, що в умовах системного збирання і обробки інформації на ПЕОМ (АРМ) принципова відмінність нової форми обліку від діючих полягає у тому, що початок автоматизованого облікового процесу починається не з обробки первинних документів, а з одноразового (у ритмі виробництва) запису на магнітний носій (пам'ять ПЕОМ) всіх первинних (фактичних) даних, які виникають при виробничо-господарській та іншій діяльності. Ці дані (разом з іншими постійними) обробляються, передаються по каналах зв'язку користувачам різних рівнів, а потім зберігаються певний час. Видача обробленої інформації користувачам усіх рівнів здійснюється у різних режимах, по різних формах і на різних носіях, за будь-який період часу і за станом на будь-яку дату, а також про різні об'єкти управління. При цьому, фактична інформація може видаватись після її обробки не тільки повністю і самостійно, але і в порівнянні із зведеними нормативними, плановими та іншими даними, із записаними відхиленнями і відносними величинами та іншими показниками. Така інформація може використовуватись при економічному аналізі та прийнятті управлінських рішень.

Інформатика та удосконалення форм і методів контролю і аналізу

Маючи єдину базу даних, що включає нормативну і планову, довідкову, договірну та іншу постійну інформацію, а також фактичні показники (які збираються у ритмі виробництва) і установлений єдиний крок для інформаційних сукупностей в плануванні і обліку, можна автоматично і оперативно порівнювати планові (інколи і зведені нормативні) дані з фактичними в різних аспектах і отримувати відхилення (кількісні і вартісні) з відносними величинами цих відхилень. Інформація про відхилення потім використовується користувачами різних рівнів. Робляться й інші обчислення та порівняння. Треба указати, що записи (позиції), у яких планові і фактичні показники збігаються за тією чи іншою ознакою, як правило, користувачам можуть не видаватись.

Такий метод автоматичного контролю і економічного аналізу дозволяє не тільки звільнити персонал управління від ручного виконання технічних операцій, але і підняти управління на більш високий ступінь, прискорити вироблення і прийняття оптимальних рішень тощо.

У цих умовах є можливість робити пофакторний аналіз і таким чином встановлювати фактори, що впливають на виконання тих чи інших процесів і

операцій. При розрахунках показників для оперативного і поточного аналізу використовуються також не тільки традиційні прийоми та засоби економічного аналізу (абсолютні, відносні і середні величини; методи порівняння, групування і ланцюгові підстановки; балансовий, індексний та інші методи), але і математичні методи (графічні, кореляційний і регресійний аналіз, лінійне і динамічне програмування, матричні. методи імітації та ін.).

Організація економічного аналізу таким чином значно скорочує час не тільки для відбору необхідних для цієї мети даних і оперативного проведення аналізу, але й для прийняття управлінських рішень. При цьому, оброблена інформація про відхилення буде тією, що слугуватиме при організації ' управління за методом відхилень.

Інформатика та удосконалення методів складання зведеної звітності

Автоматизована система збирання і обробки, а також зберігання інформації у базі даних істотно впливає на методи складання установленої зведеної звітності. Існуючі типові форми бухгалтерської, статистичної та іншої звітності абсолютно не пристосовані для машинного заповнення; в них включається багато дублюючих, постійних і похідних показників. Тому ці форми слід замінити іншими, більш придатними для машинного заповнення, а також значно спрощеними. При цьому треба виходити а того, що складання звітності повинно здійснюватись автоматично не тільки на підприємствах, але і на всіх зовнішніх (вищестоящих, територіальних) рівнях.

Автоматично складати звітність на рівні підприємства можливо тому, що є, з одного боку, база даних, в якій фактична інформація може бути згрупованою в потрібних напрямках і узагальненою необхідними доповненнями; з другого боку, є масиви-довідники з кодами рахунків бухгалтерського обліку, які відповідають рядкам і графам типових форм діючої звітності. Крім того, існує масив бібліотеки типових форм, які можна видавати як на екран ПЕОМ, так і на паперовий носій. Одночасно з складанням звітності на підприємстві формують машинні носії з такою ж інформацією як і в звітності, яка потім передається по каналах зв'язку (або іншим способом) у зовнішні організації, де вона також автоматично зводиться у форми.

Таким чином, організація системної обробки економічної інформації, застосування нових методів автоматизованого складання звітності на різних рівнях, а також заміна діючих типових форм звітності більш досконалими і придатними для машинного складання, при іншій їх кількості, змісту та зовнішньому вигляді приводять до повного усунення ручної праці фахівців управління.

Інформатика та удосконалення форм і методів управління

Діючі форми і методи управління виробництвом на підприємствах для умов ринкової економіки не завжди ефективні. Для цих умов при системній обробці інформації і широкого застосування ПЕОМ як АРМ користувачів різних рівнів треба створювати нові форми та більш раціональні методи управління підприємством. Одним з таких напрямків удосконалення форм управління може бути, наприклад, об'єктно-цільовий підхід із закінченим циклом управління при визначенні організаційної структури, нових прав і обов'язків, а також міри відповідальності за свої посадові функції керівників і фахівців різних рівнів.

Цільова об'єктно-функціональна структура апарату управління дозволить компетентне, комплексно, компактно і своєчасно здійснювати процеси управління, як це і потрібно при ринковій економіці. Цільовими об'єктами управління є

ресурси: матеріальні, трудові, фінансові, а також основні засоби виробництва і готова продукція. Увесь процес управління тим чи іншим ресурсом розглядається як одне ціле, в рамках якого діють взаємозалежні інформаційні потоки, що проходять через усі функції управління (планування, облік, контроль тощо). Можуть бути й інші напрямки удосконалення форм управління.

Посадові права, обов'язки і міра відповідальності працівників апарату управління тієї чи іншої структури (ресурсу) розроблюються для нових умов функціонування, а тому вони будуть основою при формуванні регламентуючої інформації. При їх складанні слід враховувати те, що всі процеси по збиранню і обробці, зберіганню та виданню користувачам інформації, а також деякі логічні операції будуть виконуватись у системному порядку і автоматизовано.

Регламентуюча інформація повинна характеризувати у формалізованому вигляді ці нові посадові права, обов'язки керівників і фахівців даної структури, їх міру відповідальності за прийняте (або не прийняте) рішення а також включати відповіді на запитання кому, яка, в якому вигляді і в які строки повинна надходити після обробки необхідна для управління інформація. Програма-диспетчер забезпечить користувачів необхідною інформацією автоматично.

При системній обробці інформації створюються умови і для удосконалення методів управління виробництвом. Так наприклад, можливо буде застосувати метод управління по відхиленнях. Суть його полягає ось у чому. Коли виробничі процеси і господарські операції виконуються за встановленими завданнями, то управлінські впливи на ці процеси та операції звичайно не здійснюються, а тому інформація про них у цьому випадку керівникові даної ланки буде не потрібна. Коли інформація до нього буде все ж таки надходити, то буде створюватись інформаційне перевантаження. У цьому випадку відсутність інформації у керівника буде свідчити про те, що справи ідуть за задалегідь розробленим планом або сценарієм.

Інформація буде надходити на екран ПЕОМ даного керівника лише в тих випадках, коли в указаних процесах або операціях будуть відхилення (які відносяться до компетенції даного керівника) фактичних показників від заданих, або іншим чином встановлених. При цьому інформація буде надходити тільки про відхилення від заданих параметрів і то лише по тих керованих об'єктах, в яких є наявні відхилення. Інформацію про відхилення будуть супроводжувати деякі специфічні ознаки: питома вага відхилень, причина і винуватець їх тощо. Маючи елементи порівняльного аналізу, цю інформацію можна використати для вироблення сценарію поведінки у даній ситуації, а потім і прийняття управлінського рішення.

Завдяки організації управління за методом відхилень значно зменшуються обсяги інформації, які надходять керівникові або фахівцю, і прискорюються строки підготовки та вироблення того чи іншого управлінського рішення.

Методичні основи автоматизації процесів збирання і обробки економічної інформації

Автоматизована система збирання і обробки економічної інформації починається з одноразового виміру первинної інформації, яка характеризує процеси та явища, що відбуваються у виробничо-господарській та іншій діяльності об'єкта управління і його складових ланках. Ця інформація фіксується в пам'яті ПЕОМ і негайно обробляється по розроблених алгоритмах, а потім використовується користувачами цієї ПЕОМ для виконання своїх посадових

функцій. Для централізованої обробки і зведення по підприємству цілому зазначену інформацію збирають по каналах зв'язку.

Ефективність автоматизованої системи збирання і обробки економічної інформації в першу чергу залежить від застосованих методів і засобів по вимірюванню, фіксації і передачі інформації на обробку, а також від своєчасної її обробки і передавання користувачам різних рівнів. Практика свідчить, що традиційно існуючі методи збирання інформації не забезпечують оперативність і якість управління тому, що дуже багато втрачається часу на таке збирання, а також занадто багато застосовується ручної праці. Наприклад, широко розповсюджений варіант формування фактичної інформації в первинний документ, який заповнюється в ручну, має негативні риси: він тривалий час заповнюється, підписується посадовими особами, а потім через кур'єра передається до обчислювального центру для обробки.

Затримка первинної (фактичної) інформації до її обробки на ЕОМ відбувається тому, що дуже важко і довго документально (юридично) оформляються первинні документи, на яких заснований бухгалтерський облік, а також занадто неудоконалені способи передачі цих документів на обробку. Тому в теорії і на практиці існує не тільки бухгалтерський облік, але й оперативний (оперативно-технічний) та статистичний облік, які і забезпечують в основному необхідною інформацією оперативне управління виробництвом.

Важливим елементом методичного удосконалення обліку є зближення планових і облікових одиниць вимірювання показники, що характеризують об'єкти управління. Такий підхід дозволить не тільки організувати автоматизоване збирання інформації за допомогою ЕОМ, але потім і порівнювати планові і облікові (а інколи і нормативні) показники по відповідних об'єктах і процесах у ритмі виробництва, тобто одержувати відхилення і використовувати їх як обліку, так і для прийняття оперативних управлінських рішень. Крім того, можна використовувати планові показники, що є у базі даних, як облікові. Це означає, що можна збирати як фактичні показники тільки відхилення їх від планових, додаючи при необхідності ряд ознак, що характеризують виконавців, а також причини та винуватців відхиляємих величин. Все це значно скорочує обсяги інформації, що збирається і оброблюється, виключить її дублювання, зменшує трудомісткість інформаційного процесу. Тільки у такому випадку можна чекати найбільшого економічного ефекту від застосування сучасного комплексу технічних засобів для системного збирання та обробки економічної інформації.

Щоб первинна (один раз зібрана) інформація, що характеризує фактичні процеси та явища, які проходять на об'єктах управління, після обробки і на ЕОМ використовувалась персоналом управління не тільки для обліку, аналізу та складання установленної звітності, а також і для вироблення конкретних управлінських рішень в різних режимах (оперативному, поточному тощо), потрібно організувати в ритмі виробництва автоматизоване збирання первинної (фактичної) інформації для її наступної обробки на ЕОМ в такому ж режимі.

Автоматизоване збирання єдиної первинної інформації в ритмі виробництва може бути здійснення за допомогою, наприклад, персональних ЕОМ, які використовуються як АРМ, на основі так званого бездокументного методу з послідовним підтвердженням переданої до обробки інформації первинними документами, або такими, що їх замінюють (наприклад, щоденні накопичувальні

відомості). Технологія формування і оформлення в ритмі виробництва первинних показників, що характеризують масові процеси або явища, здійснюється автоматизовано завдяки попередньому створенню бібліотеки форм первинних і звітних (результатних) документів, що видаються на екран, або на паперовій носії, а також масивів з умовно – постійною (довідковою, нормативною, плановою, договірною та іншою), а інколи і розрахунковою інформацією.

Відомі два основні способи організації технології автоматизованого збирання єдиної первинної інформації у ритмі виробництва: 1) з попереднім автоматичним формуванням документа – заготовки (форма документа, що заповнена умовно–постійною інформацією) на екрані ПЕОМ, та наступним друкуванням заготовки на паперовому носії; 2) з послідовним друкуванням первинного документа (або такого що його замінює) на паперовому носії, форма якого спочатку автоматично формується і заповнюється умовно – постійною інформацією на екрані ПЕОМ.

Перший спосіб характеризується тим, що первісно з бібліотеки на екран ПЕОМ видається форма первинного документа, в яку автоматизовано з бази даних вписується відповідно умовно-постійна або розрахункова інформація. Заповнена таким чином форма перетворюється в документ-заготовку. Ця заготовка за допомогою принтера друкується на папері в потрібній кількості примірників, котрі передаються потім за призначенням. По мірі того, як здійснюється виробничо-господарська чи інша операція, фактична інформація про цю операцію проставляється (записується) від руки на папері у відповідну графу цієї форми. Потім цю інформацію також записують (за допомогою клавіатури дисплея) у відповідну графу цієї форми, що є на екрані дисплея. Одночасно ця інформація (у вигляді форми, первинного документа, тощо) вводиться до пам'яті ПЕОМ і стає здобутком (завдяки каналам зв'язку) відповідних користувачів. І так далі до повного заповнення форми первинного документа фактичною інформацією.

Отже первинний документ, складений цим способом, створюється так би мовити в два заходи: первісно готується документ-заготовка на папері, а потім остаточно складається первинний документ. Наприклад, складений таким чином на папері у двох примірниках багаторядковий лімітний (лімітно-заборний) документ-заготовка передається до складу матеріальних цінностей і до цехової комори. Там ці документи-заготовки використовуються як основа при видаванні (одержанні) відповідних цінностей. Фактично видана (одержана) кількість матеріалів проставляється вручну завідуючим складом у відповідній графі цієї форми. Водночас з цим проставляються (за допомогою клавіш дисплея) фактичні показники в аналогічну форму цього документа, яка виведена на екран ПЕОМ, що встановлена на складі і використовується як АРМ зав. складом. Введену в пам'ять ПЕОМ інформацію можуть уже використовувати (завдяки каналам зв'язку) будь-які користувачі як в управлінні виробничо-господарською діяльністю, так і в обліку, аналізі тощо. Так само формується і заповнюється маршрутно-технологічна документація (лист або карта), що використовується при обліку виготовляємої партії деталей або виробів, що збираються тощо.

Другий спосіб характеризується тим, що із бібліотеки форм на екран ПЕОМ, яка використовується також як АРМ, видається форма відповідного документа, ця форма, завдяки серії багаторівневого меню, автоматизовано заповнюється відповідно умовно-постійною інформацією, що знаходиться у базі даних. Потім за допомогою клавіш дисплея у відповідних графах і рядках цієї форми

проставляються фактичні (кількісні та інші) реквізити. Після перевірки вірності уведених на екранну форму даних, принтер автоматично друкує на паперовому носії первинний документ. Уведені в пам'ять ПЕОМ дані можуть зараз же використовуватись (завдяки каналам зв'язку) відповідними користувачами.

Переваги зазначених способів автоматизованого збирання фактичної інформації і формування первинних документів очевидні; вони не трудомісткі, оперативні, прості за виконанням тощо.

Автоматизація збирання первинної (фактичної) інформації дозволяє організувати її обробку як в оперативному, так і в поточному режимах.

Оперативна обробка первинної (фактичної) інформації на ПЕОМ, зібраної в ритмі виробництва, полягає в перетворенні її по спеціально розробленим для цих цілей алгоритмах зараз же після введення її в пам'ять ПЕОМ. Результати обробки звичайно видаються (відповідно до регламентуючої інформації) на екран ПЕОМ, яка встановлена на робочому місці користувача-фахівця, керівника, тощо, і використовуються для спостереження та контролю за виробництвом, коректування оперативних планів, проведення оперативного аналізу процесів та явищ, які відбуваються у виробничо-господарській діяльності об'єктів з наступним виробленням і прийняттям управлінських рішень тощо. Отже, ця інформація призначена в основному для оперативного управління і економічного аналізу.

Поточна обробка первинної (фактичної) інформації полягає в тому, що вона спочатку збирається, накопичується, а інколи і групується (разом з умовно-постійною інформацією), а потім обробляється по типових алгоритмах і видається на паперові носи у вигляді машинограм (таблиць), а інколи і на екран ПЕОМ, яка встановлена на робочому місці користувача. Використовується ця інформація як бухгалтерські введення (документи) і довідки, а також для заповнення установлених форм бухгалтерської та статистичної звітності за рівні періоди часу, проведення комплексного економічного аналізу виробничо-господарської та іншої діяльності як об'єкта управління в цілому, так і його складових частин. Отже, ця інформація використовується для господарського обліку, складання звітності, та інших поточних справ.

Методичні основи організації економічного аналізу

Підсистема економічного аналізу і прийняття управлінських рішень відноситься до групи функціональних підсистем управління виробництвом. Основне призначення цієї підсистеми - організація системної обробки інформації на ПЕОМ з метою використання її для одержання і всебічної оцінки результатів виробничо-господарської і іншої діяльності підприємства, а потім і прийняття рішень по управлінню виробництвом, випуском, відвантаженням і реалізацією готової продукції, основними засобами, матеріальними і енергетичними ресурсами, трудовими ресурсами, фінансовими (у тому числі затрати на виробництво і калькулювання собівартості продукції) ресурсами по всіх видах діяльності підприємства.

Наявність засобів і методів, які застосовуються для системного аналізу в даний період часу, дозволяють всі розрахунки, які виконуються в цій підсистемі, поділити на два основні класи.

До першого класу відносяться розрахунки, які виконуються у ритмі виробництва, а одержані результати цих розрахунків використовуються в оперативному економічному аналізі і при прийнятті оперативних управлінських

рішень. У цих випадках результати розрахунків видаються на екрани ПЕОМ, які установлені на робочих місцях користувачів – керівників і фахівців різних рівнів. При цьому, на екран звичайно видається інформація тільки про відхилення фактичних показників від планових (договірних, нормативних тощо). Ця інформація видається лише по тих позиціях, де ці відхилення формуються. Крім цього, у цих позиціях зазначаються також реквізити, які характеризують питому вагу цих відхилень, причини відхилень і їх винуватців. В тих же випадках, коли показники порівняльних позицій збігаються, ці позиції взагалі на екран не видаються, через те, вони будуть створювати інформаційний надлишок. Це значить, що відсутність інформації свідчить про задовільне положення справ, процесів тощо.

До другого класу відносяться розрахунки, які виконуються звичайним порядком за більш довгий час – місяць, квартал тощо. У цих випадках результати обробки інформації для економічного аналізу виробничо-господарської та іншої діяльності можна видавати користувачам як на екран ПЕОМ, так і на паперовий носій у вигляді машинограм (таблиць). При цьому, екран ПЕОМ застосовується лише тоді, коли такий аналіз вибіркоким методом і для цього використовується невелика кількість показників. При суцільному економічному аналізі необхідні дані, як правило, виводяться на паперовий носій.

Ринкові відношення змушують керівників і фахівців підприємства значно розширювати межі проведення економічного аналізу як за часом, і у пооб'єктному напрямку. Першочергового значення набуває контроль за рівнем цін на товари, особливо на ті з них, які користуються широким попитом, або виробляються на підприємстві. Крім того, треба оперативно контролювати виконання договорів на постачання готової продукції і покупцям, а також слідкувати за постачанням постачальниками матеріальних цінностей для виробництва, грошовим станом на рахунках у банках тощо.

У звітний період виконуються також і традиційні розрахунки для економічного аналізу. Це аналіз: ходу виконання цільових завдань (по кінцевому продукту) госпрозрахунковими або орендними бригадами, ділянками цехами; використання засобів праці (виробничого устаткування) і їх ремонту; використання предметів праці як у виробництві, так і в наявності їх у запасі (наднормативні залишки); стану і руху кадрів; використання робочого часу, дотримання трудової дисципліни робітниками і службовцями; як виконуються норми виробітку робітниками-відрядниками; виробництва, випуску, відвантаження готової продукції; затрат на виробництво і собівартість продукції; оборотності оборотних коштів; прибутків і рентабельності виробництва; фінансового стану, а також визначення загальної оцінки виробничо-господарської та іншої діяльності підприємства і його складових частин, утворення і використання фондів економічного стимулювання та інших резервів.

Таким чином, узагальнюючи вищезазначене, слід виділити такі основні комплекси задач, що розв'язуються підсистемою по обробці даних за різні періоди часу або за станом на будь-яку дату для економічного аналізу:

- виконання плану виробництва по випуску деталей, складальних з'єднань, готових виробів, запасних частин та інструменту, а також виконаних робіт і послуг за межі виробництва як дільницями і цехами, так і підприємством у цілому у кількісних і вартісних показниках;
- виконання договірних поставок готових виробів, виконання робіт і послуг, а також розрахунків за цю продукцію;
- наявність браку деталей, складальних з'єднань і готових виробів у різних аспектах;

- динаміки незавершеного виробництва деталей, складальних з'єднань і готових виробів;
- становища фактичних запасів готової продукції на складах, а також продукції, яка довгий час на складі знаходиться без руху;
- використання основних засобів, у тому числі виробничого устаткування, а також даних про хід капітального ремонту і будівництва об'єктів;
- забезпеченість основним виробничим устаткуванням процесів виготовлення готової продукції, яка передбачена планами виробництва;
- виконання договірних поставок постачальниками матеріальних цінностей, а також розрахунки з ними за ці поставки;
- використання матеріальних цінностей у виробництві, їх доля витрат у виготовленні готової продукції, а також динаміки витрат при заготовленні цих матеріалів;
- становище фактичних запасів сировини, матеріалів, покупних напівфабрикатів і комплектуючих виробів, тари, палива, запасних частин, спецодягу і спецвзуття, інструменту, пристроїв та інше;
- динаміку руху, наявність складу працівників, та забезпеченість кадрами виробництва у різних аспектах;
- використання робочого часу, порушення трудової дисципліни у різних аспектах;
- використання фонду заробітної плати, виконання норм виробітку робітниками-відрядниками, а також доля заробітної плати у затратах при виготовленні готової продукції;
- використання коштів, кредитів та позик банків, даних про хід розрахунків з постачальниками і покупцями тощо;
- формування затрат на виробництво і калькулювання собівартості продукції в різних аспектах;
- формування різних фондів та прибутків і їх використання, а також показник результатів виробничо-господарської та іншої діяльності підприємства;
- формування даних для інформування керівників і фахівців різних рівнів про результати проведеного економічного аналізу виробничо-господарської та іншої діяльності структурних підрозділів та підприємства в цілому за різні періоди часу.

Розрахунки для економічного аналізу виконуються не тільки за визначеним графіком часом, але і за запитом користувача за будь-який період часу як по підприємству в цілому, так і по його структурних підрозділах. При виконанні цих розрахунків використовуються як традиційні способи і прийоми економічного аналізу (абсолютні, відносні і середні величини; методи порівняння, групування ланцюгових підстановок; балансовий, індексний та інші методи), так і математичні методи економічного аналізу (графічні методи, кореляційний і регресійний аналіз, лінійне і динамічне програмування, матричні методи, методи моделювання і імітації тощо).

Для розрахунків, які виконуються при економічному аналізі, використовуються, у першу чергу, дані оперативних масивів, що формуються при збиранні первинної (фактичної) інформації для господарського обліку. Використовується і умовно-постійна інформація – нормативна, планова, договірна, довідкова, а також та, що нагромаджена з початку року, фактична інформація, яка сформована у базу (бази) даних загального призначення. Крім того, використовується і похідна інформація, яка одержана при складанні установленої зведеної звітності на підприємстві.

РОЗДІЛ 3

СИСТЕМНЕ ЗАБЕЗПЕЧЕННЯ ЕКОНОМІЧНИХ ПРОЦЕСІВ

Сучасне уявлення про економіку як науку про суспільне чи приватне господарство, способи його ведення, відносини між людьми у процесі виробництва й обміну товарів та послуг, закономірності протікання господарчих процесів, адаптоване до принципів системного підходу, дозволяє виокремлювати серед розмаїття притаманного їй понятійного тезаурусу поняття «економічна система» з характерними для такої системи «економічними процесами», отже – сукупністю елементів, функціонування яких забезпечує протікання економічних процесів та функціонування економічної системи як цілісного утворення з них.

У такому розуміння економіка як складна система є підсистемою суспільства з притаманними йому суспільними відносинами, складаючи, в свою чергу, єдність двох своїх підсистем – виробничої та невиробничої, які взаємодіють між собою. Виробничо-технологічні зв'язки між ними утворюють виробничо-технологічну структуру економічної системи, функціонування якої, здійснюване у формі протікання економічних процесів, забезпечується відповідною організаційно-господарською структурою.

Цілеспрямованість протікання економічних процесів та притаманні суспільним відношенням вимоги щодо їх ефективності об'єктивно приводять до необхідності системного забезпечення умов, необхідних для цього. З точки зору інформації, яка супроводжує такі процеси у формі вхідних, проміжних, вихідних, нормативних, довідникових даних, керівних вказівок, наказів, інструкцій, розпоряджень тощо, це означає, що забезпечення економічних процесів інформаційними даними повинно здійснюватися точно (безпомилково), швидко (своєчасно), повноцінно (у повному обсязі). Постільки функції точності, швидкості, повноцінності все більшою мірою покладаються на комп'ютерну та іншу офісну техніку, керовану спеціально створеними для неї програмами, системність забезпечення економічних процесів вимагає знань як апаратної частини такої техніки, так і її програмного забезпечення.

Технічне забезпечення обробки інформації про економічні процеси

Уявлення про варіативність моделей обчислювальної техніки на сучасному етапі її розвитку дає таблиця 1, де ЕОМ класифіковані за їх габаритними характеристиками.

Таблиця 1

Клас ЕОМ	Основне призначення	Основні технічні дані	Приклади моделей чи виробників
Супер-ЕОМ	Складні наукові обчислення	Інтегральна швидкодія - до десятків мільярдів операцій за секунду; до 100 процесорів, які працюють паралельно	Cray 3, Cray 4, VAX-1000, MULTICON, Cyber 205, Fujitsu VP2000
Великі ЕОМ (мейнфрейми)	Оброблення великих обсягів інформації в банках, на великих підприємствах	Мультипроцесорна архітектура; підключення до 200 робочих місць	IBM360, IBM370, IBM ES/9000, EC1068, EC1170
Супер-міні-ЕОМ	Системи управління підприємствами;	Мультипроцесорна архітектура; підключення до 200 терміналів;	Сімейство VAX, SPARC (SUN Microsystems),





	багатопультові обчислювальні системи	дисківі запам'ятовуючі пристрої, що нараховуються до кількох терабайтів	AS/400 (IBM)
Міні-ЕОМ	Системи управління підприємствами середнього розміру; багатопультові обчислювальні системи	Однопроцесорна архітектура, розгалужена периферія	ES/9370 (IBM), серії А та 2200 (Unisys)
Робочі станції	САПР, системи автоматизації експериментів, комп'ютерна графіка, відеомонтаж	Одно- та двопроцесорна архітектура, висока швидкодія, ОЗП - 128 Мбт і більше; спеціалізована периферія	MERVA-2 (IBM), Sun UltraSPARC, NeXT
Мікро-ЕОМ (персональні)	Індивідуальне обслуговування користувача; робота в локальних автоматизованих системах управління	Однопроцесорна архітектура, гнучкість конфігурації, можливість підключення різноманітних зовнішніх пристроїв	Широкий перелік моделей і виробників

Класифікацію персональних ЕОМ за їх класами наведено у таблиці 2:

Таблиця 2

Клас персональних ЕОМ	Маса, кг	Джерело живлення	Примітки
Настільні (desktop)	5 - 10 (без монітора)	Побутова електромережа	Використовуються в приміщеннях для обладнання робочих місць; забезпечують широкі функціональні можливості
Переносні (laptop)	2,5 - 5	Побутова електромережа або акумулятори	Призначені для використання в поїздах. Забезпечують широкі функціональні можливості, зокрема, підключення до обчислювальних мереж. Останнім часом цей клас ЕОМ все більше тяжіє до блокнотних ЕОМ
Блокнотні (notebook)	0,7 - 2,5	Акумулятори або перетворювачі напруги	Призначені для використання в поїздах. Забезпечують широкі функціональні можливості, зокрема, підключення різноманітної периферії та підключення до обчислювальних мереж
Електронний записник (PDA - Personal Digital Assistant)	менше 0,7	Батареї або перетворювачі напруги	Зміщається у кишені. Набір функцій дає можливість записувати тексти, виконувати обчислення, використовувати довідники та перекладачі

Поняття базового технічного комплексу комп'ютерної техніки

Базовий комплект комп'ютерної техніки складають: процесорний блок , монітор , клавіатура  і миша , з'єднані між собою так, що в процесі функціонування являють собою єдине ціле (власне комп'ютер):



Основні характеристики компонентів процесорного блоку:

1	Тип процесора	Intel (Celeron, Athlon, Duron, інші)
2	Тактова частота	1000 - 2000 Меггерц (МГц)
2	Об'єм оперативної пам'яті (RAM)	128 - 512 Мегабайт (Мб)
3	Місткість жорсткого диску (HDD)	40 - 80 Гігабайт (Гб)
4	Об'єм відеопам'яті (VRAM)	64 - 256 Мегабайт (Мб)

Примітка: комп'ютерна техніка розвивається швидко, тому наведені значення характеристик компонентів процесорного блоку слід розуміти як орієнтовні, які постійно змінюються (в сторону збільшення).

У спеціальній літературі та у прайс-листах продавців комп'ютерної техніки згадані (та інші) характеристики представлені записами типу:

Cel 1,7 / 256 RAM / 20,0 HDD / 128 VRAM,

які слід розуміти так: *ідеться про процесорний блок, в якому встановлено процесор Celeron (Cel) з тактовою частотою 1,7 ГГц, 256 Мб оперативної пам'яті (RAM - Random Access Memory - пам'ять з довільним доступом), HDD - жорсткий диск (вінчестер) місткістю 20 Гб та 128 Мб відеопам'яті VRAM.*

Дізнаватись про стан розвитку комп'ютерної техніки найзручніше або за публікаціями в періодичних виданнях, або на сайтах виробників і продавців комп'ютерів в Internet. Там наводяться також характеристики комплектуючих, які встановлюються у процесорному блоку.



Які фірми-виробники сьогодні є лідерами комп'ютерної техніки ?

Крім наведених вище характеристик вживаними вважаються й інші, а саме: тип корпусу (mini-tower, middle-tower - вертикальні корпуси різного розміру), кількість вільних роз'ємів (слотів) на материнській платі для встановлення додаткових блоків, наявність роз'ємів для підімкнення пристроїв USB (Universal Serial Bus - універсальна шина для паралельного підімкнення), наявність та тип роз'ємів для підімкнення периферії, на шумові характеристики вентиляторів охолодження процесора, відеокарти і блоку живлення тощо.

Один із можливих варіантів загальної оцінки процесорних блоків:

Фірма	"Хелп Ко."	Compass	"Квазар-Мікро"
Марка	Maestro Vivaldi	Delfics	Advantis KT3d
Ціна, грн.	1940	1637	1842
Гарантія, міс.	36	24	36
Документація (бали)	4	2	5
Упакування (бали)	1	3	5
Коротка конфігурація	Cel 1,7/256/20	Cel 1,7/128/30	Cel 1,0/128/20
Роз'єми USB	+	+	+
Тип корпусу	middle-tower	middle-tower	mini-tower
Шумність (бали)	4	3	5



Яке значення, на Вашу думку, може мати показник шумності ?

Усі основні компоненти в процесорному блоці змонтовані на материнській платі: процесор з вентилятором охолодження, блоки оперативної пам'яті, відеокарта, звукова карта. Останнім часом спостерігається тенденція до інтегрування відео- та аудіокарт у материнську плату. Окремо в спеціальному контейнері встановлюється жорсткий диск (диски) - вінчестер.

Крім цього, всередині процесорного блоку знаходяться: блок живлення, дисководи для компакт-дисків (CD) та гнучких дисків (FD). Там же знаходиться системний гучномовець.

Взаємодія користувача з процесорним блоком відбувається з використанням монітора, клавіатури і миші.

На **монітор** виводиться інформація про результати роботи процесорного блоку та інших складових частин комп'ютера, а також інформація про дані, які користувач вводить в комп'ютер.

Монітори характеризуються типом та розміром екрана.

Найпоширенішими (на сьогодні) є монітори на кольорових електронно-променевих трубках з розміром діагоналі 15" - 21". Стандартний ряд розмірів діагоналі екрана монітора має наступний вигляд: 15", 17", 19", 21".

Бувають монітори як менших (14", 11"), так і більших (22", 25") розмірів. Для використання в офісі при відстані очей користувача від екрана в межах 60 - 90 см. цілком достатньо монітора з розміром діагоналі екрана 15". Для використання окремих комп'ютерних технологій (графічні редактори, видавничі системи, геоінформаційні системи), пов'язаних з обробкою графічної інформації, використовують монітори з розміром діагоналі екрана 21". Для спеціальних робіт використовують монітори, довша сторона екранів яких розташована вертикально.



Де, на Вашу думку, могли б знайти застосування чорно-білі монітори?

Для моніторів на електронно-променевих трубках важливою характеристикою є частота вертикальної розгортки. При низьких значеннях частоти (до 70 Гц) спостерігається мерехтіння зображення на екрані монітора. Достатньою слід вважати частоту вертикальної розгортки в межах 70 - 100 Гц.

Налагоджено серійний випуск рідкокристалічних панелей (LCD) - TFT-моніторів. Такі панелі мають абсолютно плоский екран, не мерехтять, споживають мало електроенергії та не нагріваються під час роботи.

Рідкокристалічні панелі випускаються таких самих розмірів, що й монітори на електронно-променевих трубках.

Клавіатура і миша - пристрої для введення даних і команд в комп'ютер.

Клавіатура призначена для введення в комп'ютер команд і даних. Зазвичай використовується так звана 101-клавішна клавіатура, хоча насправді на сучасних клавіатурах клавіш завжди більше, ніж 101. Це зумовлене дублюванням клавіш та використанням різними виробниками спеціальних клавіш, наприклад: клавіші відкриття меню, клавіша пуску в операційній системі Windows, клавіша відкриття контекстного меню, клавіша переходу в режим очікування, клавіша програмного вимикання комп'ютера тощо.

Великого поширення набули мультимедійні клавіатури зі спеціальними клавішами для управління мультимедійними пристроями та спеціальним програмним забезпеченням.

З погляду ергономіки клавіатури можуть відрізнитись розташуванням клавіш у вертикальній і горизонтальній площинах. Такі клавіатури розробляються для професійного набирання тексту.

Миша (офіційно - ручний маніпулятор "миша") є пристроєм для роботи спеціально з графічним інтерфейсом сучасного програмного забезпечення, для якого характерне використання для введення команд та запуску їх на виконання графічних образів, висвітлених на екрані монітора. З використанням такого інтерфейсу створена найпоширеніша (сьогодні) операційна система Windows. Використання графічного інтерфейсу передбачають також інші операційні системи, про які мова йтиме дещо пізніше.

Принцип роботи електронно-механічної миші доволі простий. Під час пересування миші по плоскій поверхні кулька, розміщена в корпусі миші, обертається і приводить в рух валики оптоелектронних датчиків переміщення, які до неї дотикаються. Сигнали від датчиків миші (горизонтального і вертикального) перетворюються у команди переміщення спеціального графічного символу (курсору) по екрану монітора. За допомогою клавiш миші (зазвичай їх є дві, хоч може бути і більше) ініціюється виконання команд, пов'язаних з графічними об'єктами, над якими в даний момент знаходиться курсор миші. Якщо не вказано інше, то одноразове короткочасне натискання (клацання) лівої кнопки миші виділяє графічний об'єкт (переводить його у активний стан). Подвійне швидке клацання лівою кнопкою миші запускає на виконання програму, пов'язану з об'єктом, на який вказує курсор. Одноразове клацання правою кнопкою миші викликає на екран контекстне меню (меню, пов'язане з графічним об'єктом, над яким в даний момент знаходиться курсор миші).

При використанні графічного інтерфейсу в операційній системі Windows в принципі можна обійтись без одного з пристроїв уведення: або без клавіатури, або без миші. Однак, в цьому випадку робота з комп'ютером стає складною, незручною і, як наслідок, неефективною.



Як Ви уявляєте собі роботу в операційній системі Windows без клавіатури або без миші?

Тому ми й пропонуємо вважати, що до базового комплекту персонального комп'ютера входять також клавіатура і миша.

До інших складових комп'ютера (так званої периферії) належать:

- **принтери та плотери** - пристрої для виведення даних на паперові носії;
- **сканери** - пристрої для уведення даних з друкованих оригіналів у комп'ютер;
- **модеми** - пристрої для передачі даних з комп'ютера у зовнішні комунікаційні мережі та отримання сигналів із них;
- зовнішні **приводи для компакт-дисків, магнітооптичних дисків та блоки флеш-пам'яті**;
- мультимедійна периферія - **проектори, web-камери, мікрофони, колонки**.

Примітка: наведений вище перелік, звісно, неповний, оскільки комп'ютерна техніка розвивається швидкими темпами і постійно з'являються нові пристрої, які розширюють області та способи застосування комп'ютерних технологій.

Периферійні комп'ютерні пристрої (периферія) підключаються до процесорного блоку за допомогою сигнальних кабелів і часто мають власні пристрої живлення.

Найчастіше результати виконання роботи з використанням комп'ютера виводяться на друк у вигляді паперової копії (друкувати можна також на інших не паперових матеріалах - прозорій синтетичній плівці, фользі, тканині тощо). Найпоширенішим периферійним комп'ютерним пристроєм є **принтер**.

В залежності від технології створення зображення на твердих гнучких матеріалах (носіях) принтери бувають матричні, струменеві, лазерні та оптоелектронні.



Принтери яких типів використовуються сьогодні найчастіше?

В **матричних принтерах** (Рис. 1) зображення формується з крапок, які утворюються на папері після удару голками друкувальної головки по паперу через фарбуючу стрічку.

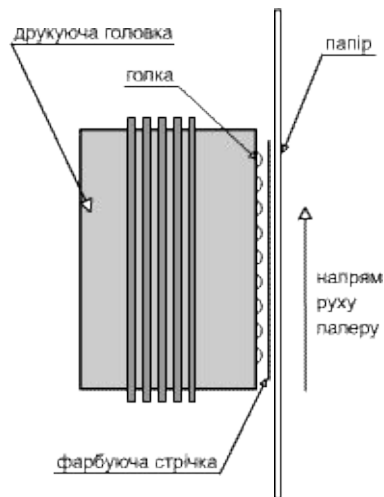


Рис.1.

На цьому ж принципі заснована робота звичайної друкарської машинки із тою лише різницею, що там зображення знаків формується за один удар повністю. В матричному ж принтері зображення формується як комбінація окремих крапок (растрове зображення). У використанні є матричні принтери з 9-ма і 24-ма голками. Якість друку тим вища, чим більша кількість голок. Витратним матеріалом є фарбуючи стрічка у картриджі або без нього.

В **струменевих принтерах** зображення також формується з крапок, але фарбуючою речовиною є спеціальне чорнило, яке дрібними краплями подається з форсунок друкуючої головки (рис.2).

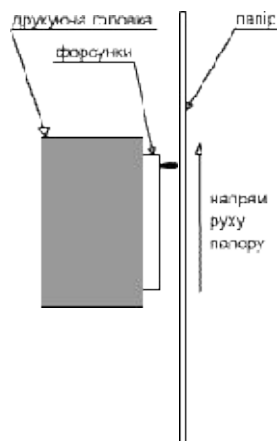


Рис.2.

Спосіб формування крапель, склад чорнила, спосіб розпилення крапель на папір різний у принтерів різних виробників.

Струменеві принтери дозволяють друкувати кольорові зображення за рахунок простої зміни кольору чорнила (точніше - за рахунок одночасного використання картриджів з чорнилами різних кольорів). Витратним матеріалом є чорнило (чорне і кольорове) у спеціальних картриджах або без них.



Комбінації яких кольорів можуть використовуватись для створення повнокольорового зображення?

Лазерні принтери використовують складну технологію створення проміжного зображення на світлочутливому барабані променем малопотужного лазера. З барабана зображення за допомогою тонкодисперсного порошку (тонера) переноситься на папір, де закріплюється термічним способом. Витратним матеріалом є тонер у картриджі або без нього. Замінним вузлом є світлочутливий барабан у спеціальному картриджі.

В оптоелектронних принтерах використовується спеціальний світлочутливий папір (на зразок паперу для факсів), а зображення переноситься на папір за допомогою лінійки з оптичних діодів, які керуються сигналами спеціального сканера зображення. Витратним матеріалом є світлочутливий папір.

До основних характеристик принтерів належать:

- швидкість друку;
- якість друку;
- можливість кольорового друку;
- шумність;
- кількість копій з одної заправки витратних матеріалів;
- використовувані для копій тверді носії;
- вартість.

В табл.3 наведено порівняльні характеристики матричних, струменевих та лазерних принтерів. Оптоелектронні принтери із-за невисокої надійності, відсутності відчутних переваг та необхідності використання спеціального паперу в практиці зараз широко не використовуються.

Таблиця 3

Порівняльні характеристики принтерів

Характеристика	Матричний	Струменевий	Лазерний
Швидкість друку	найнижча	середня	найвища
Якість друку	найнижча	висока	найвища
Різнокольоровий друк	немає	є	є
Шумність	найвища	низька	низька
Кількість копій з одної заправки	найбільша	найменша	велика
Вартість	найнижча	висока	найвища



Які характеристики принтерів видаються Вам найважливішими ?

Принтери використовуються для виведення на друк документів розміром А4 та А3.

Для виведення на папір штрихових документів великого розміру (креслення, карти розміром до А0) використовуються **плотери**. Вузлом, за допомогою якого наноситься зображення в плотерах, є головка, що переміщується за командами з комп'ютера. Зображення може наноситись олівцем, фломастером або рапідграфом.

За способом переміщення головки відносно паперу розрізняють плотери планшетні і барабанні.

В планшетних плотерах папір кріпиться на нерухомому столі, а головка переміщується в обох взаємоперпендикулярних напрямках (рис.3).

В барабанних плотерах головка переміщується лише в одному напрямі, а переміщення паперу в перпендикулярному до руху головки напрямі забезпечує барабан, який обертається (рис.4).

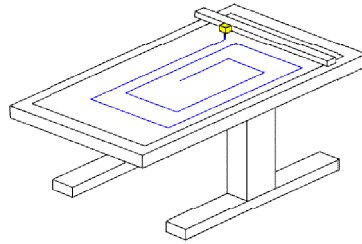


Рис.3.

На плотер можна вивести текстовий документ з ілюстраціями (графіками, діаграмами, таблицями), використовуючи результат як плакат для демонстрації у великій аудиторії. Але цей спосіб вже практично не використовується в зв'язку з розвитком технологій демонстрації невеликих зображень (як прозорих, так і непрозорих) проектуванням їх на екран за допомогою оптичних або комп'ютерних проекторів.

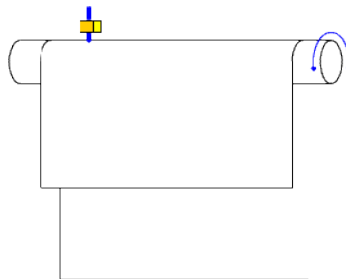


Рис.4.



Де у вашій практиці можна застосувати плотер?

Часто використання комп'ютерних технологій вимагає уведення в комп'ютер даних з друкованих оригіналів. Для цього використовуються **сканери** - ручні або планшетні.

В сканері оригінал освітлюється світлом потужної електричної лампи (галогенної), а відбите світло сприймається спеціальним датчиком і отриманий сигнал перетворюється в цифрову форму. Отримане таким чином зображення в цифровій формі доступне для обробки в комп'ютері спеціальними програмами.

Ручний сканер малогабаритний і дешевий, але якість отриманого зображення залежить від натренованості оператора у рівномірному пересуванні сканера по поверхні оригіналу. Крім цього, його складно використовувати для сканування великих оригіналів. В планшетному сканері процес сканування повністю автоматизований і його результат практично не залежить від людини, а лише від якості оригіналу та якості сканера.

Робота сканера характеризується швидкістю і роздільною здатністю.

Доступні зараз сканери сильно відрізняються між собою за цими характеристиками. Так, наприклад, сканування повнокольорового зображення розміром А4 може тривати від 10 с до 3 хв. Роздільна здатність сканерів коливається від 600 до 2400 dpi (dot per inch - крапок на дюйм). В офісах використовують в основному планшетні сканери з роздільною здатністю 1200 - 1800 dpi.



Який вплив на якість сканованого зображення має роздільна здатність сканера?

Під час сканування неплоских оригіналів (розгорнутих книг - особливо рідкісних, та іншої типографської продукції) використовують сканери спеціальної конструкції та спеціальні програми, здатні коригувати зображення, отримані з неплоских оригіналів.

Модем - пристрій для передачі даних між комп'ютерами засобами зовнішніх ліній зв'язку.

Для того, щоб передати дані з одного комп'ютера на інший, вони повинні бути з'єднані між собою каналами зв'язку.

Канали зв'язку можуть бути організовані з використанням ліній різних типів. У найпростішому випадку використовуються проводові телефонні лінії (комутовані або виділені). Все частіше використовуються лінії зв'язку з оптичного волокна (оптичні, чи оптоволоконні лінії). Нерідко для зв'язку між комп'ютерами використовується радіозв'язок. Тоді говорять про радіолінію.

У будь-якому випадку проблема полягає в тому, що характеристики сигналів передачі даних в комп'ютері і в лінії не співпадають. Тому ці сигнали потрібно перетворювати з одного виду в інший як на стороні передавача, так і на стороні приймача сигналу (даних) без втрати самих даних (рис.5).

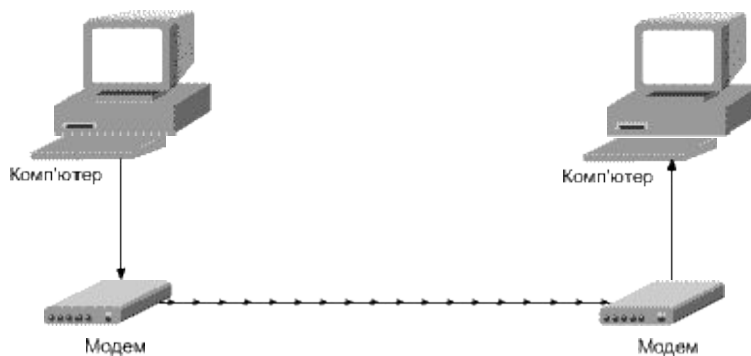


Рис.5.

Це і є основне призначення модему. Крім цього модеми виконують і інші важливі для комп'ютерного зв'язку функції і є доволі складними технічними пристроями.

У відповідності до конкретних умов (наявних ліній зв'язку) використовують цифрові або кабельні модеми. Є різновиди модемів для безпроводових інфрачервоних, радіочастотних і супутникових ліній зв'язку. За способом під'єднання модему до комп'ютера розрізняють модеми зовнішні, внутрішні і програмні.

Для уведення в комп'ютер даних з оптичних і магнітооптичних дисків інколи використовують **зовнішні приводи**. Досить часто для тривалого зберігання даних використовують блоки **флеш-пам'яті**. Це зручні засоби перенесення даних, які мають невеликий об'єм - близько 10 куб.см (5x2x1 см) і вагу, тим більше, що ємність цих засобів зберігання даних постійно зростає (256 Мб і більше).

Автономні (зовнішні) приводи та флеш-пам'ять підмикаються до комп'ютера через USB-роз'єми, що дозволяє операційній системі відразу розпізнавати їх наявність без перезавантаження комп'ютера.

Використання флеш-пам'яті дозволяє також, хоча і посередньо, вирішувати у певних межах і проблему захисту даних.



Які носії даних Вам відомі?

Як носії даних поряд із вже традиційними гнучкими дисками (дискетами) та дисками CD-R дедалі ширшого застосування набувають сучасні оптичні носії інформації: диски CD-RW, CD-DVD та ZIP. Компакт-диски випускаються двох типорозмірів: 120 та 90 мм. Диски CD-RW допускають багаторазовий запис даних на одному й тому ж диску (диски CD-R допускають лише одноразовий запис даних). Що ж до технологій запису дисків DVD, то вона ще недостатньо усталена. Зараз використовується близько 30 технологій запису DVD, які не завжди сумісні між собою, тому час широкого розповсюдження цієї технології збереження даних ще не настав.

Для цифрових фото- та кінокамер як носії даних використовуються спеціальні чіпи пам'яті - smart drive. Дані з таких носіїв зчитуються у звичайних дисководів для гнучких дисків з використанням спеціальних перехідних касет.

Для резервного копіювання та архівування як носії даних використовуються стримери - накопичувачі на магнітній стрічці. Місткість таких накопичувачів дуже велика, але швидкість доступу до даних в них дуже низька, оскільки стримери належать до пристроїв постійної пам'яті з послідовним доступом (для того, щоб зчитати необхідні дані, потрібно перемотати стрічку від кінця до місця розташування на ній цих даних).

Особливу групу периферії складають мультимедійні пристрої, які дозволяють значно розширити можливості представлення оброблюваної комп'ютером інформації.

Мультимедійні проектори дозволяють проектувати на великий екран зображення або безпосередньо з комп'ютера, або з автономних носіїв. Таким чином доступ до сприймання пропонованої інформації одночасно може мати велика аудиторія.

Web-камери (цифрові відеокамери) є засобами для переведення в електронну форму рухомого зображення. Таке зображення можна обробляти в комп'ютері на місці, або передавати на інший комп'ютер засобами комп'ютерного зв'язку. Web-камери дозволяють організувати т.зв. відеоконференції.

Мікрофон та звуковідтворюючі пристрої (телефони, колонки) є засобами уведення в комп'ютер та виведення з нього звукових даних. Обробка в комп'ютері звукової інформації - найдавніший приклад використання комп'ютера для обробки мультимедійної інформації (звуку і рухомого зображення), і, напевно, добре відомий хоча б з музики, яку можна слухати за допомогою комп'ютера.

Програмне забезпечення обробки інформації про економічні процеси

Другою необхідною складовою комп'ютерних технологій, на чому наголошувалося раніше, є **програмне забезпечення**.

Програмне забезпечення прийнято спрощено поділяти за його призначенням (рис.б).

Тепер дещо детальніше про компоненти цієї структурної схеми.

Операційна система (ОС) - це сукупність програм, які забезпечують керування апаратними і програмними ресурсами комп'ютера та їх взаємодією з іншими програмними засобами і користувачами.

З погляду користувача, все, що виконується на комп'ютері, відбувається під управлінням операційної системи.

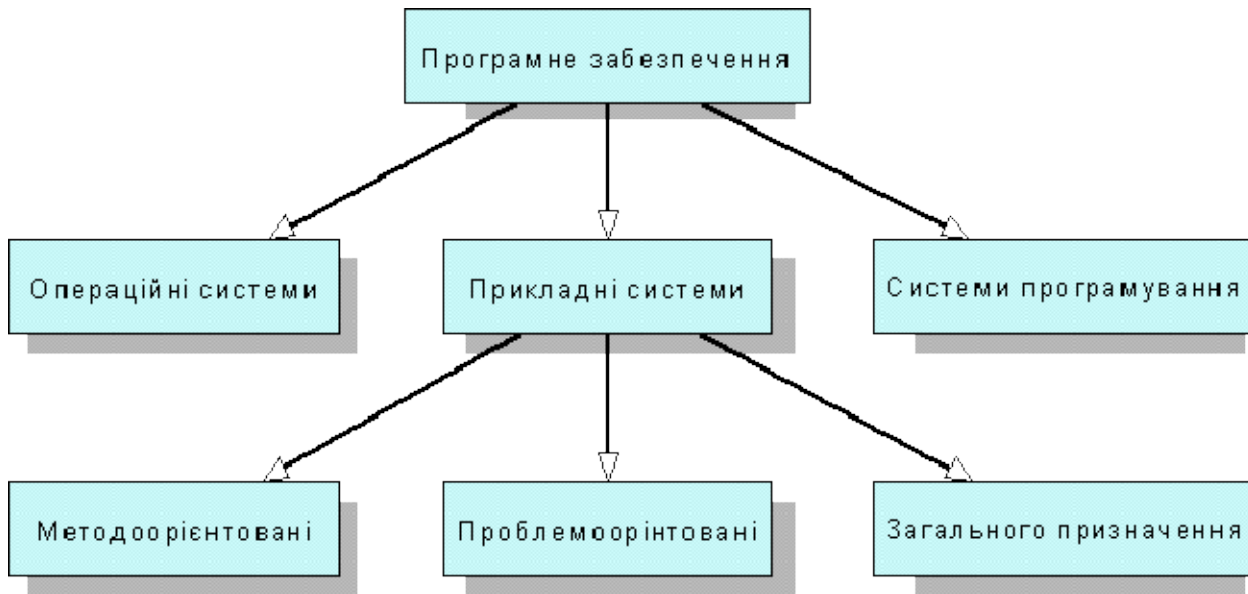


Рис.6.



Які операційні системи використовуються в комп'ютерах, з якими з них ви мали справу? Хто займається інсталяцією програмного забезпечення та підтриманням його в робочому стані?

Загалом, в комп'ютері не обов'язково повинна бути операційна система. Приклади таких комп'ютерів - ігрові приставки. В них також є процесор (нерідко дуже потужний), оперативна пам'ять, засоби введення інформації (миша або джойстик). Але операційна система їм не потрібна. Ігрові програми і дані до них містяться або в мікросхемі, яка знаходиться в ігровому картриджі, або на лазерному диску. Коли картридж або диск вставляється у приставку, програма запускається автоматично, і ніякі дії, не передбачені сценарієм гри, виконати неможливо.

Однак, для комп'ютера універсального призначення вона необхідна. Така операційна система повинна відповідати наступним вимогам:

- бути загально визнаною і використовуватись як стандартна система на багатьох комп'ютерах;
- працювати з різними апаратними засобами, які виготовляються різними фірмами;
- забезпечувати можливість запуску і використання програм різних виробників і різного призначення;
- підтримувати мережеві комп'ютерні технології;
- надавати засоби для перевірки, налаштування та обслуговування комп'ютера, його пристроїв та програм, які на ньому інстальовані.

В зв'язку з тим, що в світі існує і використовується багато різних типів і класів персональних комп'ютерів, які дуже швидко розвиваються, а розробкою програмного забезпечення займаються численні виробники, стратегії яких підпорядковані ринковим умовам, існує і велика кількість операційних систем. Для користувача у сфері державного управління реальний інтерес становлять три з них, які відповідають наведеним вище вимогам - Windows, MacOS та т.зв. Unix-подібні системи (наприклад, Linux або FreeBSD).



Слід підкреслити, що подібні твердження (про реальний інтерес) актуальні лише зараз і протягом нетривалого майбутнього. До принципової зміни ситуації та наших поглядів на питання вигоди практичного використання тих чи інших операційних систем можуть призвести наступні чинники:

- стрімкий розвиток елементної бази персональних комп'ютерів, розробка нових типів комп'ютерів (наприклад, нейронних);
- значні досягнення фундаментальних досліджень в царині інформатики, теорії інформаційного суспільства;
- швидка комп'ютеризація усіх сфер діяльності людини і, як наслідок, перетворення засобів комп'ютерної обробки даних у суттєвий фактор поступального розвитку економіки і суспільства;
- ринкові процеси у сфері виробництва програмного забезпечення.

Операційна система Windows

Найпоширенішою зараз в Україні є операційна система Windows, яка використовується на так званих IBM/PC-сумісних комп'ютерах. Це зумовлене поширеністю саме таких комп'ютерів, різноманітністю та високою якістю прикладного програмного забезпечення, відносною простотою інсталювання (встановлення на комп'ютер) системного та прикладного програмного забезпечення.

Ще однією причиною донедавна була доступність прикладного програмного забезпечення для цієї ОС, оскільки масовою практикою було використання неліцензованого програмного забезпечення. Ця доступність стала причиною сильної залежності поширення комп'ютерних технологій в Україні від розвитку IBM/PC-сумісних комп'ютерів та програмних продуктів виробництва фірми Microsoft.

Операційна система Windows з самого початку була орієнтована на кінцевого користувача. Цьому сприяв графічний інтерфейс, ідея якого спочатку вдало була реалізована П.Нортоном в його оболонці Norton Commander для ОС MS-DOS (Microsoft Disk Operation System). Перші версії Windows (2.0, 3.1, 3.11) також були лише оболонкою (надбудовою) для однієї з перших ОС для персональних комп'ютерів - DOS, і повністю використовували ядро цієї операційної системи.

Перетворення Windows у повноцінну операційну систему відбулось з випуском версії Windows95, де користувацькі переваги графічного інтерфейсу виявились у повній мірі. Наступним логічним розвитком ОС Windows стали версії Windows'98, Windows NT, Windows ME, Windows 2000, Windows XP, Windows Service 2003 тощо.

Графічний інтерфейс разом з продуманим і зручним інсталятором самої Windows та розробленого під неї прикладного програмного забезпечення, дозволили перетворити комп'ютер у інструмент справді персонального використання як на робочому місці, так і вдома. Таке поширення Windows стимулювало також розробку найрізноманітнішого програмного забезпечення для вирішення конкретних задач обробки даних. Перелік прикладного програмного забезпечення під Windows зараз містить не одну тисячу назв.

Починаючи з версії Windows95 в ОС підтримуються інтегровані сервіси підключення до глобальних комп'ютерних мереж, а з версії Windows'98 - організація однорангових локальних комп'ютерних мереж, а також доступ до сервісів глобальних мереж з практично всіх прикладних програм (інколи ці можливості навіть обтяжливі).

Для інсталяції більшості апаратних засобів в ОС Windows використовується механізм PnP (Plug and Play), який дозволяє здебільшого автоматично розпізнати апаратне забезпечення і комплектує та підібрати і встановити відповідний драйвер.

Особливістю сучасного стану ОС Windows є дві обставини, які безумовно вплинуть на подальший її розвиток, а відтак, і на розповсюдженість:

- по-перше, несумірне нагромадження графічних елементів ОС і прикладних програм викликає необхідність використання дедалі потужніших

апаратних засобів (швидших процесорів, відеокарт, значних обсягів оперативної пам'яті, потужних систем охолодження), а, отже, і значно дорожчих.

- по-друге, розвиток як операційної системи, так і прикладних програм під неї відбувається шляхом їх універсалізації, а, отже, ускладнення, що викликає зростання їх вартості з одночасним значним недовикористанням можливостей та зниженням їх надійності та захищеності від несанкціонованого доступу через комп'ютерні мережі.



Операційна система Windows має багато так званих сервісних програм. Які з цих програм Вам відомі? Як Ви їх використовуєте?

Операційна система MacOS

Другою найпоширенішою операційною системою в Україні є MacOS. Вона встановлюється на комп'ютерах фірми Apple Macintosh (Apple історично були першими персональними комп'ютерами).

Інтерфейс сучасних систем MacOS також графічний і з вигляду нагадує Windows, хоча деякі особливості в роботі з цим інтерфейсом є. На клавіатурі немає звичної клавіші Ctrl, натомість є клавіша Cmd (Command). Дискковод для флоппі-дисків не має кнопки повернення дискети - ця процедура виконується програмно. Є і низка інших особливостей. До того ж ціна комп'ютерів Apple Macintosh майже у півтора рази вища, ніж аналогів на платформі IBM/PC.



Чи доводилось Вам колись працювати за комп'ютером, на якому інстальована операційна система MacOS? В чому її особливість?

Загалом слід відзначити, що до недавнього часу комп'ютери Apple Macintosh з MacOS мали переваги перед комп'ютерами з Windows в обробці графічної інформації з погляду швидкості такої обробки. Але останнім часом технічні характеристики процесорів Intel, Duron, Celeron, AMD стали такими, що різниця в часі обробки великих масивів інформації стає зникаючою.

Unix-подібні операційні системи

Історія виникнення та розвитку операційної системи Unix цікава і багата. Ця система стала основою для створення комп'ютерних мереж і зараз є основою в глобальних комп'ютерних мережах.

З використанням Unix працюють більшість хостів та серверів Internet. Це потужна і гнучка система, яка дозволяє повністю контролювати як апаратні, так і інформаційні ресурси комп'ютерних мереж. Поки що - це найзахищеніша ОС, яку складно зіпсувати і отримати несанкціонований доступ до ресурсів, що знаходяться під її управлінням.

Однак Unix - це операційна система не для кінцевих користувачів. Це система для професіоналів. Вона розрахована на використання командного інтерфейсу та детальне знання принципів її роботи і команд управління нею. Крім цього, вона вимагає точних знань технічних характеристик апаратного забезпечення та його компонентів. Саме з цих причин, та ще із-зі її високої вартості, ця система не набула широкого розповсюдження серед непрофесіоналів.

Однак, останнім часом у ширшому використанні цієї операційної системи сформувались цікаві тенденції, розвиток яких може докорінно змінити уподобання багатьох користувачів, як індивідуальних, так і корпоративних.

Мова йде про так звані Unix-подібні системи, які розробляються на громадських засадах колективами ентузіастів в цілому світі (точніше колективами ентузіастів, до яких входять учасники з цілого світу, оскільки розробка цих систем відбувається в Internet). До найвідоміших розробок відносяться операційні системи Linux та FreeBSD.

Ентузіастів приваблює те, що, по-перше, системи є відкритими і їх коди можна безкоштовно отримати на багатьох сайтах в Internet, а, по-друге - ці системи повністю сумісні з Unix, а, отже, дають можливість значно комфортніше почуватись в глобальних мережах (точність запитів, підвищена швидкість реагування серверів на запит, менший за обсягом трафік і більша швидкість пошуку).

За ці переваги, однак, доводиться розплачуватись доволі складним для неспеціаліста процесом інсталювання системи, складностями в інсталюванні прикладного програмного забезпечення, використанням переважно командного інтерфейсу, складностями в процесі кирилізації системи.



Що Ви чули про приклади практичного використання операційних систем FreeBSD, Linux? В чому їх особливості?

І усе-таки, використання Unix-подібних систем має свої перспективи в Україні, і зокрема, в органах державного управління та місцевого самоврядування.

Перший крок на шляху формування таких перспектив був зроблений, коли у Верховну Раду України було внесено проект закону "Про використання вільного (неліцензованого) програмного забезпечення". Минаючи всі спеціальні обґрунтування, зазначимо, що прийняття такого закону у запропонованій редакції стимулюватиме розвиток вітчизняного ринку програмного забезпечення. Для кінцевого користувача це означатиме необхідність освоєння нових операційних систем та нового прикладного програмного забезпечення.

Цим проектом передбачається законодавча підтримка заходів у наступних основних напрямках:

- розробка власних локалізованих дистрибутивів операційних Unix-подібних систем для корпоративних кінцевих користувачів;
- розробка прикладного програмного забезпечення для Unix-подібних систем;
- заборона використання комерційних програмних продуктів в державних установах у тих випадках, коли є їх аналоги серед вільного програмного забезпечення;
- державна підтримка розробників операційних систем і прикладного програмного забезпечення та торговельних організацій, які продають вільне програмне забезпечення (в цьому випадку ціна програмного продукту складається з вартості носія і торговельної націнки).

Досвід розробки локалізованих (кирилізованих) дистрибутивів вже є. Це, наприклад, дистрибутив ASP Linux версій 7.2 і 7.3.

Продумана і опрацьована інсталяція системи на двох компакт-дисках дозволяє відносно просто інсталювати систему на серійний комп'ютер. При цьому передбачена українізація системи. Під час інсталяції пропонується встановити графічну оболонку для системи (X Window). Все це сприяє простішому і

швидшому знайомству користувача з основами роботи в новій операційній системі. На інсталяційних дисках знаходиться також великий і досить повний набір прикладного програмного забезпечення, необхідного користувачеві в офісі.

Свідченням зростаючої популярності Unix-подібних систем є випуск каталогу програмного забезпечення Linux, який з'явився у першому номері "Каталогу програмного забезпечення" від фірми Softline (Росія) в липні 2003 року.

Прикладне програмне забезпечення

Хоча операційні системи необхідні для роботи комп'ютера універсального призначення, кінцевому користувачеві доводиться, в основному, працювати з **прикладним програмним забезпеченням**. Операційна система за налагодженого в установі адміністрування інформаційної системи - це сфера професійних і службових інтересів системного адміністратора.

Перелік використовуваного в органах державного управління та місцевого самоврядування програмного забезпечення визначається обсягом та змістом тих задач, які вирішуються цим органом.

Всі органи регіонального управління мають спільні риси: проблеми соціально-економічного розвитку регіону повинні вирішуватись на основі комплексного підходу до прогнозування та планування, з поєднанням галузевого та територіального принципів, з використанням багатоаспектної інформації. Це в свою чергу потребує вирішення значної кількості питань, пов'язаних з особливостями інформаційного забезпечення управлінських процесів, а також з:

- великими обсягами інформації;
- великою кількістю джерел інформації;
- складним та багаторазовим перехрещенням великої кількості

інформаційних потоків, часто зі слабкою їх взаємодією або взагалі її відсутністю в зв'язку з тим, що управління регіонами в більшості випадків виконується не одним, а сукупністю територіальних, галузевих та багатогалузевих органів;

- відсутністю уніфікованих структур повідомлень в багатьох інформаційних потоках;
- відсутністю єдиної регламентованої системи показників функціонування регіонів.

В своїй більшості регіони являють собою складні народногосподарські комплекси. В кожному з них, як правило, складається своя специфічна соціально-економічна структура, що може відрізнити його від інших в своїй класифікаційній групі і в деякій мірі обумовлювати його місце в структурах управління вищого рівня. Зокрема, особливості окремих регіонів, відрізняючи їх один від одного, можуть полягати у відмінностях:

- географічного розташування;
- галузевої структури народного господарства регіону;
- структури та технічного оснащення господарств, підпорядкованих регіональним органам управління;
- структури народногосподарського комплексу по територіальній підпорядкованості (державним, обласним, міським, районним, сільським, селищним органам управління);
- економічних показників;
- демографічної структури населення;

- забезпечення населення регіону культурно-побутовими, транспортними та іншими послугами;
- навколишніх (природних) умов проживання населення; використання території;
- наявності та використанні трудових ресурсів, їх професійно-кваліфікаційній структурі, рівні безробіття;
- наявності та використанні природних ресурсів.

Всі ці відмінні особливості безпосередньо відбиваються на організаційних структурах (складі підрозділів та служб) і складі функцій та задач органів регіонального управління.

З функціональної точки зору всі регіональні органи управління умовно можна розділити на два напрямки своєї діяльності, які визначають два види функцій: управлінські, які включають безпосередньо функції управління регіоном (тобто ті, для виконання яких і створені органи регіонального управління), і офісні, що забезпечують власну життєдіяльність органу управління (управління кадрами, матеріально-технічне постачання, бухгалтерський облік, діловодство, контроль виконавчої дисципліни, робота зі зверненнями громадян та інші).

Офісні функції - це функції секретаріату. Незважаючи на той факт, що на ринку програмної продукції представлена значна кількість комерційних програмних продуктів, призначених для автоматизації офісних задач, є об'єктивна необхідність оптимального сполучення комерційних продуктів загального призначення та спеціалізованих продуктів, орієнтованих безпосередньо на конкретного користувача чи задачі на конкретному робочому місці (АРМ).



Яке програмне забезпечення використовується у вашій установі для виконання офісних функцій?

До комерційних продуктів слід віднести текстові редактори, програми автоматизованого перекладу з української мови на іноземну і навпаки, орфографічні редактори тощо. До спеціалізованих програмних продуктів слід віднести такі, що автоматизують основні функції секретаріату і, по-перше, не вимагають, як у випадку з комерційним продуктом, значної кваліфікованої адаптації до конкретних умов, а по-друге, не потребують високої комп'ютерної кваліфікації від особи, що має працювати з продуктом (практично без адаптації). Комплексними управлінськими функціями слід вважати:

- комплексний аналіз та контроль задоволення регіону ресурсами і умовами його життєдіяльності;
- прогнозування соціально-економічного розвитку регіону на періоди різної тривалості, у т.ч. й великої;
- визначення потреб регіону в ресурсах та умовах, які необхідні для його життєдіяльності;
- реалізація заходів щодо задоволення потреб регіону;
- формування та ведення паспорту регіону (умовно-постійна інформація, тобто така, що незначно оновлюється не частіше, ніж за 2-5 років). Паспорт регіону - це системно-організований комплекс цифрової, текстової та графічної інформації про територію, населення та інфраструктуру, яка характеризує усі сфери життєдіяльності регіону.



Як Ви уявляєте використання комп'ютерів для виконання управлінських задач?

В переважній більшості випадків задачі, які вирішуються на робочих місцях в органах державного управління та місцевого самоврядування за допомогою комп'ютерної техніки, залишається традиційним, а саме:

- створення текстових документів (простих документів довільної форми, документів на бланках, складних документів довільної форми, аналітичних документів з необхідними ілюстраціями) - текстові процесори;
- виконання обчислень для аналізу числових даних та графічне представлення результатів цього аналізу - табличні процесори;
- доступ до розподілених баз даних або робота з власними (зазвичай невеликими за обсягом) базами даних з метою отримання даних чи їх поновлення - системи управління базами даних;
- доступ до Web-ресурсів засобами локальних комп'ютерних мереж з обробкою 2D графіки та рухомого зображення (статична графіка, найпростіша анімація).

Окремо слід виділити задачі збирання й аналізу даних для прийняття управлінських рішень. Хоча в основному для вирішення цих задач і використовують поширені інформаційні технології, підготовка, аргументування і моніторинг результатів реалізації рішень вимагає творчого підходу і необхідних теоретичних знань саме в царині прийняття управлінських рішень.

Для виконання більшості задач з **програмного забезпечення загального призначення** традиційно використовують офісні програмні продукти з пакету MS Office, розроблені для операційної системи Windows. Можливостей цього програмного забезпечення цілком достатньо для виконання більшості перелічених вище задач.

Це твердження не стосується створення і супроводу великих за обсягом розподілених баз даних, для яких, зрештою, завжди використовувалось спеціальне програмне забезпечення FoxBase, Oracle тощо.

Розвиток вітчизняного ринку програмного забезпечення може створити таку ситуацію, коли у персональних комп'ютерах використовуватимуть Unix-подібні операційні системи типу Linux чи FreeBSD та відкрите офісне програмне забезпечення на взірець Open Office.

Є побоювання, що такий розвиток подій викличе незаплановані витрати коштів і часу на освоєння персоналом нового (не завжди простого, наприклад, системного) програмного забезпечення, однак у цьому зв'язку слід звернути увагу на дві наступні обставини:

- кінцевому користувачу неважливо, в якій операційній системі працює потрібне йому прикладне програмне забезпечення. Робота операційної системи - це завдання для спеціальних служб чи окремого спеціаліста організації.
- освоювати нове (вільне) програмне забезпечення можна вже зараз, використовуючи те, що нові програмні продукти є крос-платформовими, тобто придатними для використання під управлінням більшості використовуваних зараз операційних систем, зокрема Windows.

Для використання в органах державного управління та місцевого самоврядування рекомендується використовувати програмне забезпечення, перелік якого наведено в табл.4. Там же наведено і аналоги, які використовуються зараз.



Чи доводилось Вам особисто працювати з програмним забезпеченням, наведеним у другому стовпці таблиці чи бачити приклади його використання?

Відразу зазначимо, що наведений перелік - не єдино можливий варіант. Ринок програмного забезпечення розвивається дуже швидко і, зокрема, в царині прикладного програмного забезпечення.

Таблиця 4

<u>Перелік програмного забезпечення</u>		
Використовують зараз	Рекомендоване	Призначення
Microsoft Word	Open Office.org Writer	створення текстових документів з ілюстраціями, на бланках, за шаблонами
Microsoft Excel	Open Office.org Calc	табличний процесор (електронні таблиці)
Power Point	Open Office.org Impress	створення та транслявання комп'ютерних презентацій
Photo Shop, Photo Paint, Corel Draw	Gimp	створення растрової та векторної графіки
Internet Explorer, Netscape Navigator	Mozilla	навігація в Internet та доступ до сервісів Internet

Основною перевагою використання пропонованого програмного забезпечення є те, що воно розповсюджується на умовах GPL безкоштовно. Вартість інсталяційних матеріалів становить вартість носія, на якому вони записані (як правило, компакт-диска). Одночасно вирішується питання про незаконне використання не ліцензованого програмного забезпечення.

Ще одним з аргументів на користь проведення пропонованої політики щодо використовуваного програмного забезпечення є те, що продукти основного комерційного виробника на українському ринку - фірми Microsoft - стають дедалі громіздкішими та ресурсоємними. Це стосується як операційних систем, так і прикладного програмного забезпечення.

Що стосується використовуваного **спеціального програмного забезпечення**, то сюди слід віднести графічні редактори, видавничі системи, геоінформаційні системи, системи управління базами даних (СУБД). Перелік деяких з них наведено в табл.5.

Таблиця 5


<u>Перелік деяких спеціальних програмних продуктів</u>	
Призначення	Програмний продукт
Графічні редактори	PhotoShop, PhotoPaint, CorelDraw, AdobeIllustrator, Gimp, FreeHand
Системи управління базами даних	FoxBase, FoxPro, Oracle, MySQL
Видавничі системи	PageMaker, QuarkPress
Системи автоматизації розрахунків	Derive, MathCad
Системи (середовища) програмування	VisualBasic, TurboPascal, Delphi, C++, Java

Однак, купуючи ці програмні продукти потрібно враховувати наступні міркування:


- робота з цими програмними продуктами вимагає не лише доброго знання прийомів роботи з програмними продуктами, але й спеціальних навичок та умінь (художнього смаку, знання спеціальних видавничих термінів та понять, предметної галузі баз даних та принципів їх проектування);

- виконання непрофесіоналом певної роботи за допомогою цих програмних продуктів займе значно більше часу, ніж виконання тої ж роботи у спеціалізованих підприємствах чи організаціях, та й якість виконаної роботи буде не найкраща.

Наведені міркування дають підстави для наступного висновку: роботу, яка вимагає використання спеціального програмного забезпечення, краще довіряти спеціалістам з-поза установи. Це буде і швидше, і якісніше. Якщо спробувати оцінити витрати установи на виконання таких робіт сторонніми організаціями, то варто пам'ятати, що комерційне спеціальне програмне забезпечення коштує дуже дорого, а використовується в державному управлінні за прямим призначенням дуже рідко.

 Згадаємо, ще одну обставину, яка часто залишається поза увагою користувачів. Дуже часто в офісі є потреба у створенні, наприклад, простіших текстових документів або виконанні нескладних математичних обчислень.

В першому з цих випадків немає ніякої необхідності використовувати текстовий процесор Word - достатньо найпростіших вбудованих у Windows текстових редакторів: NotePad (Блокнот) або WordPad. Перший дозволяє створювати найпростіші текстові документи без ілюстрацій та різноманітних шрифтів. Другий - доволі потужний текстовий редактор, який дозволяє використовувати вбудовані об'єкти (рисунок, діаграми, таблиці) і формувати текст з використанням різних шрифтів та їх стилів. Робота з цими редакторами принципово нічим не відрізняється від роботи у будь-якій програмі під управлінням Windows. NotePad створює файли з розширенням .txt, а WordPad - з розширенням .wrt. У другому випадку нема потреби використовувати електронні таблиці: часто достатньо вбудованої у Windows програми Calculator (Калькулятор). Результати обчислень через буфер обміну можуть бути перенесені у довільний документ.

 Чи маєте Ви практичний досвід використання спеціального програмного забезпечення? Яким він є?

В практиці роботи органів державного управління та місцевого самоврядування дедалі ширше використовуються спеціальні програмні продукти для реалізації сучасних інформаційних технологій: підтримки прийняття управлінських рішень, електронного документообігу, електронний переклад з іноземних мов та навпаки, підготовки інформаційних мультимедійних матеріалів.

Програмне забезпечення для цих технологій випускається багатьма фірмами серійно, але складності у його використанні зумовлені необхідністю його адаптації та налаштування для потреб кожної конкретної організації чи установи та вирішуваних ними задач окремо у кожному випадку.

З розвитком та поширенням застосування мережевих технологій обробки даних особливого значення набуває оволодіння користувачами основами створення за допомогою універсального та спеціалізованого програмного забезпечення Web-документів для забезпечення доступу до цих документів з глобальних мереж. Для цього необхідні базові знання мови гіпертекстової розмітки документів HTML та прийомів роботи зі спеціальним програмним забезпеченням (HTML-редактори). Один з таких редакторів - FrontPageExpress - є інтегральною частиною операційної системи Windows. Потужніша і повніша версія цього редактора - FrontPage - входить до складу програмного продукту MS Office. Крім цього, не слід забувати, що всі компоненти MS Office дозволяють досить просто зберігати документи у форматі .asp (.aspl) та компоувати матеріали для їх розміщення на потрібному сайті (виконувати публікація матеріалів).

РОЗДІЛ 4

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ОБРОБКИ ЕКОНОМІЧНОЇ ІНФОРМАЦІЇ, МЕРЕЖНІ ТЕХНОЛОГІЇ

Технологія - це комплекс наукових та інженерних знань, реалізованих у прийомах праці, наборах матеріальних, технічних, енергетичних, трудових факторів виробництва, засобах їх об'єднання для створення продукту або послуги, що відповідають певним вимогам. У такому розумінні технологія нерозривно пов'язана з машинізацією та автоматизацією виробничого або невиробничого, насамперед управлінського процесу. Управлінські технології ґрунтуються на застосуванні комп'ютерів і телекомунікаційної техніки, їх об'єднання у комп'ютерні мережі.

Відповідно до визначення, прийнятого ЮНЕСКО, **інформаційна технологія** - це комплекс взаємозалежних, наукових, технологічних, інженерних дисциплін, що вивчають методи ефективної організації праці людей, зайнятих опрацюванням і збереженням інформації; обчислювальну техніку і методи організації і взаємодії з людьми і виробничим устаткуванням, практичні додатки, а також пов'язані з усім цим соціальні, економічні і культурні проблеми.

Самі інформаційні технології вимагають складної підготовки, великих початкових витрат і наукомісткої техніки. Їхнє введення повинно починатися зі створення математичного забезпечення, формування інформаційних потоків у системах підготовки спеціалістів.

Існує декілька точок зору щодо шляхів розвитку інформаційних технологій. Загальним для них (усі вони викладені нижче) є те, що з появою тих чи інших засобів обробки інформації, тих чи інших їх можливостей, розвиток здійснювався по різному. Основною ж метою розвитку було і залишається задоволення особистих та суспільних інформаційних потреб людини як для фахової, так і для побутової сфери її життєдіяльності.

За видом задач і процесів опрацювання інформації розрізняють два етапи розвитку інформаційних технологій: 1-й етап (60 - 70-і рр.) - опрацювання даних в обчислювальних центрах у режимі колективного користування. (основним напрямком розвитку інформаційної технології була автоматизація операційних рутинних дій людини); 2-й етап (з початку 80-х рр.) - створення інформаційних технологій, спрямованих на розв'язання стратегічних задач.

За проблемами, які стоять на шляху інформатизації, розрізняють 4 етапи розвитку інформаційних технологій: 1-й етап (до кінця 60-х рр.) характеризувався проблемою опрацювання великих обсягів даних в умовах обмежених можливостей апаратних засобів; 2-й етап (до кінця 70-х рр.) характеризувався відставанням програмного забезпечення від рівня розвитку апаратних засобів, переважно ЕОМ серії ІВМ/360; 3-й етап (з початку 80-х рр.) - комп'ютер стає інструментом непрофесійного користувача, а інформаційні системи - засобом підтримки прийняття його рішень (проблемами цього етапу є максимальне задоволення потреб користувача і створення відповідного інтерфейсу для роботи в комп'ютерному середовищі); 4-й етап (з початку 90-х рр.) - створення сучасної технології міжустановних зв'язків і інформаційних систем з масовим використанням комп'ютерних мереж (проблеми цього етапу: стандарти комп'ютерного зв'язку; організація доступу до стратегічної інформації; організація захисту і безпеки інформації).

За перевагою, яку надає комп'ютерна технологія, розрізняють 3 етапи розвитку інформаційних систем: 1-й етап (з початку 60-х рр.) характеризувався досить ефективним опрацюванням інформації при виконанні рутинних операцій з орієнтацією на

централізоване колективне використання ресурсів обчислювальних центрів; 2-й етап (з середини 70-х рр.) був пов'язаний з появою персональних комп'ютерів (орієнтація зміщається убік індивідуального користувача для підтримки прийнятих ним рішень); 3-й етап (з початку 90-х рр.) був пов'язаний з поняттям аналізу стратегічних переваг у бізнесі і був заснований на досягненнях телекомунікаційної технології розподіленого опрацювання інформації (інформаційні системи мають своєю метою не просто збільшення ефективності опрацювання даних і допомогу керівнику, але й допомогу організації вистояти в конкурентній боротьбі й одержати перевагу).

За видами інструментарію інформаційної технології розрізняють 5 етапів їх розвитку: 1-й етап (до другої половини XIX ст.) - "ручна" інформаційна технологія, інструментарій якої складали ручка, чорнильниця, книга, а комунікації здійснювалися ручним способом шляхом переправки через пошту листів, пакетів, депеш; 2-й етап (з кінця XIX ст.) - "механічна" технологія, інструментарій якої складали друкарська машинка, телефон, диктофон, оснащена більш досконалими засобами доставки пошта; 3-й етап (40-60-ті рр. XX ст.) - "електрична" технологія, інструментарій якої складали великі ЕОМ і відповідне програмне забезпечення, електричні друкарські машинки, ксерокси, портативні диктофони; 4-й етап (з початку 70-х рр.) - "електронна" технологія, основним інструментарієм якої стають великі ЕОМ і створені на їхній базі автоматизовані системи керування (АСК) і інформаційно-пошукові системи (ІПС), оснащені широким спектром базових і спеціалізованих програмних комплексів; 5-й етап (з середини 80-х рр.) - "комп'ютерна" технологія, основним інструментарієм якої є персональний комп'ютер із широким спектром стандартних програмних продуктів різного призначення, починають широко використовуватися в різноманітних галузях глобальні і локальні комп'ютерні мережі.

Інформаційна технологія, як і будь-яка інша, повинна забезпечувати високий рівень розчленування всього процесу опрацювання інформації на етапи (фази), операції, дії, включати весь набір елементів, необхідних для досягнення поставленої мети, мати регулярний характер.

Технічні засоби виробництва інформації – апаратне, програмне математичне забезпечення цього процесу, яке включає текстовий процесор (редактор), настільні видавничі системи, електронні таблиці, системи керування базами даних, електронні записні книжки, електронні календарі, інформаційні системи функціонального призначення (фінансові, бухгалтерські, для маркетингу та ін.), експертні системи тощо.

За видами серед інформаційних технологій виділяються: інформаційні технології опрацювання даних; інформаційні технології керування; інформаційні технології підтримки прийняття рішень; інформаційні технології експертних систем.

Інформаційна технологія опрацювання даних використовується для розв'язання добре структурованих задач, стосовно яких є необхідні вхідні дані і відомі алгоритми та інші стандартні процедури їх опрацювання. Ця технологія застосовується на рівні операційної (виконавчої) діяльності персоналу невисокої кваліфікації з метою автоматизації деяких рутинних постійно повторюваних операцій управлінської праці. Впровадження інформаційних технологій і систем на цьому рівні істотно підвищує продуктивність праці персоналу, звільняє його від рутинних операцій.

На рівні операційної діяльності вирішуються такі задачі:

- опрацювання даних про операції, які здійснює фірма;
- створення періодичних контрольних звітів про стан справ у фірмі;
- одержання відповідей на всілякі поточні запити й оформлення їх у вигляді паперових документів або звітів.

Прикладом може послужити щоденний звіт про надходження і видачу готівки банком, який формується з метою контролю балансу готівки, або ж запит до бази даних по кадрах, який дозволить одержати дані про вимоги, що висуваються до кандидатів на певну посаду.

Існує декілька особливостей, пов'язаних з опрацюванням даних, що відрізняють дану технологію від усіх інших:

- виконання необхідних фірмі задач по опрацюванню даних. Від кожної фірми закон вимагає наявності та збереження даних про свою діяльність, які можна використовувати як засіб забезпечення і підтримки контролю на фірмі. Тому в будь-якій фірмі обов'язково повинна бути інформаційна система опрацювання даних і розроблена відповідна інформаційна технологія;

- вирішення тільки добре структурованих задач, для яких можна розробити алгоритм;

- виконання стандартних процедур опрацювання. Існуючі стандарти визначають типові процедури опрацювання даних і регламентують їхнє дотримання організаціями усіх видів;

- виконання основного обсягу робіт в автоматичному режимі з мінімальною участю людини;

- використання деталізованих даних. Записи про діяльність фірми мають докладний характер, що допускає проведення ревізій. У процесі ревізії діяльність фірми перевіряється хронологічно від початку періоду до його кінця і від кінця до початку;

- акцент на хронологію подій;

- вимога мінімальної допомоги у вирішенні проблем з боку спеціалістів інших рівнів.

Багато даних на рівні операційної діяльності необхідно зберігати для наступного використання або на цьому ж рівні, або на іншому. Для їхнього збереження створюються бази даних.

Стосовно створення звітів (документів) слід зазначити, що у інформаційній технології опрацювання даних необхідно створювати документи для керівництва і працівників фірми, а також для зовнішніх партнерів. При цьому документи можуть створюватися як за вимогою у зв'язку з проведеною фірмою операцією, так і періодично наприкінці кожного місяця, кварталу або року.

Метою інформаційної технології керування є задоволення інформаційних потреб усіх без винятку співробітників фірми, що мають справу з прийняттям рішень. Вона може бути корисна на будь-якому рівні керування.

Ця технологія орієнтована на роботу в середовищі інформаційної системи керування і використовується при більш поганій структурованості розв'язуваних задач, якщо їх порівнювати з задачами, які розв'язуються за допомогою інформаційної технології опрацювання даних.

Інформаційна технологія керування ідеально підходять для задоволення подібних інформаційних потреб працівників різноманітних функціональних підсистем (підрозділів) або рівнів керування фірмою. Інформація, що поставляється нею, містить відомості про минуле, дійсне і ймовірне майбутнє фірми. Ця інформація має вигляд регулярних або спеціальних управлінських звітів.

Для прийняття рішень на рівні управлінського контролю інформація повинна бути подана в агрегованому вигляді, так, щоб проглядалися тенденції зміни даних, причини відхилень, що виникли, і можливі варіанти вирішення проблеми. На цьому етапі розв'язуються такі задачі опрацювання даних:

- оцінка планованого стану об'єкта керування;

- оцінка відхилень від планованого стану;
- виявлення причин відхилень;
- аналіз можливих рішень і дій.

Інформаційна технологія керування спрямована на створення різноманітних видів звітів.

Регулярні звіти створюються відповідно до встановленого графіка, що визначає час їхнього створення, наприклад місячний аналіз продажів компанії.

Спеціальні звіти створюються за вимогою керівників, або коли в компанії відбулося щось незаплановане. І ті, і інші види звітів можуть мати форму підсумкових, порівняльних і надзвичайних звітів.

У підсумкових звітах дані об'єднані в окремі групи, відсортовані і подані у вигляді проміжних і остаточних результатів по окремих полях.

Порівняльні звіти містять дані, отримані з різноманітних джерел або класифіковані по різноманітних ознаках і використовуються для порівняння.

Надзвичайні звіти містять дані виняткового (надзвичайного) характеру.

Використання звітів для підтримки керування є особливо ефективним при реалізації так званого керування по відхиленнях. Керування по відхиленнях припускає, що головним змістом одержуваних спеціалістом даних повинні бути відхилення стану господарської діяльності фірми від деяких встановлених стандартів (наприклад, від її запланованого стану). При використанні на фірмі принципів керування по відхиленнях до звітів, які створюються, висуваються такі вимоги:

- звіт необхідно створювати тільки тоді, коли відхилення відбулося;
- відомості у звіті повинні бути відсортовані за значенням критичного для даного відхилення показника;
- усі відхилення бажано показати разом, щоб спеціаліст міг уловити існуючий між ними зв'язок;
- у звіті необхідно показати, кількісне відхилення від норми.

Вхідна інформація надходить із систем операційного рівня. Вихідна інформація формується у вигляді управлінських звітів у зручному для ухвалення рішення вигляді. Зміст бази даних за допомогою відповідного програмного забезпечення перетворюється в періодичні і спеціальні звіти, що надходять до спеціалістів, які беруть участь у прийнятті рішень в організації. База даних, що використовується для одержання зазначеної інформації, повинна складатися з двох компонентів: даних, що накопичуються на основі оцінки операцій, проведених фірмою; планів, стандартів, бюджетів та інших нормативних документів, що визначають планований стан об'єкта керування (підрозділи фірми).

Ефективність і гнучкість інформаційної технології підтримки прийняття рішень багато в чому залежать від характеристик **інтерфейсу**. Інтерфейс визначають: мова користувача; мова повідомлень комп'ютера, що організує діалог на екрані дисплея; знання користувача.

Мова користувача - це ті дії, які користувач робить по відношенню до системи шляхом використання можливостей клавіатури; електронних олівців, що пишуть на екрані; джойстика; "миші"; голосових команд, тощо. Найбільш простою формою мови користувача є створення форм вхідних і вихідних документів. Отримавши вхідну форму (документ), користувач заповнює його необхідними

даними і вводить у комп'ютер. Система підтримки прийняття рішень робить необхідний аналіз і видає результати у вигляді вихідного документа заданої форми.

Мова повідомлень - це те, що користувач бачить на екрані дисплея (символи, графіка, колір), дані, надруковані принтером, звукові вихідні сигнали і т.п. Важливим показником ефективності інтерфейсу, який використовується, є обрана форма діалогу між користувачем і системою. В даний час найбільш поширеним є такі форми діалогу: режим "запитання-відповідь", командний режим, режим меню, режим заповнення пропусків у виразах, запропонованих комп'ютером. Кожна форма в залежності від типу задачі, індивідуальності користувача і рішення, яке приймається, може мати свої переваги і недоліки. Довгий час єдиною реалізацією мови повідомлень був надрукований або виведений на екран дисплея звіт або повідомлення. Тепер з'явилася нова можливість представлення вихідних даних - машинна графіка. Вона дає можливість створювати на екрані і папері кольорові графічні зображення в тривимірному вигляді. Використання машинної графіки, яка значно підвищує наочність і інтерпретованість вихідних даних, стає усе більш популярним в інформаційній технології підтримки прийняття рішень.

Знання користувача - це те, що користувач повинен знати, працюючи з системою. До них відносяться не тільки план дій, що знаходиться в голові у користувача, але і підручники, інструкції, довідкові дані, які видає комп'ютер.

Удосконалення інтерфейсу системи підтримки прийняття рішень визначається успіхами у розвитку кожного з трьох зазначених вище компонентів.

Інтерфейс повинен мати такі можливості:

- маніпулювати різноманітними формами діалогу, змінюючи їх у процесі ухвалення рішення за вибором користувача;
- передавати дані до системи різноманітними засобами;
- одержувати дані від різноманітних пристроїв системи в різних форматах;
- гнучко підтримувати (надавати допомогу за вимогою, підказувати) знання користувача.

Найбільший прогрес серед комп'ютерних інформаційних технологій спостерігається у галузі розробки експертних систем. Експертні системи дають можливість спеціалісту одержувати консультації експертів стосовно будь-яких проблем, про які ці системами накопичили знання.

Розв'язання спеціальних задач вимагає спеціальних знань. Проте не кожна компанія може собі дозволити тримати у своєму штаті експертів по всім пов'язаним із її роботою проблемам або навіть запрошувати їх щоразу, коли виникає якась проблема. Головна ідея використання технології експертних систем полягає в тому, щоб одержати від експерта його знання і, загрузивши їх у пам'ять комп'ютера, використовувати їх кожного разу, коли в цьому виникає необхідність. Все це надає можливість використовувати технологію експертних систем у якості систем, що дають поради.

Подібність інформаційних технологій, які використовуються в експертних системах і системах підтримки прийняття рішень, проявляється в тому, що обидві вони забезпечують високий рівень підтримки прийняття рішень.

Основними компонентами інформаційної технології, яка використовується в експертній системі, є: інтерфейс користувача, база знань, інтерпретатор, модуль створення системи.

Спеціаліст використовує інтерфейс для введення інформації і команд в експертну систему та одержання вихідної інформації з неї. Команди містять у собі параметри, що спрямовують процес опрацювання знань. Інформація звичайно видається у формі значень, що присвоюються певним змінним.

Технологія експертних систем передбачає можливість одержувати в якості вихідної інформації не тільки рішення, але і необхідні пояснення.

Розрізняють два види пояснень:

- пояснення, що видаються за вимогою. Користувач у будь-який момент може зажадати від експертної системи пояснення своїх дій;
- пояснення отриманого рішення проблеми. Після одержання рішення користувач може зажадати пояснень того, як воно було отримано. Система повинна пояснити кожний крок своїх міркувань, що ведуть до розв'язання задачі. Хоча технологія роботи з експертною системою не є простою, інтерфейс користувача цих систем є дружнім і звичайно не викликає труднощів при веденні діалогу.

База знань містить факти, що описують проблемну галузь, а також логічний взаємозв'язок цих фактів. Центральне місце в базі знань належить правилам. Правило визначає, що варто робити в даній конкретній ситуації, і складається з двох частин: умова, яка може виконуватися або ні, і дія, яку варто виконати у випадку виконання умови.

Всі правила, які використовуються в експертній системі, утворюють систему правил, яка навіть для відносно простої системи може містити у собі декілька тисяч правил.

Інтерпретатор виконує у певному порядку опрацювання даних, які знаходяться в базі знань. Технологія роботи інтерпретатора зводиться до послідовного розгляду сукупності правил (правило за правилом). Якщо має місце дотримання умови, що міститься в правилі, то виконується певна дія, і користувачу надається варіант вирішення його проблеми.

Крім того, у багатьох експертних системах вводяться додаткові блоки: база даних, блок розрахунку, блок введення і коректування даних. Блок розрахунку необхідний у ситуаціях, пов'язаних із прийняттям управлінських рішень. При цьому важливу роль грає база даних, де містяться планові, фізичні, розрахункові, звітні та інші постійні або оперативні показники. Блок введення і коректування даних використовується для оперативного і своєчасного відображення поточних змін у базі даних.

Модуль створення системи служить для створення набору (ієрархії) правил. Існують два підходи, що можуть бути покладені в основу модуля створення системи: використання алгоритмічних мов програмування і використання оболонок експертних систем.

Для представлення бази знань спеціально розроблені мови Лісп і Пролог, хоча можна використовувати і будь-яку відому алгоритмічну мову.

Оболонка експертних систем являє собою готове програмне середовище, що може бути пристосоване для вирішення певної проблеми шляхом створення відповідної бази знань. У більшості випадків використання оболонок дозволяє створювати експертні системи швидше і легше в порівнянні з програмуванням.

Для інформаційних технологій є цілком природним те, що вони застарівають і замінюються новими. Так, наприклад, на зміну технології пакетного опрацювання програм на великий ЕОМ в обчислювальному центрі прийшла технологія роботи

на персональному комп'ютері на робочому місці користувача. Телеграф передав усі свої функції телефону. Телефон поступово витісняється службою експрес-доставки. Телекс передав більшість своїх функцій факсу й електронній пошті.

При впровадженні нової інформаційної технології в організації необхідно оцінити ризик відставання від конкурентів у результаті її неминучого старіння, тому що інформаційні продукти, як ніякі інші види матеріальних товарів, мають надзвичайно високу швидкість змінюваності новими видами або версіями. Періоди змінюваності коливаються від декількох місяців до одного року. Якщо в процесі впровадження нової інформаційної технології цьому фактору не приділяти належної уваги, цілком можливо, що до моменту завершення переходу фірми на нову інформаційну технологію вона вже застаріє і прийдеться вживати заходів щодо її модернізації. Такі невдачі з впровадженням інформаційних технологій звичайно пов'язані з недосконалістю технічних засобів, в той час як основною причиною невдач є відсутність або слабка пропрацьованість методології використання інформаційної технології, зокрема, мережних.

Централізоване опрацювання інформації на ЕОМ обчислювальних центрів було першою історично сформованою технологією. Створювалися великі обчислювальні центри колективного користування, оснащені великими ЕОМ. Застосування таких ЕОМ дозволяло опрацьовувати великі масиви вхідної інформації й одержати на цій основі різноманітні види інформаційної продукції, яка потім передавалася користувачам. Такий технологічний процес був обумовлений недостатнім оснащенням обчислювальною технікою підприємств і організацій у 60 - 70-і рр.

Децентралізоване опрацювання інформації пов'язане з появою в 80-х рр. персональних комп'ютерів і розвитком засобів телекомунікацій. Вона дуже істотно потіснила попередню технологію, оскільки дала користувачу широкі можливості в роботі з інформацією без обмеження його ініціатив.

Переваги і недоліки централізованої і децентралізованої інформаційної технології призвели до необхідності притримуватися лінії розумного застосування і того, і іншого підходу.

Такий підхід називають **раціональною методологією**. За цією методологією:

- центр обробки інформації (обчислювальний центр) повинен відповідати за створення загальної стратегії використання інформаційної технології, допомагати користувачам як у роботі, так і у навчанні, установлювати стандарт і визначати політику застосування програмних і технічних засобів;
- персонал, який використовує інформаційну технологію, повинен дотримуватися вказівок обчислювального центру, здійснювати розробку своїх локальних систем і технологій відповідно до загального плану організації.

Раціональна методологія використання інформаційної технології дозволить досягти більшої гнучкості, підтримувати загальні стандарти, здійснити сумісність інформаційних локальних продуктів та знизити дублювання діяльності.

Програмне забезпечення інформаційних технологій має недоліки, якими можуть скористатися сторонні особи (хакери) і використати їх на свою користь. Для попередження несанкціонованого доступу використовуються дуже дорогі системи захисту, а також вдосконалюється програмне забезпечення.

При використанні програмного забезпечення існує можливість втрати інформації, спричинена дією вірусів, які використовують його недоліки. У зв'язку з тим, що

вартість інформації росте, втрати можуть бути суттєвими. Для захисту доводиться використовувати спеціальні програми – антивіруси. Беручи до уваги те, що зараз відбувається концентрація у сфері інформаційних технологій, перед користувачем постає дилема вибору платформи інформаційної технології, так як в майбутньому він буде залежати від свого постачальника програмного забезпечення.

Інформаційна технологія може бути впроваджена на об'єкті управління двома способами: в локальні інформаційні структури, які засновані на адаптації нової інформаційної технології до діючої організаційної структури; в глобальні (корпоративні) інформаційні структури, які передбачають удосконалення докорінним чином діючої організаційної структури управління.

Перший спосіб впровадження нової інформаційної технології приводить до локального удосконалення методів обробки інформації, які діють на об'єкті управління і не міняють діючу організаційну структуру управління. Комунікації розвинуті недостатньо, то ж раціоналізуються тільки методи збору і обробки інформації на робочих місцях фахівців шляхом організації окремих автоматизованих робочих місць. У цьому випадку здійснюється розподіл обов'язків поміж оператором ПЕОМ (технічним працівником) і фахівцем. При цьому відбувається зливання операцій по збиранню і обробці первинної (інколи з умовно-постійною інформації, а в ряді випадків і з функцією прийняття управлінських рішень).

За другим способом впровадження нової інформаційної технології удосконалюється організаційна структура управління таким чином, щоб ця технологія принесла як найбільшу ефективність управлінню виробництвом. У цьому випадку відбувається розвиток комунікацій і розробляються нові інформаційні і організаційні взаємозв'язки, створюються комплекси взаємозв'язаних АРМ керівників і фахівців різних рівнів, впроваджуються розподілені бази даних. При цьому способом продуктивність організаційної структури управління збільшується за рахунок раціонального розподілення обробки інформації, відбувається зниження обсягів інформаційних потоків по каналах зв'язку, приближення обробленої інформації до користувачів різних рівнів, збирання і обробка первинної інформації в ритмі виробництва, яка передається відповідним користувачам в такому ж ритмі.

Таким чином, перший спосіб впровадження нової інформаційної технології орієнтований на діючу організаційну структуру управління. При цьому ступінь ризику від впровадження цієї технології зводиться до мінімуму, так як зроблені затрати незначні і організаційна структура не змінювалась. Другий спосіб спрямований на принципову зміну організаційної структури управління в бік її удосконалення. Однак для обох способів принципово змінюються методи використання обчислювальної, периферійної і організаційної техніки: замість централізованого впроваджується децентралізований збір і первинна обробка інформації, процес управління наближається до місць, де виникає і використовується первинна (фактична) інформація. На практиці і в літературі ці методи використання одержали назву розподілених обчислювальних систем, в яких поряд з централізованою обробкою інформації на великих і середніх ЕОМ застосовується децентралізована, тобто розподілена обробка інформації з використанням мікро-ЕОМ (ПЕОМ).

Розподілена обробка інформації дозволяє прискорити збирання і обробку її за рахунок максимального наближення засобів обчислювальної і периферійної

техніки до місць, де виникає ця інформація, обробляється і використовується керівниками і фахівцями різних рівнів, більш ефективно задовольнити різноманітні часто мінливі інформаційні потреби управлінського персоналу, забезпечити необхідною інформацією процеси прийняття рівень для оперативного управління, знизити затрати на утримання всієї обчислювальної системи, збільшити гнучкість і підвищити живучість системи (вихід однієї ПЕОМ не приведе до відмови всієї системи), безпосередньо приймати участь керівникам різних рівнів в процесі управління (в діалоговому режимі) і підвищити їх відповідальність за прийняття (або не прийняття) управлінських рішень; поступово (модульно) створювати нову систему за рахунок зміни окремих периферійних технічних засобів (добавлення нових, в тому числі і ПЕОМ, зв'язуючи їх у єдину систему – мережу).

Щоб організувати системну автоматизовану обробку економічної інформації, яка використовується для управління об'єктом, потрібно створити таку систему, в якій достовірна первинна (фактична) інформація, яка характеризує виробничо - господарську і іншу діяльність, один раз в мінімальному складі в ритмі виробництва фіксувалась би і в такому ж ритмі передавалась і оброблювалась (разом з умовно-постійною) на ЕОМ таким чином, щоб одержана результатна інформація повністю задовольняла усім вимогам управління об'єктом, у тому числі інформуванню керівників і фахівців різних рівнів, а також процесам формування і планування, обліку і контролю, складання установленої зведеної звітності і проведення комплексного економічного аналізу за різні періоди часу і по різних структурних ланках об'єкта управління.

Зрозуміло, що основу такого способу обробки інформації складає мережа персональних комп'ютерів, офісної техніки та засобів комунікації.

Зміни в інформаційних технологіях обумовили нове відношення до інформації. Щоб зрозуміти це, потрібно чітко розрізняти повідомлення (або послання), інтерпретацію (або сприйняття) і комунікацію.

Повідомлення (message) - це "річ" або ж "матеріалізований продукт" інтелектуальної діяльності людини. *Інтерпретація* - це "думка", тобто знання, що одержується.

Комунікація - це лише операція передачі, трансляції. Сьогодні в комп'ютерних мережах саме ця операція трансляції стала визначальною ланкою в тріаді "повідомлення" – "комунікація" – "інтерпретація". Дивлячись на сучасних політиків, біржових брокерів, менеджерів, неважко зауважити, що більше поінформована людина - це не та, яка найбільше знає, а та, яка здійснює найбільше число своєчасних комунікацій. Тільки інформація як комунікація, а не, як знання або предмет, спроможна викликати нові операції. Це відбувається тому, що інформація не стільки ресурс, скільки мотив діяльності. Тому в сучасному суспільстві інформація - це двигун та основа діяльності, тоді, як у традиційному суспільстві, вона ніяк не могла претендувати на таку роль. Отже, на сьогодні в інформації найважливіше - це комунікація, що спонукає до дії. Головним феноменом нашої епохи стали не гігантські електронні банки даних і штучний інтелект, а Internet, у якому не створюється ніякого знання, але який зате багаторазово збільшує можливості здійснення комунікацій.

Мережею як засобом комунікації охоплено всі серйозні суб'єкти господарської діяльності економічно розвинених країн. Електронне спілкування через мережу стало незамінним засобом ділового спілкування, перевищивши інформативність звичайного.

Мета створення, функціонування та широкого розповсюдження мережних технологій полягає у вирішенні проблем інформатизації суспільства і всієї життєдіяльності в країні. Під інформатизацією суспільства будемо розуміти повсюдне впровадження комплексу заходів, які спрямовуються на забезпечення повного і своєчасного використання достовірної інформації, узагальнених знань у всіх соціально значимих видах людської діяльності. Ці заходи забезпечать не тільки зростання економічних показників розвитку народного господарства, але і отримання якісних наукових досягнень у фундаментальних та прикладних науках, які спрямовані на розвиток виробництва, створення нових робочих місць, підвищення життєвого рівня населення, захист навколишнього середовища.

Інформатизація через мережні технології – об'єктивний і закономірний етап, через який в тій чи іншій формі повинно пройти кожне суспільство, яке вступило на шлях інтенсивного розвитку. Сучасна інформатизація - це глобальний процес, який пов'язаний з кардинальними змінами структури і характеру світового і соціального розвитку, з переходом до нових поколінь наукомістких технологій, технічних систем, матеріалів, а також нових видів інформаційного обміну, які дозволяють вирішальним чином змінювати характер праці і умови життя людини.

Інформатизація в усіх сферах діяльності людини дозволяє активно і широко застосовувати все багатство інтелектуального наслідування і уроків розвитку нашого суспільства. З цією метою все ширше розгортається використання інформаційних технологій в усіх сферах людської діяльності, як от проведення референдумів, опитування суспільної думки за різними аспектами життя суспільства, впровадження інформаційно-довідкових систем в різноманітних галузях (наприклад, транспорту, зв'язку, торгівельних організацій наявність товарів у формі систем замовлень товарів чи послуг для виявлення, обліку і використання ресурсів).

Інформаційні технології стали широко застосовуватись в сфері відпочинку і розваг. Використання електронних систем в бібліотеках, музеях, відеофільмотеках і створення для будь-яких груп населення інтелектуальних відеоігор відкриває якісно нові форми творчого розвитку людини, збереження багатств вітчизняної та світової культури. Інформаційні технології в галузі медицини дозволяють підвищити якість медичного обслуговування.

В галузі освіти вони створюють умови для безперервної освіти, розвитку індивідуальних форм і інтенсифікації технологічних процесів навчання. Циклічний розвиток поколінь техніки стає причиною необхідності оперативно підвищувати кваліфікацію і змінювати сферу професійної діяльності. Доступність міжнародної системи баз даних і знань для кожного члена суспільства дозволить безперервно виявляти і розвивати творчі здатності індивідуума.

В сфері матеріального виробництва створюються і впроваджуються інтегровані системи автоматизованою проектування, виробництва і контролю, які охоплюють весь життєвий цикл виробу “науковий пошук - проектування - виробництво - експлуатація - модернізація - зняття виробу з виробництва“. Крім того застосування інформаційних технологій дозволяє ефективно проводити роботи з екологічного контролю., так наприклад, регіональна система моніторингу, яка включатиме засоби експрес - інформації про стан і динаміку водних об'єктів, повітряне середовище, ґрунту, системи моделювання метеорологічних процесів і вплив широкомасштабних виробничих комплексів на оточуюче середовище,

експертні системи діагностики і прогнозування процесів забруднення, очистки та зруйнування, а також відновлення екологічного середовища забезпечує, можливість формування єдиного екологічного паспорта регіону, дозволить оперативно передбачати і тим самим передбачати екологічні катастрофи.

Відбуваються значні зміни в управлінні організаціями, модернізуються їхні структури управління, йде перерозподіл повноважень та відповідальності. Почала втілюватися концепція “управління інформаційними ресурсами“, в якій інформація розглядається як ще один важливий ресурс такого ж порядку, як фінанси, матеріали, обладнання і персонал. Будь-якому підприємству потрібна інформація про ринки збуту і заготівель – галузева чи загальноекономічна, про природні ресурси, науково-технічна, соціально-культурна, політико-правова інформація про реакцію навколишнього середовища на дії підприємства, інформація про реакцію підприємства на дії навколишнього середовища. Такий хід розвитку обумовлює зростання ролі інформатизації на всіх рівнях управління організацією. Успіх комерційної і підприємницької діяльності безпосередньо пов’язується з муніципальними, банківськими, біржовими інформаційними системами, інформатизацією оптової і роздрібною торгівлі, торгових домів, служб управління працею і зайнятістю, створенням банків даних ринку товарів і послуг, розвитком центрів довідкової та аналітико-прогнозної інформації, електронної пошти, систем електронного обміну даними і, так далі., як правило, робота розглянутих систем базується на локальних обчислювальних мережах різноманітної архітектури або їх об’єднаннях.

Сьогодні значний прошарок керівників організацій набув широких знань в сфері інформатики, сформовано новий еталон працівників організацій, які вже природно ставляться до застосування нових інформаційних технологій в управлінні. Цим, зокрема, пояснюється той факт, що керівництво організацій проявляє розуміння в необхідності застосування інформаційних технологій і саме це обумовлює зацікавленість у застосуванні інтегрованих інформаційних систем управління. Такі системи дозволяють на основі інформації, що поступає і обробляється в реальному режимі часу, приймати ефективні управлінські рішення.

Найновіші досягнення в галузі мікроелектроніки привели до нових концепцій організації інформаційних служб. Завдяки новим розробкам компактною і високопродуктивною і економічною комп’ютерною технікою інформаційно-обчислювальні ресурси органічно вписуються на робочих місцях менеджерів, бухгалтерів, плановиків, адміністраторів, інженерів та інших категорій інтелектуальних працівників. Удосконалюються персональні системи обробки даних, автоматизовані робочі місця на базі персональних комп’ютерів.

Користувачам інформації України наданий доступ до віддалених інформаційних ресурсів, закордонних інформаційних мереж. Все це створює надійну основу для широкого залучення сучасних інформаційних технологій до управління виробництвом.

Темпи впровадження та застосування сучасних інформаційних технологій найчастіше обумовлюються фінансовим становищем підприємств. Проте успішна робота будь-якої організації в сучасних умовах швидкого науково-технічного прогресу неможлива без серйозного відношення до впровадження інформаційних технологій, а також вирішення питань інформатизації різних сфер діяльності, в першу чергу в сферах удосконалення самоорганізації та підтримки виробничої діяльності підприємства..

РОЗДІЛ 5

ЗАСТОСУВАННЯ ІНТЕРНЕТ В ЕКОНОМІЦІ

Internet - найбільша глобальна комп'ютерна мережа, що зв'язує десятки мільйонів абонентів у більш як 150 країнах світу. Щомісяця її поширеність зростає на 7-10%. Internet забезпечує взаємодію інформаційних мереж, що належать різним установам у всьому світі. Якщо раніше вона використовувалася виключно як середовище для передачі файлів і повідомлень електронної пошти, то сьогодні вирішуються більш складні завдання, які підтримують функції мережного пошуку та доступу до розподілених інформаційних ресурсів й електронних архівів. Таким чином, Internet можна розглядати як деякий глобальний інформаційний простір.

Мережа Internet, що служила спочатку дослідницьким і навчальним групам, стає все популярнішою в ділових колах. Компанії спокушують дешевий глобальний зв'язок і його швидкість, зручність для проведення сумісних робіт, доступні програми, унікальна база даних цієї мережі. Вони розглядають глобальну комп'ютерну мережу як доповнення до своїх власних локальних мереж. Уже кілька років розвиваються і встигли широко ввійти в практику в розвинених країнах технології Intranet, що є інформаційними технологіями "великої" мережі в корпоративних мережах і навіть у дуже невеликих мережах ПК підприємств малого бізнесу. При низькій вартості послуг (часто це тільки фіксована щомісячна плата за лінії зв'язку або телефон) користувачі можуть дістати доступ до комерційних і некомерційних інформаційних служб США, Канади, Австралії, європейських країн, а тепер уже України та Росії. В архівах вільного доступу мережі Internet можна знайти інформацію практично з усіх сфер людської діяльності, починаючи з нових наукових відкриттів до прогнозу погоди на завтра. В Internet можна знайти рекламу багатьох тисяч фірм і розмістити (часто безкоштовно!) свою рекламу. Крім того, Internet надає унікальні можливості дешевого, надійного та конфіденційного глобального зв'язку. Це виявляється дуже зручним для фірм, що мають свої філіали по всьому світу, транснаціональних корпорацій і структур управління. Як правило, використання інфраструктури Internet для міжнародного зв'язку коштує набагато дешевше від прямого комп'ютерного зв'язку через супутниковий канал або телефон.

Стандарти Internet забезпечують можливість групової роботи над спільним проектом за допомогою електронної пошти, гіпертекстових документів (служба WWW), а також за допомогою теле-, аудіо- і навіть відеоконференцій у масштабі реального часу. Для забезпечення інформаційної безпеки в мережі застосовуються різні протоколи шифрування конфіденційної інформації, електронні підписи, сертифікація інформації. Заборона на несанкціоноване переміщення даних між локальною мережею підприємства і глобальною мережею може забезпечуватися спеціальними комп'ютерами або програмами (брандмауерами).

Варіант зв'язку між комп'ютерами в локальній мережі з підключенням до Інтернет показано на рисунку 7: робочі місця у комп'ютерних класах з'єднані між собою та через концентратори приєднані до центрального серверу, який має вихід на Інтернет-провайдера.

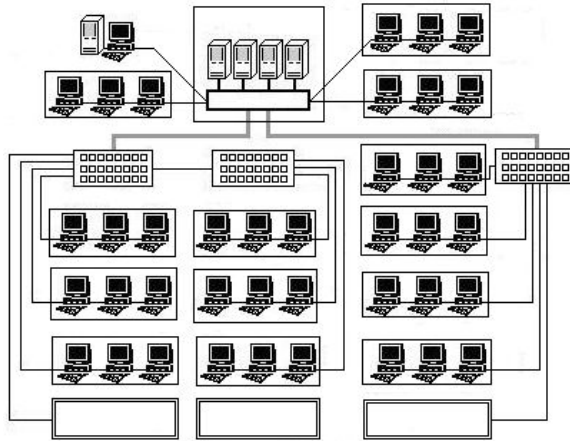


Рис.7

Основні мережні сервіси

Практично всі послуги мережі Internet побудовані на принципі "клієнт-сервер".

Сервер (у мережі Internet) - це комп'ютер або програма, здатні надавати клієнтам (у міру надходження від них запиту) деякі мережні послуги.

Клієнт - прикладна програма, завантажена в комп'ютер користувача, яка забезпечує передачу запитів до сервера й одержання відповідей від нього.

Різні сервіси мають різні прикладні протоколи. У міру розвитку мережі з'являються нові протоколи (сервіси), змінюючи її вигляд і стрімко розширюючи коло користувачів. Таким чином, щоб скористатися якоюсь із служб мережі Internet, необхідно встановити на комп'ютері клієнтську програму, здатну працювати за протоколом цієї служби. Деякі клієнтські програми входять до складу ОС Windows 98, NT, а також до складу програм-браузерів, наприклад, Microsoft Internet Explorer та Netscape Communicator. Розглянемо деякі сервіси, які забезпечує Internet.

Сервіс FTP (File Transfer Protocol). Це протокол передачі файлів, один із перших сервісів Internet. Цей сервіс дає можливість абоненту обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі. Для вузлів FTP характерною є наявність процедури входу (login). При цьому користувачеві надається доступ до безкоштовно поширюваної інформації. Даний сервіс може бути використаний для комерційного поширення програмних продуктів, баз даних, моделей, рекламних презентацій, великих за обсягом документів (книг) тощо.

Сервіс E-mail (електронна пошта). Електронна пошта є одним із перших і, мабуть, найпоширенішим сервісом Internet. Цей сервіс забезпечує обмін поштовими повідомленнями з будь-яким абонентом мережі Internet. Існує можливість відправлення як текстових, так і двійкових файлів. Електронна пошта є найдешевшим і доступним Internet-сервісом в Україні. Можна навести такі переваги електронної пошти в організації ділової діяльності:

- реалізується дешево і майже моментальне розсилання;
- не витрачається час на візити до посадових осіб із дрібних питань;
- не треба переписувати (передруковувати) копії для розсилання;
- дуже просто використати цитати, відповідаючи на повідомлення;
- архів листування зберігається в комп'ютері в зручному вигляді;
- можна задавати списки розсипки, псевдоніми (alias), вести адресні записники;
- можна передавати двійкові файли (схеми, ілюстрації, програми, архіви тощо).

Поштові сервери одержують повідомлення від клієнтів і пересилають їх по ланцюжку до поштових серверів адресатів, де ці повідомлення накопичуються. При встановленні сполучення між адресатом і його поштовим сервером, за командою відбувається передача повідомлень, що надійшли на комп'ютер адресата. Серед клієнтських поштових програм можна виділити The Bat, Microsoft Outlook та інші.

Сервіс Mail Lists (списки розсилки). Його створено на підставі протоколу електронної пошти. Підписавшись (безкоштовно) на списки розсилки, можна регулярно одержувати електронною поштою повідомлення про певні теми (науково-технічні й економічні огляди, презентація нових програмних та апаратних засобів і т. д.).

Сервіс Usenet (групи новини або телеконференції). Він забезпечує обмін інформацією (повідомлення, статті) між усіма, хто користується ним. Це щось на зразок електронної дошки оголошень, на яку будь-який бажаючий може помістити своє повідомлення, і воно стає доступним для всіх інших. Цей сервіс дає змогу поширювати й одержувати комерційну інформацію, дізнаватися про новини ділового світу. Новини поділяються за темами на групи, що якоюсь мірою їх упорядковує. На певні групи можна оформити підписку і періодично, як і електронною поштою, одержувати всі повідомлення, що проходять за темою групи. Для реалізації цього сервісу існують клієнтські програми, наприклад, Microsoft Internet News.

Сервіс WWW (World Wide Web - всесвітня павутина). WWW - це єдиний інформаційний простір, який складається із сотень мільйонів взаємозв'язаних гіпертекстових електронних документів, що зберігаються на Web-серверах. Окремі документи всесвітньої павутини називаються Web-сторінками. Групи тематично об'єднаних Web-сторінок утворюють Web-вузол (жаргонний термін - Web-сайт, або просто сайт). Web-сторінка - це текстовий файл, що містить опис зображення мультимедійного документа на мові гіпертекстової розмітки - HTML (Hyper-Text Markup Language). Сторінка може містити не тільки форматований текст, а й графічні, звукові та відео об'єкти.

Найважливішою рисою Web-сторінок є *гіпертекстові посилання*. З будь-яким фрагментом тексту або, наприклад, із малюнком, можна пов'язати інший Web-документ, тобто встановити гіперпосилання. У цьому разі під час клацання лівою клавішею миші на тексті або рисунку, що є гіперпосиланням, відправляється запит на доставку нового документа. Цей документ, у свою чергу, також може мати гіперпосилання на інші документи. Таким чином сукупність величезного числа гіпертекстових електронних документів, які зберігаються в серверах WWW, утворює своєрідний гіперпростір документів, між якими можливе переміщення.

Для передачі інформації у WWW використовується протокол HTTP (HyperText Transfer Protocol - протокол передачі гіпертексту). Перегляд Web-сторінок і переміщення через посилання користувачі здійснюють за допомогою програм браузерів (від слова "to browse" - переглядати). Найпопулярнішими Web-браузерами в Україні є Microsoft Internet Explorer та Netscape Communicator.

Сервіс IRC (Internet Relay Chat). Він забезпечує проведення телеконференцій у режимі реального часу. Переваги: можна анонімно поговорити на цікаву тему або швидко одержати консультацію. На відміну від системи телеконференцій, в якій спілкування між учасниками обговорення теми відкрито для всього світу, в системі IRC беруть участь, як правило, лише кілька чоловік. Іноді службу IRC називають чат-конференціями, або просто чатом.

Існує кілька популярних клієнтських програм для роботи з серверами і

мережами, що підтримують сервіс IRC, наприклад, програми mIRC і mIRC32 для Windows. Ці, а також подібні до них програми застосовуються для ділового й особистого спілкування персоналу фірм у реальному часі, для проведення групових консультацій і нарад.

Служба ICQ. Вона призначена для пошуку мережної IP-адреси людини, комп'ютер якої приєднано в даний момент до мережі Internet. Назва служби є акронімом виразу I seek you - я тебе шукаю. Необхідність у подібній послугі пов'язана з тим, що більшість користувачів не мають постійної IP-адреси. Їм видається динамічна IP-адреса, що діє тільки протягом цього сеансу. Цю адресу видає той сервер, через який відбувається приєднання. У різних сеансах динамічна IP-адреса може бути різною, причому заздалегідь невідомо якою. При кожному приєднанні до мережі Internet програма ICQ, встановлена на комп'ютері користувача, визначає поточну IP-адресу і повідомляє його центральній службі, яка, в свою чергу, оповіщає партнерів користувача. Далі партнери (якщо вони також є клієнтами цієї служби) можуть встановити з користувачем прямий зв'язок. Після встановлення контакту зв'язок відбувається в режимі, аналогічному сервісу IRC.

Сервіс Telnet (віддалений доступ). Він дає можливість абоненту, працювати на будь-якому комп'ютері мережі Internet, як на своїй власній. Часто використовується режим роботи - доступ до віддаленого сервера бази даних. У минулому цей сервіс також широко використовувався для проведення складних математичних розрахунків на віддалених суперкомп'ютерах. У наші дні у зв'язку зі швидким збільшенням обчислювальної потужності ПК необхідність у подібній послугі скоротилася, але служби Telnet у мережі Internet продовжують існувати. Часто протоколи Telnet застосовують для дистанційного керування технічними об'єктами, наприклад, телескопами, відеокамерами, промисловими роботами. Прикладом програми, що реалізує доступ до Telnet-сервісу, може бути програма Net Term.

Доступ користувачів до мережі Internet

Для роботи в мережі необхідно:

- фізично приєднати комп'ютер до одного з вузлів мережі Internet;
- одержати IP-адресу на постійній або тимчасовій основі;
- встановити і настроїти програмне забезпечення - програми-клієнти тих сервісів, послугами яких має намір скористатися.

Організаційно доступ до мережі користувачі дістають через провайдери.

Провайдер - це організація (юридична особа), що надає послуги у приєднанні користувачів до мережі Internet.

Як правило, провайдер має постійно ввімкнений досить продуктивний сервер, сполучений з іншими вузлами каналами з відповідною пропускною здатністю, і засоби для одночасного підключення кількох користувачів (багатоканальний телефон, багатопортова плата тощо).

Провайдери роблять подібну послугу на договірній основі, найчастіше орієнтуючись на час роботи користувача або обсяг даних, які пересилаються по мережі. При укладанні договору провайдер повідомляє клієнту всі атрибути, необхідні для підключення та настройки з'єднання (ідентифікатори, номери телефонів, паролі тощо). Як правило, користувачі навчальних закладів, великих організацій, фірм, підприємств приєднуються до мережі Internet через свою локальну мережу. На один із комп'ютерів локальної мережі покладається

вирішення завдань проху-сервера - управління локальною мережею й виконання функцій "посередника" між комп'ютерами користувачів та мережею Internet (проху - представник, довірена особа).

Всі технічні й організаційні питання взаємодії з провайдером вирішує адміністратор мережі. Для користувачів розробляється інструкція, в якій наводиться перелік дій, які треба виконати для приєднання до мережі Internet. Технічно для приєднання до комп'ютера провайдера потрібні ПК, відповідне програмне забезпечення й модем - пристрій, що перетворює цифрові сигнали від комп'ютера на сигнали для передачі по телефонних лініях і навпаки. Комп'ютер провайдера може виконувати функції хост-машини або звертатися до більш потужних хост-машин для доступу до глобальних ресурсів мережі Internet через високопродуктивний канал передачі даних - магістраль.

Хост-машина (від англ. host - господар) - це комп'ютер, що виконує мережні функції, реалізуючи повний набір протоколів. Крім мережних функцій, хост-машина може виконувати завдання користувача (програми, розрахунки, обчислення). Деякі хост-машини можуть виконувати функції шлюзів - апаратних і програмних засобів для передачі даних між несумісними мережами, наприклад, між мережею Internet та мережами FidoNet. Роль шлюзу між мережею Internet і локальними мережами відіграє проху-сервер.

Система адрес у мережі Internet

Адреси потрібні для ідентифікації об'єктів, які можуть цікавити користувача в мережі. Найчастіше такими об'єктами є вузли мережі (сайти), поштові скриньки, файли, Web-сторінки. Для кожного з них існує свій формат адреси. Однак, оскільки об'єкти зосереджено у вузлах мережі, в їхніх ідентифікаторах обов'язково присутня адреса вузла.

Для ідентифікації вузлів і маршрутизації пакетів служить IP-адреса. **IP-адреса** - це чотирибайтне число, перших два байти якого визначають адресу підмережі, а два інших - адресу вузла в ній. За допомогою IP-адреси можна ідентифікувати більш як 4 млрд. вузлів. На практиці ж через особливості адресації до деяких типів локальних мереж кількість можливих адрес становить понад 2 млрд. Для користувача працювати з числовим зображенням IP-адреси незручно, тому йому пропонується більш проста логічна система доменних імен DNS (Domain Name System) - послідовність імен, сполучених крапками, наприклад, microsoft.com, rambler.ru, itl.net.ua, lviv.ua і т.д.

Домен - група вузлів, об'єднаних за деякою ознакою (наприклад, вузли навчальних закладів, вузли якої-небудь країни, вузли однієї організації і т. д.). Система доменів має ієрархічну деревоподібну структуру, тобто кожний домен проміжного рівня містить групу інших доменів. Кореневий домен є умовним, на верхньому рівні можуть бути розташовані початкові (територіальні) домени різних країн. Ім'я вузла (машини) становить нижній рівень доменного імені та позначається крайнім лівим доменом.

Наведемо доменні імена деяких країн та організацій: us- США, au- Австралія, fr- Франція, ca- Канада, jp- Японія, ru- Росія, ua- Україна, de- Німеччина, com- комерційні організація, edu - навчальні заклади, gov - урядові установи, net - постачальники мережних послуг, org - неприбуткові організації. Слід зазначити, що IP та DNS - різні форми запису адреси одного й того самого мережного комп'ютера.

Для переведення доменних імен у IP-адресу служить сервіс DNS.

Для ідентифікації ресурсів мережі (файлів, Web-сторінок) використовується адреса URL (Uniform Resource Locator - уніфікований покажчик ресурсу), яка складається з трьох частин:

- Зазначення сервісу, що забезпечує доступ до ресурсу (як правило, це ім'я протоколу). Після імені йдуть двокрапка: і два знаки / (коса риска): <http://...> ;
- Зазначення DNS імені комп'ютера: <http://www.itl.net.ua...> ;
- Зазначення повного шляху доступу до файлу на даному комп'ютері: <http://www.itl.net.ua/Faes/Arcbiv/pagel.html>

Як роздільник у повному імені використовується знак /. Вводячи ім'я, потрібно точно дотримувати регістр символів, оскільки в Internet малі та великі літери вважаються різними.

В електронній пошті адреса складається з імені одержувача (поштової скриньки), знака "@" та доменної адреси поштового сервера (локальної мережі), до якого приєднано одержувача. Наприклад: kovalenko@polynet.lviv.ua .

Интернет-адреси (довідково)

<http://info.riprn.net/relam/members/vmo/workop.html/> - гранти та стипендії.

<http://www.expressinform.com/> - політична, економічна інформація по Україні.

<http://www.liga.kiev.ua/> - законодавство України.

<http://www.search.global.epnet.com/> - електронний формат наукових журналів.

<http://altavista.digital.com//> - пошукова система.

<http://rambler.ru/> - пошукова система.

<http://weblist.gu.net/> - каталог українських Web ресурсів

<http://www.meta-ukraine.com/> - пошукова система.

<http://www.uar.net/> - Українська Наукова та освітня мережа.

<http://gpntb.ippi.ras.ru/> - державна публічна науково-технічна бібліотека (Росія).

<http://www.nbu.gov.ua/> - національна бібліотека України.

<http://www.economica.ru/> - віртуальна бібліотека з економіки.

<http://www.nau.kiev.ua/> - нормативні акти України

<http://www.rada.kiev.ua/> - верховна Рада України.

<http://www.rainbow.gov.ua/> - сайт Ради національної безпеки України.

<http://www.un.org/> - організація Об'єднаних Націй.

<http://www.unesco.org/> - сайт ЮНЕСКО.

<http://www.europa.eu.com/> - сайт Європейського Союзу.

<http://www.csr.ru/> - сайт центру стратегічних розробок.

<http://www.niss.gov.ua/> - інститут стратегічних досліджень (Україна).

<http://www.catalog.da.ru/> - бізнес, фінанси, економіка.

Телеконференція (сервіс Usenet news)

Так називають групи новин, наділених здатністю передаватися між комп'ютерами (серверами новин) за спеціальним протоколом NNTP. Користувач через провайдера, отримавши доступ до сервера новин, вибирає потрібну йому групу новин та знайомиться з повідомленнями і відповідями на них. За бажанням користувач може і сам надіслати відповідь на будь-яке повідомлення або ж надіслати повідомлення, на яке може надійти відповідь з будь-якого куточка світу. Повідомлення разом з відповідями на них носять спеціальну назву – потоки. При надсиланні нового повідомлення до групи новин створюється новий потік, а при відповіді на повідомлення відповідь додається до існуючого потоку.

Головні групи новин:

- Comp – апаратне і програмне забезпечення комп'ютерів,

- Sci – наукові дослідження,
- Soc – соціальні теми,
- News – мережні новини,
- Rec – розваги,
- Talk – розмови будь-які теми,
- Misc – різноманітні теми,
- Alt – інші теми.

Повідомлення мають свою адресу. Наприклад: soc.culture.ukraine (новини соціальної тематики, що стосуються культури народу України). До більшості груп повідомлення потрапляють відразу після відправлення, але існують групи новин, надходження повідомлень до яких контролюється: до таких груп новин повідомлення можуть потрапити із деяким запізненням у часі, або не потрапити взагалі.

Телеконференції створюють на багатьох Web-сторінках. Їх називають форумами. Для участі в них досить клацнути на кнопці сайту з відповідною назвою.

Правила поведінки учасників телеконференцій:

- перед відправленням повідомлення слід з'ясувати, чи відповідає воно тематиці групи новин;
- повідомлення має бути коротким і зрозумілим;
- текст повинен бути витриманим з точки зору граматики;
- не варто направляти повідомлення до групи, якщо немає впевненості щодо прийняття там положень, які висловлені;
- повідомлення мають викликати загальний інтерес.

Найбільш розповсюджені програми які підтримують участь у телеконференції: Microsoft Outlook Express, Netscape Communicator, Forte Agent.

Інтерактивне спілкування

Через електронну пошту та групи новин відбувається спілкування і асинхронному режимі, коли між відправленням повідомлення і отриманням відповіді проходить значний проміжок часу. Інтернет надає можливість спілкуватися по-іншому – синхронно, у режимі реального часу голосом, відео зображеннями чи текстом.

Для участі у спілкуванні голосом необхідно підключення мікрофону, динаміка, відеокамери тощо. Більш дешевими є текстові спілкування – чати. Спілкування відбувається у групі, яка поєднує співрозмовників за певним напрямком. Кожен користувач має можливість створити для спілкування власний канал, контролюючи правила спілкування: ввічливість, лаконізм, конфіденційність.

Служби, які забезпечують спілкування – IRC, ICQ. Перша являє собою групу глобальних та локальних мереж: Eenet, UnderNet, DALnet, RUSnet. Для користування ними необхідно мати програму-клієнт, наприклад mIPS. Після активізації цієї програми вибирається канал, читаються та уводяться повідомлення, які, власне, і являють собою спілкування. Іншою програмою для цього є Microsoft Chat. Служба ICQ надає можливість при реєструванні отримати свій оригінальний номер та, знаючи номери інших співрозмовників, викликати їх на спілкування і спілкуватися обміном текстових повідомлень.

Internet-економіка

В останні роки в інформаційній індустрії відокремлюється ще одна, поки ще не дуже масштабна за обсягами, але найперспективніша структура – цифрова, або

Internet-економіка. Нині світ бурхливо переживає черговий бум – зсунення акцентів з комунікаційної і інформаційно-пошукової функцій Internet на ведення з її допомогою сучасного бізнесу. Це відбувається через властивість мережних технологій докорінно змінювати спосіб взаємодії між людьми і компаніями, методи дослідницької діяльності, купівлі-продажу тощо. Крім того, глобальна мережа стала неперевершеним засобом для проведення маркетингу і здійснення прямих он-лайнних продажів, підвищення рівня обслуговування клієнтів, найпотужнішим інструментом управління фірмою і корисним джерелом інформації для наукових і практичних розробок. Internet породжує нові форми соціальної і економічної діяльності людей: телероботу, віртуальні підприємства, дистанційне навчання та інші.

Протягом найближчого десятиріччя глобальна тенденція використання можливостей Internet для бізнесу вплине на десятки секторів світової економіки. Комунікаційні технології змінюють саму сутність бізнес-моделей – базових процесів створення продуктів і послуг виробниками та надання їх кінцевим споживачам.

Якщо західні компанії практично вичерпали усі резерви традиційних способів підвищення ефективності і з допомогою запровадження переваг Internet-економіки борються за чергові кілька відсотків зростання, у наших національних підприємств можливості більші. Україна й інші країни на пострадянському просторі можуть отримати значні переваги від запровадження Internet-технологій ведення бізнесу, а саме:

- від створення єдиних територіально-інформаційних систем для автоматизації управління діяльністю корпоративних підприємств;
- від доступу до експортного ринку, до інвестиційного західного капіталу;
- від створення робочих місць для кваліфікованої робочої сили;
- від зростання податкових надходжень від застосування електронних платежів тощо.

В будь-якому разі, цифрова економіка сформувалася як світове явище і формується як галузь національної української економіки, що потребує певного ставлення науковців до досліджень у цій галузі.

Необхідність роботи з документами, матеріалами, базами даних конкретної організації або установи вдома, у готелі, у транспортних засобах привела до появи так званих віртуальних офісів. Вони базуються на роботі локальної мережі, з'єднаної з територіальною або глобальною мережею. Абонентські системи працівників установ незалежно від місця їхнього перебування вмикаються в спільну для всіх мережу.

Останнім часом мова вже йде про віртуальні підприємства, основу якої складає електронний бізнес, побудований на спільних діях бізнес-процесу в особі бізнесмена та комп'ютера або іншого автоматизованого засобу зв'язку з обміну інформацією.

Використання Internet (а також Intranet та Extranet) спільно з новітніми інформаційними технологіями роблять віртуальний бізнес привабливим і прибутковим.

Отримали право на існування так звані віртуальні підприємства як нова форма економічної організації. Структурно таке підприємство складається з територіально роз'єднаних фірм чи співробітників, що обмінюються продуктами своєї праці і спілкуються винятково електронними засобами при мінімальному або цілком відсутньому особистому контакті. Це вже не просто департаменти, відділи, цехи і групи, а сукупність бізнес-процесів, і робота організовується навколо бізнес-процесів, тому що не товари, а процеси їх створення приносять компаніям довгостроковий успіх.

Internet- трейдинг

Internet-трейдинг (internet trading) – це спосіб віддаленого доступу до торгів на біржових майданчиках (на валютній, фондовій або товарній біржі) з використанням інтернету як засобу зв'язку.

Загалом вважається, що інтернет-торгівля розпочалася в 90-их роках, коли доступ до фондових ринків став можливим завдяки закритим брокерським мережам, тож тоді американські фінансисти та інвестори познайомилися з терміном «онлайн-торгівля». Та все ж вперше про онлайн-трейдинг заговорили в 1971 році, коли американська біржа Nasdaq провела першу угоду в мережі. Останнім часом такий спосіб торгівлі отримав дуже широке поширення і в Україні.

Інтернет-трейдинг, як інструмент керування інвестиціями на торгівельному майданчику, являє собою механізм, що надає можливість інвестору особисто проводити угоди з покупки-продажу на біржі в режимі реального часу. Інтернет-торгівля ведеться через онлайн-брокера – професійного учасника торгівельного майданчика з ліцензією на брокерську діяльність (торгівця валютою, цінними паперами, ...). Онлайн-брокер – це зв'язкова ланка між трейдером (керуючим інвестиційним портфелем) та фондовим майданчиком (біржею).

Існує два основних способи надання брокерських послуг через Інтернет:

– коли як технічні інструменти торгівлі використовуються браузерні системи (інвестиційний портфель складається безпосередньо на Web-сайті компанії-посередника, при цьому використовується звичайні Web-браузери й використання іншого додаткового програмного забезпечення не вимагається, доступ здійснюється з любого терміналу (ПК, КПК, планшету, мобільного телефону та т.ін., рис.8) ;

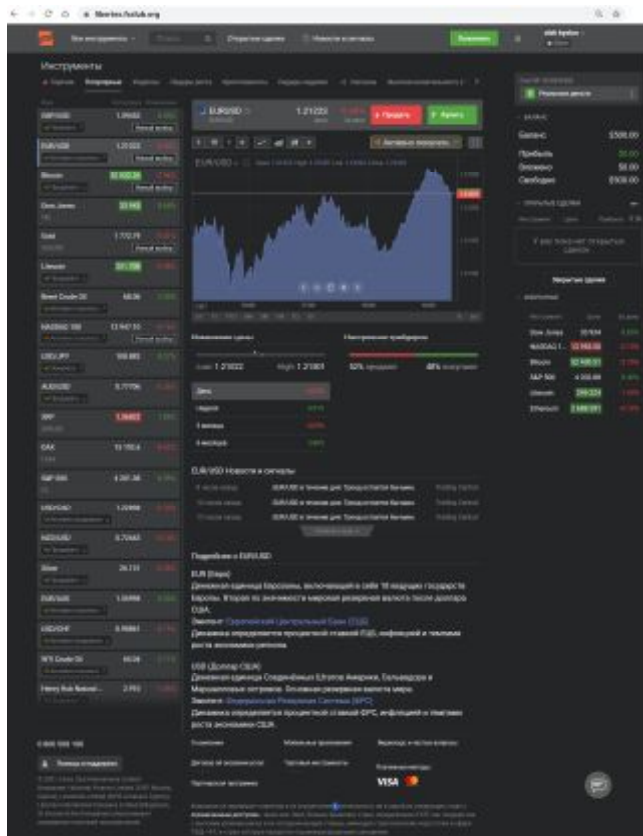


Рис.8

– коли як технічні інструменти торгівлі використовуються торгівельні платформи (для забезпечення надання брокерських послуг використовуються інсталяційні системи - спеціалізоване програмне забезпечення засобами якого надається інформація та проводяться транзакції, при цьому користувач такого програмного забезпечення має можливість налагоджувати програмне забезпечення в залежності від особистих вподобань під потреби реального часу з можливістю проведення повної або часткової автоматизації процесу торгівельної діяльності, рис.9).



Рис.9

Інтернет-трейдинг - найбільш швидко зростаюча технологія в області економіки. Ця технологія комерції, що породжує практично повну автоматизацію оформлення і проведення угод, при відсутності необхідності утримувати офіси і фахівців, з можливостями практично необмеженого числа клієнтів і угод. Інтернет-трейдинг ставши основою сучасного фондового ринку, являє собою інструмент, що дозволяє отримувати ціни фінансових активів (цінних паперів, валют, індексів, товарних контрактів), а також здійснювати операції з їх купівлі-продажу через інтернет. Це дає можливість брати безпосередню участь у біржових торгах не тільки професіоналам, але і всім зацікавленим особам. Інвестор може самостійно керувати своїми фінансовими активами в режимі реального з будь-якого куточку світу часу лише за допомогою спеціального програмного забезпечення без посередництва.

Перевагою інтернет-трейдингу як технології є оперативність з якою можна проводити прийняті рішення (присутнім на декількох торгівельних майданчиках та миттєва швидкість виконання). Недоліком інтернет-трейдингу є більш високі ризики, миттєві дії вимагають підвищеної концентрації та уваги,

В даний час можна виділити три напрямки розвитку проектів інтернет-трейдингу: пошук нових споживачів послуг, розробка комплексних інтернет-технологій обслуговування клієнтів та створення нових форм фінансового бізнесу. В Україні інтернет-трейдинг менш розвинений, ніж в Європі або США.

РОЗДІЛ 6

ОРГАНІЗАЦІЯ КОМП'ЮТЕРНОЇ БЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Види комп'ютерних злочинів, суб'єкти їх вчинення та міри запобігання

В умовах науково-технічного прогресу інформація стає об'єктом специфічних суспільних відносин, що виникають з приводу її створення, накопичення, зберігання, обробки та використання, набуваючи товарного вигляду у формі інформаційного ресурсу. Формування, обробка та використання такого ресурсу здійснюється в процесі інформатизації різних галузей людської діяльності шляхом застосування сучасних інформаційних технологій. Однак розширення застосування таких технологій призводить не лише до позитивних наслідків: реалії сьогодення зумовлюють поширення правопорушень у сфері інформації, зокрема, такого явища як інформаційні злочини. Наявність останніх ставить актуальним питання інформаційної безпеки та захисту від злочинів і зловживань, скоєних в інформаційній сфері.

Тут і далі по тексту йдеться про безпеку та захист, які стосуються лише зловживань, що вчиняються з використанням інформаційних технологій. Суб'єктами вчинення діянь, що порушують безпеку та змушують вдаватися до захисту інформації є, як правило, співробітники фінансово-господарських установ та особи, що не працюють в цих установах - хакери, фрікери, крєкери, кардери.

Хакерами вважаються особи, які проникають в систему без корисливих цілей.

Фрікери (українською мовою “частотники”) - це особи, які займаються розшифруванням супутникових сигналів з метою безкоштовного перегляду платних (кодованих) телевізійних каналів, “переадресуванням” мобільних телефонів з метою їх безоплатного використання.

Крєкери - це різновид хакерів-правопорушників, що вчиняють протиправні діяння з корисливою метою або з хуліганських мотивів. Крєкери по своїй суті нічим не відрізняються від звичайного злодія, що зламує чужі квартири і краде чужі речі. Крєкери зламують чужі обчислювальні системи і крадуть чужу інформацію, а при доступі до банківських рахунків чи систем бухгалтерії підприємств - чужі гроші або матеріальні цінності. Саме в цьому виражається відмінність між тими, кого називають хакерами і крєкерами: перші – дослідники комп'ютерних мереж, інколи фахівці з економічної безпеки підприємств, другі – просто злодії чи злодії.

Кардери – особи, що розкрадають кошти з чужих кредитних карток. Кардер встановлює номер того чи іншого власника кредитної картки, може замовити по цих картках товари по телефону або через комп'ютерну мережу. До них можна віднести також виготовлення чи підроблення фальшивих кредитних карток, за допомогою яких також розкрадаються кошти.

Серед найбільш небезпечних комп'ютерних злочинів: емісія фіктивних безготівкових коштів; використання накопичених безготівкових коштів; складання псевдо підприємницьких контрактів; придбання на фіктивні кошти реальних матеріальних цінностей; конвертація безготівкових коштів у готівкову вільноконвертовану валюту; підміна даних бухгалтерського обліку; моделювання ситуацій; підприємницьке шпигунство.

Зазначені злочини (як і інші злочинні дії в інформаційній сфері) також можуть бути вчинені фізичним (безпосередньою дією людиною) чи технічним

(організація витоку інформації технічними каналами) способами. Під каналами витоку інформації слід розуміти джерела інформації, середовища розповсюдження сигналів та апаратуру зйому інформації.

У сучасних умовах широкого розповсюдження набуває технічний спосіб незаконного збирання інформації шляхом використання інформаційних мереж як глобальних (Internet), так і локальних. Досить різноманітними є також засоби вчинення злочину технічними каналами. Для досягнення злочинної мети шляхом зйому інформації за допомогою технічних засобів застосовуються: акустичні засоби (спрямовані або вмонтовані мікрофони, вібродатчики), лазерні засоби зйому мовної інформації, засоби зйому інформації з дротів та комунікацій, засоби перехоплення побічних випромінювань, закладні пристрої, комп'ютерна техніка, в тому числі її інформативні частини.

Входження України у глобальні міжнародні комп'ютерні системи і особливо "INTERNET" – один з чинників появи і зростання кількості названих вище правопорушень, пов'язаних з використанням електронно-обчислювальних машин. Система дозволяє мати доступ до оригінальних інформаційних матеріалів у всьому світі, нею можуть користуватись не лише вчені, педагоги, бізнесмени, але й інші особи, які не мають до неї доступу, в тому числі й згадані вище хакери, фрікери, крєкери, кардери.

Виходячи зі сказаного, комп'ютерна безпека сьогодні стає пріоритетним напрямком діяльності не лише правоохоронних органів, але й кожного користувача інформаційних систем.

З метою запобігання можливості комп'ютерного злочину необхідно втілювати в практику:

- захист від несанкціонованого доступу до системи;
- резервування особливо важливих комп'ютерних підсистем;
- організація обчислювальних мереж з можливістю перерозподілу ресурсів у разі порушення працездатності окремих ланок;
- вжиття конструктивних заходів захисту від крадіжок, шантажу тощо;
- встановлення резервних систем електроживлення;
- обладнання приміщень, в яких знаходяться комп'ютери та комп'ютерна інформація системами сигналізації;
- обладнання приміщень технікою захисту від спроб копіювання інформації з зовні або сторонніми особами;
- організацію охорони обчислювальної техніки;
- створення особливого перепускного режиму в приміщеннях, де знаходиться комп'ютерна техніка або комп'ютерна інформація;
- ретельний добір персоналу;
- виключення випадків проведення особливо важливих робіт лише однією людиною;
- наявність типових планів дій керівників та обслуговуючого персоналу при виявленні порушень у роботі комп'ютерних мереж, спробах несанкціонованого доступу до інформації, пожежах, стихійному лиху тощо;
- розробка функціональних обов'язків робітників, допущених до роботи та обслуговуванню комп'ютерної техніки, комп'ютерної інформації;
- наявність плану відновлення працездатності обчислювального центру після виходу його з ладу;
- організація обслуговування та контроль за роботою сторонньої

організації або особи, запрошених для обслуговування приміщень, техніки, обладнання комп'ютерних центрів;

- універсальність засобів захисту від усіх користувачів (в тому числі вищих посадових осіб та співробітників контрольних, правоохоронних органів);
 - встановлення відповідальності та її розмежування між особами, які повинні забезпечувати безпеку комп'ютерної інформації;
 - вибір місця розташування обчислювального (комп'ютерного) центру тощо;
 - періодичне обстеження приміщень, в яких розміщена комп'ютерна техніка з метою виявлення встановлених невідомими особами пристроїв, що пристосовані для можливого зняття інформації, доступу до неї;
 - розробка законів і нормативних актів, які передбачали б відповідальність за комп'ютерні злочини;
 - захист авторських прав;
 - контроль за розробками комп'ютерних систем.
- Забезпечення безпеки комп'ютерних мереж можна без перебільшення назвати однією із найважливіших задач, що стоять перед розробниками та користувачами мережних технологій.

Сучасні моделі та методи організації безпеки комп'ютерів у мережах

Розрізняють три рівні, на яких може здійснюватися атака на електронний інформаційний ресурс: термінали, локальні мережі і, нарешті, сама мережа Інтернет.

Виходячи з контексту даного розділу, обмежимося розглядом питань, що стосуються безпеки окремих комп'ютерів, залучених до Інтернету, тому що саме ці місця є найбільше актуальними для користувачів, що працюють із ресурсами World Wide Web. Тут, у свою чергу, має сенс проводити розмежування між захистом даних, що утримуються безпосередньо в комп'ютері, включеному в мережу, і захистом даних, що відправляються за допомогою Інтернету (зокрема, електронної пошти).

Однією з найпоширеніших моделей обходу захисних механізмів, передбачених у сучасних браузерях, є атака віддаленого комп'ютера за допомогою особливих вірусів, що одержали назву "троянських коней". Типовий "троянський кінь" складається з двох частин, що називають відповідно "сервером" і "клієнтом". "Сервер" у даному випадку - це програма, що завантажується в пам'ять інфікованого комп'ютера і спроможна одержувати і виконувати команди, що надходять через Інтернет від віддаленого "клієнта - зловмисника". Команди, що може виконувати сервісний компонент, залежать від того, який саме "троянський кінь" проник у комп'ютер. Один із підкласів "троянських коней" призначений для передачі "клієнту" мережних паролів користувачів; інші види "троянців" спроможні завдати ще більшу шкоду: наприклад, відслідковувати інформацію, виведену на монітор комп'ютера, копіювати вміст вінчестера і навіть форматувати його. Проникнення "троянських коней" у комп'ютери може стати наслідком безтурботності при роботі з Інтернетом. Користувач найчастіше і не здогадується про те, який "данайський дарунок" одержав він разом з програмою, що йому сподобалася, завантаженої з мережі, або разом із отриманим на перший погляд незагрозливим електронним листом. Антивірусне програмне забезпечення, включаючи спеціалізовані програми, названі "антигенами", далеко не завжди дають можливість цілком позбутися таких "дарунків".

Інший метод, взятий на озброєння тими, хто намагається проникнути в чужі

комп'ютери, полягає у використанні особливостей організації доменної системи імен - DNS., як відомо, для навігації у всесвітньому павутинні достатньо знати лише мнемонічно легко запам'ятовуюче ім'я мережного комп'ютера, до якого звертаються. Цьому імені відповідає 32-розрядне число - так звана IP-адреса, яка також дозволяє однозначно ідентифікувати кожний комп'ютер в мережі, але який, як правило, не відомий користувачу. Оскільки при передачі даних на мережному рівні використовуються винятково IP-адреси, виникає задача перетворення в ці адреси імен, привласнених комп'ютерам. Спочатку, коли число залучених до Інтернету комп'ютерів було порівняно невеликим, пари імен і IP-адрес централізовано заносилися в спеціальний файл, доступний із мережі. Проте, з розвитком Інтернету від цього підходу довелося відмовитися.

Для вирішення задачі перетворення була створена окрема система, у яку ввійшли спеціалізовані інформаційно-пошукові сервери, які діють на основі мережного протоколу, що одержав назву DNS. Доменна система імен у тому вигляді, в якому вона була спроектована, виявилася не спроможною забезпечити безпеку. Зокрема, у ній є суттєві вади в організації захисту від перехоплення запитів, що посилаються на сервер, який відноситься до доменної системи імен. Цим може скористатися зловмисник, здійснивши переадресування відповідних запитів на власний комп'ютер, який машина користувача буде приймати за "безпечний" сервер DNS. У результаті комп'ютер користувача може виявитися мішенню для впровадження шкідливої інформації.

Існують і інші типи "атаки через Інтернет", пов'язані з недосконалістю програмного забезпечення. У цьому зв'язку варто згадати мережну атаку, здійснену у свій час студентом Корнельського університету Робертом Моррісом. Розроблений Моррісом особливий різновид комп'ютерного вірусу, що одержав назву хробак, використовуючи пролом в операційній системі UNIX і в утиліті електронної пошти Sendmail, зумів за короткий час вразити понад 6000 машини, у результаті чого стратегічні комп'ютерні мережі США виявилися на декілька днів виведеними з ладу.

Які ж заходи безпеки потрібно приймати для того, щоб зводити до мінімуму мережну загрозу?

Перше за все, це використання антивірусних програм: AntiVir, Nod32, DrWeb, Avast тощо.

Поміж тим, слід знати, що антивірусні програми дозволяють виявляти й усувати найпоширеніші віруси, проте цілком надіятись на ці програми не можна.

Перед роботою в мережі варто переконатися в правильній конфігурації серверу. Використання опцій протоколювання, відслідковування за допомогою спеціальних програм стану відкритих портів і вмісту системного реєстру є подальшими кроками в напрямку підвищення безпеки роботи в мережі.

Важливо не забувати про те, що причиною прикостей найчастішою стає інформація, що завантажується з Інтернету. Тому по можливості варто відмовитися від завантаження програм із ненадійних джерел і виявляти граничну обережність при роботі з електронною поштою, зокрема з додатками електронних листів.

Найфундаментальнішими є рішення, застосовувані для захисту локальних мереж, що мають доступ до всесвітнього павутиння. Для забезпечення безпеки мереж використовуються різноманітні технології, такі, як системи виявлення атак і міжмережні екрани.

При організації практичної діяльності по забезпеченню безпеки виникає складна для користувача задача вибору відповідних технічних засобів у відповідності з конкретними обставинами. Тому, приступаючи до вирішення цієї складної задачі, необхідно максимально використати конкретні умови експлуатації апаратури і можливі стратегії протидіючої сторони.

Аналіз опублікованих в останній час матеріалів дозволяє окреслити такі основні напрямки несанкціонованих дій:

- модифікація програмного забезпечення шляхом непомітного додавання нових функцій;
- отримання несанкціонованого доступу, тобто порушення секретності або конфіденційності інформації;
- видавання себе за іншого користувача для того, щоб зняти з себе відповідальність, або ж використати його повноваження;
- відмова від факту отримання інформації, яка насправді була отримана, або брехливі відомості про час її отримання;
- відмова від факту формування інформації;
- твердження про те, що адресату в певний момент часу була послана інформація, яка насправді не посилалась;
- твердження про те, що інформація була отримана від деякого користувача, хоча насправді вона сформована самим же порушником;
- несанкціоноване розширення своїх законних повноважень;
- несанкціонована зміна інших користувачів (свідомо недостовірний запис інших осіб, обмеження або розширення існуючих повноважень);
- під'єднання до лінії зв'язку між іншими користувачами в якості активного ретранслятора;
- приховування фактів наявності деякої закритої інформації (прихована передача) в іншій відкритій інформації;
- вивчення того, хто, коли і до якої інформації отримує доступ;
- заява про сумнівність протоколу забезпечення безпеки зв'язку, підкріплена розкриттям певної інформації, яка згідно умов протоколу повинна залишатись секретною;
- примусове порушення протоколу за допомогою введення свідомо недостовірної інформації;
- піддрив довіри до протоколу шляхом введення свідомо недостовірної інформації.

Сучасна технологія забезпечення безпеки зв'язку рекомендує всю роботу по захисту інформації з врахуванням розглянутих стратегій протидіючої сторони проводити по наступних напрямках:

- вдосконалення організаційних і організаційно-технічних заходів;
- блокування несанкціонованого доступу до оброблюваної та інформації, що передається;
- блокування несанкціонованого отримання інформації з допомогою технічних засобів.

Будь-яка робота по забезпеченню безпеки зв'язку в каналах телекомунікації повинна починатися з організаційних таких заходів захисту:

- встановлення персональної відповідальності за забезпеченням захисту інформації;
- обмеження доступу в приміщення, де проходить підготовка і обробка інформації;

- доступ до обробки, збереження і передачі конфіденційної інформації тільки перевіреним посадовим особам;
- призначення конкретних зразків технічних засобів для обробки цінної інформації і подальша робота тільки на них;
- збереження магнітних носіїв, жорстких копій і реєстраційних матеріалів в закритих міцних шафах (бажано в сейфах);
- виключення перегляду сторонніми особами змісту оброблюваної інформації за рахунок невідповідного встановлення дисплея, клавіатури, принтера тощо;
- постійний контроль пристроїв виводу цінної інформації на фізичні носії;
- збереження цінної інформації на зовнішніх носіях тільки в засекреченому вигляді;
- використання криптографічного закриття при передачі по каналах зв'язку цінної інформації;
- знищення фізичних носіїв або матеріалів, що можуть містити фрагменти цінної інформації;
- заборона ведення переговорів про безпосередній зміст цінної інформації особам, які зайняті її обробкою;
- чітка організація робіт і контроль виконання.

Врахувати специфіку каналу обліку і методу передачі або обробки інформації дозволяють організаційно-технічні заходи, які при цьому не вимагають для своєї реалізації нестандартних прийомів і обладнання:

- організація електроживлення обладнання, що обробляє цінну інформацію, від окремого джерела живлення і від загальної електромережі через стабілізатор напруги (мережний фільтр) або мотор-генератор (бажаніше);
- обмеження доступу сторонніх осіб всередину в приміщення, в якому знаходиться обладнання, шляхом встановлення механічних заборів або замків;
- при обробці і ввіді-виводі інформації використовувати для відображення дисплеї, а для реєстрації – принтери;
- при відправці в ремонт технічних засобів обов'язкове знищення всієї інформації, що містяться на них;
- розміщення обладнання для обробки цінної інформації на віддалі не менше 2,5 м від пристроїв освітлення, конденсації, зв'язку, металевих труб, теле- і радіоапаратури, а також іншого обладнання, що використовується для обробки цінної інформації;
- встановлення клавіатури і друкарських пристроїв на м'які прокладки з метою зниження відходу інформації по акустичному каналу;
- при обробці цінної інформації на персональному комп'ютері, крім випадку передачі цієї інформації по мережі, відключення комп'ютера від локальної мережі або мережі віддаленого доступу;
- знищення інформації після її використання або передачі.

Блокування несанкціонованого отримання інформації за допомогою технічних засобів є досить складним завданням, а його вирішення вимагає суттєвих матеріальних затрат. Тому перед тим, як прийняти конкретні заходи, необхідно проаналізувати стан справ і врахувати наступні рекомендації:

- найнадійнішим методом обмеження електромагнітного випромінювання є повне екранування приміщення, в якому знаходяться засоби обробки і передачі цінної інформації. Екранування здійснюється сталевими або алюмінієвими листами або листами із спеціальної пластмаси товщиною не менше 2 мм з надійним заземленням;

- при обробці цінної інформації основним джерелом високочастотного електромагнітного випромінювання є дисплей персонального комп'ютера. Необхідно пам'ятати, що зображення з його екрану можна приймати на віддалі кількох сотень метрів. Повністю нейтралізувати відхід інформації з екрану можна, лише використовуючи генератор шуму. Для обробки особливо важливої інформації рекомендується використання плазмових або рідкокристалічних дисплеїв;

- джерелом потужного низькочастотного випромінювання є друкуючий пристрій. Для блокування відходу інформації в цьому випадку рекомендується використати зашумлення або термодрук чи струменевий принтер;

- у багатьох випадках існують небезпечні наводки на провідники, що виходять за межі приміщення, яке підлягає охороні. Необхідно слідкувати, щоб всі з'єднання обладнання із “зовнішнім світом” здійснювалися через електричну розв'язку;

- дуже небезпечні спеціально внесені в схему обладнання обробки і зв'язку мікропередавачі або радіомаяки (закладки). Тому, після повернення обладнання з ремонту необхідно переконатися, що в ньому немає подібних закладок. Не рекомендується обробляти цінну інформацію на випадкових персональних комп'ютерах;

- якщо у вас з'явилися сумніви відносно безпеки інформації, запросіть спеціалістів - вони визначать можливий канал витоку і запропонують ефективні заходи для захисту.

При виборі технічних засобів захисту інформації потрібно враховувати наступні фактори:

- режим шифрування-дешифрування повинен бути простим і зручним для санкціонованого користувача;

- ефективність і надійність алгоритму шифрування не повинна залежати від змісту інформації, що передається;

- не варто віддавати перевагу тим системам, в яких криптографічні алгоритми є комерційною таємницею організації-розробника. Набагато краще, коли алгоритм відомий до деталей і відповідає деякому стандарту, а необхідний рівень стійкості визначається, наприклад, довжиною ключа;

- аналогові скремблери не забезпечують гарантований захист переговорів, оскільки в каналі зв'язку присутні частини вихідного аналогового сигналу. Використання їх має зміст лише в тих випадках, коли використання цифрових засобів захисту мови неможливе або економічно не вигідне.

Програмний захисту інформації від комп'ютерних вірусів

Оптимальне вирішення складної проблеми забезпечення безпеки зв'язку можливе лише при комплексному підході з використанням, як організаційних, так і технічних заходів. Досягнення сучасної мікроелектроніки, обчислювальної техніки і методів криптографічного перетворення дозволяють оптимістично дивитись на перспективи забезпечення безпеки зв'язку. Цьому сприяє і основна тенденція розвитку сучасних систем зв'язку – перехід до цифрових методів обробки інформації, які забезпечують безпеку зв'язку за рахунок високої стійкості криптографічного перетворення.

Один із варіантів захисту інформації – виявлення і знешкодження комп'ютерного вірусу. Для виявлення, знищення і захисту від комп'ютерних вірусів розроблено кілька видів спеціальних програм, що дозволяють виявляти й знищувати віруси. Такі програми називаються антивірусними. Розрізняють такі

види антивірусних програм:

- програми-детектори;
- програми-доктори, або фаги;
- програми-ревізори;
- програми-фільтри;
- програми-вакцини, або імунізатори.

Програми-детектори здійснюють пошук характерної для конкретного вірусу сигнатури в оперативній пам'яті й у файлах і при виявленні видають відповідне повідомлення. Недоліком таких антивірусних програм є те, що вони можуть знаходити тільки ті віруси, що відомі розробникам таких програм.

Програми-доктори, або фаги, а також програми-вакцини не тільки знаходять заражені вірусами файли, але й «лікують» їх, тобто видаляють із файлу тіло програми-вірусу, повертаючи файли у вихідний стан. На початку своєї роботи фаги шукають віруси в оперативній пам'яті, знищуючи їх, і тільки потім переходять до «лікування» файлів. Серед фагів виділяють поліфаги, тобто програми-доктори, призначені для пошуку й знищення великої кількості вірусів. Найвідоміші з них: Kaspersky Antivirus, Norton Antivirus, Doctor Web.

Але нові віруси постійно з'являються, тому програми-детектори і програми-доктори швидко застарівають і необхідно регулярно оновлювати версії.

Програми-ревізори належать до найнадійніших засобів захисту від вірусів. Ревізори запам'ятовують вихідний стан програм, каталогів і системних областей диска тоді, коли комп'ютер не заражений вірусом, а потім періодично або за бажанням користувача порівнюють поточний стан із вихідним. Виявлені зміни виводяться на екран монітора. Як правило, порівняння станів здійснюють відразу після завантаження операційної системи. При порівнянні перевіряються довжина файлу, код циклічного контролю (контрольна сума файлу), дата й час модифікації, інші параметри. Програми-ревізори мають досить розвинуті алгоритми, виявляють стелс-віруси і можуть навіть очистити версії перевірених програм від змін, внесених вірусом. До числа програм-ревізорів належить широко розповсюджена в країнах СНД програма Kaspersky Monitor.

Програми-фільтри, або «сторожі», являють собою невеликі резидентні програми, призначені для виявлення підозрілих дій при роботі комп'ютера, характерних для вірусів. Такими діями можуть бути:

- спроби корекції файлів із розширеннями COM, EXE;
- зміна атрибутів файлу;
- прямий запис на диск за абсолютною адресою;
- запис у завантажувальні сектори диска;
- завантаження резидентної програми.

При спробі якої-небудь програми здійснити вказані дії «сторож» посилає користувачеві повідомлення і пропонує заборонити або дозволити відповідну дію. Програми-фільтри досить корисні, тому що здатні знайти вірус на ранній стадії його існування, до розмноження. Однак вони не «лікують» файли й диски. Для знищення вірусів треба застосувати інші програми, наприклад фаги. До недоліків програм-сторожів можна віднести їх «настирливість» (наприклад, вони постійно видають попередження про будь-яку спробу копіювання виконуваного файлу), а також можливі конфлікти з іншим програмним забезпеченням.

Вакцини (імунізатори) – це резидентні програми, що запобігають зараженню файлів. Вакцини застосовують, якщо відсутні програми-доктори, що знищують цей вірус. Вакцинація проводиться тільки проти відомих вірусів. Вакцина модифікує програму або диск у такий спосіб, щоб це не позначалося на їхній роботі, а вірус буде сприймати їх зараженими і тому не проникне. Сьогодні програми-вакцини мають обмежене застосування.

Своєчасне виявлення заражених вірусами файлів і дисків, повне знищення виявлених вірусів на кожному комп'ютері дозволяють уникнути поширення вірусної епідемії на інші комп'ютери.

Основні заходи для захисту від вірусів

Для того щоб не піддати комп'ютер зараженню вірусами і забезпечити надійне зберігання інформації на дисках, необхідно дотримуватися таких правил:

- обладнайте свій комп'ютер сучасними антивірусними програмами, наприклад Kaspersky Antivirus, і постійно оновляйте їхні вірусні бази;
- перед зчитуванням із дискет інформації, записаної на інших комп'ютерах, завжди перевіряйте ці дискети на наявність вірусів, запускаючи антивірусні програми свого комп'ютера;
- при переносі на свій комп'ютер файлів в архівованому вигляді перевіряйте їх відразу ж після розархівування на жорсткому диску, обмежуючи область перевірки тільки наново записаними файлами;
- періодично перевіряйте на наявність вірусів жорсткі диски комп'ютера, запускаючи антивірусні програми для тестування файлів, пам'яті й системних областей дисків із захищеної від запису дискети, попередньо завантаживши операційну систему із захищеної від запису системної дискети;
- завжди захищайте свої дискети від запису при роботі на інших комп'ютерах, якщо на них не буде здійснюватися запис інформації;
- обов'язково робіть архівні копії на дискетах цінної для вас інформації;
- не залишайте в кишені дисководу А дискети при вмиканні або перезавантаженні операційної системи, щоб запобігти зараженню комп'ютера завантажувальними вірусами;
- використовуйте антивірусні програми для вхідного контролю усіх виконуваних файлів, одержуваних із комп'ютерних мереж;
- щоб забезпечити більшу надійність, застосування антивірусу необхідно поєднувати з повсякденним використанням ревізора диска.

Віруси й Інтернет – правила роботи у мережах

Спочатку найпоширенішим способом зараження вірусами були дискети, тому що саме з їхньою допомогою переносилися програми між комп'ютерами. Інтернет призвів до появи ще одного каналу поширення вірусів, за допомогою якого вони часто обходять традиційні методи боротьби з ними.

Імовірність зараження вірусами пропорційна частоті появи нових файлів або додатків на комп'ютері. Зміни в конфігурації для роботи в Інтернеті, для читання електронної пошти й завантаження файлів із зовнішніх джерел – усе це збільшує ризик зараження вірусами.

Чим більше значення комп'ютера або даних, що знаходяться в ньому, тим ретельніше треба дбати про заходи безпеки проти вірусів. Потрібно також врахувати і

витрати на видалення вірусів з ваших комп'ютерів, а також із комп'ютерів ваших клієнтів, яких ви можете заразити. Витрати не завжди обмежуються тільки фінансами, репутація організації й інші речі теж мають значення.

Важливо також пам'ятати, що віруси зазвичай з'являються в системі внаслідок дій користувача (наприклад, інсталяції програми, читання файлу через FTP, читання електронного листа). Тому з метою профілактики треба звертати особливу увагу на обмеження на завантаження потенційно заражених програм і файлів. А в середовищі з високим ризиком перевірка на віруси особливо ретельно повинна здійснюватися для нових файлів.

Нижче наводяться **правила** для роботи в мережі Інтернет для користувачів корпоративних комп'ютерних мереж.

✓ Адміністратор безпеки повинен дозволити використання програм перед їх інсталяцією на комп'ютер. Забороняється встановлювати неавторизовані програми на комп'ютери. Конфігурації програм на комп'ютері повинні перевірятися щомісяця на предмет виявлення установки зайвих програм.

✓ Програми повинні встановлюватися тільки з дозволених внутрішніх серверів для обмеження ризику зараження. Не можна завантажувати програми з Інтернету на комп'ютери. За допомогою брандмауера необхідно заборонити операцію GET (завантаження файлу) із зовнішніх серверів.

✓ На файлові сервери треба встановити антивірусні програми для обмеження поширення вірусів у мережі. Повинна здійснюватися щоденна перевірка всіх програм і файлів даних на файлових серверах на віруси. Робочі станції повинні мати в пам'яті резидентні антивірусні програми, сконфігуровані так, що усі файли перевіряються на віруси при завантаженні на комп'ютер. Забороняється запускати програми і відкривати файли за допомогою додатків, уразливих до макровірусів, до проведення їхньої перевірки на віруси. Усі листи й файли, отримані з мережі, треба перевіряти на віруси при одержанні. По можливості перевірка на віруси повинна виконуватися на брандмауері, що керує доступом до мережі. Це дозволить централізувати перевірку на віруси для всієї організації і зменшити витрати на паралельне сканування на робочих станціях. Це також дозволить централізувати адміністрування антивірусних програм, обмежити кількість місць, куди повинні встановлюватися останні оновлення антивірусних програм.

✓ Програма навчання співробітників комп'ютерній безпеці обов'язково включає такі заходи про ризик зараження вірусами.

✓ Антивірусні програми можуть знайти тільки ті віруси, які вже були кимось виявлені раніше. Постійно розробляються нові, більш витончені віруси. Отже, антивірусні програми повинні регулярно оновлятися (щомісяця або щокварталу) для того, щоб можна було знайти найновіші віруси. Важливо повідомляти системному адміністраторові про будь-яке незвичайне поведіння комп'ютера або програм. Важливо відразу ж від'єднати комп'ютер, заражений або такий, що підозрюється в зараженні, від мережі, щоб зменшити ризик поширення вірусу.

Недотримання цих заходів повинно призводити до покарання співробітника відповідно до стандартів організації.

Усі програми повинні бути встановлені на тестову машину і перевірені на віруси перед початком їх використання в робочому середовищі. Тільки після одержання дозволу адміністратора безпеки можна встановлювати програми на машинах співробітників.

Крім використання комерційних антивірусних програм, використовують емулятори віртуальних машин для виявлення поліморфних вірусів. Усі нові методи виявлення вірусів повинні використовуватися на цій тестовій машині.

Перевірку усіх файлових систем треба здійснювати щодня в обов'язковому порядку. Результати перевірок, як правило, протоколюють, автоматично збирають й аналізують системні адміністратори.

Усі дані, імпортовані на комп'ютер тим або іншим способом (із дискет, з електронної пошти і т. д.), також перевіряються на віруси. Співробітники повинні інформувати системного адміністратора про виявлені віруси, зміни в конфігурації або дивне поведіння комп'ютера чи програм-додатків.

При одержанні інформації про зараження вірусом системний адміністратор зобов'язаний інформувати всіх користувачів, що мають доступ до програм і файлів даних, які могли бути заражені вірусом, що вірус, можливо, заразив їхні системи. Користувачам повідомляють про порядок визначення, чи заражена їхня система, і видалення вірусу із системи.

Видалення вірусу з зараженої комп'ютерної системи інколи вимагає переінсталяції ОС з нуля, знищення файлів або видалення вірусу із зараженого файлу.

Будь-яка машина, підозрювана в зараженні вірусом, повинна бути негайно відключена від мережі. Машина не підключається до мережі доти, поки системні адміністратори не переконаються у знищенні вірусу. По можливості повинні використовуватися комерційні антивірусні програми для знищення вірусу. Якщо такі програми не можуть видалити вірус, усі програми в комп'ютері треба видалити, включаючи завантажувальні записи при необхідності. Усі ці програми повинні бути повторно встановлені з надійних джерел і повторно перевірені на віруси. Зареєстровані користувачі антивірусів можуть звернутися через електронну пошту до фірми – виробника програми й одержати оновлення програми із засобами видалення вірусу.

РОЗДІЛ 7

ОСНОВИ WEB-ДИЗАЙНУ

Основні поняття

Web-сторінка – елементарна частина Всесвітньої Павутини (World Wide Web, www), представлена на конкретному комп'ютері у вигляді документа, що містить посилання на інші документи, розміщені у комп'ютерній мережі. Такі посилання називаються гіперпосиланнями (hyperlink), вони дозволяють при перегляді швидко переходити від одного документа до іншого, і тим самим зв'язують усі документи в єдине ціле.

Для формування Web-сторінки використовують HTML (Hyper Text Markup Language) - спеціальну мову форматування текстових електронних документів.

Формування документів мовою HTML подібне до програмування. Вихідний текст документа складається з послідовності команд, які називають тегами (tag). Під час перегляду теги спрацьовують, і завдяки їм на екран виводяться запрограмовані елементи зображення. Теги HTML дозволяють вирівнювати рядки документа, керувати кольором фону та кольором тексту, дозволяють вставляти в текст ілюстрації, задавати заголовки різних рівнів, будувати таблиці, створювати гіпертекстові посилання на інші документи.

Web-документ зберігається і пересилається як файл з розширенням html або htm (HTML-файл). Це звичайні текстові файли-програми, написані мовою HTML. Відображаються такі файли на екрані інакше, ніж вони виглядають у HTML-файлі.

Типовий Web-документ складається з текстових блоків, рисунків, таблиць, ліній, гіперпосилань. Більш складні Web-документи містять фрейми (вкладені Web-сторінки), елементи керування та вводу інформації (кнопки, перемикачі, поля вводу тексту), динамічні об'єкти (Java Applet, Java Script, ActiveX).

Є три типи текстових блоків: текстові абзаци, списки, таблиці. За замовчування текст відображається чорним кольором, але при потребі можна змінити колір, розмір, стиль та шрифт тексту. Абзаци на сторінці за замовчуванням розмежовуються переводом рядка і не мають відступів у першому рядку. Абзац може бути вирівняний до одного з країв або по центру вікна. Зручність перегляду Web-сторінок зумовлена тим, що браузер автоматично масштабує елементи по горизонталі, враховуючи розміри вікна, і при потребі відображає тільки вертикальну смугу прокрутки. інформація на сторінці традиційно відображається на білому фоні, але його колір можна змінити на інший або використати фонову картинку.

Основи програмування Web-документів

Всі програми, написані мовою HTML (HTML-файли) мають однакову структуру. Можна зекономити час при створенні своїх Web-сторінок, якщо використовувати цю структуру як еталон.

Обов'язковими при формуванні Web-сторінки повинні бути такі команди (теги):

```
<html>
```

```
<head>
```

```
<title> .
```

.Назва вікна Web-сторінки..

```
</title>
```

```
</head>
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

Теги `<html>` і `</html>` обмежують WWW-документ і записуються лише задля форми. Команди `<head>` і `</head>` обмежують заголовок сторінки. Ці теги також не є обов'язковими, вони допомагають краще структурувати сторінку. Нарешті, `<title>` виконує важливу функцію. Текст, введений між `<title>` і `</title>`, з'являється в браузері в рядку заголовку. Ще важливішим є тег `<body>`, який дозволяє описати власне Web-документ. В цьому місці розміщується певна кількість параметрів, що визначають зовнішній вигляд документа. Ці параметри визначають, як виглядає фон і колір тексту і яке зображення повинно бути на задньому плані Web-сторінки.

Для створення Web-сторінок використовують гіпертекстові редактори, наприклад, Macromedia Dreamweaver, Microsoft FrontPage та інші, хоча за достатнього володіння мовою HTML можна писати Web-сторінки в будь-якому текстовому редакторі, наприклад, Блокнот (NotePad), Word тощо.

Після написання програм файл потрібно зберегти на диску з певною назвою та розширенням `.html` чи `.htm`.

Сучасні текстові процесори, наприклад, MS Word, Lotus WordPerfect дають змогу створити Web-сторінки методом конструювання без застосування команд HTML, оскільки вони генерують ці команди автоматично. Загальна назва таких програм, які мають можливість автоматизованого створення Web-сторінок без явного застосування користувачем мови HTML - *Web-композери*.

Розглянемо дещо детальніше можливості композера, які має програма *MS Word*.

Щоб створити Web-сторінку або цілий Web-сайт, потрібно під час створення нового документу перейти на закладку Web-сторінки і скористатись Майстром Web-сторінки чи створити нову сторінку. У другому випадку алгоритм дій дизайнерів такий:

- Командою меню **Формат->Фон** задають колір фону чи спосіб замальовування екрана певною текстурою.
- Уводять текст, вибираючи шрифт, його розмір, колір, вирівнювання тощо.
- Створюють списки, як у звичайному текстовому редакторі.
- Вставляють горизонтальні лінії командою меню **Вставка ->Горизонтальна лінія**, і вибирають тип лінії.
- За допомогою команди **Таблиця** створюють і форматують таблиці.
- Використовують команду **Вставка** для вставлення картинок, фотографій, звуку, гіперпосилань тощо.
- Записують створений файл як документ HTML, і переглядають його браузером.

Наведемо деякі теги з їх призначенням.

Для переходу на новий рядок використовується тег `
` .

Для виділення тексту в абзаци існують теги `<P>` та `</P>` .

Текст, розміщений між тегами `` та `` буде жирнішим. **bold**

Текст, розміщений між тегами `<I>` та `</I>` буде написаний курсивом. *italics*

А текст, розміщений між тегами `<U>` та `</U>` буде підкресленим. underlined

Текст, розміщений між тегами `<S>` та `</S>` буде перекресленим. ~~stricken~~

Текст, розміщений між тегами `_{` та `}` буде написаний як нижній

індекс. індекс

Текст, розміщений між тегами `^{` та `}` буде виглядати як верхній індекс.

Тег `<HR>` вставляє у документ горизонтальну лінію.

Цікавого оформлення сторінці надає тег `<MARQUEE>` та `</MARQUEE>` - він додає так звану "рухому стрічку". У саму стрічку, яка переміщатиметься екраном, можна вставити як текст, так і малюнки, таблиці тощо.

Крім цього, можна використовувати різне форматування шрифту за допомогою тегу ``, який має параметри `<FACE>` - тип шрифту (наприклад: 'Lucida Sans', 'Arial', 'Monotype Corsiva', 'Times New Roman'), `<SIZE>` - розмір шрифту (1-7), та `<COLOR>` - колір шрифту.

Для змістового виділення елементів сторінки використовуються теги-заголовки. Заголовки в HTML бувають шести типів. Текст, розміщений всередині тегу заголовка `<H1>` `</H1>` відображатиметься відповідно до типу цього заголовка. Найбільшим є заголовок H1, найменшим - H6. Заголовок може розміщуватись на екрані по центру, або вирівнений по правому або лівому краю. `<H2 align=center \ right \ left >` `</H2>`

Заголовок першого рівня_по центру

Заголовок другого рівня_по правому краю

Заголовок третього рівня_по лівому краю

Заголовок четвертого рівня_по центру

Заголовок п'ятого рівня_по правому краю

ЗАГОЛОВОК ШОСТОГО РІВНЯ_ПО ЛІВОМУ КРАЮ

Списки у HTML бувають трьох видів: марковані, нумеровані та довільної форми.

Марковані списки оформлюються та виглядають наступним чином:

<code><ul type=circle\disk\square ></code>	○ Перша група	• Перша група	▪ Перша група
<code></code> Перша група <code></code>	○ Друга група	• Друга група	▪ Друга група
<code></code> Друга група <code></code>			
<code></code>			

Як бачимо, можливий вибір маркування списку (*circle/disk/square = коло/круг/квадрат*)

Нумеровані списки оформлюються та виглядають наступним чином:

<code><ol start=n type="1 \ A\ a\ I\ i" ></code>	1. Перша група	A. Перша група	e. Перша група	I. Перша група	x. Перша група
<code></code> Перша група <code></code>	2. Друга група	B. Друга група	f. Друга група	I. Друга група	x. Друга група
<code></code> Друга група <code></code>					
<code></code>					

Як видно - можна вказувати різні типи нумерування: арабські цифри, римські цифри великі й маленькі, латинські букви великі й маленькі. Крім того, можна вказувати номер, з якого починається нумерування (для третього списку це №5, для останнього - №9).

Останній тип списків - довільної форми. Приклад їх складення та вигляду:

<code><dl ></code>	
<code><dt ></code> Перша група <code></dt ></code>	Перша група
<code><dd ></code> Староста А <code></dd ></code>	Староста А
<code><dt ></code> Друга група <code></dt ></code>	Друга група
<code><dd ></code> Староста Б <code></dd ></code>	Староста Б
<code></dl ></code>	

У цьому випадку ми вказуємо не лише елементи списку (старости), але й спосіб їх переліку (за групами)

Для оформлення документів можна використовувати графічні об'єкти. Як вже згадувалось, можна вказати малюнок або колір, які будуть слугувати тлом для документу. Можна також вставляти малюнки прямо в документ. Для цього використовується тег ``. Він має наступні параметри:

`src="*:.*"` - файлова адреса малюнка, який потрібно вставити

`width=` - ширина малюнка (в пікселях)

`height=` - висота малюнка (в пікселях)

`align="center \ right \ left"` - спосіб вирівнювання малюнка по центру \ по правому \ по лівому краю

`alt="picture"` - альтернативний текст для випадків, коли малюнок не завантажується

Інтернет базується на документах, що пов'язані між собою, тобто на гіперпосиланнях між документами. Їх можна створювати за допомогою мови HTML. Між тегами `` та `` вставляється деякий текст, клацнувши на якому можна перейти до вказаного у href документу (в даному випадку - до index.html).

Іноді виникає потреба перейти до підтеми в межах одного документу. Тоді використовується так званий якір. У тому місці документа, куди треба буде перейти, вставляється мітка ``, а посилання на неї створюється за допомогою тегів `` та ``

Часто для структурованого представлення інформації використовуються таблиці. Вони створюються тегами `<TABLE>` (таблиця), `<TR>` (рядки таблиці), `<TD>` (стовпці таблиці) і виглядають наступним чином:

`<table>`

`<tr >`

`<td >` рядок 1 стовпець 1 `</td >`

`<td >` рядок 1 стовпець 2 `</td >`

`<td >` рядок 1 стовпець 3 `</td >`

`</tr >`

`<tr >`

`<td colspan=2 >` рядок займає 2 стовпці `</td >`

`<td rowspan=2 >` рядок 2 і 3, стовпець 3 `</td >`

`</tr >`

`<tr >`

`<td >` рядок 3 стовпець 1 `</td >`

`<td >` рядок 3 стовпець 2 `</td >`

`</tr >`

`</table >`

рядок 1 стовпець 1	рядок 1 стовпець 2	рядок 1 стовпець 3
рядок займає 2 стовпці		рядок 2 і 3, стовпець 3
рядок 3 стовпець 1	рядок 3 стовпець 2	

Як бачимо - мова HTML надає можливість об'єднання комірок таблиці: тег `<td colspan= >` поєднує стовпці, а тег `<td rowspan= >` - рядки.

Тег таблиці `<TABLE>` має наступні параметри:

`bgcolor` - колір таблиці

border= - ширина рамки таблиці (0-невидима рамка)
width= - ширина таблиці (в пікселях або у відсотках)
height= - висота таблиці (в пікселях або у відсотках)
align="center \ right \ left" - спосіб вирівнювання таблиці по центру \ по правому \ по лівому краю

Схожі параметри мають теги *<TD>* та *<TR>* :

b bgcolor - колір комірки / всього рядка
background=.** - колір комірки
width= - ширина комірки (в пікселях або у відсотках)
height= - висота комірки (в пікселях або у відсотках)
align="center \ right \ left" - спосіб вирівнювання вмісту комірок: по центру \ по правому \ по лівому краю
valign="top \ bottom \ middle" - спосіб вирівнювання вмісту комірок по вертикалі: згори \ внизу \ посередині
nowrap - блокує автоматичне перенесення слів в межах комірки
colspan= - кількість стовпців комірки
rowspan= - кількість рядків комірки

Форми дозволяють організувати інтерактивність для web-сторінки. Форми в документах HTML розміщуються між тегами *<FORM >* та *</FORM >* .

Всередині самої форми може бути розміщений звичайний текст, інші HTML-документи (таблиці), а також елементи форм (випадаючі списки, текстові поля, кнопки). Заповнені форми передаються певній програмі, яка обробляє отримані дані, але можна також відправити результати електронною поштою. Форми у HTML мають наступні атрибути:

ACTION Цей атрибут задає те, як оброблятиметься форма. Зазвичай йому присвоюється адреса програми, яка оброблятиме дані, отримані з форми. Тут можна також вказати електронну адресу, на яку треба відсилати відомості з форми. *<form action="index.pl">* *<form action="x@x.lviv.ua">*

METHOD Задається те, як саме форма передаватиметься за адресою, вказаною в попередньому атрибуті(варіанти: get/post)

ENCTYPE Задає спосіб кодування (application/x-www-form-urlencoded)

Форма може складатись з різних елементів: текстових полів, випадаючих меню, списків, переключачелів та кнопок.

Текстові елементи. Для створення найпростішого текстового поля потрібно вставити елемент *<INPUT>* і присвоїти його атрибуту значення *text*. Всі елементи форми повинні мати ім'я. *<input type=text name=name>* Значенню атрибуту *size* елементу *<INPUT>* відповідає кількість символів, які будуть введені у це поле. Якщо користувач введе більше символів, то текст прокручуватиметься. Якщо треба обмежити розміри поля, можна визначити атрибут *maxlength* – максимальна кількість символів, яку можна ввести у поле.

Поле для паролів. Дані, введені у це поле показуються на екрані у вигляді “зірочок” для того, щоб введена інформація залишалась приватною. Для цього атрибуту *type* присвоюється значення *password*.

Багатострічкове текстове поле. У цього елемента є атрибути *rows* (кількість рядків), *cols* (кількість стовпців), *name* (ім'я). `<textarea name=t1 rows=6 cols=60>` та `</textarea>`. Для цього елемента можна задати значення по замовчуванню між тегами `<textarea>` та `</textarea>`.

Меню. Випадаюче меню дає користувачеві можливість вибрати один варіант з кількох можливих. Це меню створюється елементом `<select>` та `</select>`. Всередині його розміщуються кілька елементів `<option>`. При вказанні в атрибуті *value* іншого значення, ніж після `<option>`, на обробку відправлятиметься саме це значення. Можна вказати який елемент буде вибраним по замовчуванню через атрибут *selected*. Можна також поставити атрибут *size* – тоді на екран буде виводитися лише певна кількість стрічок списку.

```
<select name="s1">
<option value="Ukraine"> Ukraine
<option value="Russia"> Russia
<option value="Other"> Other country
</select>
```

Мітки. Якщо користувачеві потрібно обрати кілька пунктів з великої групи варіантів можна використати поле для міток. Для цього атрибуту *type* елемента `<input>` присвоюється значення *checkbox*. За допомогою атрибута *value* можна вказати “своє” значення замість того, що мало б передаватись далі.

```
<input type=checkbox value="Cat" name=a1> Cat
<input type=checkbox value="Parrot" name=a1> Parrot
```

Кнопки-перемикачі. Користувач може обрати лише один варіант з запропонованих. Для цього атрибуту *type* елемента `<input>` присвоюється значення *radio*. Тут обов'язково потрібно вказувати однакове ім'я *name* і різні значення *value*.

```
<input type=radio name=r1 value="yes"> Yes
<input type=radio name=r1 value="no"> No
```

Кнопки reset та submit. Ці кнопки організовуються також через завдання значення атрибуту *type* елемента `<input>` як *reset* або *submit* відповідно. Кнопка Reset дозволяє користувачеві очистити заповнені поля і перевести форму у початковий стан. Submit – дає команду браузеру відправити вміст форми за адресою, вказаною у атрибуті *action* елемента `<form>`.

Стилі. Таблиці стилів, так звані CSS, Cascading Style Sheets містять опис формату частини або цілого тексту. Завдання стилю відбувається за допомогою тегу `<style>`

```
<head>
<style>
тег {властивість1:значення1;
властивість2:значення2;...;властивістьN:значенняN}
</style>
</head>
```

Таблиці стилів можуть розміщуватися в окремому файлі, що дозволить звертатися до них всім сторінкам певного сайту - а це полегшить цілісність його оформлення. Тоді в межах тегу `<head>` вставляється посилання на файл зі стилями:

```
<link rel="stylesheet" type="text/css" href="x.css">
```

За допомогою стилів ми можемо змінити вигляд тегів *h*, *p*, *body*, *table* тощо. Наприклад, завдання стилю *body {background:honeydew}* означає, що лише вказуючи тег *body* він завжди позначатиме фон кольору *honeydew*. Можна також вводити власні стилі. Наприклад, стиль *.blue {color:blue}* означає, що текст буде написаний блакитним кольором.

При використанні стилів доцільно користуватись тегом `<div>` - він не форматує документ, але помічає фрагмент тексту, який розглядатиметься як єдиний об'єкт. Тоді для цього об'єкту можна вказати певний стиль форматування - `<div class=blue>` При цьому всередині тегу `<div>` можуть бути розміщені й інші елементи та теги.

Стильових ефектів є досить багато. З їх допомогою можна оформити веб-сторінку оригінально, різноманітніше, ніж за допомогою звичайних тегів. Наприклад ефектами оформлення можуть бути:

- background колір фону
- font-family вид шрифту
- font-size розмір шрифту
- color колір шрифту
- text-align вирівнювання тексту
- text-decoration оздоблення тексту
- font-weight - жирність шрифту
- margin-top - відстань від верхнього рядка
- line-height - висота рядка

Приклад створення власної Web-сторінки

Використовуючи один із текстових редакторів, наприклад, *NotePad*, створюється HTML-файл такого змісту:

```
<html>
<head>
<title> Homepage </title>
</head>
<body>
...
</body>
</html>
```

Цей файл записується під певним іменем з розширенням html (наприклад, *user1.html*).

Поміж `<body>` і `</body>` вставляється певний текст, наприклад:

```
<h1> Homepage користувача № Intraneta </h1>
```

```
Я радий з вами познайомитись.
```

```
<p> Це мої перші спроби створення власної сторінки. </p>
```

```
<p> Як студент я багато працюю з інформацією. </p>
```

```
<p>Мої захоплення: література, музика, комп'ютери, мандрівки рідним краєм. </p>
```

```
<p>Збережіть ці зміни у файлі user1.html. </p>
```

Змінений файл знову зберігається під тим же іменем.

Не закриваючи вікна текстового редактора, тепер можна запустити Internet Explorer і завантажити файл user1.html, використовуючи команди головного меню **Файл ->Відкрити (File -> Open)**.

На екрані буде висвітлено результат вашої першої спроби створення власної Web-сторінки.

Тепер можна повернутися до вікна текстового редактора, де виведено зміст файлу user1.html і доводити його до потрібного кінцевого результату, використовуючи потрібні для цього теги.

Створення Web-сторінки з використанням програми FrontPage

Якщо ви запустили *FrontPage*, то це означає, що нова Web-сторінка у вас вже створена. При запуску FrontPage відразу відкривається нова порожня Web-сторінка, готова для розміщення тексту та будь-яких інших елементів. Приклад такої сторінки наведений на рис.10.

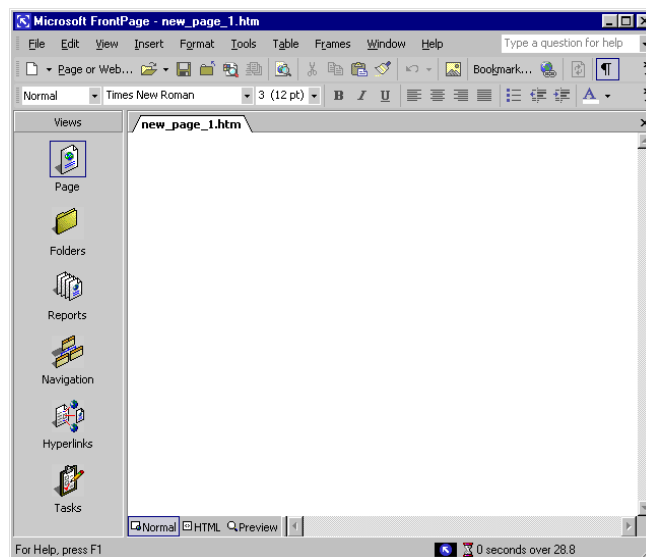


Рис. 10

Створити Web-сторінку можна й іншими способами:

- за допомогою кнопки *New Page* на панелі інструментів *Standard*;
- з меню **File** (виберіть спочатку **New**, а потім - **Page**);
- за допомогою комбінації клавіш **Ctrl+N**;
- з використанням правої кнопки миші та контекстного меню. Працює в усіх режимах за виключенням *Tasks View*.

У всіх режимах (за виключенням *Page View* і *Tasks View*) під створенням нової сторінки мається на увазі створення нового HTML-файлу в поточній папці, присвоєння цьому файлу імені за умовчанням і відкриття імені цього файлу для перейменування. При цьому Front Page не переключиться відразу ж в режим *Page View* для редагування цього файлу. Ви можете зробити це самі, двічі клацнувши по ньому мишкою.

В *Page View* натиснення клавіші *New Page* на панелі інструментів відразу ж створює та відкриває нову Web-сторінку. Три методи, що залишились, дозволяють вибрати шаблон для сторінки, що створюється. Вікно вибору шаблонів показано на рис.11.

На вкладці **General** виводиться перелік доступних на даний момент шаблонів. Обравши будь-який шаблон, можна переглянути його опис в полі *Description* та зображення для попереднього перегляду в полі *Preview* в правій частині вікна.

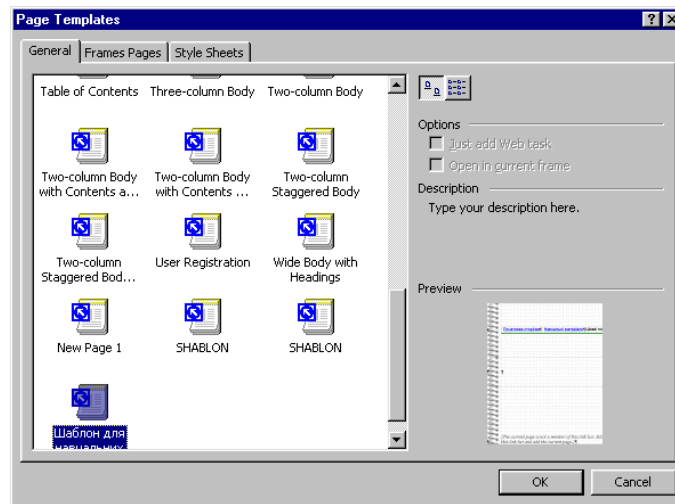


Рис. 11

На вкладці *Frames Pages* можна вибрати шаблон для створення Web-сторінки з фреймами.

Вкладка **Style Sheets** відкриває файл каскадних таблиць стилів (.css) для теми FrontPage. Файл .css – текстовий файл в спеціальному форматі. При створенні нового файлу .css він успадкує всі риси старого. Ви можете змінювати будь-які наладки стилів в цьому файлі.

Кнопки *Large Icons* і *List* змінюють представлення списку стилів у вікні.

Прапорець *Open in current frame* доступний тільки тоді, коли у *FrontPage* відкрита сторінка з набором фреймів. На даному етапі його можна проігнорувати.

При встановленому прапорці *Just add web task* крім створення сторінки буде створене і пов'язане з нею завдання в **Tasks List**.

Пам'ятайте, що створення нової сторінки у **Page View** реально не створює нового файлу. Щоб цей файл був створений, збережіть сторінку. Якщо ж ви закриєте нову пусту сторінку без будь-яких змін, *FrontPage* не збереже її і навіть не дасть запит, щоб ви це зробили.

Є декілька способів відкрити Web-сторінку, які залежать від того, в якому режимі ви працюєте.

З режиму **Page View** є чотири способи відкрити існуючу сторінку:

- В меню **File** виберіть **Recent Files**. Буде відображений список файлів, що недавно відкривалися. Просто виберіть потрібний файл з цього списку;
- Клацніть правою кнопкою миші по гіперпосиланню на вже відкритій сторінці, а потім в контекстному меню виберіть **Follow Hyperlink**. Буде відкрита та сторінка, на яку вказує гіперпосилання;
- Якщо ви працюєте з *FrontPage Web*, просто виберіть потрібний файл у *Folder List* і два рази клацніть по ньому мишею;
- Використайте діалогове вікно **Open File**, яке показано на рис.12.

Є три способи відкрити діалогове вікно **Open File**. Ви можете вибрати **Open File** з меню **File**, натиснути **Ctrl+O** чи застосувати кнопку **Open** на панелі інструментів *Standard*. На панелі в лівій частині діалогового вікна можна вибрати п'ять способів як відкрити Web-сторінку:

- **History** відображає список документів Microsoft Office, що недавно відкривалися;

- **My Documents (Personal в Windows NT)** дозволяє відобразити файли у відповідній папці;
- **Desktop** відображає файли, які знаходяться на робочому столі вашого комп'ютера. До них відносяться значки **My Computer**, **Network Neighborhood**, **My Briefcase** та інші;
- **Favorites** показує те ж саме, що ви побачите в меню **Favorites** (Вибране) Internet Explorer;
- **Web Folders** відображає список Web Folders, зареєстрованих для Microsoft Office на даному комп'ютері, і файли в них.

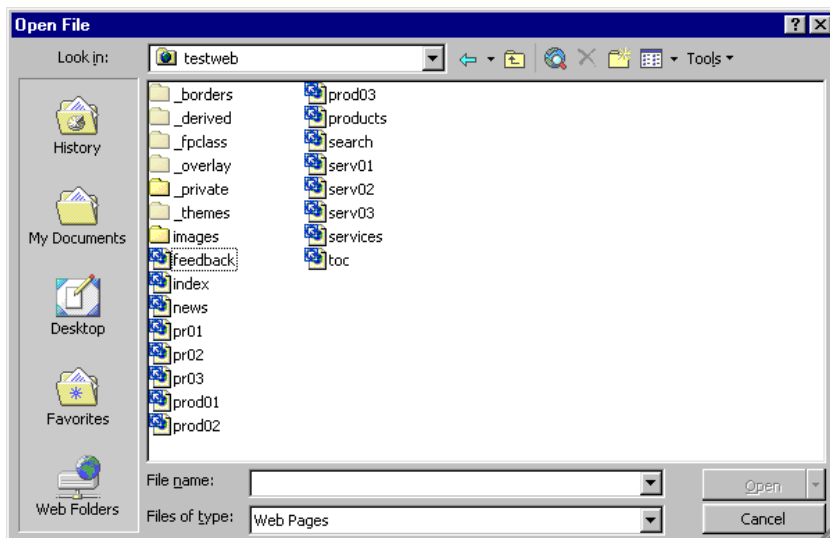


Рис. 12

Список *Look In* у верхній частині екрана дозволяє вибрати місцезнаходження потрібного файлу.

Для визначення файлу можна скористатися і полем *File Name*, при цьому можна вказати як шлях до файлу у файлової системі (наприклад, *\My Documents\mypage.htm*), так і *URL* (*http://www.interlacken.com/default.htm*). При використанні *URL* файли відкриваються за допомогою того ж механізму, який використовується в браузері. В панелі інструментів діалогового вікна **Open File** є декілька корисних кнопок.

При відкритті Web-сторінки, що створена в іншому додатку Office, FrontPage може здивувати вас, відкривши файл з з допомогою того додатку, в якому він був створений. Щоб уникнути цього, один раз, у списку, що відкривається при кнопці **Open** виберіть **Open in Microsoft FrontPage**. Щоб відключити таку можливість зовсім, в меню **Tools** виберіть **Configure Editors** і зніміть прапорець *Open web pages in the office application that create them*.

Так як **Tasks View** призначений для роботи з завданнями, а не Web-файлами, відкривати файли з цього режиму доводиться не так часто. Але, якщо вам все-таки знадобилось це зробити, ви можете використати будь-яким з двох способів:

- клацніть правою кнопкою миші на задачі, що пов'язана з Web-сторінкою, і в контекстному меню виберіть **Start Task**;
- виберіть задачу, зв'язану з Web-сторінкою, а потім в меню **Edit** виберіть **Task-Start**.

В режимах **Folders View**, **Reports View**, **Navigation View** і **Hyperlinks View** відкриття сторінки відбувається абсолютно однаково:

- можна двічі клацнути мишкою по імені файлу чи його значку;
- можна клацнути по імені файлу чи значку правою кнопкою мишки і в контекстному меню вибрати **Open**;
- можна вибрати ім'я файлу чи значок а потім натиснути **Enter**, **Ctrl+O** чи вибрати **Open** в меню **File**.

Крім кнопок, які застосовуються для створення і відкриття Web-сторінок, в панелі інструментів **Standard** знаходяться й інші корисні кнопки. Як і всі інші панелі інструментів, її можна сховати чи показати з меню **View**.

Якщо ви не зберегли сторінку, вважайте, що всіх ваших змін більше немає. Щоб зберегти файл, відкритий в режимі **Page View**, в меню **File** виберіть **Save**. Якщо сама сторінка чи будь-який з її компонентів ще ні разу не зберігався, **FrontPage** відкриє діалогове вікно **Save As**. Щоб зберегти сторінку під другим іменем, потрібно скористатися сусіднім пунктом меню **Save As**. Діалогове вікно **Save As** показано на рис.13.

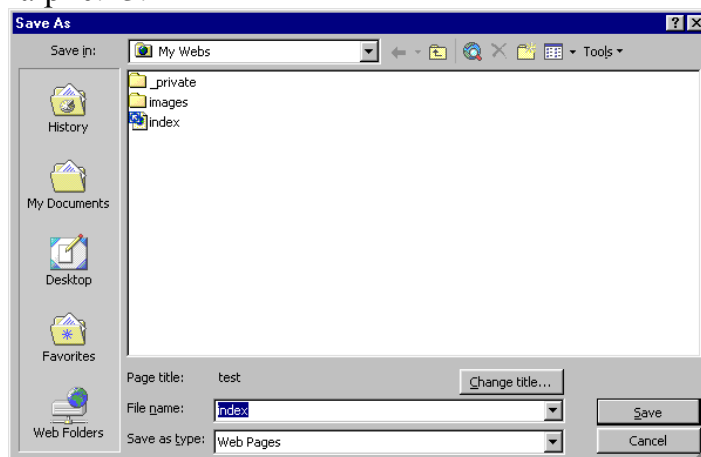


Рис. 13

Як ми бачимо, воно виглядає практично однаково з діалоговим вікном **Open**, за виключенням двох нових елементів:

- **Page title**. Відображає назви сторінок. Хоча це поле можна і не заповнювати, дуже раджу все-таки не ігнорувати його. Назва сторінки – те, що бачить відвідувач, коли відкриває її в браузері. Крім того, назва застосовується у **FrontPage** для багатьох ситуацій;

- **Change**. Натиснення на цю кнопку відкриває діалогове вікно, в якому ви можете змінити назву сторінки

- Як і при відкритті файлу, можна вказувати різні шляхи до файлу. Зберігати файл можна на локальному диску, на сервері чи у **FrontPage Web**. Якщо **FrontPage Web** ще не відкритий, **FrontPage** його відкриє автоматично.

Час від часу при збереженні сторінки **FrontPage** виводить діалогові вікна, подібні до тих, що представлені на рис.14.

Це означає, що в оперативній пам'яті знаходиться об'єкт, місце зберігання якого **FrontPage** поки що не знає. В даному випадку - це малюнок, вставлений з буферу обміну, але це може бути і результат інших дій, наприклад, використання засобів праці з графічними зображеннями. Початкове зображення може залишатися там же, де воно було розміщене до цього, але файл зі змінами потрібно зберегти окремо.

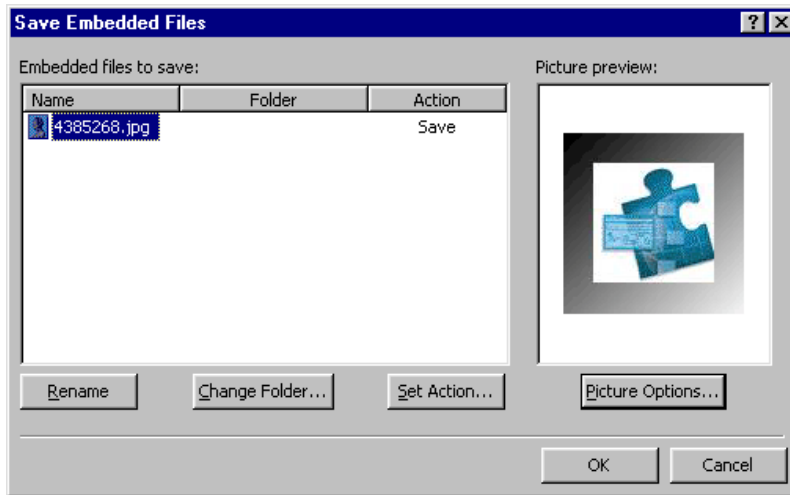


Рис. 14

При роботі з Web-сайтом рано чи пізно трапиться так, що якісь сторінки вам будуть більше не потрібні і вам захочеться їх знищити. Існують чотири способи видалення Web-сторінки (при цьому ви повинні бути в режимах **Folders View**, **Reports View**, **Navigation View** або **Hyperlinks View**):

- виберіть файл (клацнувши по його назві або значку) і натисніть на кнопку **Delete**;
- виберіть файл і потім виберіть **Delete** в меню **Edit**;
- клацніть по назві файлу або значку правою кнопкою миші і в контекстному меню виберіть **Delete**;
- виберіть файл і натисніть на кнопку **Delete** в панелі інструментів *Standard*.

Який би спосіб ви не вибрали, FrontPage попросить вас підтвердити видалення.

Атрибути Web-сторінки в основному використовуються для загального управління її зовнішнім виглядом. Щоб відкрити властивості сторінки (всі шість вкладок), клацніть правою кнопкою миші по сторінці в режимі *Page View* і виберіть в контекстному меню **Page Properties**. Відкриється діалогове вікно **Page Properties**, в якому ви побачите п'ять вкладок: **General**, **Background**, **Margins**(ця вкладка з'являється тільки коли в меню *Tools-PageOptions* вказано, що сорінки розраховані тільки на браузер Microsoft Internet Explorer), **Custom**, **Language** і **Workgroup**.

Вкладка **General** зображена на рис.15.

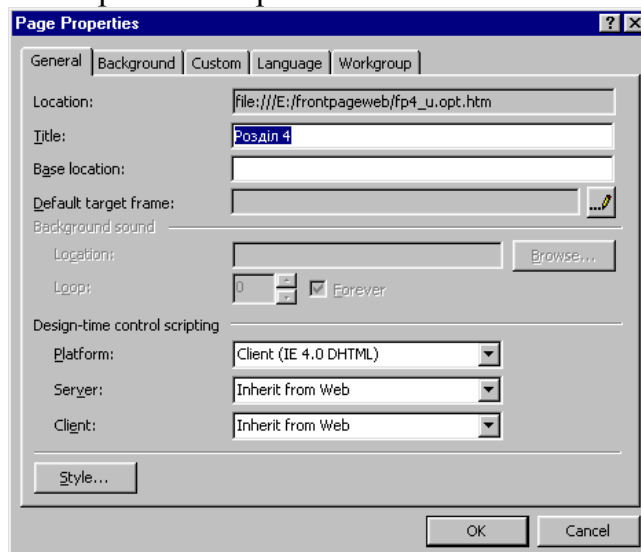


Рис 15.

З усіх полів на цій вкладці вам обов'язково треба вказати значення для поля **Title** (заголовок), всі інші поля можна не заповнювати або залишити значення за умовчанням.

- *Location*. Звичайно, це URL для даної сторінки, по якому до неї може звернутися, наприклад, браузер. Якщо файл відкривається не з Web-сервера, а з диска, URL буде починатися з *file:///*. Для щойно створеної і ще не збереженої сторінки Location буде виглядати як *unsaved:///new_page_1.htm*. Редагувати це поле ви не зможете - воно призначене тільки для читання.

- *Title*. Це дуже важливий атрибут, хоч його часто випускають з уваги. Він використовується в багатьох меню і діалогових вікнах FrontPage, крім того, назва сторінки відображається в заголовку вікна браузера і в описі результатів пошуку на пошукових серверах. Пересвідчіться, що ви присвоїли зрозумілу назву кожній сторінці.

- *Base location*. Це поле, яке використовується відносно рідко, дозволяє задати URL, який буде автоматично підставлятися до гіперпосилань, в яких вказані сторінки без повної адреси (наприклад, в Base location вказане *www.pfew.com/info*, а в гіперпосиланні - *contact.htm*; адреса для переходу по гіперпосиланню виходить *www.pfew.com/info/contact.htm*). Детальніше про це - трохи далі. Як правило, це поле залишають незаповненим.

Для гіперпосилань на Web-сторінці вказувати повні адреси зовсім не обов'язково. Якщо в ній не вказано ім'я хосту, браузер автоматично підставить ім'я поточного хосту. Якщо в гіперпосиланні не вказане ім'я папки, браузер скористається поточною папкою. Такий підхід називається використанням відносних адрес (*relative addressing*), оскільки повна адреса для гіперпосилання залежить від того, на якій сторінці вона розташована. Ось приклад:

Поточна сторінка - *http://www.pfew.com/info/default.htm*

Гіперпосилання - */products/toasters.htm*

Адреса для браузера *http://www.pfew.com/products/toasters.htm*

Звичайно використання відносної адресації зручніше. Використання цього підходу дозволяє легко переміщувати великі групи Web-сторінок з одного Web-сервера (або папки) на інший. При цьому вам не доводиться міняти всі гіперпосилання.

Але іноді може трапитися так, що в якості адреси за умовчанням для гіперпосилань вам треба буде використати адресу, відмінну від поточної. Для таких ситуацій і призначене поле *Base location*. Ось як воно використовується:

Поточна сторінка - *http://www.pfew.com/info/default.htm*

Base URL - *http://www.xxxx.com/info/*

Гіперпосилання - *contact.htm*

Адреса для браузера - *http://www.xxxx.com/info/contact.htm*

- *Default target frame*. Якщо ви розділили вікно браузера на фрейми, це поле буде відображати інформацію про фрейм, в якому буде показуватися дана сторінка.

- *Background sound*. Три поля в цій групі призначені для інформації про звуковий файл, який буде програвати браузер відвідувача під час перегляду файлу. Вказується ім'я і шлях до звукового файлу. Це може бути як шлях до локального файлу, так і URL. Щоб можна було вибрати файл, натисніть на кнопку **Browse**;

- *Locaton*. Вказує ім'я файлу. Це може бути, як шлях до локального файлу, так і URL. Щоб можна було вибрати файл скористайтеся кнопкою **Browse**.

- *Loop*. Скільки разів буде програватися файл;

- *Forever*. Встановіть прапорець, щоб звуковий файл грав постійно. Відповідно, те, що ви вказали в полі Loop, значення мати не буде.

– *Design-time control scripting*. Ця група призначена для налагодження параметрів виконання скриптів на сторінці.

– *Platform*. Вказується, де будуть виконуватися скрипти в браузері або на сервері;

– *Server*. Яка мова за умовчанням буде застосовуватися для інтерпретації скриптів, призначених для виконання на сервері. Можна вибрати JavaScript, VBScript або настройки за умовчанням для Web;

– *Client*. Яка мова за умовчанням буде застосовуватися для інтерпретації скриптів, призначених для виконання в браузері. Можна вибрати JavaScript, VBScript або настройки за умовчанням для Web;

– *Style*. За допомогою цієї кнопки можна настроїти параметри стильового оформлення (каскадних таблиць стилів) для основного тексту сторінки.

Вкладка **Background** призначена для управління загальними параметрами колірної схеми сторінки. Вона наведена на рис.16.

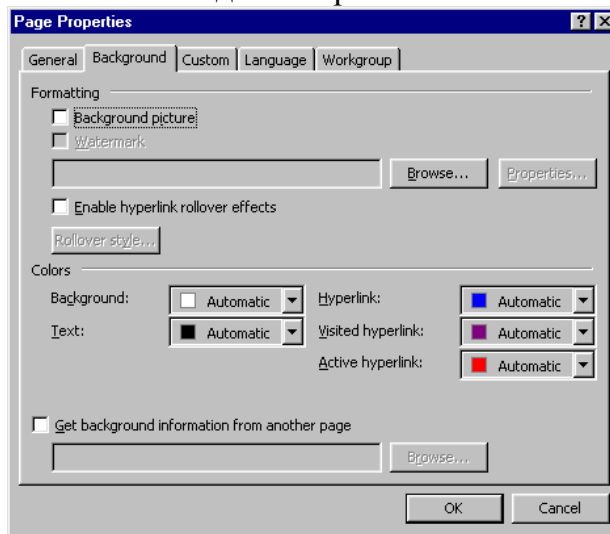


Рис.16

На вкладці Background можна провести настройки фону і колірної схеми сторінки. Врахуйте, що ці настройки будуть відноситися виключно до сторінки, що ви вказали.

Background Picture. Установка перемикача показує, що у сторінки буде фонове зображення. У полі трохи нижче можна вказати шлях до файлу фонового зображення (або вибрати його, натиснувши кнопку Browse).

Фонові зображення показуються на сторінці під будь-якими іншими зображеннями або текстом. Якщо зображення менше, ніж сторінка, воно буде повторюватися, щоб заповнити весь простір зліва направо, і зверху вниз. Такий підхід називається мозаїчним розміщенням (tiling). У результаті невелике зображення, яке можна завантажити швидко, заповнює весь простір екрана.

Якщо вам не треба, щоб фонове зображення повторювалося, зробіть його ширшим, ніж стандартний екран комп'ютера. Звичайно більш ніж вистачає ширини в 1200 пікселів. У більшості редакторів зображень можна скористатися засобом Add Margin або Extend Canvas, які дозволяють дуже просто збільшити розмір зображення. Додані пікселі можна заповнити кольором фону або зробити їх прозорими. Пікселі того ж кольору, що повторюються звичайно стискаються дуже непогано, і розмір файлу (і час його завантаження) істотно не збільшиться.

Зображення великої ширини часто використовуються для розміщення графіки у лівого кордону екрана. Звичайно сам малюнок займає 20 30 пікселів зліва, а все інше заповнюється фоном або робиться прозорим.

Уникайте насичених кольорів для фонових зображень, які можуть утруднити читання тексту.

– *Watermark*. Перекладається як “водяний знак”. Установка цього прапорця означає, що при прокрутці сторінки фонове зображення залишиться нерухомим і не буде переміщатися разом з іншими елементами сторінки. Така можливість підтримується Internet Explorer, але багато інших браузерів її не підтримують. Цікаво, що не підтримує показ поведінки цього елемента і сам FrontPage. Тому, щоб переглянути, як буде виглядати цей елемент, відкрийте сторінку в Internet Explorer.

– *Properties*. Натиснення цієї кнопки відкриває діалогове вікно, в якому можна настроїти властивості фонового зображення.

– *Background*. Визначає колір фону сторінки. Цей колір буде відображатися, якщо немає фонового зображення, якщо яка-небудь частина фонового зображення прозора і якщо браузер відображає текст ще до завантаження фонового зображення.

– *Text*. Настройки кольору звичайного тексту.

– *Hyperlink*. Настройки кольору для гіперпосилань.

– *Visited hyperlink*. Настройки кольору для гіперпосилання, що вказує на сторінку, яку користувач недавно відкривав в браузері. Колір повернеться до звичайного для гіперпосилань стану, якщо протягом часу, вказаного для кеша браузера, користувач не відвідає сторінку-мету.

– *Active hyperlink*. Настройки кольору для гіперпосилання, по якому користувач щойно клацнув мишею.

– *Enable hyperlink rollover effects*. Установка цього прапорця дозволяє настроювати зміну зовнішнього вигляду гіперпосилання при проходженні над ним курсору миші. Настройки проводяться із звичайного діалогового вікна Font.

– *Get background and colors from page*. Установка цього прапорця відмінняє всі індивідуальні настройки вкладки Background для сторінки і дозволяє отримати настройки від іншої сторінки. Така можливість може виявитися корисною при роботі з набором уніфікованих сторінок.

На вкладці **Margins** вказуються x-y координати для першого елемента на сторінці (див. рис.17). Як точка відліку приймається верхній лівий кут сторінки.

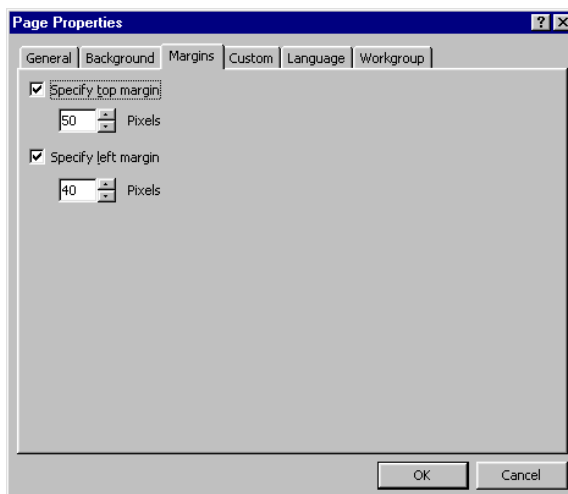


Рис.17

Відступ по висоті вказується за допомогою прапорця *Specify Top Margin* і вказівки розміру відступу в пікселях, а відступ зліва - за допомогою прапорця *Specify Left Margin* і аналогічного поля.

У FrontPage і Internet Explorer все відображається, як потрібно. Netscape Navigator використовує свої команди для роботи з відступом, тому в ньому виходить інша картина.

Вкладка **Custom** використовується для роботи із системними і призначеними для користувача змінними (див. рис.18).

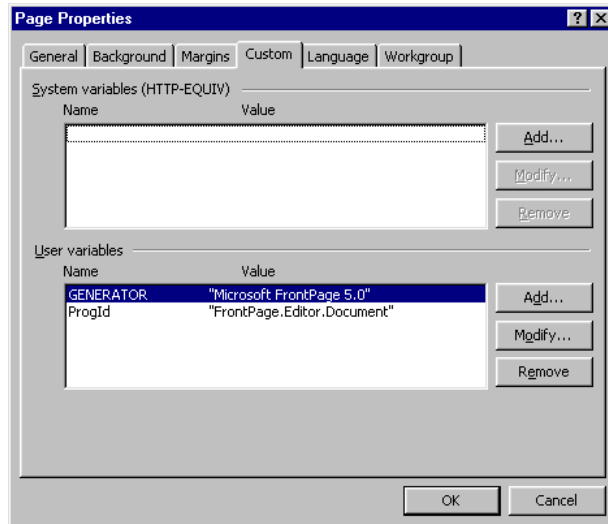


Рис.18.

Системні змінні - це, як правило, стандартні заголовки HTTP (HTTP headers), а в якості призначених для користувача ви можете вказувати все що завгодно. Детальніше про системні змінні - у вірці нижче, а змінні користувача дуже зручно використовувати, наприклад, щоб автоматично підставляти ім'я відповідального за щось співробітника або адресну інформацію. Якщо на цю посаду прийде інша людина або адреса зміниться, вам досить буде усього раз змінити змінну, замість того, щоб вишукувати всі згадки про предмет на сторінках. Додавати змінні просто: натисніть кнопку **Add**, а потім введіть ім'я і значення змінної (див. рис.19).



Рис.19.

Для зміни вже існуючої змінної або її значення призначена кнопка **Modify**.

- **Copyright.** Заява про авторські права.
- **Description.** Пропозиція з описом змісту сторінки.
- **Distribution.** Звичайно використовується одне з двох слів: *global* або *local*. *Local* означає, що сторінка не представляє інтересу для користувачів за межами організації - власника Web-сайту.

- **Expires.** Дата, після якої зміст сторінки вважається застарілим.

Використовуйте наступний формат:

- *Tue, 02 Dec 1997 21:29:02 GMT*

– **Keywords.** Це - ключові слова, які полегшують знаходження цієї сторінки при пошуку. Якщо ключових слів декілька, вони розділяються комами.

– **Robots.** Інструкції для “роботів” - пошукових серверів Інтернету. Єдиних стандартів не існує. Деякі інструкції підтримуються одними серверами і не підтримуються іншими. Ось найпоширеніші змінні:

– **None.** Роботам пропонується проігнорувати цю сторінку. Еквівалентне *Noindex, Nofollow*;

– **All.** Роботам пропонується індексувати цю сторінку і всі сторінки, до яких можна перейти по гіперпосиланням, без будь-яких обмежень. Діє так само, як і *Index, Follow*;

– **Index.** Запрошення всім роботам включити цю сторінку в результати пошуку;

– **Noindex.** Цю сторінку індексувати не рекомендується;

– **Follow.** Запрошення роботам слідувати по гіперпосиланням цієї сторінки;

– **Nofollow.** Слідувати по гіперпосиланням цієї сторінки не рекомендується.

Системна змінна *Refresh* в деяких ситуаціях може бути дуже корисною. Вона наказує браузеру зачекати декілька секунд (в нашому прикладі - 20), а потім автоматично перейти на новий URL. Так можна влаштувати, наприклад, щось на зразок автоматичного показу слайдів. Інше застосування - при переміщенні всього сайту на нове місце залишити на старому сторіночку з оголошенням про перехід на нову адресу. При цьому відвідувач буде переноситися на нову адресу автоматично. Краще не використовувати значення затримки, що рівне нулю. Відвідувач може не зрозуміти, в чому справа, і повторно спробує потрапити на ту ж адресу або скористається кнопкою Back, а сторінка буде повертати його назад.

Про деякі інші корисні системні змінні розказано нижче. Ці змінні часто використовуються пошуковими роботами типу Yahoo, Lycos, AltaVista і InfoSeek для більш точного індексування сторінок.

Вкладка **Language** призначена для налаштувань кодування для Web-сторінки (див. рис.20.).

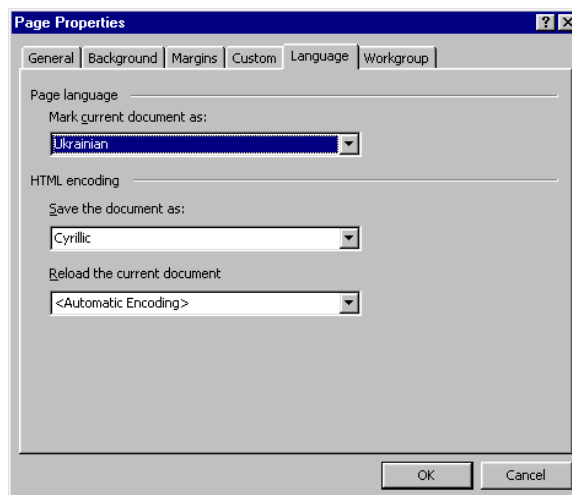


Рис.20

На цій вкладці є три поля:

– *Mark current document as.* Встановлене в цьому полі значення буде використовуватися FrontPage для перевірки правопису (якщо стоїть *None*, то FrontPage буде керуватися тим, яка розкладка клавіатури у вас була включена під час набору тексту). Значення, введене в це поле, використовується також пошуковими серверами;

– *Save the documents as.* Можна вибрати кодування, в якому буде збережена Web-сторінка. Наприклад, для сторінки на російській мові можна вибрати *Cynllic* або *Cyrillic (KOI8-R)*. Якщо ви вибрали значення за умовчанням (*None*), то буде використовуватися мова за умовчанням вашої операційної системи;

– *Reload the current document as.* Налаштування для браузера. Керуючись ними, браузер буде відображати вашу Web-сторінку. За умовчанням використовується значення *<Automatic Encoding>*.

Вкладка **Workgroup** призначена для організації роботи з Web-сторінками в колективі (див. рис.21).

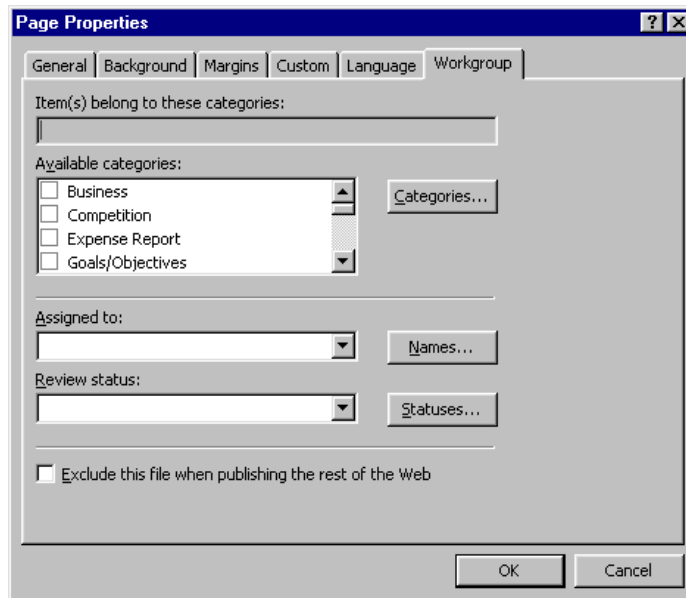


Рис.21

На цій вкладці можна скористатися наступними полями:

– *Available Categories.* Можна встановити прапорець для віднесення сторінки до певної категорії. Кнопка *Categories* призначена для створення нової категорії, якщо ті, що вже є, вас не влаштовують. Цей атрибут призначений для спільного використання з компонентом *Categories* (детальніше про нього і про інші компоненти буде розглянуто пізніше);

– *Assigned To.* У цьому полі належить вказувати відповідального за Web-сторінку. Поповнити список відповідальних можна за допомогою кнопки *Names*;

– *Review Status.* Сюди можна ввести (або вибрати зі списку) поточний етап проходження Web-сторінки по інстанціях. Інстанції можна вказувати і свої, скориставшись кнопкою *Statuses*;

– *Exclude This File When Publishing the Rest of the Web.* При публікації вашого Web ця сторінка передаватися не буде. Така можливість може виявитися корисною, якщо сторінка ще не готова.

Вибір колірної гамми - одне з найважливіших рішень, які приймаються при створенні Web-сторінок. Крім того, іноді варто поміняти кольори на Web-сторінці, щоб освіжити її сприйняття. У будь-якому випадку Web-дизайнеру необхідно знати основні концепції роботи з кольором.

Будь-який колір можна отримати за допомогою поєднання трьох кольорів: червоного, зеленого і синього. Наприклад, світло-жовтий колір вийде при певному наборі відтінків червоного і зеленого. Для людського сприйняття те, що цей колір

був отриманий за допомогою комбінації інших кольорів, не має значення. Вона сприймає його як чисто жовтий колір.

Цей ефект використовується для відображення кольорів на екрані комп'ютерних моніторів. Для кожного пікселя екранного зображення додаток повідомляє (наприклад, в діапазоні від 0 до 255) бажану кількість червоного, зеленого і синього кольору. Внаслідок множення 256 можливих значень інтенсивності червоного кольору на 256 - зеленого і на 256 - синього виходить 16 777 216 можливих колірних відтінків. Такий підхід отримав назву колірної моделі RGB (від red - червоний, green - зелений і blue - синій). Наприклад, колір RGB 255-0-0 означає найінтенсивніший, який тільки можливий. 0-0-0 - абсолютно чорний колір, а 255-255-255 - повністю білий.

У значній кількості відвідувачів Web відеоадаптери працюють тільки з 8 бітами на піксель, що дає тільки 256 можливий значень. Технологія така, що їх системи можуть відображати всі 16 777 216 кольорів, але не більше за 256 кольорів одночасно на екрані. У результаті можуть виникнути деякі проблеми.

Звичайно не всі з 256 кольорів можна одночасно використати. Наприклад, для комп'ютерів під управлінням Windows 20 кольорів зарезервовано для використання з елементами вікон (кордони вікон, заголовки і інші). У результаті в розпорядженні додатків залишається тільки 236 кольорів.

Web-сторінки можуть містити будь-яку кількість зображень у форматі GIF. Кожне з цих зображень, в свою чергу, може використати 256 кольорів. Однак два зображення можуть запросити вже 512 кольорів, три - 768 і т.д. Але одночасно на екрані можуть відображатися тільки 256 кольорів, що може привести до проблем при перегляді Web-сторінок.

Зображення у форматі JPEG (JPG) може містити до 16 777 216 кольорів. Зрозуміло, що при спробі використати для такого зображення тільки 256 кольорів можуть виникнути проблеми.

Більшість браузерів вирішують ці проблеми, використовуючи для відеосистем з 256 кольорами фіксовану палітру з 216 кольорами. Ця кількість кольорів виходить шляхом множення шести певних рівнів червоного на шість рівнів зеленого і на шість рівнів синього. Оскільки це - єдині кольори, відносно яких можна бути упевненим, що браузер відобразить їх так, як ви задумали, вони отримали назву безпечних кольорів браузерів (browser-safe colors).

Для відображення кольорів з RGB-інтенсивністю, відмінною від значень, наведених в таблиці, браузер використовує змішення (dither) або підстановку (substitutes). При застосуванні змішення області з нестандартним кольором замінюються сумішшю пікселів стандартних кольорів. У теорії око майже не повинно помічати відмінностей, але на практиці таке зображення часто виглядає зернистим. Найкраще змішення працює на зображеннях з плавними переходами тонів (наприклад, фотографіях). Однак результат його застосування до тексту, заливки або зображень, створених за допомогою ліній, може бути дуже неприємним.

При використанні підстановки браузер просто замінює нестандартний колір на найближчий до нього стандартний. Для фону звичайно застосовується підстановка, оскільки нерівномірність фону може привести до проблем із сприйняттям тексту.

Висновки прості. Так як при змішенні та підстановці відвідувачі будуть бачити зовсім не те, що ви планували, використовуйте для Web-сторінок тільки безпечні кольори.

У деяких випадках, однак, можна забути про відвідувачів з 8-бітним кольором і розраховувати тільки на відвідувачів з 24-бітними відеосистемами (наприклад, всі відвідувачі вашого сайту - співробітники підприємства зі стандартними робочими станціями). Однак, якщо ви відкриваєте сайт для публічного доступу, потурбуватися про 8-бітних відвідувачів слід обов'язково.

Для вибору більшості кольорів у FrontPage використовуються три діалогових вікна:

Діалогове вікно, що розкривається з 16 стандартними кольорами VGA, кнопкою для кольору за умовчанням (Automatic) і кнопкою для додаткової настройки (More Colors). Це діалогове вікно наведено на рис.22.

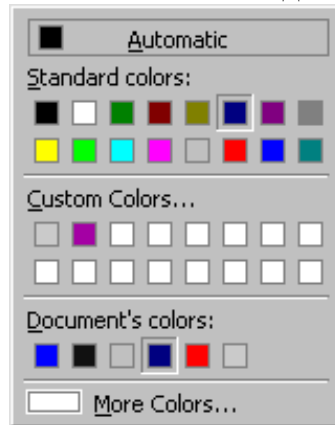


Рис. 22

При натисненні на кнопку More Colors відкривається наступне діалогове вікно, в якому вам вже пропонується вибір з 127 кольорових зразків, зразків білого і чорного кольорів і 15 відтінків сірого кольору. Крім того, там є декілька інформаційних полів для відображення інформації про колір і дві кнопки. Натиснення на кнопку Select дозволяє вибрати будь-який колір з екрана вашого монітора, а натиснення на кнопку Custom відкриває ще одне діалогове вікно для найбільш точного вибору кольору.

Абсолютно будь-який з доступних в системі кольорів можна вибрати в діалоговому вікні Color. Щоб відкрити це вікно, в діалоговому вікні More Colors натисніть кнопку Custom.

Розглянемо кожне вікно для вибору кольорів по порядку.

Найперше діалогове вікно надає у ваше розпорядження 16 кольорів (кнопка **Automatic** передбачає вибір кольору, що використовується браузером по умовчанням). На жаль, з цих 16 кольорів лише половина є безпечною. Вісім, що залишилися, може не вистачити для створення ефективного оформлення вашого сайту.

Тому для вибору кольорів краще використовувати наступний рівень - діалогове вікно **More Colors** (див. рис.23). У ньому використовується шестикутник з набором із 127 кольорів, а внизу - чорний і білий зразки і шість градацій чорно-білого зображення. Переважна більшість кольорів з набору є безпечними, однак зрідка попадаються і такі кольори, які не відповідають вимогам до безпечних кольорів. Наприклад, такий колір вибраний на малюнку. Пам'ятайте, що якщо будь-яке з трьох шістнадцатеричних значень відрізняється від 00, 33, 66, 99, CC або FF, то колір не є безпечним.

У цьому ж діалоговому вікні можна скористатися кнопкою Select, щоб вибрати потрібний колір з будь-якої точки екрана монітора.

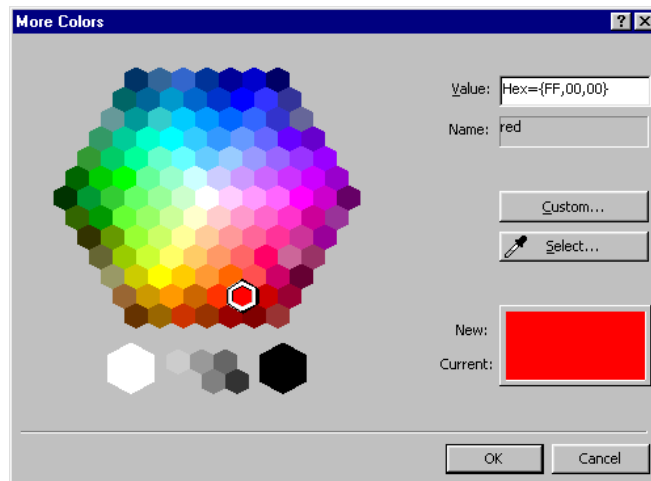


Рис. 23

Будь-який колір, доступний в системі, можна вибрати в наступному діалоговому вікні Color (див. рис.24). При цьому у вашому розпорядженні три способи: можна вибрати колір на панелі, можна скористатися звичайною колірною моделлю RGB, а можна також скористатися моделлю відтінок-насиченість-яскравість: Hue-Saturation-Luminance (в цій книзі ми її не будемо розглядати). Пам'ятайте, що для створення будь-якого з 216 безпечних кольорів в полях Red, Green і Blue можна використати тільки десятиричні значення 0, 51, 102, 153, 204, 255 - і ніякі інші.

Вибір кольору – це певного роду мистецтво. Однак є декілька порад, які можуть допомогти Web-дизайнерам:

- вибирайте кольори, які поєднуються між собою. Наприклад, сторінку можна виконати в небесних тонах, морських тонах, вибрати для неї теплі кольори землі або щось інше. Ті кольори, які звичайно зустрічаються разом і красиво виглядають в природі, швидше усього, будуть добре виглядати всюди;
- при виборі кольорів для Web-сторінок пам'ятайте про те, що вони повинні поєднуватися з кольорами на зображеннях;
- між текстом і фоном завжди повинен бути чіткий контраст;
- краще використовувати темний текст на світлому фоні, ніж навпаки;
- кольори повинні контрастувати, але не дисгармоніювати;
- краще не використовувати таких фонових зображень, які будуть відволікати увагу від тексту. Ідеальні зображення для фону - дуже світлі і зі слабким контрастом.

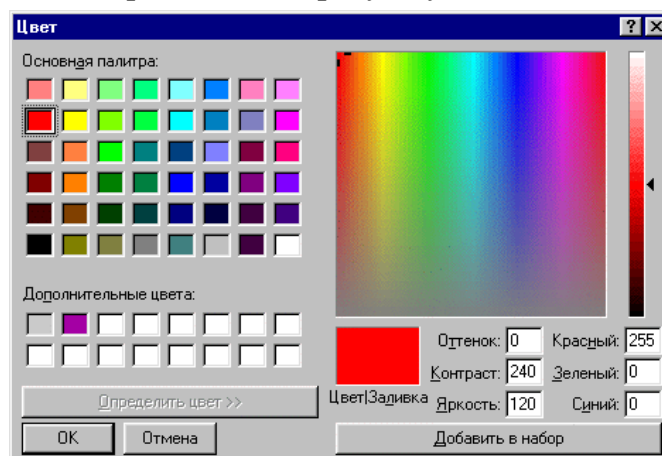


Рис.24

Критерії оцінки ефективності Web-сторінок та Web-сайтів

Визначення критеріїв ефективності є ключовим пунктом при побудові та супроводженні сайту. Це, зокрема:

- економічна вигода;
- формування суспільної думки;
- ознайомлення суспільства з нормативною інформацією, результатами наукових досліджень тощо;
- зміцнення особистого авторитету чи авторитету організації.

З перерахованих вище мотивів лише бажання отримати економічну вигоду відображається в критеріях якості – розмір прибутку відображає ефективність Веб-сайту (проводиться чітка аналогія зі звичайним підприємством). Проте, з другого боку, Інтернет є настільки великим та динамічним середовищем, що й економічні критерії не є достатньо точними та об'єктивними.

Серед неекономічних критеріїв оцінки ефективності Веб-сайтів є часто вживаними такі типи критеріїв:

- технічні критерії швидкодії, оптимізації мережевого трафіку та вимог до апаратних ресурсів;
- технічні критерії надійності та безпеки й ефективності технічної підтримки;
- естетичні та художні критерії;
- психологічні критерії;
- системні критерії глобальних середовищ (відвідуваність сайту, рейтинг сайту, Link Popularity та ін.).

Важливими також є естетичні, психологічні критерії. Винятково важливими стають також системні критерії.

Врахування всіх критеріїв породжує ряд суперечностей. Між тим сьогодні все ще не існує єдиного обґрунтованого погляду на їх пріоритетність. При побудові реально діючих WWW-систем розробники одночасно застосовуються критерії усіх наведених вище типів.

Наведені вище проблеми побудови сайтів, що є ефективними за багатьма критеріями, можуть частково або повністю розв'язані при виконанні ряду наступних рекомендацій.

Раннє визначення критеріїв. Критерії оцінки якості та ефективності сайту повинні бути визначені на ранніх етапах Інтернет-проекту. При запізнілому визначенні критеріїв необхідно заново пройти ряд основних етапів створення сайту. Використання спеціальних допоміжних технологій не є запорукою успіху та може нести різні приховані небезпеки.

При визначенні критеріїв слід враховувати їхню синтетичну природу, наявність суб'єктивних та нестабільних складових. Це дає можливість визначати основні вимоги до сайту, задоволення яких забезпечуватиме достатній рівень якості та ефективності по усіх критеріях.

Встановлення пріоритетності критеріїв. Після визначення усіх критеріїв ефективності необхідно встановити їхню пріоритетність. Найпріоритетніші критерії повинні формувати функції мети (їхнє покращання повинно здійснюватися постійно). Решта критеріїв повинна формувати систему обмежень. Недотримання певних обмежень не допускається, проте після досягнення певного рівня ефективності по таких критеріях подальшого покращання можна не здійснювати.

Визначення стратегії розвитку. Потрібно визначити стратегію побудови та розвитку сайту із врахуванням критеріїв та їхніх пріоритетів. Ця стратегія повинна бути спланована на довготривале використання. Зміна стратегії після проходження великої частини процесу побудови ефективного сайту є дуже дорогим і трудомістким заняттям. Наприклад, системні критерії відвідуваності цільовою аудиторією та ранжування за ключовими словами у пошукових системах великою мірою залежать від вибраної стратегії реєстрації ресурсу. Після виконання процедур реєстрації ефективна зміна цільової аудиторії або ключових слів може для даного сайту бути практично неможлива (без повторної реєстрації під іншою Інтернет-адресою – що знову ж призводить до втрати бренду).

Використання усталених технологій. Використання експериментальних, не дуже поширених, не стандартизованих певним чином технологій без вагомих причин є недопустимо. Набір технологій, що використовуються при створенні сайту, повинен бути зведений до абсолютного мінімуму, достатнього для розв'язання поставлених задач.

У кожній із застосованих технологій необхідно, у першу чергу, використовувати максимально спеціалізовані засоби (наприклад, типова помилка Веб-дизайну – використання лише HTML-тегів із параметрами для форматування призводить до значного погіршення якості сайту згідно і з технічними і з системними критеріями).

Постійний контроль за показниками якості. Важливою особливістю, що відрізняє Веб-сайти від інформаційних систем локального призначення, є їхнє функціонування в глобальному середовищі. Як наслідок, не можливо зберегти високі показники якості протягом навіть недовгого часу (реальним часом стабільності є декілька місяців), якщо не здійснювати постійний контроль та вдосконалення сайту. Динамічність Інтернет суттєво ускладнює побудову сайту з високими показниками ефективності за багатьма критеріями.

Проте, з іншого боку, динамізм World Wide Web та Інтернет при постійному контролі та вдосконаленні сайту надає додаткові можливості щодо збільшення комплексної ефективності сайту. Зокрема, є можливість знаходження нових узгоджень за критеріями, уточнення системи пріоритетів. Показники по деяких критеріях можуть покращуватися без особливих зусиль (Наприклад, покращуватися технічні характеристики).

Крім того, WWW є достатньо відкритими середовищем, що дає змогу відслідковувати методи вирішення певних задач по побудові сайту та оцінювати їхню ефективність за вибраними критеріями.

Сьогодні практично для кожного нового WWW-проекту (крім експериментальних та науково-дослідницьких) можна знайти діючі аналоги, що дозволяє врахувати їхній досвід досягнення або втрати ефективності. Розробникам сайтів практично неможливо приховати використані ними технічні рішення (на концептуальному рівні). Цілком відкритими є естетичні та психологічні якості сайту. Системні показники сайтів здебільше випадків також можуть бути отримані.

Таким чином, Інтернет для розробників сайтів є не лише джерелом проблем, а й джерелом ідей щодо їхнього вирішення.

Застосування математичних та формальних підходів. Задачі, що постають при розробці ефективних сайтів, є складними, багатокритеріальними та слабоформалізованими. Проте застосування математичних та формальних підходів

до цих задач дозволяє побудувати об'єктивну систему вимірювання ефективності сайту, визначити багатокритеріальні задачі оптимізації та координації, визначити основні підходи до їхнього розв'язання. Звичайно, у загальному випадку не завжди можна отримати повністю формалізовані математичні задачі оптимізації або координації. Однак, у будь-якому разі, формальний аналіз задачі покращання сайту дозволяє точніше визначити пріоритетність і співвідношення між критеріями, числові характеристики критеріїв, конфлікти та колізії у цій задачі покращання.

Побудова ефективних сайтів для деяких типових класів

Протягом розвитку Веб-систем та Інтернет-технологій пріоритетність критеріїв якості Веб-сайтів зазнавала змін. Проте і донині не вироблено єдиної шкали (хоча б описової) щодо пріоритетності тих чи інших критеріїв. Відсутність таких пріоритетів є головним обмеженням при побудові сайтів, що відповідають сучасним вимогам якості та ефективності.

Проте для певних типових класів сайтів така пріоритетність може бути визначена.

Для сайтів усіх категорій (за винятком першої та другої) характерним є порівняно дешевий трафік. Як правило, існує значний невикористаний резерв трафіку. Верхній рівень обмеження по трафіку досягається рідко.

Тому цільова відвідуваність для таких сайтів стає основним критерієм. Це справедливо і для другої категорії сайтів, які мають значні витрати на трафік, проте витрачають достатньо фінансових ресурсів.

Прибутковість таких сайтів не є великою (за винятком успішних Інтернет-магазинів).

Таким чином, пріоритетними критеріями ефективності для кількох основних груп сайтів на сьогодні є системні критерії. Основним таким критерієм є відвідуваність сайту цільовою аудиторією за контрольний період - та їхня кількість.

Визначення цільової відвідуваності сайту та постійної аудиторії у загальному випадку є складною задачею, що може бути розв'язана лише за допомогою спеціальних інтелектуальних аналізаторів доступу до Веб-сайту. Крім згаданих вище критеріїв цільової відвідуваності та постійної аудиторії, важливими критеріями ефективності є загальна аудиторія сайту та загальна множина запитів до сайту. Ці показники інколи вважаються несуттєвими, інколи навпаки – основними. Насправді, цей показники, є другорядними порівняно з вище згаданими, проте істотно на них впливають.

Критична оцінка методик покращання сайту

Існує багато методик покращання системних показників сайтів. Більшість із них є комерційними та передбачають цілий комплекс заходів та постійний контроль і поліпшення результату. Однак, деякі заходи покращання показників сайту, не завжди виправдовують себе, інші навпаки – при невеликих затратах дають значний ефект. Наведемо деякі з них.

- Основну частину відвідувачів (понад 50 відсотків) сайти загального характеру отримують з пошукових машин, значна – із каталогів. Лише незначна частина відвідувачів приходить на сайт із прямої неінтернетівської реклами (виняток – сайти телевізійних, радіомовних та друкованих ЗМІ).

- Для сайтів із не дуже широкою цільовою аудиторією рідко виправдовує себе участь у банерних мережах та рейтингах.

- Системні показники сайту значною мірою залежать від його структури та наповнення.

- Ключовим фактором “затримування” відвідувача на сайті є не художній дизайн, а інформаційне наповнення.
- Художнє оформлення сайту може стати фактором відштовхування відвідувачів від сайту (зокрема, при використанні нетрадиційних прийомів та засобів та при повільному завантаженні сторінок).
- Усе популярнішими стають схеми розрахунків за Інтернет-послуги, що орієнтуються на використаний трафік, а не на час роботи в Інтернеті.

Рекомендації

Щодо наповнення та структурування сайту визначимо такі рекомендації:

- Інформаційне наповнення сайту є ключовим фактором, що забезпечує високі системні показники.
- Важливим фактором збереження постійної аудиторії та ефективної індексації сайту пошуковцями є постійне оновлення сайту.
- Якщо існує достатньо велика постійна аудиторія сайту ефективним засобом підвищення системних показників сайту є форуми, онлайн-дискусії, інші види інтерактивної взаємодії між відвідувачами.
- Наявність надлишкових функцій на сайті (**електронна пошта, SMS** тощо) що не належить до першого та другого класів (див. вище) є недоцільною. Вони незначно покращують системні показники сайтів, проте часто істотно погіршують показники інших типів.
- Структура сайту повинна враховувати вимоги провідних пошукових машин та каталогів.
- Структура сайту повинна відповідати вимогам usability для різних користувачів та при виконанні ними різних задач (онлайн перегляд, офлайн перегляд локальних копій, роздрук тощо).

По дизайну та технічній реалізації сайту визначимо наступні рекомендації:

- Графічний дизайн повинен ґрунтуватися на принципі “мінімального задоволення всіх ” а не “максимального задоволення декого ” (винятки – сайти дизайн-студій, мистецьких об’єднань тощо).
- Сторінки повинні легко та швидко завантажуватися, не містити зайвих елементів, які збільшують її фізичний об’єм. Ця існує давно, проте її актуальність, незважаючи на постійне збільшення пропускних здатностей Інтернет, лише зростає. Це обумовлено швидким ростом об’ємів ресурсів WWW та зміною форм розрахунків клієнтів за Інтернет-послуги. Оптимальним поданням текстової інформації (з огляду також на технічні критерії оптимізації трафіку та відкритості коду) є XHTML+CSS. Для дублювання інформації допускаються формати PDF, PostScript та документоорієнтовані мови XML.
- При розробці сайту потрібно остерігатися технік Веб-дизайну, що призводять до появи сторінок з обмеженими або спеціальними властивостями відображення. Це, зокрема, фрейми, скріпти, флеш-ролики тощо. Такі технології можуть використовуватися для організації спаму, орієнтованого на пошукові машини (**пошукова машина** буде “бачити” іншу сторінку, ніж звичайний відвідувач). Будь-яка техніка дизайну, що дозволяє у залежності від конфігурації клієнта міняти вигляд сторінки, заборонена або може бути заборонена у близькому майбутньому. Це ж стосується інтелектуального завантаження сайту залежно від параметрів відвідувача (зокрема, підбір мови по регіональним установкам).

– Бажано, щоб усі змістовні сторінки сервера або були статичними, або приховували від відвідувача факти свого динамічного створення на сервері. Зокрема, небажаною є явна передача параметрів у запиті на отримання сторінки.

– Щодо адаптації сайту, його змісту та структури для кращого подання в пошукових машинах та каталогах визначимо наступні рекомендації:

– Повинна бути окреслена цільова аудиторія сайту. Визначені її ключові характеристики: мова, регіон, уподобання, узагальнені особисті дані (вікова категорія, стать тощо).

– Уже на початкових етапах створення сайту (при системному плануванні та аналізі) необхідне визначення ключових слів та фраз (з урахуванням вимог цільової аудиторії), що повинні ідентифікувати сайт. Мають бути проаналізовані синоніми та слова, що йдуть за аналогіями до визначених ключових слів. Для таких слів повинні бути проаналізовані частотні показники запитів по них на пошукових машинах.

– Необхідно визначити розділи відомих каталогів (зокрема, DMOZ та Yahoo!), у яких могли б бути розміщені посилання на сайт (з урахуванням вимог цільової аудиторії).

– Повинні бути проаналізовані характеристики сайтів, що конкурують по тематиці, по ключових словах та позиціях у каталогах. Зокрема, потрібно аналізувати відвідуваність сайтів (якщо інформація доступна), відвідуваність із пошуковців по вибраних ключових словах (якщо інформація доступна), відвідуваність із каталогів по вибраних розділах (якщо інформація доступна), системні показники сайтів (Page Ranking, Link Popularity) та можливість гідно з ними змагатися по вибраних словах. Якщо очевидно, що шанси скласти достойну конкуренцію (наприклад, увійти у першу десятку на пошуковцях по вибраних словах та по рейтингу в рейтингових системах) відсутні, потрібно поміняти ключові слова та розділи вибраних каталогів-рейтингів.

– Формуючи інформаційне наповнення сайту потрібно використовувати вибрані ключові слова та фрази (у тексті, заголовках, підписах тощо) у різних модифікаціях. Проте, використання ключових слів повинно бути коректним та не порушувати семантику та прагматику текстів.

– Виконуючи процедури реєстрації сайту у пошуковцях, каталогах, рейтингах, інших сайтах потрібно використовувати вибрані ключові слова та фрази, орієнтуватися на розділи, які відповідають вибраним в основних світових каталогах.

– Процедура реєстрації сайту у пошуковцях та каталогах повинна плануватися та відбуватися поступово. Вона повинна враховувати існуючі залежності між відомими Інтернет-сервісами та не мати надлишкового характеру (наприклад, окрема реєстрація сайту в пошуковці Google не доцільна і не бажана, якщо сайт зареєстровано в DMOZ).

РОЗДІЛ 8

ПРОГРАМНІ ЗАСОБИ РОБОТИ ЗІ СТРУКТУРИЗОВАНОЮ ІНФОРМАЦІЄЮ

Термін “структуризація”, застосований відносно інформації, відображає її особливість набувати різної форми, маючи одне і те ж змістове наповнення: скажімо, “виробництво тканини” як інформаційне поняття може мати різну форму відображення – озвучену голосом, записану на папері, зображену схематично малюнком тощо.

Поява першого графічного інтерфейсу користувача і технології обробки зображень наприкінці 80-х прискорило поширення концепції структуризації на електронну форму документа, особливо це стосувалося розробки різноманітних підходів для переведення в електронну форму великої за обсягом інформації на паперових носіях.

За короткий термін робота з електронними документами була вдосконалена, поширившись від звичайного графічного образу документа до ідеї управління документами. Ідея маніпуляції інформацією в електронній, а не в паперовій формі стала узвичаєною. Обсяг такої інформації в офісах подвоюється кожні три роки.

На початку 90-х років виникли технологічні розробки, пов'язані з індексацією й пошуком документів, що використовують результати, отримані в галузі нейронних мереж і штучного інтелекту. Вони дозволили сформулювати принципово нові концепції побудови систем управління інформацією в електронному вигляді.

Електронний документ являє собою форму, вигляд якої подібний до звичайного паперового документа й обробка якої відбувається за допомогою послідовного застосування взаємозалежних технологій у межах, так званих, систем управління електронними документами - Electronic Document Management Systems (EDMS). Системи EDMS, що працюють переважно на розподілених архітектурах, реалізують і використовують різноманітні комбінації технологій збирання, індексування, збереження, пошуку й перегляду електронних документів.

Нині ринок систем EDMS пропонує більш 500 програмних продуктів. Згідно з прогнозами часопису “SIGNAL Magazine”, обсяг внутрішнього ринку США систем EDMS сягає близько 6 мільярдів доларів і має тенденцію зростати.

Система електронного документообігу являє одну з найважливіших складових корпоративної інформаційної системи і відповідає за управління процесами введення документів у систему, їх збереження, пошуку, маршрутизації, обробки, збирання та аналізу інформації про поточний стан виконання ділових та адміністративних процедур.

Сучасна інтегрована система управління електронними документами має бути розрахована на “синергетичний ефект”, при якому ефективність її використання для організації в цілому значно перевищує суму усіх локальних ефектів і поліпшень, отриманих від автоматизації окремих операцій.

Системи документообігу, на думку спеціалістів, беруть свій початок із кінця 19 сторіччя, коли з'явилися фотоапарат і плівка, тобто коли виникла можливість створювати зменшені копії документів. Шпигунські технології зазнали процесу конверсії і стали застосовуватися для створення мікрографічних архівів документів. Мікрофіша створила реальний якісний прорив у вартості збереження і копіювання документів, але, із точки зору швидкості пошуку і колективного використання документів, показники залишилися на рівні паперових документів.

Виникнення комп'ютерних систем із колективним доступом дозволило здійснити технологічний прорив і в цій галузі. Відбулася трансформація від мікрографічних документів до електронних образів паперових документів, від мікрографічних систем до, так званих, *imaging-систем*. Розширена система пошуку, можливість мати доступ до одного документа декільком користувачам одночасно, можливість віддаленої і розподіленої роботи при незначній собівартості зберігання й копіювання інформації зумовили тенденцію заміни мікрографічних документів на електронні.

З часом обладнання та програмне забезпечення вдосконалювалися - виникали усе більш швидкісні і невибагливі до якості паперового документа сканери, спеціальні плати, що дозволяють прискорити процес друкування, масиви зберігання інформації, оптичні й магнітооптичні роботизовані бібліотеки.

Imaging-системи дали поштовх розвитку систем розпізнавання машинописних і рукописних символів. З'явилися перші системи *workflow*, призначені для організації процесу впровадження паперових документів в електронний архів. Надалі вони поширилися на бізнес-процеси (офісну діяльність) організацій та підприємств.

На початку 90-х рр. було створено ще один новий клас систем для управління електронними документами, що відрізнялися від образів документів. Переважно це були файли текстових процесорів. Запити на подібні рішення надходили від федеральних агентств і юридичних компаній.

Розвиток функціональності, наприклад, додавання засобів роботи з образами документів, сприяли тому, що сучасні системи управління стали перевершувати за функціональними можливостями *imaging-системи*. При цьому їхня вартість була приблизно удвічі менша. Крім функцій збереження й пошуку, властивих *imaging-системам*, у системах управління документами з'явилися засоби, що дозволяють контролювати життєвий цикл документа. Іншими словами, на відміну від своїх попередників, системи управління документами орієнтовані на роботу з динамічними документами.

Основні вимоги до систем електронного документообігу

Масштабуємість. Розуміється як вимога до системи підтримувати будь-яку кількість користувачів і мати можливість нарощувати потужність за рахунок потужності апаратного забезпечення, на якому вона встановлена. Реалізація цієї вимоги може бути забезпечена за допомогою використання індустріальних систем управління базами даних таких компаній, як Sybase, Microsoft, Oracle, Informix, що існують практично на всіх можливих програмно-апаратних платформах, забезпечуючи таким чином максимально широкий спектр продуктивності.

Розподіленість. Трактуються таким чином: основні проблеми при роботі з документами виникають у територіально-розподілених організаціях, тому архітектура системи документообігу повинна підтримувати взаємодію розподілених об'єктів (або автоматизованих робочих місць). Причому вони можуть бути об'єднані найрізноманітнішими за швидкістю та якістю каналами зв'язку. Також архітектура системи має забезпечувати взаємодію з віддаленими користувачами.

Модульність. Розуміється як здатність системи складатися з окремих модулів, інтегрованих між собою. Тобто не завжди потрібне впровадження всіх компонентів системи документообігу, а коло задач, що мають бути розв'язані, можуть бути менше усього спектру задач документообігу.

Відкритість. Система документообігу повинна мати відкриті інтерфейси для можливої доробки та інтеграції. Інакше кажучи, така система не повинна існувати у відриві від інших прикладних задач. Наприклад, часто потрібно інтегрувати її з прикладною бухгалтерською програмою.

Як і будь-яка сфера людської діяльності, процеси та операції, що виконують системи електронного документообігу, потребують стандартизації.

Один з основних стандартів, призначених для передачі й представлення електронних документів, орієнтованих на діловодство (бланки, накладні, листи, звіти й ін.) є стандарт ISO 8613 part 1-6, Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format (ODIF).

Стандарт ODA має два основних призначення:

1. Це метод опису електронного представлення документа, включаючи усі види інформації (текст, графіка, таблиці), що є вхідними в нього. Такий структурний опис називається архітектурою документа.

2. Представлення змісту структурованої інформації у вигляді, зручному для автоматизації обміну між текстовими процесорами, робочими станціями, лазерними принтерами і т.д. Кодування ODA для обміну між таким обладнанням здійснюється у послідовному форматі ODIF, що зручно для використання у комунікаційних середовищах.

Крім різноманітних версій основного стандарту ODMA (Open Document Management API), розробляються і багато інших, наприклад, такі як: стандарти тільки для workflow-систем (така специфікація має назву WorkFlow Coalition API); або стандарти тільки для роботи з образами документів, що забезпечують уніфікацію форматів, що використовуються (формат TIFF Group IV, формат, розроблений компанією Adobe - PDF).

Класифікація програмних засобів роботи з електронною документацією

Один із варіантів класифікації програмних продуктів, поданих на ринку систем автоматизації при роботі з електронною документацією:

1. Засоби автоматизації карткового обліку паперового діловодства («Дело-2015», «Золушка»).
2. Системи обміну електронними документами («1С Документооборот», Exchange).
3. Системи керування електронними документами (DOCS Open, Excalibur EFS).
4. Системи автоматизації ділових процесів (Action Workflow, WorkRoute, FormFlow).
5. Засоби автоматизації карткового обліку безпаперового діловодства (OfficeMedia).
6. Продукти суміжних галузей, що нарощують базові можливості систем діловодства (FineReader, Stylus).
7. Продукти, що адаптують базові можливості західних систем до особливостей російської мови (Bit Full Text Retrieval).
8. Повнофункціональні системи автоматизації всіх задач паперового й безпаперового діловодства російською мовою (PowerDOCS).

Загальні поняття про технологію роботи з електронною документацією

У діяльності багатьох державних, акціонерних і приватних підприємств виникає необхідність обробки структурованих і неструктурованих форм документів, виконаних на папері. Звичайно, під цим розуміється початкове введення інформації з паперового документа в інформаційну систему організації і,

потім, власне опрацювання даних у СУБД і прикладних інформаційних систем. З ростом кількості документів зростає і проблема із застосуванням засобів для обробки великих обсягів даних в обчислювальних системах.

Технологія обробки документів визначається такими чинниками як вид документа й вимоги його життєвого циклу та особливості роботи окремих установ. Основу такої системи повинна складати відповідна організація програмного забезпечення, що дозволяє вирішувати широке коло задач обробки документів, незалежно від специфіки діяльності окремого закладу. Система повинна бути зручна в адмініструванні й експлуатації. З іншого боку, у будь-якій такій системі має бути передбачена можливість для основних виконавців самим здійснювати програмування маршруту проходження й режими обробки того чи іншого документа.

Завдання, що вирішують системи документообігу і потрібна для цього система автоматизації, визначаються стадією життєвого циклу документа, який необхідно підтримувати.

Життєвий цикл документу складається з двох основних стадій: розробка документу, що може включати власне розробку змісту документу, оформлення документу, затвердження документу; стадія “опублікованого” документу, яка може містити: активний доступ, архівний документ короткострокового і довгострокового збереження, знищення документу.

У тому випадку якщо документ перебуває у стадії розробки, він вважається неопублікованим, і права на нього визначаються правами доступу конкретного користувача. Коли документ переходить на другу стадію, він вважається опублікованим, і на нього залишається тільки одне право - доступ на читання.

Залежно від конкретної стадії життєвого циклу документа, системи підрозділяються на: статичні архіви документів (або просто архіви) - системи, що обробляють тільки опубліковані документи; динамічні архіви (або системи управління документами) працюють як з опублікованими документами, так і з тими, що знаходяться у розробці.

Архівна система (або архів) організації розуміється як комплекс програмного й апаратного забезпечення, призначений для вирішення перерахованих нижче задач: організація збереження електронних документів, а також їх облік та контроль, включаючи такі операції, як: видача документа, його повернення й ін.; організація пошуку документів; забезпечення захисту документів від несанкціонованого доступу й аудит роботи; можливість перегляду документів (без завантаження прикладних програм); підтримка анотування документу.

Такі задачі реалізує як статистичний, так і динамічний архів.

Як характерний приклад реалізації статистичного архіву може служити стандартна imaging-система WaterMark, PaperWise, ImagineWise.

Для динамічних архівів додатково мають бути реалізованими такі обов'язкові функції: підтримка колективної роботи з документом, що виражається забезпеченням цілісності документів; підтримка роботи зі складовим документом, тобто таким, що являє собою сукупність інших документів. За своїми характеристиками він аналогічний простому, у нього об'єднують документи за допомогою декількох типів зв'язків; поширення опублікованих документів (або через систему електронної пошти, або через Інтернет); розширений спектр прав доступу до документа, як-то: на редагування, на публікацію, на зняття публікації, на створення нової версії.

При організації систем електронного документообігу однією зі складових є система маршрутизації й контролю виконання, яка оперує документами, що зберігаються в архівній системі.

При організації систем маршрутизації застосовують два основних підходи: документоорієнтований, при якому документ розглядається як головний об'єкт системи і маршрутизується саме він, а всі інші параметри маршрутизації асоціюються винятково з документом; другий підхід орієнтований на роботу, його основний об'єкт - робота до якої може бути прикріплений різноманітний список об'єктів, включаючи документи.

Практично в будь-якому бізнесі, пов'язаному з обслуговуванням клієнтів, а також у діяльності державних органів, зайнятих збиранням даних про фізичні і юридичні особи, виникають стандартні форми. Це можуть бути: платіжні доручення, податкові і митні декларації, анкети персоніфікованого урахування громадян, страхові і статистичні анкети, форми комунальних платежів, бланки перепису населення й анкети для проведення виборів, усілякі опитувальники, що поширюються фірмами для з'ясування рівня попиту на запропоновані товари й послуги, платіжні форми, сліпи з пластикових карт і т.д.

Як приклади основних галузей застосування систем обробки форм, що визначаються спектром зарубіжних (американських, переважно) бізнес-пропозицій, можна привести такі: обробка факсів, податкових декларацій і платіжних доручень по оплаті податків, транспортних документів і накладних, гарантійних талонів і реєстраційних карт, рахунків на оплату у медицині, запровадження замовлень на товари, маркетингових досліджень і опитувань.

Проблема запровадження стандартних форм є суттєво складнішою, ніж, наприклад, використання неструктурованих документів, хоча у обох випадках є багато схожого. До основних технологій, які використовуються у електронному документообігу, відносяться: сканування; фрагментація документа на елементи з різноманітними властивостями (графіка, текст, таблиці, поля форми і т.д.); розпізнавання OCR/ICR/OMR; контекстний аналіз результатів розпізнавання (автоматичний і інтерактивний при участі оператора); зберігання даних у потрібному форматі.

У більшості систем обробки форм втручання людини потрібне на таких етапах, як: 1) заповнення форми; 2) підготування документа до автоматичної обробки; 3) сканування; 4) опрацювання виняткових ситуацій (коли оператор виконує ідентифікацію відсканованого образу документу, який не змогла ідентифікувати система); 5) редагування даних і підтвердження достовірності їхньої обробки.

Ефективність перших трьох етапів може бути значно підвищена при грамотному дизайні форм.

Користування системами управління електронними документами

Системи управління електронними документами (СУЕД) призначені для автоматизації усього комплексу робіт із документами як створеними в конкретному закладі, так і з тими, що надійшли ззовні: введення документів у систему, реєстрація, розподіл і розсилання, редагування і модифікація, оперативне збереження й архівація, пошук і перегляд, витяг і відтворення, контроль виконання, розмежування доступу, довідкова служба та ін. Суттєвим моментом є забезпечення групової (кооперативної) роботи з документами.

При впровадженні систем у практику роботи закладу ставляться такі основні цілі: підвищення продуктивності праці; підвищення оперативності у роботі з документами; поліпшення якості підготовки, обробки й відтворення документів; скорочення кількості непотрібних документів і непотрібних розсилок; поліпшення організації контролю за прямуванням і виконанням документів; розширення можливостей довідково-інформаційного обслуговування.

Розробка конкретної системи управління документами або настроювання покупної системи повинні здійснюватися на основі системного аналізу діяльності закладу, у результаті чого визначаються: основні задачі, рівні організаційної структури, групи користувачів і їхні інформаційні запити, етапи життя документів і відповідальні за них служби, схема функціонування системи, склад і структура бази даних, форми представлення вхідної інформації і вихідних документів.

Конкретний зміст і важливість задач таких систем визначаються на основі аналізу моделі діяльності закладу. У загальному випадку модель діяльності закладу може бути подана як організована сукупність взаємозалежних процесів (ділових процесів його персоналу). До них відносять: процеси обслуговування клієнтів закладу, процеси адміністративно-управлінської і господарської діяльності, організацію фінансових розрахунків, організацію й супровід документообігу, інформаційно-аналітичне забезпечення й ін. процеси.

Системи управління документами одного закладу об'єднуються в єдину систему, структура якої має відбивати організаційну структуру даного закладу. Для функціонування такої системи створюється й використовується загальне транспортно-комунікаційне середовище, наприклад, Інтернет/Інтранет, яке дозволяє здійснювати обмін електронними документами як усередині підрозділів, так і між підрозділами закладу. Таке середовище є основою для ефективного доступу до інформаційних ресурсів і систем закладу, а також до ресурсів і систем родинних і вищих організацій. Інтранет закладу може містити ряд віртуальних підмереж різноманітних підрозділів із різним ступенем захисту в залежності від специфіки їх роботи. Вихід в Інтернет здійснюється, як правило, з урахуванням вимог забезпечення безпеки роботи закладу.

Незалежно від конкретних підрозділів, у цілому по закладу для роботи з документами, як правило, виділяються три рівні користувачів: 1) *керівництво*, 2) *канцелярія*, 3) *виконавці*.

Кожний користувач системи на своєму рівні приймає рішення щодо подальшого просування документів. Йому необхідно надати зручний графічний інтерфейс, за допомогою якого він може вибрати для документів, які йому потрібні, певну "маршрутну карту" з описом етапів обробки цих документів.

Вхідні й вихідні документи підрозділяються та такі види, як: лист, указ, наказ, розпорядження, постанова, заява, договір, звіт, запит, платіжне доручення, кошторис, довідка, акт, рішення, зведення, рахунок, протокол, клопотання, службова записка, оповіщення, протест, касаційна скарга, угода і т.п. Цей перелік дає уявлення про ті документи, що повинні бути опрацьованими (або обробленими) у такій системі. Він може бути розширений шляхом додавання нових записів у таблицю видів документів.

Прогрес у застосуванні електронної технології опрацювання документів (або форм) і поступової заміни паперових форм буде визначатися розвитком електронних комунікацій, і насамперед Інтернет. Його доступність буде ключем

до успіху застосування та розвитку різних електронних документів не тільки на окремо узятому підприємстві або в установі, але й у світовому масштабі.

Автоматизована система документообігу на основі Lotus Notes

Система містить у собі засоби і правила створення документів, ведення електронного архіву (архівного зберігання), підтримки документообігу і опирається на програмно-технічні платформи. Усі інші складові управління (економіка, людські ресурси, фінанси, облік, взаємодія структур) базуються на системі ведення діловодства з метою ефективного використання інформації для досягнення поставлених задач і вирішення проблем, що стоять перед державною чи госпрозрахунковою структурою.

Система призначена для автоматизації діловодного процесу, що включає: створення (внесення, сканування і т.п.), пересилання, обробку документів (візування, ознайомлення, узгодження, затвердження і т.п.) і контроль за їхнім виконанням. Складовими частинами діловодства є: документування (методи або способи створення документації); документація (сукупність документів, взаємозалежних між собою); організація роботи з документами, контроль виконання документів (накази, завдання, розпорядження і т.п.).

Система охоплює питання організації роботи з документами в процесі здійснення управлінських дій. На процес діловодства впливає кількість задіяних структурних одиниць організації (підприємств) - користувачів і доручених їм завдань. Система дозволяє оптимізувати роботу і взаємодію управління із структурними одиницями (підрозділами) і значно поліпшує діяльність підприємства. Система підвищує ефективність контрольної служби, старанність співробітників, швидкість взаємодії підлеглих із керівництвом.

Ідея системи - створення передумов для поетапного переходу від традиційного паперового документообігу до електронного документообігу в частині автоматизації діловодного процесу.

Основні функції і принципи побудови системи, що відповідають принципами побудови інформаційно-аналітичних систем з інтегруючим електронним документообігом, в повній мірі належать платформі Lotus Notes. Lotus Notes – це ціла сукупність технологій: перш за все - це система тиражування (реплікації) та захисту даних.

Основним призначенням Notes є вирішення задач автоматизації діловодства та документообігу, або, як це все називають одним словосполученням, для автоматизації ділових процесів. Додатки для ділових процесів орієнтуються довкола визначення та координації діяльності людей, що спільно виконують деяку роботу. Ці процеси можуть залучати велику кількість людей в одній або навіть декількох організаціях, які часто при цьому територіально роз'єднані: це можуть бути співробітники різних відділів, постачальники та покупці, яким необхідно мати одночасний доступ до спільно використовуваних даних та між якими циркулює, як правило, велика кількість неоднорідної та слабо структурованої інформації.

Властивості системи:

а) корпоративність:

– користувач будь-якого територіального підрозділу організації має можливість взаємодіяти з будь-яким співробітником організації, підключеним до системи;

- методика, що змінюється - робота з всім персоналом організації;
- списки персоналу всіх організацій доступні скрізь;
- загальнодоступні списки зовнішніх організацій;
- єдина (в рамках організації) поштова – транспортна система Lotus Notes.

б) одноманітність:

- призначений для користувача інтерфейс програм рівня територіального підрозділу організації, аналогічний інтерфейсу бази даних корпоративного рівня.

в) масштабованість функціональна:

- під'єднання нових комплектів СЕДО або бази даних ;
- під'єднання зовнішніх систем на РСУБД;
- необхідні зміни в функціональності або дизайні системи можливо виконувати з єдиного робочого місця.

г) масштабованість географічна:

- зовнішня - необмежена кількість нових організацій;
- внутрішня - необмежена кількість нових підрозділів, співробітників;

д) підтримка багатоструктурності організації:

- централізоване адміністрування і управління корпоративними базами даних;
- самостійність відділень у виборі набору баз даних і управління ними ;
- можливість делегування функцій в регіональні підрозділи, визначеним співробітникам по веденню бази даних, в т.ч. по настроюванню системи.

Система автоматизації діловодства та управління документами (САД) поділяється на декілька функціональних блоків (підсистем), що визначаються структурою організації. Ці підсистеми містять у собі функції виконавця, керівників різних рангів, контрольної служби, канцелярії, відділу кадрів і т.д. При зміні структури підприємства або внесенні нових посадових інструкцій можлива модернізація кожної підсистеми окремо або системи в цілому.

Для інформаційного обміну між компонентами системи використовується локальна обчислювальна мережа (ЛОМ), глобальна обчислювальна мережа (ГЛОМ) або зв'язок за допомогою телефонних комутованих каналів. Обмін документами між компонентами системи здійснюється автоматично.

Система базового електронного документообігу відповідає наступним вимогам і структурі та виконує такі функції:

1. Прийом вхідного документу, реєстрація, введення головних параметрів (реквізитів), відкриття картки на документ.
2. Внесення відсканованого образу документу (змісту) в картку.
3. Направлення (розсилання) документів на розгляд керівництву (отримання резолюції). Внесення резолюції (зміст завдання, кому направлено, контрольний термін). Направлення фізичного документу з реєстраційною карткою.
4. Одержання розглянутих документів від керівництва з резолюцією. Введення резолюції в картку реєстрації , напрямку, відповідального виконавця і контрольного терміну виконання.
5. Направлення (розсилання) документів з резолюцією керівництва на виконання в структурні підрозділи.
6. Одержання виконаних (після виконання) документів, внесення позначок про виконання. Направлення документу згідно номенклатури справ у відповідну папку.
7. Якщо на основі виконаного документу сформовано вихідний документ - реєстрація вихідного документу.

8. Одержання оперативного звіту про роботи, що повинні бути виконані протягом 1 доби, 3 діб, 5 діб.

9. Формування оперативного звіту про документи, що повинні бути виконані протягом 3 діб по кожному підрозділу, про роботи, що вимагають виконання і розсилання у вигляді документу - нагадування.

10. Формування оперативного (узагальненого) звіту про документи, що повинні бути виконані протягом 1 доби в розрізі: підрозділ, кількість робіт.

11. Одержання оперативного звіту про документи, що повинні бути виконані протягом 1 доби, про невиконані документи з інформацією: хто не виконав, місцезнаходження та етап роботи з документом.

Система АД не замикається на управлінні загального діловодства, а має продовження в системах управління діловодством підрозділів (управлінь, відділів).

Канцелярія здійснює координацію документообігу, головна частина якого проходить через підрозділи.

На рівні підрозділів ведеться внутрішній документообіг, фіксування вхідних і вихідних документів. При цьому в системі управління документами банку робиться відповідна позначка.

З канцелярії банку здійснюється повний наскрізний контроль за усіма маршрутами проходження документів, включно із замкнутими системами документообігу підрозділів.

Головні автоматизовані функції систем автоматизованого документообігу

1. Призначення контрольних карток і завдань.
2. Автоматизоване заповнення реквізитів користувача САД.
3. Здійснення санкціонування ЕД.
4. Друк документу.
5. Архівне зберігання ЕД.
6. Перегляд документу перед друком.
7. Створення грифів доступу.
8. Повідомлення про появу документу.
9. Встановлення грифів доступу.
10. Автоматичне виконання операцій з ЕД.
11. Надходження і відправка повідомлень і нагадувань про контроль та строки виконання.
12. Перегляд схеми і/або розкладу ЕД.
13. Заповнення деяких типів документів за наявними шаблонами.
14. Введення оригіналу документу.
15. Повернення зі створенням службового ЕД.
16. Вибір типу документу.
17. Виконання завдання і контрольних карток.
18. Проведення санкціонування ЕД.
19. Відправка документу на розгляд.
20. Відправка документу на санкціонування ЕД.
21. Відправка на затвердження та засвідчення.
22. Перенесення терміну виконання.
23. Проведення розгляду.
24. Створення адресатів.

25. Створення документу-відповіді на вхідний документ.
26. Створення кореспондентів.
27. Створення нового індексу ЕД.
28. Створення підрозділів.
29. Складання списку осіб, що санкціонують ЕД.
30. Затвердження перенесення терміну виконання.
31. Передача документу для складання іншій особі.
32. Вибір типу та заповнення ЕД.
33. Виконання завдання і службових документів.
34. Відправка ЕД на розгляд.
35. Відправка ЕД на узгодження та санкціонування.
36. Відправка на підпис.
37. Перенесення терміну виконання.
38. Прийом виконання.
39. Проведення розгляду.
40. Упорядкування списку осіб, що підписують документ.
41. Затвердження перенесення терміну виконання.
42. Створення нової штатної одиниці.
43. Призначення на посаду.
44. Надання відпустки.
45. Прийом співробітника на роботу.
46. Створення службових документів.
47. Створення особової картки.
48. Звільнення та переміщення співробітника.

Система базується на програмному забезпеченні Lotus Notes, конкретній операційній системі (MS Windows, Windows NT, OS/2, UNIX...) і IBM сумісному персональному комп'ютері. Для функціонування САД на робочих станціях користувачів повинна бути встановлена програма для групової роботи - Lotus Notes 4.5 і вище.

РОЗДІЛ 9

ПРОГРАМНІ ЗАСОБИ РОБОТИ З БАЗАМИ ТА СХОВИЩАМИ ДАНИХ

Основні об'єкти СУБД Access , їх властивості, способи формування та обробки

База даних – організована структура представлення і зберігання інформації з використанням певної моделі даних. Переважна більшість баз даних побудовані на реляційній моделі. Назва “реляційна” пов’язана з тим, що кожний запис в такій базі даних має інформацію, яка відноситься тільки до одного конкретного об’єкту. В реляційній базі всі дані представлені в вигляді таблиць.

До програмних засобів, що реалізують функції баз даних відносяться Oracle, Paradox, Lotus, FoxPro, Access та інші. У подальшому розглядатимемо одну з них, а саме – Access.

В базі даних **Access** основними об’єктами є таблиці, запити, форми, звіти, макроси і модулі.

Таблиця – об’єкт, який використовується для збереження даних. Таблиця складається з **полів** (стовпчиків), в яких зберігаються різні дані, і **записів** (рядків). В записі зібрана вся інформація про деякий об’єкт.

Запит – об’єкт, який дозволяє користувачу отримати потрібні дані з одної або декількох таблиць. Для створення запиту можна використовувати бланк QBE (запит по зразку) або інструкцію SQL. Можна створювати запити на вибірку, поновлення, видалення або додавання даних. За допомогою запитів також можна створювати нові таблиці, використовуючи дані з одної або декількох існуючих таблиць.

Форма – об’єкт, призначений в основному для вводу даних, відображення їх на екрані або керування роботою додатку.

Звіт – об’єкт, призначений для створення документа, який в подальшому може бути роздрукований або включений в документ іншого додатку.

Макрос – послідовність команд, генерованих системою відповідно до дій користувача, які після їх створення можуть повторюватися як фрагменти програм, що повторюють ці дії.

Модуль – набір команд, створюваних користувачем на власний розсуд для виконання послідовності дій над об’єктами бази даних, що інтерпретуються і виконуються системою як самостійна програма.

Властивості полів бази даних:

- ім’я поля,
- тип поля,
- розмір поля,
- формат поля,
- маска уведення даних,
- підпис,
- значення за умовчужанням,
- умова на значення,
- повідомлення про помилку,
- обов’язковість заповнення,
- дозвіл на уведення порожніх рядків,
- ознака індексованості поля.

Типи даних, що використовуються при роботі з базою:

- *текстовий* (для набору тексту до 255 символів),
- *поле Мемо* (для фрагментів тексту до 65535 символів),
- *числовий* (для дійсних чисел),
- *дата і час* (для календарних дат і поточного часу),
- *грошовий* (для чисел, якими виражено грошові суми),
- *рахівник* (для автоматичного нумерування рядків),
- *логічний* (для даних типу “Так” або “Ні”)
- *поле об’єкта OLE* (для даних графічного чи мультимедійного типу),
- *гіперпосилання* (для зберігання адрес Web-об’єктів),
- *майстер підстановок* (для об’єктів, налаштуванням яких можна автоматизувати введення даних).

Об’єкти бази даних формуються внутрішніми засобами бази даних, до яких відноситься “*Конструктор*” та “*Майстер*”. При створенні таблиць необхідно передбачити зв’язок між ними через так звані “ключові” поля. Вимогою до таких полів є не повторюваність записів у них (унікальність).

Таблиці формуються і заповнюються звичним способом. Кожна з них повинна мати свою назву, назви полів (одне з них має бути визначене ключовим), записи до полів.

Будь-який об’єкт таблиці після її створення легко відформатувати на власний смак, використовуючи стандартну технологію, а саме: виставити курсор на потрібний об’єкт (назву таблиці, назву поля, запис, клітку, границю тощо), “клацнути” правою кнопкою миші, викликавши тим самим контекстне меню, активізувати серед висвітленого списку необхідну команду.

Запити дають широкі можливості для вибору, сортування і обчислення з використанням даних однієї або кількох таблиць.

Щоб відкрити вікно нового запиту в режимі конструктора, і вікні бази даних потрібно перейти на вкладку **Запрос** і натиснути кнопку **Создать**, яка міститься з правого боку від списку запитів. Access відкриє вікно діалогу **Новый запрос**. Є вибір: створити запит самостійно в режимі **Конструктора** або скористатися допомогою **Майстра** для створення одного з декількох типів запитів.

Щоб відкрити існуючий запит в режимі конструктора, треба виділити його ім’я на вкладці **Запрос** і натиснути кнопку **Конструктор**. Запит відкривається в режимі Конструктор. В верхній частині вікна запиту, що висвітлиться, знаходяться списки полів (назви стовпців таблиці), в нижній частині - бланк запиту.

Кожен стовпець бланку представляє одне поле, яке використовується в запиті. Поле може просто належати одній з таблиць, бути обчислюваним (його значення розраховується на основі одного або декількох полів таблиці), або підсумковим, яке використовує одну із вбудованих функцій Microsoft Access.

Полям запиту можна надавати імена, які будуть відображатися в заголовках стовпців при виведенні набору записів запиту, а для генерації обчислюваних полів можна використовувати вирази довільного ступеню складності.

До кожного поля запиту вноситься ім’я таблиці, поле, якого запит стосується, формується команда на виконання (чи не виконання) сортування, вказується – виводити чи не виводити на екран поля в наборі записів, формується умова відбору записів.

Вибір полів здійснюється простим перетягуванням назви поля з таблиці в потрібний стовпчик бланку.

Якщо потрібно відібрати записи з конкретним значенням поля, треба ввести його у комірку **Условие отбора** цього поля. Текстова значення, яке використовується в якості умови відбору, повинно бути вміщене в лапки.

Окрім звичайних операторів порівняння Access пропонує три спеціальних оператора, корисних для відбору даних, які виводяться в наборі записів запиту:

BETWEEN – визначає діапазон значень.

IN – задає список значень, що використовується для порівняння.

LIKE – оператор, корисний для пошуку зразків в текстових полях: в зразок пошуку можна включити символи шаблону – “?” (довільний символ в даній позиції), “*” (означає довільні символи), “#” (в даній позиції повинна бути цифра).

При формуванні обчислювальних полів використовують оператори:

- створити текстовий рядок поєднанням рядків.
- + Додає два арифметичних вирази.
- Віднімає від першого арифметичного виразу другий.
- * Перемножує два арифметичних вирази.
- / Ділить перший арифметичний вираз на другий вираз.
- \ Заокруглює два арифметичних вирази до цілих значень і ділить перше на друге. Результат округлюється до цілого.

^ Вираховує степінь першого арифметичного виразу, степінь задається другим арифметичним виразом.

MOD Округлює обидва арифметичних вирази до цілих значень, ділить перше на друге і повертає залишок.

& Створює текстовий рядок, як результат приєднання другого рядка в кінець першого.

Для побудови складних виразів Access надає утиліту, яка називається **Построитель выражений**. В верхній частині цього вікна міститься область введення, яка призначена для створення виразів. В нижній частині вікна знаходяться три списки, призначені для пошуку імен полів і функцій, необхідних для створення виразів.

У базі даних передбачено можливість використання як згаданих вище запитів на вибірку даних, та к інших, а саме: запитів з параметром, підсумкових запитів, запитів на зміну даних, перехресних запитів.

Форми, на відміну від таблиць, дозволяють вводити і виводити дані не у вигляді таблиць, а за більш привабливою для користувача схемою. Формують структуру форми, використовуючи як таблиці, так і запити. База даних має внутрішній засіб автоматичного формування форми – **Автоформу**. Остання створює форму за однією із трьох схем, які вказує користувач, – у стовпцем, стрічкою або таблицею. Проста форма стосується однієї таблиці, складна – кількох. У кожному з цих варіантів форма має три розділи: область заголовка, область даних, область поміток. Кожен із цих розділів заповнюється користувачем так, щоб мати можливість ефективно працювати з формою. Другий розділ заповнюється, зокрема, елементами керування (перемикачами, прапорцями, списками тощо), необхідними для автоматизації введення даних.

Звіти нагадують форми, але мають інше функціональне призначення – вони створюються для автоматизації виведення даних, тобто містять додаткові елементи

для урахування параметрів друкуючого пристрою та паперу, на який має виводитись результат.

Технологія створення бази даних і таблиць

Перше, з чого слід починати, це з визначення проекту таблиць для конкретної задачі чи ситуації.

Приклади проекту таблиць “Список” та “Заняття”:

Ім'я поля	Тип поля	Розмір поля	Ім'я поля	Тип поля	Розмір поля
Номер 1	Лічильник	Довге ціле	Номер 2	Лічильник	
П.І.ПБ	Текстовий	14	П.І.ПБ	Текстовий	14
Предмет	Текстовий	12	Аудиторія	Числовий	Ціле
Дата нар.	Дата/час	Короткий	День	Текстовий	12
Стаж	Числовий	Байт	Час	Текстовий	12
Навантаження	Числовий	Байт			
оклад	Числовий	Ціле			

Далі активізуємо СУБД Access.

Встановлюємо вказівник “Нова база даних”. “Ок”. Вказати ім.'я бази. “Создать”.

Відкриється вікно бази даних з надписами “Таблиці”, “Запити”, “Форми”, “Звіти”

Активізуємо “Таблиці”, “Создать”.

Вибрати один із варіантів створення таблиці: *Режим таблиці*, *Конструктор*, *Майстер таблиць*, *Імпорт таблиць*. Вибрати «**Конструктор**».

З'явиться вікно виду:

<input type="checkbox"/>				
	Имя поля	Тип данных	Описание	
▽				
Свойства поля				
Общие		Подстановка		

Вікно заповнюється відповідно сформованого раніше проекту таблиці.

Після заповнення таблиці необхідно активізувати поле «Номер 1», за ним – «Ключевое поле».

Активізувати (верхня ліва позиція), «Закреть», «Сохранить», «Да», «Ок».

Повторювати для формування кожної таблиці, передбаченої проектом.

Заповнення таблиць даними

Здійснюється після активізації створеної раніше таблиці (“Открыть”).

На екрані з'явиться порожній бланк таблиці, який слід заповнити даними.

Робота з таблицями

Після створення таблиці і заповнення її даними все, що стосується роботи з нею, може з успіхом здійснюватися з використанням контекстних меню, які активізуються після відкриття таблиці, це:

– контекстне меню **стовпців** (команди: *сортировка*, *найти*, *ширина столбца*, *закрепите столбцы*, *добавить столбец*, *удалить столбец*, *переименовать столбец*, *столбец подстановок*),

– контекстне меню **рядків** (команди: *вырезать*, *копировать*, *вставить*, *высота строки*);

- контекстне меню **таблиці** (команди: *открыть, конструктор, печать, предварительный просмотр, вырезать, копировать, сохранить, удалить, переименовать, свойства*);
- контекстне меню **комірки** (команди: *формат по выделенному, исключить выделенное, фильтр, удалить фильтр, сортировать, вырезать, копировать, вставить, добавить объект, гиперссылка*);
- контекстне меню **назви таблиці** (команди: *формат по выделенному, исключить выделенное, фильтр, удалить фильтр, изменить фильтр, применить фильтр, шрифт, ячейки, отобразить столбцы*).

Створення зв'язків між таблицями

Для створення зв'язків між таблицями використовують команду «Схема данных», вказуючи, яке з полів однієї таблиці має зв'язок з яким із полів іншої таблиці. Зміни, які стосуються зв'язків легко здійснювати за допомогою контекстного меню лінії, яка з'єднує таблиці.

Створення запитів на вибірку даних

В режимі таблиці доступні самі різні операції з даними - огляд, сортування, фільтрація, поновлення і друк. Але достатньо часто приходиться проводити обчислення і огляд даних з декількох таблиць. Відобразити потрібні дані можна за допомогою запитів.

Після виконання запиту Microsoft Access створює набір записів, які містять відібрані дані. В більшості випадків з набором записів можна працювати так само, як з таблицею: можна проглянути і відібрати інформацію, роздрукувати і поновити дані. Але на відміну від реальної таблиці, цей набір записів фізично не існує в базі даних. Access створює набір записів з даних таблиць тільки під час виконання запиту. Якщо змінити дані в наборі записів, Access внесе відповідні зміни в таблицю, на базі яких побудований запит.

При вивченні форм і звітів виявляється, що запити є найкращим способом виділення даних, необхідних для вирішення визначеного завдання. Запити можуть слугувати джерелами даних таких елементів керування, як список і поле зі списком, що спрощує введення даних.

Відкриття вікна нового запиту

Щоб відкрити вікно нового запиту в режимі конструктора, і вікні бази даних потрібно перейти на вкладку **Запрос** і натиснути кнопку **Создать**. Access відкриє вікно діалогу **Новый запрос**.

Є вибір: створити запит самостійно в режимі конструктора або скористатися допомогою майстра для створення одного з декількох типів запитів. Щоб відкрити існуючий запит в режимі конструктора, треба виділити його ім'я на вкладці **Запрос** і натиснути кнопку **Конструктор**. Запит відкривається в режимі Конструктор. В верхній частині вікна запиту знаходяться списки полів (назви стовпчиків таблиці), в нижній частині - бланк запиту.

Вибір даних з однієї таблиці

Запит на основі однієї таблиці формується так: відкрити вікно бази даних, вибрати потрібну таблицю, розкрити список кнопки **Новый объект** на панелі інструментів і вибрати пункт **Новый запрос** і натиснути кнопку **ОК** (якщо рядок **Имя таблицы** не виводиться в бланку запиту, слід вибрати команду **Вид/Имена таблиц**).

Відкривається вікно конструктора (рис.25), воно розділене на дві частини . В верхній частині знаходяться списки полів таблиць або запитів, на підставі яких створюється новий запит. В нижній розміщений бланк QBE (Query By Example - запит по зразку), в якому виконується вся робота по створенню нового запиту. Кожний стовпчик бланку представляє одне поле, яке використовується в запиті. Поле може просто належати одній з таблиць, бути обчислюваним (його значення розраховується на основі одного або декількох полів таблиці), або підсумковим, яке використовує одну із вбудованих функцій Microsoft Access.

Полям запиту можна надавати імена, які будуть відображатися і заголовках стовпчиків при виведенні набору записів запиту, а для генерації обчислюваних полів можна використовувати вирази любого ступеню складності.

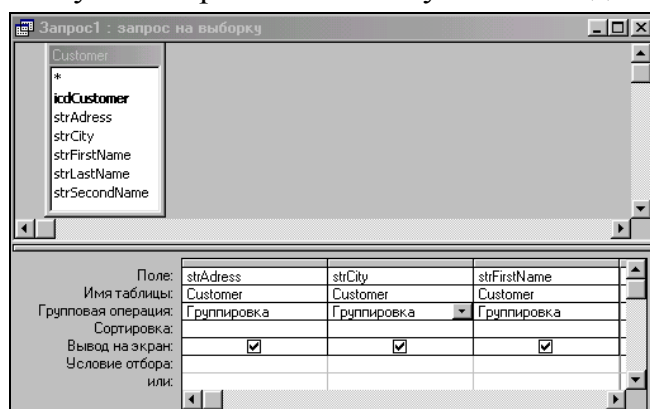


Рис. 25

В зв'язку з тим, що була виконана команда **Вид/Імена таблиць**, в даному рядку бланка запиту Access виведе ім'я таблиці, з якої вибране поле. В третьому рядку бланка можна задати, чи потрібно виконувати сортування по вибраному або обчислюваному полю.

Прапорці в бланку **Вивод на екран** відповідають за вивід на екран полів в наборі записів. По замовчуванню Access виводить на екран всі поля, які містить бланк запиту. Але деякі поля включаються в запит тільки для відбору потрібних записів, а виводити їх на екран зовсім не обов'язково. Щоб виключити таке поле з набору записів, треба зняти його прапорець в рядку **Вивод на екран**.

Для введення умов відбору записів використовується рядок **Условие отбора** і рядок **или**.

Першим кроком при створенні запиту є вибір полів, які включаються в набір записів. Це можна зробити, просто перетягнувши поле в потрібний стовпчик бланка зі списку полів в верхній частині вікна. При перетягуванні поля вказівник мишки перетворюється в маленький прямокутник.

Якщо потрібно включити в запит всі поля таблиці, то достатньо перетягнути значок "*" зі списку полів в бланк QBE.

Інший спосіб ввести в запит всі поля таблиці - це двічі клацнути на заголовку списку полів в верхній частині вікна: таким чином виділяються всі поля таблиці. Потім перетягнути виділені поля в рядок **Поле бланка** запиту. Вказівник миші перетвориться в значок з зображенням декількох прямокутників, який показує, що перетягуються декілька полів. Коли відпускається кнопка миші, Access помістить в бланк запиту всі поля таблиці.

Встановлення властивостей полів

В загальному випадку поля, які виводяться в наборі записів запиту, наслідують властивості для відповідних полів таблиці. Можна задати інші значення властивостей, а саме: **Описание** (інформація, яка виводиться в рядку стану вікна запита в режимі таблиці, коли поле стає поточним), **Формат поля** (представлення даних на екрані), **Число десятичних знаків** (для числових даних), **Маска вводу** і **Підпись** (заголовок стовпчика).

Щоб задати властивості деякого поля, потрібно клацнути налюбій чарунці відповідного стовпчика в бланку запита і натиснути кнопку **Свойства** на панелі інструментів або вибрати команду **Вид/Свойства**.

Введення умов відбору

Якщо потрібно відібрати записи з конкретним значенням поля, треба ввести його чарунку **Условие отбора** цього поля. Текстова значення, яке використовується в якості умови відбору, повинне бути вміщене в лапки.

В випадку, якщо нас цікавить декілька значень, вони вводяться в рядок **Условие отбора** і розділяються логічним оператором **OR (или)**.

Коли вводяться умови відбору для декількох полів, то всі вирази в рядку **Условие отбора** або в рядку **или** повинні приймати значення **Истина** для любого запису, який включається в набір записів запиту. Це означає, що Access виконує логічну операцію **AND** над умовами відбору, які знаходяться в одному рядку. Щоб результат операції **AND** мав значення **Истина**, умови повинні бути істинними; тільки в цьому випадку запис відбирається запитом. Наприклад, ми вибираємо записи з таблиці, в якій знаходяться дані про робітників. Умовою відбору обрано поле “Загальний стаж” і його значення:

$$>10 \text{ AND } <20$$

Це означає, що будуть відібрані тільки ті записи, значення яких відповідає обом умовам в рядку **Условие отбора** (стаж більше 10 років, але не перевищує 20). Всі інші записи в таблицю запиту не попадуть.

Коли задаються для деякого поля декілька умов відбору, які з’єднані логічним оператором **OR**, то для того, щоб запис був відібраний запитом, істинним повинна бути хоча б одна з них. Є два способи задати декілька пов’язаних оператором **OR** умов для одного поля. Можна ввести всі умови в одну комірку рядка **Условие отбора** і з’єднати їх оператором **OR**. Наприклад, з таблиці про поставників продукції запис в комірці **Условие отбора**:

$$\text{“Київ” OR “Вінниця”},$$

означає, що будуть відібрані всі записи про поставників, що знаходяться в містах Київ і Вінниця.

Інший варіант: введення кожної умови в окрему комірку рядка **или**. При використанні декількох рядків **или** для відбору запису достатньо виконання всіх умов в одному з рядків **или**.

Окрім звичайних операторів порівняння Access пропонує три спеціальних оператора, корисних для відбору даних, які виводяться в наборі записів запиту.

BETWEEN. Визначає діапазон значень. **Between 10** означає те саме, що і вираз **And 20 >=10 And <=20**.

IN. Задає список значень, що використовується для порівняння. Вираз IN(“Київ”, “Вінниця”) означає те саме, що і вираз “Київ” OR “Вінниця”.

LIKE. Оператор, корисний для пошуку зразків в текстових полях. В зразок пошуку можна включити символи шаблону, “?” заміняє любий символ в даній позиції, а “*” означає любую кількість символів в даній позиції. Символ “#” вказує, що в даній позиції повинна бути цифра.

Умови відбору для дат і часу

Microsoft Access зберігає значення дат і часу як числа з плаваючою комою і з подвійною точністю. Значення з лівого боку від десяткової коми відповідає даті, а дробова частина числа представляє час доби.

Щоб повідомити Access про те, що вводиться дата і час, значення вміщується в символи числа (#). Наприклад, #10 Квітень 2003# і #10/04/03# визначають одну і ту саму дату.

Access дає декілька функцій, які можна використовувати при завданні умов відбору для дат і часу:

Day (дата). Повертає значення дня місяця в діапазоні від 1 до 31.

Month (дата). Повертає значення місяця року в діапазоні від 1 до 12.

Year (дата). Повертає значення року в діапазоні від 100 до 9999.

Weekday (дата). Повертає значення чисел від 1 (Неділя) до 7 (Субота), які відповідають дням тижня.

Hour (дата). Повертає ціле число від 0 до 23, які представляють значення часу.

DatePart (інтервал, дата). Повертає номер кварталу або номер тижня в залежності від того, який код інтервалу задається (“q” - для визначення кварталу, “ww” - для визначення порядкового номера тижня в році).

Date(). Повертає поточну системну дату.

Використання параметрів запиту

До сих пір ми вводили умови відбору безпосередньо в бланк запиту в режимі конструктора. Але на етапі створення запиту на завжди можна визначити, які значіння повинен відшукувати Access. Потрібно включити в запит параметр, і при кожному виконанні запиту Access буде вимагати конкретні умови відбору.

Щоб визначити параметр, потрібно ввести в рядок **Условие отбора** замість конкретного значення ім'я або фразу, яка вміщена в квадратні дужки. Те, що вміщене всередині квадратних дужок, Access розглядає як ім'я параметра. Воно виводиться в вікні діалогу при виконанні запиту, тому в якості імені параметра розумно використовувати змістовну фразу. В одному запиті можна задати декілька параметрів, при цьому ім'я кожного параметру повинно бути унікальним і інформативним.

Для кожного параметра запиту можна вказати тип даних. Access використовує цю інформацію для перевірки введеного значення. Наприклад, якщо визначено параметр як числовий, Access відкине літерні символи в значенні параметра. За умовчанням Access надає параметрам запиту текстовий тип даних. Якщо потрібно змінити тип даних, треба вибрати команду **Запрос/Параметры**, і Access виведе на екран вікно діалогу **Параметры запроса**. В цьому вікні діалогу вводиться ім'я кожного параметра, тип якого ми хочемо визначити, в стовпчик **Параметр** в такому вигляді, в якому воно було вказане в бланку запиту, але без

квадратних дужок. В стовпчику **Тип данных** треба встановити потрібний тип даних, який вибирається зі списку, що розкривається. Після визначення всіх параметрів натискаємо кнопку **ОК**.

При виконанні запиту Access попросить ввести по чергово значення для кожного з параметрів, використовуючи вікно діалогу.

Багатотабличні запити

Розглянувши можливості запитів, які основані на одній таблиці, на базі отриманих знань легко організувати перегляд об'єднаних даних з декількох пов'язаних таблиць. Здатність запитів відбирати дані з декількох таблиць особливо корисна при створенні форм і звітів.

Розглянемо приклад, в якому об'єднується інформація з двох таблиць. В вікні бази даних треба перейти на вкладку **Запросы** і натиснути кнопку **Создать**. В вікні діалогу **Новый запрос** вибрати **Конструктор** і натиснути кнопку **ОК**. Access відкриє вікно нового запиту в режимі конструктора і виведе на екран вікно діалогу **Добавление таблицы**. Вікно діалогу дозволяє вибрати таблиці і запити, які будуть базовими для нового запиту. Вибираються дві таблиці і закривається вікно.

Якщо зв'язок між базовими таблицями був раніше визначений, то верхня частина вікна запиту в режимі конструктора буде виглядати так, як показано на малюнку.

Access пов'язує використовувані в запиті таблиці на основі інформації про зв'язок, яка задана при їх створенні (лінія, яка з'єднує первинний ключ одної таблиці з відповідним полем іншої). Якщо зв'язок між таблицями не визначений, Access сам прийме рішення, встановивши зв'язок між полями з однаковими іменами і співпадаючими типами даних (рис.26).

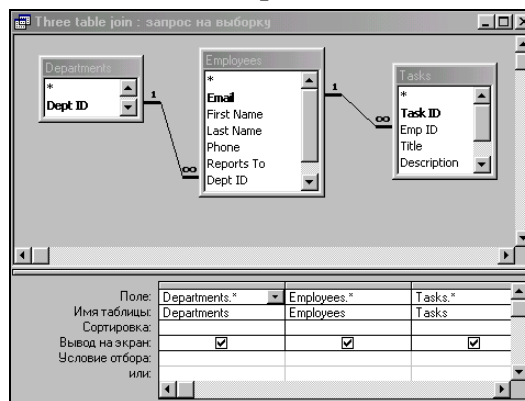


Рис. 26

Якщо зв'язок між базовими таблицями був раніше визначений, то верхня частина вікна запиту в режимі конструктора буде виглядати так, як показано на малюнку.

Access пов'язує використовувані в запиті таблиці на основі інформації про зв'язок, яка задана при їх створенні (лінія, яка з'єднує первинний ключ одної таблиці з відповідним полем іншої). Якщо зв'язок між таблицями не визначений, Access сам прийме рішення, встановивши зв'язок між полями з однаковими іменами і співпадаючими типами даних.

Користувач включає в бланк запиту необхідні поля з двох таблиць. Побачити результат запиту можна, переключившись у режим таблиці.

Як уже згадувалося, у вікні режиму таблиці можна виконувати з набором записів запиту майже всі дії, які доступні для звичайних таблиць (рис.27).

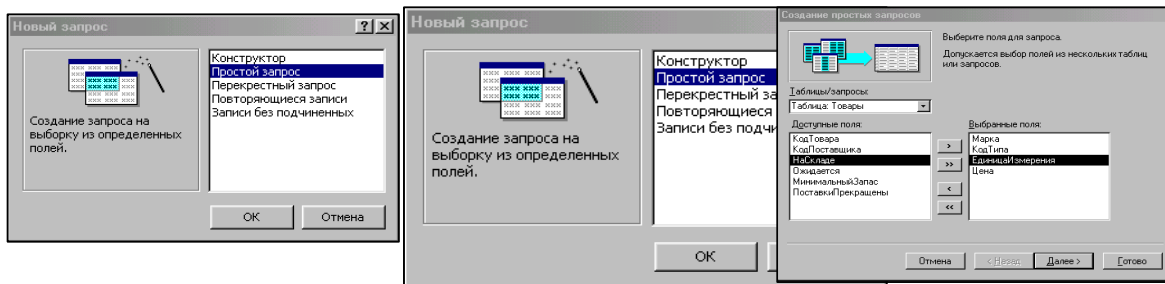


Рис.27

Одним з найцікавіших аспектів багатотабличних запитів є можливість зміни даних вихідних таблиць прямо в наборі записів.

Створення запиту на основі іншого запиту

При створенні запиту в режимі конструктора вікно діалогу **Добавление таблицы** дозволяє вибрати в якості джерела даних для нового запиту не тільки таблиці, але і запити. Дійсно, побудова одного запиту на основі іншого - це ще один спосіб роботи з даними з декількох таблиць: спочатку створюється один запит, за допомогою якого вирішується визначене коло задач і відбирається сукупність даних з декількох таблиць, а потім на його основі будується інший для отримання кінцевого набору записів.

Використання майстра запитів

В вікні бази даних перейти на вкладку **Запросы** і натиснути кнопку **Создать**.

В діалоговому вікні **Новый запрос** вибрати майстра **Простой запрос**. **ОК**.

В діалоговому вікні, що з'явилася, вказати ім'я таблиці або запиту, на якому буде збудований новий запит. Потім вибрати поля, з яких повинні бути відновлені дані.

Якщо необхідно, вказати додаткові таблиці або запит, а потім вибрати з них поля, які повинні бути використані.

Закінчивши роботу в цьому діалоговому вікні, натиснути **ОК**. Потрібно слідувати інструкціям, які виникають в наступних діалогових вікнах майстра. В останньому діалоговому вікні користувачу пропонується вибір виконати запит або продивитися його структуру в режимі конструктора. Якщо отриманий запит не відповідає вимогам, можна знову звернутися до майстра або внести зміну в запит в режимі конструктора.

Відкриття, копіювання, збереження, перейменування і видалення запитів

Користувач може відкрити в режимі конструктора різні запити: запит на вибірку, перехресний запит і запит на зміну. Запит на вибірку і перехресний запит також можна відкрити в режимі таблиці для огляду результатів.

Можна створити ярлик для відкриття об'єкта бази даних, яка знаходиться або на комп'ютері користувача, або на файловому сервері мережі або в директорії для спільного доступу. В Microsoft Windows можна створити ярлик, перемістивши за допомогою миші об'єкт з вікна бази даних в робочий стіл або папку. Інший спосіб – клацнути правою кнопкою миші потрібний об'єкт (запит, наприклад) і вибрати команду **Создать ярлык**. Щоб створити ярлик не на робочому столі, треба ввести новий шлях в поле **Размещение**.

Для копіювання вибирається об'єкт і натискається кнопка **Копировать** на панелі інструментів. Під час копіювання об'єкта в іншу базу даних, закривається поточна база і відкривається та, в яку потрібно вставити об'єкт. При відкритому вікні бази даних натиснути кнопку **Вставить** на панелі інструментів.

Збереження запиту відбувається шляхом натискання кнопки **Сохранить** на панелі інструментів.

Для збереження копії об'єкта бази даних з новим ім'ям або в іншому файлі, при умові що об'єкт відкритий або виділений, потрібно вибрати команду **Сохранить как/Экспорт** в меню **Файл**. Щоб зберегти об'єкт в поточній базі даних, треба вибрати параметр **В текущей базе данных** в діалоговому вікні **Сохранение объекта**, ввести ім'я об'єкта і натиснути **ОК**.

Для перейменування запита потрібно впевнитися, що об'єкт бази даних закритий. Далі в вікні бази даних вибрати вкладку **Запросы**, яка містить потрібний об'єкт. Натиснути кнопку миші на імені об'єкта, а потім знову натиснути кнопку миші, щоб змінити ім'я, ввести нове ім'я.

Для видалення об'єкта виділити його і натиснути кнопку **Delete**.

Виконання розрахунків в запитах

Можна виконувати обчислення злюбими полями таблиці і зробити обчислюваний вираз новим полем в наборі записів. Для цього можна використати будь-які із вбудованих функцій **Access**. Окрім цього, поля запиту можуть містити дані, які отримані за допомогою арифметичних операцій над полями таблиці.

Обчислюване поле може також містити результат конкатенації (об'єднання) значень текстових полів або рядкових (текстових) констант. Щоб створити рядкову константу, потрібно заключити текст в подвійні або одинарні лапки. В якості операції конкатенації використовується символ "&". Наприклад, можна створити поле, яке буде містити результат об'єднання поля **Прізвище** і поля **Ім'я**: *[Прізвище]& [Ім'я]*.

В виразах можна використовувати все ті ж оператори для обчислень: +, -, *, /, \, ^/ MOD, &/

Якщо після клацання в чарунці рядка **Поле** (де ми маємо ввести вираз) натиснути **SHIFT+F2**, то відкриється вікно **Область ввода**, в якому зручно задавати довгі вирази.

Використання побудови виразів

Для побудови складних виразів **Access** надає утиліту, яка називається **Построитель выражений**. Для того щоб запустити її, потрібно клацнути по пустому полю в бланку запита і натиснути кнопку **Построить** на панелі інструментів. **Microsoft Access** відкриє вікно **Построитель выражений**, яке показано на рис. 28.

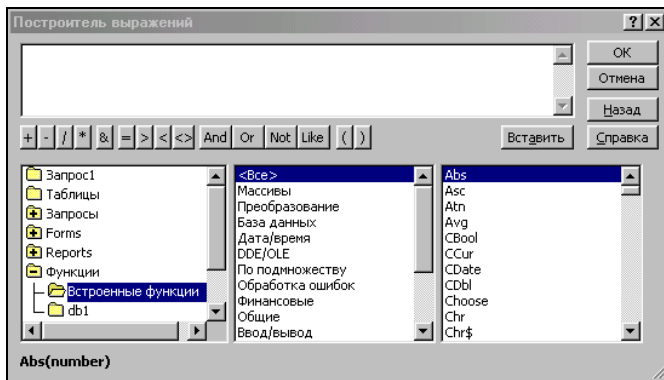


Рис. 28

В верхній частині цього вікна міститься порожня область введення, яка призначена для створення виразів. Можна самому ввести вираз, але легше використати різні кнопки операторів, які містяться прямо під областю введення. В нижній частині вікна знаходяться три списки, призначені для пошуку імен полів і функцій, необхідних для створення виразів.

Наприклад, в нас є таблиця з даними, яка складається з полів: *Прізвище, Ім'я, По батькові, Дата народження, Фізика, Хімія, Алгебра, Геометрія, Історія, Українська література, Українська мова*. Формат останніх семи полів числовий (оцінки за 12-ти бальною шкалою). Необхідно обчислити середню оцінку за навчання для кожного учня.

В бланку запити вибирається вільне поле і запускається **Построитель выражений**. В нижній частині вікна, що з'явилося, в правому полі відшукуємо таблицю, на основі якої будується запит. При виборі таблиці, у центральному списку з'являються назви полів. Саме з цього списку потрібно обрати необхідні поля. Формула, за якою пройде обчислення така:

Середня оцінка=Сума оцінок по всім предметам/Кількість предметів

У верхньому вікні **Построителя выражений** ця формула набуде такого вигляду:

Выражение1:([ФК15-1]![ИМов]+[ФК15-1]![ВСТІ]+[ФК15-1]![Информатика]+
[ФК15-1]![БЖД]+[ФК15-1]![История]+[ФК15-1]![Ділова мова]+[ФК15-
1]![Документознавство])/7

Щоб не вводити назви полів вручну, достатньо двічі клацати на них в нижньому середньому списку і вони будуть з'являтися автоматично в верхній області введення.

Якщо при побудові виразу потрібно додати стандартну вбудовану функцію, потрібно в лівому нижньому списку вибрати папку **Функции/Встроенные функции**. В результаті в правому списку з'являться назви функцій.

Створивши вираз, в бланку запити можемо дати ім'я новому стовпчику. Для цього акуратно замість слова **Выражение1** вводимо власну назву (більш змістовну).

Після всіх виконаних попередніх дій потрібно перейти в режим таблиці і оглянути отримані результати. Якщо вони не задовольняють, треба повернутися в режим конструктора і виправити помилки.

В разі потреби можна задати порядок сортування даних. Для цього клацнути по рядку **Сортировка** в стовпчику того поля, по значенням якого потрібно відсортувати записи, і вибрати **По возрастанию** або **По убыванию**.

Підсумкові запити

Іноді цікаві не окремі записи таблиці, а підсумкові значіння по групам даних. Наприклад, нам потрібно знати середню оцінку з Фізики з попереднього прикладу, або сумарну кількість балів з Хімії. Відповіді на такі питання дає підсумковий запит. Для обчислення підсумкових значень потрібно натиснути кнопку **Групповые операции** на панелі інструментів конструктора запитів, щоб у бланку QBE з'явився рядок **Групповая операция**.

Коли натиснути кнопку **Групповые операции** на панелі інструменті, Access використовує установку **Группировка** в рядку **Групповая операция** для любого поля, який занесений в бланк запиту. Тепер записи по кожному полю групуються, але підсумок не підводиться. Для отримання підсумків потрібно замінити установку **Группировка** в рядку **Групповая операция** на конкретну підсумкову функцію.

Access дає дев'ять функцій, які забезпечують виконання групових операцій. Можна ввести ім'я функції з клавіатури в рядку **Групповая операция** бланка запиту або вибрати її в списку, що розкривається. Нижче перераховані підсумкові функції Access:

Sum Вираховує суму всіх значень заданого поля в кожній групі. Використовується тільки для числових та грошових полів.

Avg Вираховує середнє арифметичне всіх значень даного поля в кожній групі. Використовується тільки для числових та грошових полів.

Min Повертає найменше значіння, яке знайдене в цьому полі всередині кожної групи. Для числових полів повертається найменше значення. Для текстових полів - найменше з символічних значень незалежно від регістру.

Max Повертає найбільше значіння, яке знайдене в цьому полі всередині кожної групи. Для числових полів повертається найбільше значення. Для текстових полів - найбільше з символічних значень незалежно від регістру.

Count. Повертає число записів, в яких значення даного поля відмінні від Null (пусто).

StDev. Вираховує стандартне відхилення всіх значень даного поля в кожній групі. Ця функція використовується тільки для числових або грошових полів.

Var Вираховує дисперсію значень даного поля в кожній групі. Ця функція використовується тільки для числових або грошових полів.

First Повертає перше значення цього поля в групі.

Last Повертає останнє значення цього поля в групі.

Якщо не потрібно включати в групи підсумкового запиту деякі записи, можна додати в бланк запиту одне або декілька полів для фільтра. Для створення фільтра треба в рядку **Групповая операция** вибрати установку **Условие**, зняти прапорець **Вывод на экран** для цього поля і ввести умови відбору.

Перехресні запити

Access підтримує особливий тип підсумкових запитів, який називається перехресними запитами. Вони дозволяють побачити обчислювані значення в вигляді перехресної таблиці, яка нагадує електронну.

Для побудови перехресного запиту потрібно виділити ім'я потрібної таблиці в вікні бази даних і вибрати **Новый запрос** зі списку кнопки **Новый объект** на панелі інструментів. В вікні діалогу вибрати **Конструктор**, а потім у вікні конструктора вибрати команду **Запрос/Перекрестный**. Access додасть в бланк

запиту рядок **Перекрестная таблица**. В цьому рядку для кожного поля перехресного запиту може бути вибрана одна з чотирьох установок: **Заголовки строк**, **Заголовки столбцов**, **Значение** (яке виводиться в чарунках перехресної таблиці) і **Не отображается**. Для перехресного запиту потрібно визначити у крайньому випадку одне поле в якості заголовків рядків, одне для заголовків стовпчиків і одне поле значень. Кожне поле, яке є заголовком стовпчиків, повинно мати в рядку **Групповая операция** установку **Группировка**. Для поля, яке використовується в якості заголовків рядків, в рядку **Групповая операция** повинна бути встановлена операція **Группировка**, вибрана одна з підсумкових функцій або введений вираз, який містить підсумкову функцію. Для поля з установкою **Значение** вибрати одну з підсумкових функцій або ввести вираз, який використовує підсумкову функцію.

Як і в інших типах підсумкових запитів, для відбору даних, які включаються у набір записів, можна використати додаткові поля. Для них необхідно вибрати установку **Условие** в рядку **Групповая операция** і **Не отображается** в рядку **Перекрестная таблица**, а потім ввести умову відбору. Умову відбору можна також задати для любого поля, яке використовується в якості заголовків стовпчиків, а дані можна сортувати полюбим полям (з мовчазної згоди Access сортує заголовки стовпчиків за зростанням).

Можна визначити порядок стовпчиків в перехресній таблиці, використовуючи вікно властивостей запиту. Для цього треба клацнути в будь-якому місці верхньої частини (ззовні списків полів) вікна запиту в режимі конструктора і натиснути кнопку **Свойства** на панелі інструментів. На екрані з'явиться вікно властивостей запиту.

Щоб задати порядок виводу стовпчиків в перехресній таблиці, потрібно ввести заголовки так, як вони виглядають в відформатованому вигляді, і в тій послідовності, в якій користувач хоче їх побачити. Потрібно впевнитися, що включені всі заголовки стовпчиків, які з'являються в підсумковій таблиці. Якщо пропущений який-небудь заголовок стовпчика або неправильно введено його ім'я, то він не з'явиться в перехресній таблиці.

Обмеження на використання запитів на вибірку для поновлення даних

Набір записів, які створює Access при виконанні запитів, виглядає і веде себе майже як реальна таблиця, яка містить дані. Дійсно, в багатьох випадках можна вставляти рядки, видаляти їх і поновлювати дані в наборі записів. При цьому Access внесе необхідні зміни в відповідні базові таблиці запиту.

Але в деяких випадках Access не розуміє, які зміни необхідно виконати. Access не дозволяє виконати зміни, які діють на декілька рядків базової таблиці. По цій причині не можна змінити дані в підсумковому або перехресному запиті. Access не дозволяє змінити дані в полі, яке має установку **Avg** або **Sum**, тому що результат у цьому випадку визначається значеннями з багатьох записів.

При роботі з набором записів, які отримані в результаті об'єднання, Access дозволяє змінювати значення любых полів таблиці, яка знаходиться на боці відношення "багатьох", але не дозволяє поновлювати ключові поля таблиці з боку "один". Наприклад, один склад може мати багато товарів. В наборі записів, який є результатом об'єднання таблиць **Склади** і **Товари**, можна поновити любе поле, яке належить таблиці **Товари**, але не можна змінити значення полів, які створюють первинний ключ таблиці **Склади**. Access дозволяє змінити інші поля цієї таблиці.

Якщо не бути уважним, можливість поновлювати поля з боку “один” може призвести до небажаних результатів.

Створення перехресних запитів за допомогою майстра

В вікні бази даних перейти на вкладку **Запросы** і натиснути кнопку **Создать**.

В діалоговому вікні **Новый запрос** вибрати майстра **Перекрестный запрос**.

Натиснути кнопку **ОК**.

Користувач слідує інструкціям, які з’являються в діалогових вікнах майстра. В останньому діалоговому вікні користувачу пропонується вибір виконати запит або проглянути його структуру в режимі конструктора.

За допомогою майстра **сводных таблиц** дані перехресних таблиць можна вивести на екран, не створюючи окремого запиту. За допомогою зведеної таблиці заголовки рядків і стовпчиків можна змінювати таким чином, щоб різними способами можна було аналізувати дані.

Настроювання властивостей запиту

Microsoft Access дозволяє встановлювати ряд властивостей запиту, які змінюють хід його виконання. Щоб відкрити вікно властивостей запиту, потрібно клацнути любуому місці верхньої частини вікна запиту в режимі конструктора ззовні списку полів і натиснути кнопку **Свойства** на панелі інструментів. На рис.29 показано вікно властивостей запиту на вибірку.

Звичайно, користувач вибирає тільки деякі поля, які при виконанні запиту на вибірку необхідно включити в набір записів. Але якщо розробляється запит для використання в формі, і потрібно, щоб всі поля з усіх базових таблиць запиту були доступні в цій формі, встановлюється для властивості **Вывод всех полей** значення **Да**. В більшості випадків краще залишити для цієї властивості встановлене з мовчазної згоди значення **Нет**, а змінювати цю установку тільки для визначених запитів.

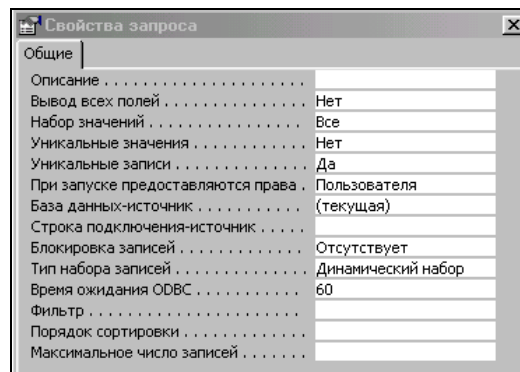


Рис. 29

В залежності від складності запиту Access може знадобитися декілька секунд (хвилин), щоб знайти всі рядки і почати вивід інформації на екран. Якщо користувача цікавлять тільки “перші” або “верхні” рядки набору записів, треба скористатися властивістю **Набор значений** для виводу інформації на екран, як тільки Access знайде перші *n* або перші % рядків. Якщо ввести ціле значення, Access виведе на екран результат відразу, щойно знайде задане число рядків. При введенні десяткового значення менше 1 Access почне вивід даних після того, як знайде приблизно такий відсоток рядків. Слід врахувати, що при виконанні сортування Access повинен спочатку відібрати всі записи, і тільки відсортувавши

їх, він може визначити перші n рядків. В цьому випадку властивість **Набор значений** не прискорить вивід інформації на екран.

При виконанні запиту часто виникають ситуації, під час яких Microsoft Access знаходить рядки, що повторюються. З мовчазної згоди він включає в результуючий набір тільки унікальні записи. Це означає, що ідентифікатор кожного рядка (первинний ключ таблиці для запиту з єдиною таблицею або сполучення первинних ключів для багатотабличного запиту) унікальний. Якщо користувач не просить Access повертати унікальні значення, в набір записів запиту будуть включені тільки відмінні один від одного рядки. Якщо є потреба бачити всі дані (враховуючи рядки-дублікати), необхідно встановити обидві властивості **Унікальные значения** і **Унікальные записи** - в значення **Нет**. Користувач не зможе змінювати поля в наборі записів запиту, якщо для властивості **Унікальные записи** встановлене значення **Нет**.

Якщо база даних буде використовуватися в мережі в режимі колективного доступу, то для забезпечення безпеки даних можна дати доступ іншим користувачам до таблиць тільки за допомогою запитів. Власник таблиці завжди має до неї повний доступ. Можна заборонити всім доступ до таблиць і дозволити проглядати деякі дані тільки визначеним користувачам. Ця умова буде працювати, якщо для властивості **При запуске предоставляются права** встановити значення **Пользователя**. Якщо для тих, хто використовує конкретний запит, потрібно дозволити “успадкувати права власника” на доступ до таблиць, треба встановити для властивості **При запуске предоставляются права** значення **Владельца**.

Якщо запит призначений для роботи в режимі колективного доступу, то для забезпечення цілісності даних при редагуванні використовується властивість **Блокировка записей**. Access скористається блокуванням тільки в тому випадку, коли в вихідну таблицю потрібно записати рядок. Щоб запис блокувався при внесенні до нього змін одним з користувачів, встановлюється для цієї властивості значення **Изменяемой записи**. Більш жорстка установка **Всех записей** блокує всі записи, відібрані запитом, до тих пір, доки запит відкритий одним з користувачів. Використовується ця установка тільки у тому випадку, коли при виконанні запиту робляться численні зміни в деякій таблиці, і інші користувачі не можуть мати доступу до даних цієї таблиці до тих пір, доки не завершиться виконання запиту.

Запити на зміну даних

Для пошуку окремого запису в базі даних і зміни в ній значення поля досить режиму таблиці. Але що робити, якщо потрібно провести однакові зміни в більшій кількості записів. Виконання заміни в кожному окремому запису недоцільно.

Перед тим, як приступити до створення запиту на поновлення записів в базі даних, потрібно створити запит на вибірку з умовою відбору, який дозволяє знайти всі записи, які підлягають поновленню. Після того, як Access вибери потрібні записи, користувачу не складно буде перетворити запит на вибірку в запит на поновлення або в інший тип запита на зміну.

Запит на поновлення

Допустимо, що запит на вибірку створений. Тепер можна змінити запит так, щоб він поновлював рядки таблиці. В режимі конструктора в меню **Запрос** існують команди для чотирьох типів запитів на зміну: **Создание таблицы**, **Обновление**,

Добавление, Удаление. Окрім цього, для перетворення типу запита можна скористатися списком кнопки **Тип запроса** на панелі інструментів. Треба вибрати пункт **Обновление**, щоб перетворити запит на вибірку в запит на поновлення.

При подібному перетворенні запита Access змінює заголовок його вікна і додає рядок **Обновление** в бланк QBE. Він використовується для введення нових значень (або виразів), які повинні замінити в відібраних записах існуючі дані.

Якщо потрібно повністю убезпечити себе від любых випадковостей, то перед виконанням запиту на поновлення робиться резервна копія таблиці. Для цього перейти в вікно бази даних, вибрати таблицю, яка буде поновлюватися, і скористатися командою **Правка/Копировать**. Потім вибрати команду **Правка/Вставить** і, коли Access виведе на екран відповідне вікно діалогу, надасть копії таблиці інше ім'я.

Щоб виконати запит на поновлення, треба вибрати команду **Запрос/Запуск** або натиснути кнопку **Запуск** на панелі інструментів. Access продивиться таблицю і визначить кількість рядків, які поновлюються, а потім виводить на екран вікно діалогу. Щоб провести поновлення даних, потрібно натиснути кнопку **Да** в вікні діалогу. Якщо користувач бачить, що число поновлюваних рядків в вікні діалогу не відповідає очікуваній кількості або не впевнений, що Access поновить потрібні записи або поля, натиснути кнопку **Нет**, щоб відмовитися від виконання запиту і поновлення даних. Після виконання запиту можна перевірити зміст таблиці, щоб впевнитися, що зміна даних проведена правильно.

Щоб повторити запит на зміну, потрібно вибрати його в вікні бази даних і натиснути кнопку **Открыть**. Після запуску запита на зміну з вікна бази даних Access виводить на екран вікно діалогу, яке запитує підтвердження для його виконання. В вікні підтвердження натиснути кнопку **Да**, щоб завершити поновлення даних. Якщо треба відмінити вивід на екран вікна підтвердження, вибирається команда **Вид/Настройка** і на вкладці **Правка/поиск** вікна діалогу **Параметры** в секції **Подтверждение** знімається прапорець **Запросов на изменение**.

За допомогою запита на поновлення можна за один раз змінити значення декількох полів, включивши їх у бланк запиту і визначивши вирази, які будуть використовуватися для поновлення цих полів. При цьому для обчислення нового значення деякого поля можна використати значіння інших полів.

Перед поновленням запису в базовій таблиці або запиті Access робить копію вихідного запису. Він використовує задані вирази до вихідного запису і вносить результат в копію. Потім він змінює зміст бази даних, переписуючи поновлену копію в таблицю.

Створення нової таблиці

Іноді потрібно зберігати в новій таблиці дані, які отримані за допомогою запита на вибірку. Створення нової таблиці прискорить роботу додатка, особливо якщо при виконанні запита об'єднуються декілька таблиць.

Збереження набору записів запита як таблиці корисно і в тому випадку, коли нашаровується підсумкова інформація і її треба довго зберігати після видалення з бази вихідних даних, на яких базувався запит.

Припустимо, існує запит на вибірку. Щоб перетворити запит на вибірку в запит на створення таблиці, треба вибрати команду **Запрос/Создание** таблиці.

Access виведе на екран вікно діалогу **Создание таблицы**. Користувач вводить ім'я для підсумкової таблиці і тисне кнопку **ОК**, щоб закрити вікно діалогу.

Закінчивши з установками в запиті на створення таблиці, можна виконати його за допомогою команди **Запрос/Запуск** або кнопки **Запуск** на панелі інструментів. Access створює записи, розміщає їх в нову таблицю і інформує користувача про кількість рядків, які вставляються в таблицю, на екран виводиться вікно діалогу для підтвердження створення таблиці. Натиснути кнопку **Да**, щоб створити нову таблицю і помістити до неї записи. Після цього потрібно переключитися в вікно бази даних і клацнути по вкладці **Таблицы**, щоб відкрити список таблиць. Вибирається в ньому тільки що створену таблицю і відкривається вона в режимі таблиці. Можна оглянути отримані дані. В режимі конструктора можна змінити імена полів або формати.

Запит на додавання

За допомогою запиту на додавання можна скопіювати вибрані записи і вставити їх в іншу таблицю. Окрім того, запит на додавання можна використати для перенесення даних з іншого джерела в свою базу даних, щоб потім відредагувати отримані дані і вставити їх в існуючу таблицю.

Запит на додавання, подібно запиту на створення таблиці, дозволяє зібрати обчислені підсумкові дані і зберегти їх в визначеній таблиці. Одна з переваг запиту на додавання в тому, що є можливість повністю визначити поля і задати їх властивості перед вставленням даних в кінцеву таблицю. Недоліком цього типу запитів є велика ймовірність помилок, оскільки дані, які додаються, можуть не відповідати типам полів кінцевої таблиці.

Перед виконанням запиту на додавання можна запустити його як запит на вибірку (як і в випадку інших запитів на зміну), щоб впевнитися в тому, що копіюються потрібні записи. Можна взагалі почати з побудови запита на вибірку, виконати його і тільки після цього перетворити його в запит на додавання. Також можна відразу створити запит на додавання і перевірити дані, що додаються, переключившись з режиму конструктора в режим таблиці.

Після перевірки правильності додавання записів можна виконати запит прямо з вікна конструктора або зберегти і потім запустити його з вікна бази даних.

Цікавим прикладом використання запитів є добування застарілих даних з робочої таблиці і копіювання їх в архівну таблицю. З часом в основній таблиці можуть накопиться тисячі записів - контракти або замовлення, які вводяться на протязі ведення справи. Скоріше за все дані річної давнини для поточної роботи на потрібні. Періодично прибираючи в архівні таблиці "старі" дані, збільшується швидкодія основних частин додатку.

Помилки при виконанні запитів на зміну

Перед внесенням відповідних змін в базу даних Microsoft Access аналізує запит і дані, які змінюються при його виконанні. При наявності помилок Access завжди дає користувачу можливість відмінити виконання запиту на зміну.

Під час виконання запита на зміну Access розрізняє чотири категорії помилок:

Дублікати первинного ключа. Ця категорія помилок виникає при спробі додавання або зміни деякого запису в таблиці, яка приводить до появи дублюючого значення первинного ключа або унікального індексу. Access не буде поновлювати

або додавати записи, які створюють такі значіння. Щоб запобігти конфлікту перед спробі додати такі записи користувач потрібен змінити значення первинного ключа в вихідній таблиці.

Помилки перетворення даних. Помилки цієї категорії виникають в тих випадках, коли вставляються дані в існуючу таблицю і при цьому видно, що тип даних полів-отримувачів не співпадає з типом даних полів-джерел. Наприклад, помилка виникає, якщо додаються текстові значіння до поля, що містить цілочисельні дані, а текстове поле містить літерні символи або надто довгий рядок цифрових символів. В запиті на поновлення помилка перетворення може виникнути, якщо використовується формула, яка пробує провести обчислення над полем, яке містить букви.

Блокування запису. Помилки цієї категорії виникають, коли при виконанні запиту на видалення або на поновлення використовується таблиця, до якої мають доступ інші користувачі мережі. Access не може провести поновлення записів, які в цей час поновлюються іншими користувачами. Необхідно зачекати деякий час і спробувати знову виконати поновлення або видалення в той момент, коли дані записи більше ніхто не використовує.

Порушення умов на значення. Якщо записи, що вставляються або поновлюються, не задовольняють умові на значення для деякого поля або для таблиці, Access повідомляє про помилку і не проводить вставку або поновлення подібних записів.

Видалення групи записів

Видалити групу записів з бази даних можна за допомогою запита на видалення. Після підведення підсумків і копіювання старих записів (наприклад, контрактів) в архівну таблицю, ці записи краще видалити з основної таблиці. Очевидно, що подібна операція буде виконуватися не однократно. Тому можна створити запит для автоматичного відбору на основі поточної дати записів, які видаляються. З іншого боку, в запит можна включити параметр, щоб користувач міг вказати потрібну дату після запуску запита на видалення. В будь-якому випадку користувачу не прийдеється змінювати визначення запиту перед кожним його запуском.

Як і у випадку запита на поновлення, корисно спочатку в'яснити, які саме записи будуть видалені запитом. Тому починають з запита на вибірку, який відбирає їх. Якщо запит на вибірку вибрав ті самі записи, що були скопійовані в архівну таблицю за допомогою запита на додавання, то можна повернутися в режим конструктора, щоб запустити запит і видалити ці рядки з основної таблиці.

У зв'язку з тим, що не можна продивитися видалені записи, розумно зробити резервну копію таблиці, особливо у тому випадку, якщо запит на видалення виконується уперше. Для створення копії таблиці виконується процедура, яка описана в розділі “Виконання запиту на поновлення”.

Запит на вибірку, відкритий у режимі конструктора, можна перетворити в запит на видалення, виконавши команду Запрос/Запуск. Вносити в запит які-небудь додаткові зміни не потрібно. Щоб Microsoft Access видалив записи, потрібно вибрати команду Запрос/Запуск або натиснути кнопку Запуск на панелі інструментів. Якщо в запит включений параметр, потрібно ввести його значення в вікно діалогу Введите значение параметра. Access відбере записи, які підлягають видаленню, і виведе на екран вікно діалогу для підтвердження видалення. Для завершення процесу видалення натиснути кнопку **Да**.

РОЗДІЛ 10

ОСНОВИ ОФІСНОГО ПРОГРАМУВАННЯ

Ринок мов програмування наповнено програмними середовищами та мовами, серед яких – C, C++, Visual C++, C# , Assembler, Delphi, Pascal, Object-Pascal, Basic, Visual Basic for Application, Java, JavaScript, HTML, XML, XSL, CSS, PHP, MySQL, Perl та багато інших. Їх вивчення та ознайомлення з можливостями прямо стосується як самої сфери розробки програм (програмування), так і сфери використання інформаційних систем та технологій. Сучасні системи електронного документообігу, принципи їх побудови, функціональні модулі, програмне забезпечення офісної діяльності та задачі офісного програмування нерозривно пов'язані з професійною діяльністю фахівців, зайнятих у цій сфері. Мови програмування представлені як створеними для роботи з ними програмними середовищами (наприклад, мова Pascal у програмному середовищі TurboPascal), так і складовими частинами програмних виробів спеціалізованого призначення (мова HTML у текстових редакторах; мова SQL у базах даних тощо). Задача фахівця – оволодіти цими засобами незалежно від їх конкретного представництва.

Поняття про розробку програм

Перший міф працівника інформаційної сфери – програміста-початківця чи співробітника певної фірми – полягає в думці, що роботою програміста є створення програм, а роботи звичайного працівника знання мов програмування не стосується. Цей міф з того ж ряду, що й міф про яблуко як випадкове явище, яке надихало Ньютона створити теорію тяжіння (немов би інакше теорія тяжіння не була б створена) і ще багато різних міфів про те, що можна чекати, чекати і дочекатись до акту творіння (чи, більш частіше, чекати, але не дочекатися його). У дійсності всі результати людської діяльності виникали і виникають як продукт, хай навіть і несвідомого, але використання певної технології до розв'язування конкретної задачі. Процес застосування технології найчастіше називають не створенням, а розробкою: програми, якщо готового для роботи з інформацією програмного виробу не існує; певних кінцевих результатів обробки інформації, якщо використовується уже розроблена програма.

Задачі, що можуть розв'язуватись з використанням інформаційних технологій, всі без винятку полягають в тому, що якийсь набір вхідних даних за допомогою комп'ютера перетворюється на відповідний набір вихідних даних. Така постановка задачі підкреслює три складові кожної задачі, незалежно від програмування як такого, - конкретний набір вхідних даних, конкретний набір вихідних даних та конкретний комп'ютер. У кожному можливому варіанті йдеться про програмування як процес побудови послідовності дій, який здатен перевести набір вхідних даних до кінцевого результату. Але в одному випадку програмування зводиться до використання можливостей уже створеної програми (користувацький варіант роботи), в іншому – до розробки програми, яка б надала такі можливості (програмістський варіант роботи).

З позиції офісного програмування є сенс оговорити обидва варіанти роботи. Почнемо з варіанту, коли розглядається робота програміста.

Перше правило програмування полягає в тому, що програма потрібна, якщо в переліку можливих операцій комп'ютера відсутня така, що може безпосередньо виконати потрібні дії.

Серед можливих операцій комп'ютера можуть бути (і гарантовано є) такі, про які знає тільки одна людина - саме той, хто розробив відповідну програму. Якби існував повний перелік всіх програм, що вже створені людством, більшість програмістів залишилася б без роботи. На щастя, сама тільки задача впорядкування вже розроблених програм потребує більшої обчислювальної потужності, ніж її мають всі комп'ютери світу. Зростання обчислювальної потужності тягне за собою зростання кількості вже розроблених програм і повне впорядкування переліку програм стає ще складнішою задачею. Саме тому досить часто одну й ту ж саму задачу роблять багато різних програм.

Другий міф програміста-початківця полягає в думці, що він може чи створити, чи хай вже розробити, але найкращу програму.

Теорія програмування стверджує, що для кожної задачі з перетворення вхідних даних на результати існує можливість розкладення на послідовність елементарних дій, а ця послідовність вже може бути здійснена процесором, якщо в ньому передбачена можливість здійснювати ці елементарні дії. Добре відомо, що навіть в тому випадку, коли результатом роботи програми може бути тільки одна з двох відповідей "Так" або "Ні", існує принаймні два варіанти розкладення програми на елементарні дії (нормальна діз'юнктна та кон'юнктна форми булівської функції). Кожному з варіантів відповідає своя програма, яка буде давати той же результат, що і всі інші. Якщо різних варіантів результату роботи програми більше двох, варіантів розкладення програми теж більше, а всі вони дають той же результат. Всілякі спроби доповнити оцінку програми за результатом ще й оцінкою за якимись якостями (точніше, кругліше, вибагливіше тощо) тільки збільшують множину можливих результатів і відповідно збільшують кількість варіантів реалізації програми.

Правило друге: гарно працює та програма, яка працює. На жаль, та ж сама теорія програмування наполягає, що переконатись в працездатності програми можна лише тоді, коли вона буде виконана для кожного можливого набору вхідних даних. Саме тому найсуттєвішу, принаймні за витратами часу, частину процесу створення програми складає перевірка її працездатності, її відповідності поставленій задачі.

Поняття про етапи роботи над програмою

В процесі створення програми виділяються чотири принципово різних етапи:

- розробка алгоритму (мається на увазі формулювання в довільній формі послідовності дій, що дозволяє отримати потрібний результат);
- переклад алгоритму на конкретну мову програмування та на виконувану програму (для початківців останнє є окремою проблемою);
- тестування програми (є проблемою не тільки для початківців);
- локалізація помилки та її виправлення.

Останні два етапи насправді є рекурсивними, оскільки кожна знайдена помилка в роботі програми потребує виправлення або на етапі перекладу алгоритму на мову програмування або навіть в самому алгоритмі. Разом з тим етап тестування є найдовшим в часі, оскільки тестування може вважатись закінченим тільки тоді, коли перевірено правильність роботи програми для всіх можливих значень її вхідних даних.

Розподіл процесу створення програми на побудову алгоритму та перекладення алгоритму на конкретну мову програмування є дещо штучним,

оскільки кваліфікований програміст найчастіше і сам алгоритм створює безпосередньо в термінах мови програмування.

Задача побудови програми сама теж може тлумачитись, як своєрідна задача програмування, оскільки має кінцевим результатом є набір кодів процесора, вхідними даними - постановка задачі. Єдине, що відрізняє задачу побудови програми - несповна відомими залишаються елементарні операції процесора, що виконує цю задачу. Разом з тим до неї застосовні принаймні перші два правила програмування. Перше переформулюється в такий спосіб, що навчити програмувати можна тільки тих, хто хоче навчитись. Друге твердить, що нема сенсу вчити програмуванню людину, яка вже розроблює потрібні їй програми. Залишається тільки винайти спосіб виокремлювати саме тих, кому доцільно навчатись програмувати і тоді не залишиться жодної проблеми.

Маючи весь час на увазі кінцеву мету роботи програми – отримання потрібних результатів з того, що було заданим, процес розробки програми можна розглядати, як процес перетворення даних. Тоді сама програма повинна поступово, крок за кроком, наближати множину вхідних даних до множини результатів. В процесі такого наближення виникають проміжні дані, що стають вхідними даними для наступної, трохи простішої задачі (якщо проміжні дані справді хоч трохи, а ближче до потрібного результату). Найсуттєвішим в цьому процесі поступового наближення є те, що на першому етапі зовсім не обов'язково мати повний опис всієї структури проміжних даних, вона може визначатись поступово, як деталізація окремих кроків. Такий процес розподілу програми на окремі частини з наступною деталізацією частин називають технологією "згори донизу". Мається на увазі, що спочатку задача розбирається "з великої відстані", коли на деталі увага не звертається. Потім відстань зменшується, кожна з підзадач бачиться окремою задачею і так само розбивається на кроки. Продовження деталізації врешті-решт зупиниться, коли наступний крок можна буде зробити одним оператором мови програмування або безпосередньо командою процесора.

В технології "згори донизу" найсуттєвішої уваги вимагає структура проміжних даних. Більш точно, потрібно ретельно контролювати, аби на кожному кроці проміжні дані, що виникають, ставали ближчими до потрібних результатів.

Оскільки програмування - є скоріше мистецтвом, ніж виробництвом, навчатись програмуванню можна тільки за допомогою детального розгляду прикладів конкретної роботи з розробки конкретних програм.

Поняття про відлагодження програми

Програма завжди виконує тільки ті дії, що їх заклав до неї програміст. На відміну від комп'ютера, програмістові властиво помилятися. Інколи, якщо помилки відсутні, або вони компенсуються (тих, у кого таке трапляється, звать програмістами від бога) і в такому разі програма виконує саме ті дії, які від неї очікувались, але найчастіше тільки-но підготовлена програма містить певну кількість помилок. Ці помилки потрібно знаходити та виправляти і в такий спосіб налагоджувати роботу програми, аби вона врешті-решт стала здатною виробляти саме ті результати, яких від неї очікують. Процес пошуку та виправлення помилок і називають відлагодженням.

Помилки, що можуть зустрічатись в тексті програми, поділяють на *синтаксичні* та *логічні*.

До *синтаксичних* відносять помилки в перекладі алгоритму обчислень на конкретну мову програмування, такі, як відсутність роздільника між операторами або непарність дужок. Ці помилки можуть бути впізнані ще під час трансляції тексту програми, якщо мова програмування побудована в такий спосіб, аби не припускати різних тлумачень.

До *логічних* відносять, в першу чергу, помилки в алгоритмі, що використовується в програмі. Але розподіл процесу створення програми на побудову алгоритму та перекладення алгоритму на конкретну мову програмування є дещо штучним, оскільки кваліфікований програміст найчастіше і сам алгоритм створює безпосередньо в термінах мови програмування.

До логічних досить часто відносять також помилки, що їх не може впізнати транслятор конкретної мови. Найтипівшим прикладом такої помилки є така помилка в позначенні циклу мовою Fortran: замість правильного циклу DO I=1,2 з комою, що відділяє початкове та кінцеве значення параметру циклу, заноситься DO I=1.2 що є привласненням дійсного значення 1.2. Оскільки Fortran підтримує неявне оголошення та ще й дозволяє використовувати проміжок в ідентифікаторі, транслятор витлумачує цей запис, саме як оголошення дійсної змінної DO I разом із привласненням і не повідомляє про помилку.

Звичайно пошук помилок не закінчується навіть після того, як програма чи навіть програмний комплекс набуває широкого розповсюдження. Всі досить великі програмні комплекси містять помилки в більшій або меншій кількості, тому для всіх комплексів загального використання створюється відповідна служба підтримки, що намагається більш або менш оперативно реагувати на повідомлення користувачів про виявлені помилки чи просто про некоректність роботи. В першу чергу це стосується найвідоміших програмних комплексів операційних систем, систем підготовки та обробки інформації.

Наприклад, Microsoft регулярно випускає пакети підтримки Service Pack для своїх операційних систем та інших програм, а ще до випуску чергового пакету оновлення розповсюджує пакети виправлення помилок в окремих програмах.

Некомерційні системи, як FreeBSD, Linux також регулярно розповсюджують засоби виправлення помилок patch, що дозволяють виправляти помилки не тільки тому користувачеві, хто знайшов конкретну помилку, а й усім зацікавленим в надійності роботи системи.

Власне відлагодження відрізняється від супроводу програми тільки тим, хто саме здійснює тестування – автор програми чи сторонній користувач. На етапі відлагодження використовуються не всі можливі значення вхідних даних, а тільки обмежений набір, а саме такі дані, для яких можна досить просто передбачити результат роботи програми.

На етапі супроводу від користувачів надходять повідомлення про те, що для якогось практично потрібного набору вхідних даних результат роботи програми не відповідає очікуваному, але це може бути наслідком не тільки помилок в програмі, а й помилок в очікуваннях конкретного користувача.

Поняття про тестування програм

Власне відлагодження поділяється на два етапи – тестування та пошук помилки. Другий етап потрібний тільки в тому разі, коли перший призвів до результату, що відрізняється від очікуваного. Перший, тобто власне тестування,

бажано здійснювати в такий спосіб, щоб якомога більше отримати інформації про можливу локалізацію помилки.

Тестування відбувається шляхом порівняння результату роботи програми з очікуваним результатом. Просто виконання програми без такого порівняння не може дати оцінки працездатності програми. Відповідно для обчислювальних задач тестування повинне спиратись на результати, що попередньо розраховані без використання цієї програми. Для таких розрахунків доцільно використовувати найпростіші з розрахункової точки зору вхідні дані, навіть такі, що дають тривіальний результат.

Головною проблемою в тестуванні є відсутність повноцінної теорії. Теорія алгоритмів стверджує, що впевненість в відсутності помилок може виникнути тільки тоді, коли програма буде перевірена для всіх можливих значень вхідних даних. Така перевірка потребує нехай скінченого, але нереального часу на виконання програми і, що найголовніше, на аналіз результатів.

Звичайно в множині можливих значень вхідних даних виділяються певні області, щодо яких можна вважати, що програма, яка правильно працює для одного значення, буде так само правильно працювати і для всіх інших з цієї області. Після цього в кожній області обирається одне значення (типовий представник), для якого результат можна отримати без використання програми (це й буде очікуваний результат) і саме для нього здійснюється порівняння результату роботи програми з очікуваним.

Розподіл множини можливих значень на області є непростою задачею, що звичайно вимагає глибоких уявлень про розв'язувану задачу. Тому ця проблема повинна вирішуватись не програмістом, а спеціалістом з прикладної області, та ще й таким, що добре розуміється на проблемах програмування. Саме тому тестування, а з ним і програмування взагалі, вважається скоріше мистецтвом, ніж наукою.

Локалізація помилок

Процес локалізації помилки є насправді процесом аналізу тексту програми і в тій його частині, що стосується синтаксису програми, може бути повністю виконаний транслятором, якщо тільки мова програмування досить добре побудована і не дозволяє подвійного тлумачення виразів. З цієї точки зору найдосконалішими мовами є Pascal та Java, хоча вони й не посідають першого місця серед професіональних програмістів.

До синтаксичних відносять помилки, що стосуються невідповідності запису дії мовою програмування очікуваному змісту цієї дії. Наприклад, синтаксичною помилкою буде запис послідовності двох операторів без розділювача. Таким розділювачем може бути символ ";" в більшості мов, або символ переходу на наступний рядок (і саме на сьому його позицію) в Fortran.

Якісні транслятори завжди дають деякий коментар до помилки, в більшості випадків цей коментар здатний допомогти в тлумаченні помилки.

Форма, що її має цей коментар, залежить не тільки від транслятора, а навіть і від режиму його роботи. Для комп'ютерів з малим об'ємом пам'яті властивим є просто позначати помилку номером, а коментар подавати окремо, часто навіть просто в посібнику до транслятора. Більш ефективним є друкування тексту повідомлення про помилку, що в ньому наводиться думка авторів транслятора щодо причин виникнення помилки.

Насправді, слід мати на увазі, що повідомлення про помилку виникає в тому разі, якщо транслятор виявився неспроможним витлумачити текст програми. Це може бути не тільки тому, що в програмі є помилка, а й тому, що транслятору не вистачає інтелекту. Мабуть, найінтелектуальнішим з усіх трансляторів є «батько» всіх трансляторів високого рівню Fortran. Він здатний шукати тлумачення навіть в тому разі, коли його просто забули повідомити про те, що в програмі буде використовуватись змінна і їй треба виділити місце в пам'яті.

Неспроможність до подвійного тлумачення та до тверджень за згодкою (замовчувань) є однією з визначальних рис сучасних мов програмування. Така неспроможність дозволяє транслятору локалізувати найпоширеніші людські помилки зокрема, помилки друку, коли замість однієї літери друкується інша, або не друкується жодної.

Поради початківцю

Швидкодія сучасних комп'ютерів дозволяє регулярно перевіряти синтаксис набраної частини тексту програми, навіть якщо цей текст ще не закінчено. Залежно від здатності сприймати спокійно чи нервуючи повідомлення про помилку, можна використовувати різні способи регулярної перевірки правильності тексту.

Найпростіше перевіряти текст, запускаючи трансляцію після кожного набраного оператора. Тоді транслятор обов'язково буде видавати повідомлення про помилку, поки текст не набраний до кінця і потрібно уважно аналізувати, яким саме було це повідомлення, чи стосувалось воно змісту набраного рядка чи просто є реакцією транслятора на незавершеність програми.

Набагато спокійніше набирати програму, одразу заготувавши можливість правильної її трансляції. Початком такого набору буде просто програма з чотирьох елементів `begin end`. Додаючи в цю програму рядок за рядком потрібні оператори, можна очікувати, що за відсутності синтаксичної помилки транслятор буде кожного разу нормально сприймати текст програми, навіть якщо вона ще й не здатна виробляти правильні дії.

Взагалі, перед розробкою програм слід ознайомитися зі змістом понять, які стосуються цієї сфери професійної діяльності. Частина з них розкрито нижче.

Оператор привласнення

Найважливішим оператором кожної мови програмування є оператор привласнення, який заносить результати обробки даних, передбачених правою частиною цього оператора, в комірку оперативної пам'яті, якій відповідає змінна, описана в його лівій частині. З огляду на існування різних типів даних оператор привласнення може бути застосований тільки в тому разі, коли тип значення, виробленого правою частиною, співпадає з типом змінної в лівій частині. Наприклад, якщо в правій частині відбувається додавання двох цілих чисел, то результат можна привласнити цілій змінній, тобто для змінної `Res:Integer` привласнення `Res:=1+1` є цілком припустимою дією, але якщо в правій частині виробляється дійсне значення (`Res:=1.0+1`), виникає протиріччя між типом результату та типом змінної. Це протиріччя може стосуватися тільки "довжини" виділеної пам'яті (наприклад, сума вимагатиме для зображення 4 байтів, а під тип `Integer` найчастіше виділяється тільки два), або навіть тільки в тлумаченні змінної (рядок із двох символів – `аб` – може бути зображений двома байтами, але не буде тлумачитись, як ціле число).

В різних мовах програмування припускаються ті або інші порушення вимоги співпадання типів. Всі мови припускають привласнення цілого значення дійсній

змінній, автоматично перетворюючи ціле число на дійсне, але навіть така стандартна дія вимагає певної обережності. Так, для змінної `res:Real` привласнення `res:=7+3` призведе до очікуваного значення 10, але привласнення `res:=7/3`; може призвести до значення 2.33333333, а може і до просто 2 залежно від того, що відбувається раніше – перетворення типів, чи ділення. Мова Pascal відрізняється найсуворішими вимогами: перетворення типів відбувається тільки з цілого на дійсне, всі інші варіанти є неприпустимими. В інших мовах контроль за перетворенням набагато слабкіший, але практично завжди працює правило, що в комірку, на яку вказує змінна, повинно бути достатньо місця для розташування отриманого значення.

Оператор привласнення не порівнює величини, як було б звично тлумачити стандартний для C++ запис $a=b$, а саме заносить нове значення в комірку пам'яті, що має назву "а". Попереднє значення, внаслідок привласнення зникає, тому має сенс, наприклад, вираз $a:=a+1$, в якому з комірки "а" вибирається значення, яке там зберігалось до виконання оператора привласнення, а після виконання там вже знаходиться збільшене значення. Саме оператор привласнення найяскравіше відбиває послідовний характер програми.

В англійській мові для виконання програми передбачене дієслово "run" ("бігти"), тобто програма "пробігає" (а інколи "проповзає") свої оператори і цей рух є незворотнім. Оператор привласнення транслюється в операцію пересилання правої частини (розрахованого значення) в комірку пам'яті, що позначається змінною в лівій частині.

Виклики підпрограм

Наступна за важливістю дія – виклик *підпрограм*. *Підпрограми*, так само як і змінні, мають кожна своє власне ім'я, то ж виклик підпрограми відбувається за назвою. Після назви в дужках мають бути розташовані аргументи підпрограми. Якщо аргументів багато, вони відділяються комами. Прикладом виклику може бути підпрограма друку. Наприклад, `Write(hello)` отримує один аргумент – рядок символів "hello" і передає на пристрій друку (найчастіше на екран) коди отриманих символів, а фрагмент `value:= World! ; Write(hello, value)` передає на пристрій друку спочатку коди символів рядка "hello" (перший аргумент) а потім коди символів, записані в комірці пам'яті, що відведена під змінну "value" (другий аргумент). Підпрограми друку теж припускають автоматичне перетворення типів. Так, `Write(res)` перетворить значення, що зберігаються в цілій змінній `res` в послідовність символів і тільки після цього видасть на пристрій друку отриману послідовність.

Послідовність операторів

Найпростіший спосіб запрограмувати декілька обчислень – перерахувати їх послідовно одне за одним. В такій послідовності оператори потрібно відокремлювати один від одного, це здійснюється за допомогою спеціального символу ";" – крапка з комою. Звичайною практикою є ставити крапку з комою наприкінці кожного оператора, хоча в прикладі "begin Write(aga) end." крапка з комою взагалі не потрібна. Послідовність операторів під час трансляції перетворюється в послідовність команд центрального процесора.

Складаний оператор

Часто виникає потреба використання послідовності операторів там, де за синтаксисом може бути присутнім тільки один. На цей випадок в усіх мовах програмування передбачені своєрідні операторні дужки. В мові Pascal такими

дужками є пара ключових слів “begin” та “end”, мова C++ ставиться до дужок економніше і використовує пару фігурних дужок “{” та “}”, Maple інвертує ключове слово – “do” – “od”, “if” – “fi”.

Умовний оператор

Досить часто програма повинна виробити одну або іншу дію залежно від певної умови. Відповідний оператор так і зветь *умовним*. Він має синтаксис

if < condition> then <Then Part> else <Else Part>.

Тут <Then Part> та <Else Part> обидва є виконуваними операторами, але в програмі виконується тільки один з цих операторів, а саме <Then Part>, якщо умова здійснюється, або <Else Part>, якщо не здійснюється. Для умовного оператора часто використовують графічне зображення у вигляді блок-схеми, з якої можна легко побачити, що умовний оператор саме розділяє програму на дві гілки. Частина умовного оператора < condition> є логічним виразом, що набуває одного з двох можливих значень, а <Then Part> та <Else Part> є саме операторами, а не послідовностями операторів. Якщо виникає необхідність в одній з гілок чи в обох виконати декілька дій, використовується складаний оператор. Частина <Else Part> може бути відсутньою, тоді мається на увазі, що ця частина складається з порожнього оператора, тобто якщо умова здійснюється, виконується <Then Part> а потім наступний оператор, якщо ж ні – одразу наступний. Слід пам'ятати, що всі три частини умовного оператора є одним оператором і між ними не повинен бути присутнім роздільник операторів. Умовному оператору відповідає команда центрального процесора (точніше, набір команд), що в ній відбувається порівняння і залежно від його результату здійснюється або ні перехід не до наступної команди, а одразу на декілька команд. В асемблері ці команди мають назву “jump” – стрибок. Безпосередньо в команду процесора перетворюється тільки редукований умовний оператор, а повний має трохи складнішу реалізацію.

Розгалуження

Для тих досить частих випадків, коли умова може приймати не два, а декілька різних значень, пристосований оператор розгалуження трохи складніший за синтаксисом. Цей оператор поділяє програму не на дві, а на декілька гілок. Для кожної гілки має бути означене своє значення селектора. Найважливішою вимогою до оператора розгалуження є необхідність в передбаченні поведінки програми для всіх можливих значень селектора (інакше програма не знатиме, що робити).

Розгалуження має такий синтаксис:

Case selector of

Case1: Branch1;

Case2: Branch2;

...

CaseN: BranchN;

Else Default_Branch

End

Можна побачити, що в операторі розгалуження одна за одною перераховуються всі можливі гілки, а перед кожною з них стоїть одне, або декілька значень (констант вибору), які може мати селектор, аби виконувалась відповідна дія. Гілка Else забезпечує працездатність програми для тих випадків, коли значення

селектора не співпадає з жодною константою вибору. Змінна - “selector”, як і константи вибору повинні належати до одного з впорядкованих типів: цілих, символьних, перерахованих або діапазонних. Якщо вони мають логічний тип, оператор розгалуження не відрізняється від умовного. Оператор розгалуження транслюється в послідовність умовних команд, а кожна гілка закінчується оператором переходу на кінець розгалуження, то ж виконується тільки одна з гілок, саме та, для якої значення селектора співпало з константою вибору.

Оператор розгалуження є базовою частиною виконавчої операційної системи. В цій системі кожне поле вікна програми отримує свій код, що він передається програмі, коли відбуваються події над цим полем (клик, подвійний клик, правий клик чи просто курсор прийшов на поле чи покинув його). Отримавши цей код, програма здійснює відповідну гілку - підпрограму відгуку на подію. В розрахункових програмах оператор розгалуження використовується рідше, коли варіантів розрахунків більше ніж три.

Цикли

Однією з найсуттєвіших рис програмування є можливість поєднувати подібні дії в один блок, в якому за одним алгоритмом, крок за кроком, виконуються всі потрібні обчислення. Такий блок має назву *цикл* і для його організації в кожній мові програмування передбачено різні засоби. Припустимо, потрібно порівняти два слова, що складаються з декількох літер кожне. Припустимо, далі, що перше слово занесене в пам'ять у вигляді послідовності літер А, в такий спосіб, що значення першої літери слова можна отримати за адресою А[1], другої А[2] і так далі. Подібно до нього літери другого слова знаходяться за адресами В[1], В[2] . . . Порівняння слів можна здійснити, порівнюючи одну за одною літери, тобто здійснюючи послідовно

A [1] B [1]

A [2] B [2]

A [3] B [3]

В усіх цих діях присутня спільна частина порівняння подібних літер. Якщо переписати послідовність дій у вигляді *A [k] B [k]* ; то легко побачити цикл. Для організації циклів кожна мова програмування має відповідні оператори, але ці оператори можуть бути зовсім різними в різних мовах і тому доцільно спочатку розглянути найпростішу версію без використання спеціалізованих операторів. Такий цикл складається з типових частин: ініціалізація змінних, виконання потрібних дій, перевірки умови завершення.

k<=1;

loop: res<=A[k] B[k];

if (res and (k<kmax)) k<=k+1;goto loop;

В першому рядку готується спеціальна змінна параметр циклу *k*. Він використовується , в першу чергу, для індексації чергової літери слів. В другому рядку власне і здійснюється порівняння чергової пари літер *res*. Додатково до цього рядка додано помітку, яка далі використовується оператором переходу. Умовний оператор наступного рядка перевіряє, чи не втратило сенс повторення дій. Адже якщо попередня перевірка призвела до негативного результату, далі порівнювати нема потреби. Порівняння *k* з *kmax* потребує попереднього визначення довжини слів. Ми не будемо на цьому зупинятись, більш важливо, що поки умова

продовження виконується, потрібно повернутися знов до порівняння, попередньо збільшивши на одиничку параметр циклу. Повернення здійснює оператор `goto loop`. Загальна структура циклу має дві частини: тіло циклу `S`, тобто послідовність дій, що їх треба виконати під час кожного проходження циклу, та умову циклу з умовний виразом, значення якого визначає, продовжувати виконання циклу, чи закінчити. В різноманітних мовах програмування послідовність дій, потрібних для організації циклу, реалізується спеціальною операторною структурою – оператором циклу.

Паскаль передбачає визначення трьох різновидів оператора циклу:

рахівний цикл – *for k=1 to kmax do S;*

цикл з передумовою – *while C do S;*

цикл з післяумовою – *repeat S until C.*

C++ передбачає три подібних оператори:

for(int k=1;k<=kmax;k++)S;

while (C) S;

do S while(C);

Maple передбачає структуру:

for k from 1 to kmax while C do S end do;

Мова JavaScript повністю відтворює синтаксис C++.

Найголовнішою проблемою програм з циклом є питання про те, чи закінчиться коли-небудь цикл, чи він виявиться нескінченним. На перший погляд здається, що створення нескінченного циклу навіть не може бути поставлено за мету. Насправді це не так. Найпростіший приклад програми, що потребує нескінченного циклу, це аналізатор командного рядка. Аналізатор приймає від системи вводу команд рядок, розкладає його на компоненти, передає компоненти виконуючій системі і повертається знов до очікування команди. Єдиною умовою завершення роботи аналізатора є завершення роботи системи, але завершення роботи теж є командою, яку повинна обробити виконуюча система, тож команда завершення є такою самою, як і всі інші і так само обробляється. Тому цикл роботи аналізатора є саме нескінченним. Відповідно нормальна мова програмування не може заборонити нескінченні цикли, а про скінченність кожного конкретного циклу повинен турбуватися програміст.

Критерії скінченності циклу спираються на співставлення властивостей умови та тіла циклу. Припустимо, що умова циклу стосується значень якого-небудь байта `b` (256 можливих значень) і цикл має закінчитись, коли цей байт набуде спеціального значення, припустимо, 0. Якщо в тілі циклу значення `b` змінюється завжди тільки в бік зменшення, то не більше ніж за 255 кроків `b` стане нулем і цикл завершиться. Якщо ж параметр циклу може і зменшуватись і збільшуватись, підстави для впевненості в тому, що цикл коли-небудь завершиться, відсутні. Обмеження розміру параметра циклу одним байтом не є суттєвим, головне те, що параметр циклу завжди має скінченну кількість можливих значень. А от монотонність зміни параметру циклу є важливою умовою його скінченності.

Більш формалізовано умова скінченності циклу є умовою на послідовність значень параметра циклу, що виробляються тілом циклу і полягає вона в тому, що ця послідовність повинна збігатись до області, що в ній здійснюється умова

завершення циклу. Цикл рахівного типу характеризується тим, що для нього начебто зарані визначено кількість повторень. Насправді це не так, якщо в тілі циклу параметр змінюється в протилежному напрямку, наприклад

```
for(int k=0;k<1;k++){k ;};
```

В цьому прикладі цикл починається з $k=0$ і має завершитись, коли стане $k=1$. Але умова завершення перевіряється перед виконанням тіла циклу. Перше ж виконання тіла циклу зробить $k=1$, а потім оператор кроку циклу надасть параметру значення $k=0$ і тільки після цього буде перевірятись умова завершення. Зрозуміло, що такий цикл ніколи не завершиться. Транслятори мов високого рівня намагаються відслідковувати зміни параметра циклу в тілі і повідомляти про можливі неприємності. Цикл з передумовою та цикл з післяумовою не залишають транслятору можливості контролювати можливі зміни параметру в тілі циклу. Навпаки, для цих циклів зміна параметру є необхідною.

Розділ оголошень та ініціалізації

Місце в пам'яті відводиться, найчастіше, шляхом оголошення змінної потрібного типу. Деякі (орієнтовані на спрощення етапу розробки) мови програмування дозволяють створювати змінні навіть серед виконуваних операторів, деякі, що звертають більше уваги на полегшення роботи над помилками, вимагають, аби змінні оголошувались в окремому розділі оголошень. Ми надалі завжди користуватимемось цим правилом.

Паскаль. Оголошення змінних є частиною розділу оголошень і відбувається в форматі *Name [,Next_Name]:Type;*

Наприклад, *Var s, q, x :Real; n :Integer; const eps=0.001;*

Maple Оголошення змінної є одним з виконуваних операторів і співпадає з ініціалізацією змінних..

Наприклад, *x:=0.1:eps=0.001: s:=1:q:=1:n:=0;*

C++. Оголошувати змінні можна в виконуваній частині одночасно з ініціалізацією, а можна винести в розділ оголошень.

Наприклад, *float s=1; float x; float q=1; int n=0; float const eps=0.001;*

Розділ оголошень використовується транслятором.. Під час трансляції звернення до змінної замінюється зверненням до пам'яті за адресою, вказаною в цій таблиці. Відтак трансльована програма може нічого не знати про назви змінних, якщо тільки транслятор не додасть в програму таблицю змінних з метою полегшення пошуку помилок.

За розділом оголошень має бути розділ ініціалізації, в якому змінні отримують початкові значення. Автоматичного надання змінним стандартного значення не відбувається. Тому неініціалізована змінна може отримати зовсім випадкове значення, що залишилось від попередніх програм.

Розділ ініціалізації мовою Паскаль: *BEGIN S:=1;q:=1;n:=1;*

Розділ ініціалізації Maple та C++ може співпадати з розділом оголошень.

Типи даних та ефективність програми

Однією з центральних концепцій програмування мовами високого рівня є концепція типу даних. Вона з'явилася ще в першій мові програмування і виявилась настільки важливою, що знайшла своє відображення навіть в архітектурі

процесорів. Зміст цієї концепції полягає в тому, що множина всіх можливих даних та множина типових дій з даними розділяються на підмножини - типи даних. Кожному з типів даних властиве своє окреме тлумачення можливих значень та свій окремий набір типових (припустимих) дій з даними. Можна сказати, що тип даних є поєднанням способу збереження даних з правилами їх обробки. В найпростіших виконуючих машинах процесор може робити які завгодно дії з якими завгодно даними. Для однорозрядного процесора, що обробляв біт за бітом, всіх можливих дій є двома бітами-операндами небагато – всього 4, тому такий процесор залишався універсальним. З появою багаторозрядних процесорів універсальність стала неможливою, оскільки загальна кількість різних команд стає неприпустимо великою, тим більше що на практиці використовуються лише деякі з них. Стандартний перелік команд процесора вміщує майже тільки операції порівняння та цілої арифметики, а в багатопроцесорних системах один чи навіть декілька процесорів виділяються під дії з дійсними числами. Всі інші потрібні дії програмуються, як комбінації цих простих дій. Оскільки прості дії в різних процесорах є різними, такі комбінації також повинні бути різними для різних процесорів. Разом з тим, заміна процесорів на потужніші не повинна повністю відкидати старі програми, інакше розробка програм не буде встигати за розробкою процесорів. Тому найчастіше в процесорах забезпечується сумісність знизу вгору, тобто більш потужні процесори однієї серії звичайно можуть виконувати програми, що розроблені для менш потужних. Чи призводить така сумісність до ефективного використання потужніших процесорів? Розглянемо, наприклад, процес порівняння двох текстових рядків. В одnobайтовому процесорі він програмується, як послідовність процесів порівняння байтів з однаковими номерами. Якщо тепер ту саму програму виконувати на 64 розрядному процесорі, вона буде за один такт завантажувати в 64 розрядний акумулятор 8 бітів потрібного символу і порівнювати з 8 бітами другого рядка, а на порівнювання всіх 64 розрядів використовувати 8 таких тактів. Якщо ж програма транслюється з урахуванням можливостей процесора, вона буде використовувати один такт для порівняння одразу 64 розрядів, тобто працювати у 8 разів швидше, але тільки після того, як буде перетрансльована з урахуванням можливостей процесора. Це одна з причин уповільнення DOS-програм на 32-розрядних комп'ютерах.

Найзручніша для використання довжина даних залежить від архітектури процесора. Стандартне уявлення про байт виникло в зв'язку з задачами обробки текстів ще в ті часи, коли процесори були однорозрядними, але надалі для багаторозрядних процесорів стали використовувати процесорні регістри довжиною, що є кратною байту – 16-розрядні, 32-розрядні, 64-розрядні. Більш точно – використовується степінь двійки. Поле даних з довжиною, що дорівнює довжині типових регістрів процесора, називають машинним словом. Більш точно, машинним словом вважається квант обміну даними між процесором та оперативною пам'яттю – кількість бітів, що пересилається між процесором та пам'яттю за один такт. При використанні процесорів з великою розрядністю часто ефективніше використовувати не всі біти слова, ніж ділити машинне слово на частини, в яких зберігалися б різні дані. Наприклад, для збереження логічних значень, що можуть цілком вміститись в одному біті, досить часто використовується повне машинне слово.

Ефективність програмування суттєво залежить від того, наскільки повно програма використовує можливості конкретного процесора. Тому ефективні

програми завжди є платформно-залежними, а платформна незалежність може бути отримана тільки за рахунок втрати ефективності. Одним з сучасних напрямків досягнення компромісу між ефективністю та платформною незалежністю є проект GNU, в якому серед інших програмних засобів розробляється уніфікований транслятор G++. Цей транслятор має платформно-залежну частину власного коду, але забезпечує ефективну трансляцію кожної програми на кожній платформі, до якої він адаптований. З поширенням багатопроцесорних систем в ньому була врахована й необхідність паралельного виконання частин програми на різних процесорах. Крім мови C++, що є базовою, транслятор G++ підтримує, хоча й не так ефективно, практично всі сучасні мови програмування. Найсуттєвішим обмеженням, що його накладає використання універсальних трансляторів, є заборона використання конкретних властивостей процесора. Ця заборона не є абсолютною, її можна у той чи інший спосіб обходити, але в такому разі втрачається можливість переносу програм на більш потужні комп'ютери. Конкретні уявлення про конкретний комп'ютер корисні тільки в тому разі, коли розробляється програма, що має забезпечувати взаємодію з апаратурою конкретного комп'ютера. Для програм загального користування набагато доцільніше використовувати уявлення про стандартні методи збереження даних, що є частиною загальної концепції типів даних. Навіть коли виникає потреба в оцінці припустимого інтервалу значень якої-небудь змінної, мови високого рівня завжди надають можливість використати функцію `sizeof()`, що дає кількість байтів, виділених для змінної. Серед типів даних виділяються прості та складні, стандартні та програмовані. Прості типи даних мають зарані визначений розмір. Стандартні мають також зарані визначений набір дій. Серед програмованих можна виділити підтипи стандартних (діапазони), на які переносяться стандартні дії базового типу.

Стандартні типи даних

За час розвитку теорії програмування відпрацьовані певні уявлення про те, які саме типи даних найчастіше використовуються на практиці, і саме вони й складають базовий, тобто присутній практично у всіх мовах високого рівня, набір стандартних типів. Слід зауважити, що уявлення про стандартні типи дещо відрізняється від мови до мови, або навіть від реалізації до реалізації однієї і тієї ж мови. Так, базова частина мови Pascal містить тільки загальне уявлення про ціле число, не передбачаючи конкретної реалізації цього типу. Водночас кожен з трансляторів Pascal, оскільки він орієнтований на конкретну архітектуру процесора, має декілька підтипів цілого типу, що відрізняються довжиною комірки пам'яті. Навіть найуніверсальніша, як на сьогодні, мова C++ має в різних реалізаціях різні комплекти стандартних типів даних. Найчастіше під стандартними маються на увазі типи даних, що відомі транслятору без використання додаткових бібліотек. До них, зазвичай, відносять прості (або базові) типи та тип `String`. Останній тлумачиться, як масив літер, тобто не є простим, тому ми розглядатимемо його окремо.

Boolean – логічний тип, найпростіший стандартний тип даних. Він має всього два значення `True` та `False`. Для збереження даних цього типу можна, теоретично, використовувати один біт, але на практиці виділення одного біту з оперативної пам'яті є настільки неефективним, що використовуються байт, або два, або 4, або навіть і 8, залежно від розрядності процесора. Точніше сказати, що для

логічного типу за замовчуванням використовується одне машинне слово. Найчастіше нульове значення тлумачиться, як False, а всі інші як True.

Тип Boolean мають всі результати операцій порівняння, в які входять $>$ (більше), $<$ (менше), \leq (не більше), \geq (не менше), $=$ (дорівнює). Найважливішим місцем використання даних логічного типу є умовний оператор та оператори циклів. В цих операторах умова є виразом логічного типу, тобто не обов'язково попередньо оголошена логічна змінна, а вираз, що має саме цей тип. Дії, що припустимі для логічного типу: однооперандна дія not (змінює значення на протилежне); логічне об'єднання or (або або); логічне перехрещення and (і-і); виключне “або” – xor (або-або, але не обидва).

До стандартних відносяться цілі типи даних. Кожна архітектура процесорів використовує свої особисті уявлення про цілий тип, оскільки базовим цілим типом завжди є такий, що його довжина співпадає з розрядністю процесора. Він завжди має позначення Word, хоча й може бути різної довжини 16, 24, 32, 48, 64 розряди. З метою стандартизації виділяють *Integer*, *Longint*, *Short*, що мають стандартизовану довжину 16, 32, 8 розрядів.

Використання того чи іншого типу цілих даних пов'язане з суттєвою залежністю швидкості роботи програми від довжини цілих значень, адже якщо ціле не вміщується в процесорі, кожна операція виконується в 2 або й 4 етапи відповідно. Якщо ж програма написана з урахуванням короткого процесорного слова, вона досить часто не прискорюється на потужнішому процесорі.

Цілі типи є впорядкованими, для них визначена операція отримання попереднього $pred(x)$ або наступного $succ(x)$ значення. Ця операція виробляє ознаку помилки, якщо відповідне значення виходить за межі діапазону.

Найпростішим з цілих типом даних є *Byte*. Відповідно до назви він використовує для збереження даних один байт, а біти, що його складають, тлумачаться за двійковою арифметикою: біт 00000001 тлумачиться, як 1; біт 00000010 – як 2; 00010000 – як 16 і так далі, біт 11111111 – як 256.

Тип *ShortInt* відрізняється від *Byte* тим, що в ньому зберігаються не тільки додатні, а й від'ємні значення.

Типи *Integer* та *LongInt* відрізняються від *ShortInt* довжиною, а відповідно й місцем розташування знакового біта.

Дійсні типи. Специфіка комп'ютерного зображення дійсних чисел полягає в тому, що дійсні числа розташовані на числовій вісі надто щільно, адже кількість дійсних чисел між якими завгодно двома числами така ж сама, як і на всій вісі. Неприємним наслідком цього твердження є те, що в комп'ютері не можна повністю зобразити не тільки числову вісь, а й який-небудь її відрізок. З теорії чисел відомо, що раціональні числа, тобто відношення цілих чисел утворюють всюди щільну підмножину множини дійсних чисел, тобто в околі довільного дійсного числа завжди знайдеться принаймні одне раціональне число. Цей факт дозволяє наближено зображувати дійсні числа досить близькими до них раціональними, покриваючи числову вісь мережею раціональних чисел і обираючи за зображення дійсного числа найближче до нього значення з цієї мережі.

В 16-розрядній архітектурі розділяються дійсні різної довжини (*Real*, *Complex*, *Extended*). Транслятори Pascal за замовчуванням використовують тип *Real*, інші типи тільки у разі вмикання опції транслятора Pentium III є тип *Extended*.

Мова C++ використовує типи Short, float та double. Останній тип повністю відповідає можливостям 64 розрядних співпроцесорів і саме його найдоцільніше використовувати при обчисленнях.

Символьний тип – Char. Значення цього типу тлумачаться, як номери в таблиці малюнків зображень літери на вивідному пристрої. Історично склалося, що на цей тип відведено 8 бітів (один байт), хоча для відтворення всіх літер всіх алфавітів світу потрібно 2 байти. Існує стандартна кодова таблиця для базової частини 127 символів та її розширення, що забезпечують відтворення всіх інших наборів символів. Ці розширення базової кодової таблиці відбувається в різних операційних системах у різні способи. Відповідні засоби отримали назву засобів локалізації.

З програмної точки зору тип *Char* є впорядкованим типом, який можна перетворити на число функцією *ord(x)*, а можна зробити й обернене перетворення *chr(x)*. Символи можна також друкувати або ж читати *Write(char)*, *Read(char)*, порівнювати на співпадіння або впорядковувати. Тлумачення номерами дозволяє використовувати для символьних змінних ті ж самі операції порівняння, що й для цілих чисел і так само переходити до наступного, але тільки за допомогою функцій *succ()*, *pred()*. Мова C взагалі не робила різниці між символьним типом та коротким цілим, тому і в C++ для типу char застосовні всі цілі дії.

Символьний тип використовується лише в окремих випадках, найчастіше в програмах зустрічається похідний від нього рядковий тип String, що є послідовністю символів.

Дії чисто символьного типу залежать від апаратної реалізації, тому Pascal винесить їх в окрему бібліотеку CRT, до якої входять типові функції *ReadKey:Char* та *KeyPressed:Boolean*. Перша з них зупиняє проходження програми до натискування клавіші, а потім повертає код натисненої клавіші. Друга повертає логічне значення True, якщо клавіша вже була натиснена, але ще не прочитана. Тому програма, що працює без зупинок, але перевіряє, чи не була натиснена певна клавіша, має містити комбінацію цих двох функцій: *If Keypressed then res:=ReadKey else res:=chr(0);*

Після цього рядка змінна *res* матиме значенням або нуль, або код натисненої клавіші.

Вказівний тип *Pointer*. Цей простий за термінологією програмування тип є базою для створення найскладніших структур даних. За означенням, значеннями цього типу є адреси в пам'яті. Оскільки організація пам'яті є апаратно-залежною, тлумачення значень суттєво залежить від реалізації комп'ютера і навіть від того, в якому саме режимі працював транслятор. В базовому Паскалі вказівний тип вважається невпорядкованим, тоді, як в усіх різновидах він може тлумачитись, як число. Це часто використовується в C-програмах. Вказівний тип може бути нетипований, тоді він припускає тільки привласнення та порівняння (тільки на співпадіння), а може бути вказівним на певний тип даних, тоді припускається ще така корисна дія, як розіменування, перехід від вказівника на той об'єкт, що на нього вказує цей вказівник. Оголошення типізованого вказівника здійснюється за допомогою префікса перед назвою типу вказуваного об'єкту. Наприклад, якщо потрібно означити вказівник на ціле, це можна зробити за допомогою оголошення змінної *p_int: ^Integer*. Використання цієї змінної може здійснюватись або безпосередньо (привласнити значення іншому вказівнику), або за допомогою суфікса *^*, в останньому разі *^p_int* тлумачиться як ціла змінна і може використовуватись в діях цілої арифметики.

Специфічними для вказівного типу діями є відведення пам'яті та звільнення пам'яті. Для нетипованого вказівника `p:Pointer` відведення пам'яті здійснюється запитом `getmem(p,size)`, якому потрібно вказувати розмір потрібної пам'яті `size`, а звільнити цю пам'ять можна запитом `freemem(p,size)`. Типований вказівник використовується із запитом `New(p)`, `Dispose(p)`. Використання вказівного типу в програмах C++ відрізняється головним чином заміною знаку на * та перестановкою префіксів та суфіксів. Означення вказівника на ціле має такий вигляд `int* p_int` (зірочка і є ознакою вказівного типу), а використання значення збереженої за цією адресою величини здійснюється за допомогою звернення `*p_int`. Виділення та звільнення пам'яті здійснюється подібними процедурами.

Складні типи даних

Складні типи даних використовуються, коли потрібно розробляти дії, які відсутні серед припустимих для стандартних типів. Серед таких дій може бути навіть просто привласнення та порівняння, але для даних, для яких відводиться нестандартна область пам'яті.

Утворення складного типу даних потребує визначення розміру пам'яті, що виділяється для даних, та означення припустимих дій. Завжди можна вигадати багато різних способів реалізації і першої, і другої задачі, але існує ряд стандартних методів, вигаданих саме для створення типів даних.

Виділення пам'яті може відбуватись у різні способи, залежно від того, які саме дії передбачається використовувати для обробки даних утвореного типу. Найпростішим є використання масивів - послідовностей даних однакового типу. Масиви корисно використовувати, коли передбачається виконувати подібні дії з усіма елементами масиву, перебираючи ці елементи, оди за одним, в циклі з переліком.

Досить часто до різних частин складного типу потрібно застосовувати різні дії. В такому разі доцільно використовувати структуру даних, що має назву запис (*record*) мовою *Pascal*, або *structure* мовою C++, або *table* мовою *Maple*. Ця структура означає тип, як перелік полів, кожне з яких є змінною простішого, вже означеного типу, і до якого можна застосовувати дії цього простішого типу.

Методи означення дій теж можуть бути найрізноманітнішими, але й вони поділяються на два різних способи. За одним, логічно простішим, означення дії здійснюється у вигляді загальної підпрограми, що отримує за аргумент змінну означуваного типу і здійснює над нею всі потрібні дії. Більш розвинутою є техніка об'єктного програмування, за якою тип створюється, як суцільний об'єкт, що містить разом і поля даних, і методи їх обробки. В цій конструкції даними для методів обробки послуговуються поля об'єкту і зменшується можливість переплутування даних з різних частин програми.

Метод об'єктно-орієнтованого програмування використовується для великих за обсягом програм або для програм з досить заплутаною логікою, оскільки цей метод дозволяє ефективно здійснювати принципі програмування згори донизу, поділяючи обробку вхідних даних на етапи обробки частин даних. Саме тому цей метод активно розвивається, остання зі створених універсальних мов програмування Java взагалі навіть не визнає типів даних чи змінних, що не є частиною певного об'єкту. Створювана програма, за Java-ідеологією, теж є об'єктом, що передається на обробку виконуючій системі.

Отже, серед складних типів даних можна розрізнати *масиви* - послідовності даних однакового типу, *записи* - комплекси даних різних типів, та *об'єкти* - комплекси даних разом з методами їх обробки.

Впорядкування даних однакового типу найефективніше здійснюється за допомогою спеціалізованого способу створення типу даних масиву. Масив за означенням є послідовністю даних однакового типу з визначеною кількістю елементів. Масиви найзручніше використовувати для виконання однакових дій над усіма елементами. Це можна робити, використовуючи в циклі замість окремих даних елементи масиву, номер яких змінюється в циклі.

Найточніше означення *масиву* здійснюється мовою Pascal за допомогою конструкції *Array[Type of indexes] of <Type of elements>*. Тут мається на увазі, що *Type of indexes* є одним з простих впорядкованих типів з такою кількістю елементів, щоб загальний розмір масиву не виходив за межі припустимого. Загальний розмір обчислюється, як добуток кількості елементів на розмір одного елемента, а припустимий розмір даних обмежений апаратною реалізацією і складає 64 кілобайти для DOS-трансляторів і 2 гігабайти для трансляторів в середовищі Win 32, хоча в цьому середовищі можуть виникати проблеми часу виконання, якщо загальної оперативної пам'яті (фізичної разом з віртуальною) не вистачить для розташування всього масиву. Операційні системи 64 розрядної архітектури позбавлені таких обмежень етапу трансляції. Транслятори C++ орієнтовані на означення масивів за допомогою перерахування кількості елементів, а Maple динамічно збільшує довжину масиву за потребою. Використання масивів найчастіше здійснюється шляхом перебору елементів. Для масиву, як такого, можна застосовувати дві загальні дії: привласнення масиву *a* значення масиву *b*, тобто $a:=b$, та порівняння на збіг $a=b$ ($a \langle \rangle b$). Привласненням є й передача значення масиву підпрограмі *subroutine(a)*.

Динамічні масиви використовуються з метою зменшення розміру файлу з програмою. Статичні змінні, тобто змінні, місце для зберігання яких виділяється ще на етапі трансляції, розташовуються в зарані виділеній для програми частині оперативної пам'яті, що має назву область даних. Під час трансляції вся ця область копіюється на диск, хоча дані, що будуть в ній зберігатись, ще не визначені (виключенням є тільки Java). Тому, якщо для масиву даних потрібно 2 кілобайти пам'яті, ці 2 кілобайти, заповнені вщент нулями, копіюються з диску в ОП кожного разу, як тільки програма завантажується для виконання. Якщо ж програмі потрібно обробляти масив $100 \times 100 \times 100$ дійсних чисел, копіюються вже 8 мегабайтів суцільних нулів. Саме в цих випадках корисно використовувати динамічний масив, оголошуючи замість змінної з типом масиву змінну, що є вказівником на масив, адже така змінна займає всього одне машинне слово.

Мовою Pascal цей метод мав би такий вигляд: в розділі оголошень означається стала `Const dim=100;` далі означаються тип масиву `Type mas=Array[1..dim] of Extended;` `MasP= Mas;` далі означається змінна `Data:MasP,` яка й буде забезпечувати доступ до масиву; потім в тілі самої програми або в розділі ініціалізації до цієї змінної застосовується оператор виділення пам'яті `New(Data);` після цього можна працювати з масивом, елементи якого доступні за допомогою вказівника `Data`. Звернення до конкретного елемента відбувається за допомогою двох дій: по-перше, потрібно перейти від вказівника на масив до самого масиву (операція розіменування) `data`, по-друге, визначити конкретний номер елемента

масиву (операція індексації) `data [k]`. Цей вираз і визначає *k*-тий елемент динамічного масиву.

Записи

Записи або структури є поєднанням даних різних типів. Зразком запису може бути інформація про студента, що вміщуватиме такі частини: прізвище, ім'я та по-батькові, рік (або повна дата) народження, рік вступу, номер групи, перелік зданих іспитів з оцінками, рік закінчення (або відрахування), номер диплому (якщо так вийшло). Для цих даних потрібні різні поля для збереження: рядковий тип для персональних даних та для назв предметів, числові поля для оцінок та номерів і таке інше. Тому зібрати всі дані в масив просто неможливо, і вони групуються в запис. Поєднання даних в запис корисне з програмної точки зору, оскільки після такого поєднання спрощується обробка записів про студентів.

Запис мовою Pascal:

```
Type Student=record Name, Parent, Family :String; Number :LongInt; End;
```

Запис мовою C++:

```
struct Student {
char* Name, Parent, Family;
longint Number;
}
```

Мова Maple не має спеціального типу для записів, як таких, натомість вона використовує загальний тип `table` для конкретних даних. Наприклад, запис про конкретного студента може мати вигляд: `st[1] := table ([Name = Петро, Parent = Іванович, Family = Сидорко, Number = 1234567])`.

Об'єкти

Використання записів пов'язане з розподілом програми на окремі блоки підпрограми, і саме підпрограми обробки даних є ще одним типом даних, що входять в більш загальну структуру об'єкт. Мова C++ використовує для такого поєднання даних та методів трохи інші терміни - "структура" та "клас", а програмування з використанням об'єктів отримало назву "об'єктно-орієнтоване програмування". Справжній зміст використання об'єктів полягає не в поєднанні, а в відокремленні програміста від турботи про внутрішню реалізацію об'єктів, оскільки найчастіше доступ до полів об'єкта відсутній, а потрібні значення встановлюються відповідними методами цього об'єкта.

Файли

Під *файлом* розуміють логічну одиницю збереження інформації: програму, лист, статтю, малюнок тощо. Уявлення про файли виникло з появою можливості зберігати на одному носії блоки даних, що мають використовуватись незалежно один від одного. Крім змісту, файли відрізняються один від одного назвою та місцем розташування. В файлах можна зберігати результати розрахунків, результати вимірювань, що потім потрібно буде обраховувати, просто вхідні дані для розрахункової програми і навіть проміжні результати обчислень. В файлах також зберігаються тексти ділової документації, наукових публікацій, курсових, дипломних робіт тощо. Якщо в науковій роботі використовується комп'ютерна обробка даних, то використовувати результати такої обробки набагато простіше, коли вони вже

існують в комп'ютері у вигляді файлу результатів. Саме тому головним результатом розрахункової програми є скоріше не зображення на екрані, а файл.

В роботі з файлами слід відзначити власне обробку даних від зчитування даних з файлу або записування даних в файл. Обробка даних є внутрішньою справою програми і насправді не має відношення до роботи з файлами. Досить часто запису даних в файл передуює підготовка цих даних перетворення у форму, зручнішу для запису, а зчитування даних найчастіше супроводжується перетворенням до форми, зручнішої для подальшої обробки, але ці дії тільки супроводжують роботу з файлами. Власне робота з файлом полягає тільки в передаванні даних, тобто записуванні даних в файл або зчитуванні даних з файлу. Ці дії виконуються процедурами запису – write – або зчитування – read – з відповідними доповненнями. Разом з тим до роботи з файлом залучаються ще процедури підготовки файлу до роботи та завершення роботи з файлом. Зміст цих додаткових процедур легше зрозуміти, якщо трохи детальніше розібрати ті дії, що їх потрібно виконувати кожного разу, як тільки відбувається обмін даними з файлом.

На пристроях зберігання інформація, зазвичай, розташована у вигляді блоків фіксованого розміру – секторів, що вміщують найчастіше 512 байтів. Сектори поєднуються в доріжки, циліндри тощо, але розмір кожного такого угруповування завжди кратний 512 байтам, тоді як розмір файлів може бути довільним. Самі пристрої приєднуються до комп'ютера через спеціалізовані процесори – адаптери. Наприклад, адаптер жорсткого диску досить часто навіть не виділяє при зчитуванні одного сектора, а копіює собі в власну пам'ять цілу доріжку, а вже з цієї доріжки виділяється і передається далі один сектор (або декілька, якщо потрібно). До того ж для обміну даними між адаптером та оперативною пам'яттю часто використовується так званий канал прямого доступу, що пересилає дані між адаптером та ОП без втручання процесора.

Такі спеціалізовані дії суттєво відрізняються від комп'ютера до комп'ютера, але вони є цілком стандартними за своїми результатами, тому існує спеціалізований комплекс програмних засобів, що перетворює файли на послідовність блоків (в процесі запису) або навпаки, перетворює послідовність блоків в файл (під час зчитування). Цей комплекс має назву файлової підсистеми, що є частиною операційної системи. Програми, що входять до складу файлової системи, приховують від інших програм специфіку організації збереження файлів на різних пристроях, тому для використання необхідного файлу достатньо тільки зазначити його повну або скорочену назву.

Назва файлу складається з декількох різних за змістом частин власне імені файлу, назви каталогу, а можливо й назви комп'ютера, на якому зберігається файл. Назву каталогу називають часто шляхом до файлу. Саме шлях до файлу й можна скорочувати. Сам цей термін виник у зв'язку з уявленням про робочий каталог – той, в якому можна працювати з файлами безпосередньо за іменами. Колись давно, з метою впорядкування інформації про розташування файлів на пристроях було використано спеціалізовані файли – каталоги, що вміщували табличку відповідності між назвою файлу та місцем його розташування. Оскільки сам каталог теж є файлом, такий файл можна розташувати серед інших файлів каталогу. Так утворюються вкладені каталоги, що для кожного пристрою утворюють ціле дерево файлової структури. Операційна система, за умов недостатнього розміру оперативної пам'яті, тримала в ОП тільки один каталог - він і називався робочим.

Насправді, найчастіше в ОП утримувався зміст двох каталогів – робочого та системного. Перехід з каталогу до каталогу супроводжувався зміною змісту буфера робочого каталогу, а забезпечити доступ до конкретного файлу можна було тільки в тому разі, якщо цей файл знаходився в робочому каталозі. Саме з тих часів і з’явилося уявлення про те, що для доступу до файлу потрібно зробити певну кількість переходів з каталогу до каталогу – пройти певний шлях до файлу. Наприклад, приватні дані студента першої групи першого курсу можуть знаходитись на диску, що позначається літерою D: , в каталозі “students” , підкаталозі “group11”, в якому ще й розташовано персоніфікований підкаталог LastName . Шлях до всіх приватних даних буде задаватись, як D:\students\group11\LastName\ .

Якщо для зручності відділити дані програми “first” та “second” в різні підкаталоги, файл з текстом програми “first.pas” буде мати повну назву D: \ students \ group11 \ LastName \ first\first.pas, а програми “second.pas” – D: \ students \ group11 \ LastName \ second \ second.pas .

Звернення до файлу здійснюється за допомогою назви, що може бути повною (абсолютною) або відносною. Відносна відрізняється тим, що в ній використовується припущення про те, який саме каталог є робочим на момент звернення до файлу.

Імена, каталоги, розширення – всі ці ознаки файлу потрібні тільки тому, що на пристроях зберігання знаходиться багато різних файлів. Конкретні програми, якщо вони звертаються до файлів з даними, використовують далеко не так багато файлів одночасно, тому складна система файлових імен їм не потрібна. Більш за те, складна система файлового імені заважатиме роботі з файлом. Припустимо на хвилинку, що кожного разу для запису даних в файл ми повинні були б вказувати повну або скорочену назву файлу, а під час роботи програми з’ясувалось, що нам відтепер буде потрібно працювати з іншим файлом. Тоді було б необхідно виправляти назву в кожному рядку з операцією запису. Тому в програмах замість назви файлу використовується файловий дескриптор (сукупність даних про файл). Не менш важливою частиною процесу роботи з файлами є буферизація – попереднє накопичення змісту файлу в оперативній пам’яті, якщо файл потрібно читати, або підготовка даних в оперативній пам’яті перед записом з наступним перенесенням на записуючий пристрій блоків даних стандартної довжини. Всі ці дії виконуються за допомогою стандартних наборів – підпрограм, а місце для буферів виділяється за допомогою оголошення змінних спеціального типу – файлових змінних. Спеціальна процедура – assign – встановлює зв’язок між файлом на пристрої та файловою змінною. Після цього назва файлу в програмі більше не використовується, а звернення до файлу здійснюється, як звернення до файлової змінної.

Іноколи програмам потрібно отримувати більш детальну інформацію про файлову структуру, наприклад, шукати потрібні файли, перебирати каталоги тощо. З цією метою система програмування звичайно доповнюється комплексом процедур обробки каталогів (модуль DOS, Windows, тощо). Найчастіше, в досить простих програмах, що мають на меті тільки зберігати результати розрахунків або проміжні дані, ці процедури просто не потрібні. Такі процедури винесені в окремі бібліотеки.

Типи файлів

Файлова змінна насправді є не просто змінною, а певним об’єктом, що з ним пов’язані уявлення про конкретну структуру даних, що зберігатимуться в файлі.

Відповідно до цієї структури файлові змінні можуть бути типізованими в різний спосіб. Наприклад, оголошення *fb: File of Byte* створює файлову змінну, що з її допомогою можна читати з файлу або записувати в файл дані розміром в один байт, тоді як *fr: File of Real* забезпечує можливість обміну з файлом даними в одне дійсне число.

Слід зауважити, що типізована файлова змінна дозволяє здійснювати обмін з файлом даними тільки того типу, що вказаний в описі змінної.

Своєрідним типом файлових змінних є *Text*. Цей тип призначено для роботи з текстовими рядками – послідовностями символів, що можуть відділятися одна від одною спеціальною ознакою кінця рядка. Цією ознакою в різних типах операційних систем є різні спеціалізовані символи: в системах DOS та Windows використовується два символи з кодами 0D та 0A (десяткові 13 та 10), в системах Unix – тільки один символ з кодом 0D, в MacOS - символ 0A. Відповідно, програми обробки текстових файлів повинні враховувати особливості обробки текстів, в тому разі і під час пересилання текстових файлів з одного комп'ютера на інший.

Текстовий тип файлів відрізняється також тим, що в файлах цього типу інформація має такий самий вигляд, що й на екрані, або принтері, тобто її можна читати. Стандартні дані несимвольних типів відтворюються у текстових файлах за допомогою стандартних процедур перетворення, які навіть не потрібно спеціально викликати.

Файлова змінна і певний файл ставляться у відповідність одне одному за допомогою процедури `assign(var fd:(File of SomeType),name:String)`. Ця процедура тільки ставить у відповідність файл та файлову змінну, нічого іншого не здійснюючи з самим файлом. Вона також готує файловий буфер – область пам'яті, в якій будуть накопичуватись дані.

Початок роботи з файлом здійснюється разом з означенням напрямку обміну даними з файлом. В інших мовах програмування ця дія тлумачиться, як відкриття файлу, Pascal використовує ключове слово `reset` – встановити на початок для процедури `reset(var fd:FileType)`, що готує файл для зачитування, та ключове слово `rewrite` для процедури `rewrite(var fd:FileType)`, що готує файл для запису.

Оскільки читати з файлу, якого не існує, не можливо, процедура *reset* виробляє помилку, якщо файл із зазначеною назвою не знайдено. Процедура *rewrite* просто замінює файл, якщо він існує і його можна модифікувати, або ж створює новий файл. Слід зауважити, що створення нового файлу або модифікація того, що існує супроводжується зміною змісту файлу каталогу, в якому знаходиться інформація про оброблюваний файл. Тому програма повинна мати повноваження не тільки на модифікацію файлу, а й на модифікацію каталогу.

Файл, що його підготовлено до зчитування процедурою `reset`, копіюється в буфер, тому зчитування даних практично не затримує роботи програми. Якщо файл не вміщується в буфері, копіюється тільки його частина, а наступні частини файлу операційна система намагається копіювати раніше, ніж буде використано весь зміст буферу. Наслідком такого поступового зчитування є те, що повернутись до вже прочитаних даних і знову зчитати їх ще раз можна тільки з самого початку файлу, використовуючи знов процедуру `reset`.

З файлом, що його підготовлено до запису, ситуація складніша. Головною відмінністю є те, що під час запису система насправді не переписує нові дані замість старих, а створює новий файл з тією ж назвою, одночасно вивільнюючи місце на диску для наступного використання. Відповідно одразу після виконання процедури `rewrite` втрачається можливість звертатись до даних, що були записані в файлі.

Замість того, файлова система утримує буфер в ОП, в якому поступово накопичує дані для запису, копіюючи їх на диск в так званому “лінивому” режимі, тобто тоді, коли це стане зручно для самої системи.

Для текстових файлів передбачена додаткова можливість дописування даних в кінець файлу, що вже існує. Вона здійснюється за допомогою процедури `append` (`fd:Text`). Деталі реалізації цієї процедури змінюються від одного покоління операційних систем до другого, але вона насправді дописує дані саме в кінець файлу.

Нарешті, коли обмін даними з файлом закінчено, його потрібно закрити процедурою `close(fd)`. Ця процедура просто забороняє доступ до буфера і знімає контроль за наповненістю буфера, якщо файл відкривався для зчитування. Якщо ж файл використовувався для запису, закриття файлу супроводжується, по-перше, записом всього змісту буфера на диск, а по-друге, записом в каталог інформації про справжній розмір файлу. Тільки після цього файл стає придатним для наступного використання. Раптове вимкнення комп'ютера навіть в потужних операційних системах, як правило, приводить до зникнення всіх файлів, що були підготовлені на запис, але не були закриті. Режим дописування текстового файлу, якщо його використання супроводжується закриттям після запису наступної порції даних і відкриванням безпосередньо перед записом, забезпечує можливість зберігати проміжні дані навіть в досить складних ситуаціях. Саме цей режим використовується, наприклад, для системних журналів, що відслідковують ефективність та безпечність використання комп'ютера.

Adobe Photoshop

Безперечним лідером галузі сучасного видавництва електронних документів вже багато років є компанія Adobe Systems Inc. Саме вона перша реалізувала програмний і апаратний інтерпретатори мови опису сторінок PostScript, який використовують усі високоякісні пристрої виводу зображень, і, насамперед, фотоскладальні автомати, що складають технологічну основу сучасного циклу підготовки поліграфічних оригіналів-макетів. Програми *Photoshop*, *Illustrator*, *FrameMaker*, *InDesign*, *Acrobat* і ін. утворюють чудовий і могутній комплект програмного забезпечення сучасного видавництва. Інтеграція і єдність інтерфейсу цих програм прискорює роботу і полегшує їхнє освоєння.

Adobe Photoshop є безумовним лідером серед професійних графічних редакторів за рахунок своїх широких можливостей, високій ефективності і швидкості роботи. Програма надає всі необхідні засоби для корекції, монтажу, підготовки зображень до друку і високоякісного виводу.

Друга, не менш велика сфера застосування програми – Web-дизайн і електронні публікації. Можна зовсім без перебільшення сказати, що Photoshop знаходиться в арсеналі кожного професійного комп'ютерного дизайнера.

Adobe Photoshop призначений для редагування і створення растрової графіки (*bitmapped images*). Програма використовується для роботи з фотографіями і колажами з них, мальованими ілюстраціями, слайдами і мультиплікацією, зображеннями для Web-сторінок, кінокадрами. Photoshop має практично безмежні можливості. Його з успіхом використовують фотохудожники для ретуші, колірної і тонової корекції, підвищення різкості і створення художніх ефектів. Добре продуманий набір інструментів для роботи з частинами зображення незамінний для оформлення монтажів. Великий набір спеціальних фільтрів активно застосовується при створенні як комерційного дизайну, так і художніх творів. Вмонтована в Photoshop

програма ImageReady дозволяє створювати анімовані зображення. Adobe Photoshop відноситься до професійних програм і орієнтована на високий апаратний рівень.

Версія Photoshop 7.0 з вмонтованим редактором ImageReady 7.0 надає такі додаткові можливості:

- швидка і зручна навігація по зображеннях;
- сортування зображень за категоріями (наприклад “Не закінчені”, “Готові” і т. п.), іменем файлу, висотою і шириною зображення, розміром та типом файлу, роздільною здатністю, колірним профілем, дато створення і модифікації, автором зображення;
- керування і робота з файлами і каталогами – створення, перейменування і видалення папок, копіювання, переміщення видалення й перейменування файлів зображень, перейменування групи зображень;
- повертання зображень на 90 або 180 градусів;
- наявність нових інструментів ретушування зображень (Healing Brush та Patch Tool);
- порівняння якості зображення і розмірів файлу до і після оптимізації;
- гнучке керування прозорістю;
- збереження чіткості тексту і векторної графіки при стисканні для областей, що містять такі елементи;
- створення персональних робочих просторів, що зберігають установки для подальшого використання;
- створення унікальних інструментів шляхом налаштування існуючих і збереження змін.

Photoshop дозволяє виконувати обробку фотографій на професійному рівні, що, звичайно, спричиняє ускладнення програми. Для роботи з ним недостатньо знати, що роблять конкретні команди чи інструменти. Необхідно уявляти собі суть процесів обробки зображення, володіти хоча б основами теоретичних знань про растрову і векторну графіку, про тонову і кольорову корекцію та інше.

Робоча область Photoshop включає *рядок меню, рядок стану, інструментальні палітри та інформаційне поле.*

Рядок меню розміщується в верхній частині вікна Photoshop і містить дев'ять основних меню: *File (Файл), Edit (Редагування), Image (Зображення), Layer (Шар), Select (Вибір), Filter (Фільтр), View (Вид), Window (Вікно), Help (Допомога).*

Рядок стану, що розміщується в нижній частині робочої області, містить основну інформацію про файл і ту інформацію про комп'ютер, яка стосується Photoshop: масштаб, меню інформаційного поля, повідомлення.

Меню інформаційного поля містить п'ять команд:

- *Document Sizes* (Розміри документів) – при наявності в документі шарів вказує два значення – розмір файлу зображення в згладженому виді і розмір файлу зображення з шарами;
- *Scratch Sizes* (Розміри робочих дисків) – зліва вказується розмір оперативної пам'яті, що використовується в даний час, а справа – загальний обсяг доступної оперативної пам'яті;
- *Efficiency* (Ефективність) – відсоток вказує достатність оперативної пам'яті для обробки поточного зображення;
- *Timing* (Хронометраж) – показує, скільки часу необхідно Photoshop для обробки останньої команди;

- *Current Tool* (Поточний інструмент) – відображає просте позначення інструмента, що використовується в даний час.

Повідомлення містять інформацію про поточне зображення, призначення команди і т.п.

Інструментальні палітри являють собою діалогові вікна, призначені для налаштування параметрів основних інструментів та проведення деяких операцій з зображеннями. Вони розміщуються зверху робочої області, надаючи швидкий доступ до команд і можливостей програми. Як правило, відкритими залишають тільки ті палітри, які найчастіше використовуються. В Photoshop використовуються палітри: *Tools* (Інструменти), *Navigator* (Навігатор), *Info* (Інфо), **Палітра параметрів**, *Color* (Синтез), *Swatches* (Каталог), *Brushes* (Пензлі), *Layers* (Шари), *Chennels* (Канали), *Paths* (Контури), *History* (Протокол), *Actions* (Операції).

Вікна програм ImageReady і Photoshop дуже схожі. Різниця полягає в вилученні та переміщенні деяких інструментів та появі нових вікон: Animation (Анімація), Rollover (Ролловер), Image Map (Карта ссылок).

Програму ImageReady можна запустити зі списку Программы системного меню Пуск, за допомогою ярлика, або безпосередньо з програми Photoshop.

Вікно Animation (Анімація), яке можна відкрити, клацнувши на відповідній вкладці, спочатку містить один кадр, інструменти створення і видалення кадрів та тестового програвання анімації. Після того, як відкрито або створено зображення (бажано, щоб всі його елементи знаходились в різних шарах), можна додавати кадри і змінювати в них положення окремих елементів зображення, їх колір та інші параметри.

Створення простих анімацій в Photoshop за допомогою програми ImageReady передбачає такі етапи:

- · створення початкового зображення;
- · створення кількох ключових кадрів (створюється новий кадр і робиться переміщення об'єкта);
- · створення переходів з одного стану в інший за допомогою команди Tween (Промежуточный);
- · проведення оптимізації за допомогою спеціальної опції;
- · визначення часових інтервалів для кожного кадру чи для анімації в цілому;
- · збереження файлу з потрібними якістю і розміром.

В діалоговому вікні, яке викликається командою **Tween** (Промежуточный), можна задати напряму формування послідовності проміжних кадрів (до кадру «до» - Previous Frame, та до кадру «після» – Next Frame). Можна також визначити кількість проміжних кадрів і те, які параметри зображення будуть враховуватись при створенні проміжних кадрів:

- · *Position* (Позиция) – положення елементів зображення в кадрі;
- · *Opacity* (Непрозрачность) – поступовість перетворення кольору елементів зображення;
- · *Effects* (Эффекты) – будь-які стилі шарів: тіні, рельєфи, відблиски, що застосовані до елементів зображення.

Ролловер, що створюється в програмі ImageReady, являє собою набір зображень, які змінюють одне одного при взаємодії з курсором миші. Зовнішній вигляд ролловера пов'язується з такими подіями: *Over* (Над), *Down* (Вниз), *Up*

(Вверх), *Click* (Щелчок), *Out* (Наружу), *None* (Отсутствие). Ролловер може включати не всі шість, а тільки частину станів. Більшість ролловерів для Web мають два стани: *None* (Отсутствие) і *Over* (Над). Це забезпечує менший обсяг файлу і полегшує сприйняття Web-сторінки користувачем. Створюють ролловери в вікні *Rollover* (Ролловер).

Палітра карт посилань *Image Map* (Карта ссылок) надає інструменти малювання форм на поверхні зображення, призначення цим зонам посилань на Web-сторінки, файли, звуки або анімацію. Можна також задати текст підказки, яка впливає при наведенні курсору на посилання.

За допомогою палітри *Optimize* (Оптимизация) можна вибрати Web-безпечний формат файлу для його збереження, задати якість вибраного формату і вибрати різновид завантаження зображення (прогресивний чи черезрядковий).

При експорті зображення в HTML-формат *ImageReady* автоматично генерує сценарій мовою JavaScript. Для того, щоб читач Web-сторінки побачив ролловерні ефекти, треба, щоб ця мова не була відключена в його браузері.

Flash-технології

Набули розповсюдження так звані Flash-технології (технології інтерактивної Web-анімації), започатковані компанією *Macromedia*, яка розвинула графічну програму для Web (*FutureSplash*). Орієнтація на векторну графіку, як основний інструмент розробки, дозволила реалізувати всі базові елементи мультимедія – рух, звук та інтерактивність.

До основних позитивних сторін Flash можна віднести:

- · невеликий розмір файлів, що пов'язано з використанням векторної графіки і потужних алгоритмів стиснення інформації;
- · усунення проблем сумісності між браузерами;
- · відсутність проблеми невідповідності розмірів екрану і сторінки (якщо задати розміри об'єктів через відсотки від розмірів самого екрану, масштабуються окрім елементів векторної графіки також вмонтовані графічні зображення);
- · наявність автоматичної підтримки антиаліазингу (згладжування контурів шляхом змішування сусідніх кольорів), що забезпечує красоту зображень;
- · наявність вмонтованої мови опису сценаріїв;
- · зручність, що забезпечується простотою інтерфейсу;
- · наявність засобів експортування зображень в найбільш розповсюджені графічні формати;
- · поширеність.

Недоліками Flash є те, що вона поки-що не підтримує тривимірну графіку, програвач анімації забирає достатньо багато ресурсів процесора, а швидкість програвання залежить від розмірів екрану (для повільних процесорів невеликою буде швидкість роботи). Окрім цього Flash поступається за можливостями створення і редагування зображень, які надають поширені програми *Photoshop* та *CorelDraw*.

У *Macromedia Flash* існує два принципово різних способи анімувати щонебудь. Перший – прорисовувати кожен кадр самому, використовуючи Flash тільки як засіб, що дозволяє швидко перегортати ваші зображення, і другий – змусити Flash автоматично прораховувати проміжні кадри.

Базовими поняттями Flash є **кадри** (frames), **символи** (symbols), **шари** (layers) і **часова шкала** (timeline).

Часова шкала – основний інструмент при роботі з анімацією в Flash. На ній відображається інформація про шари, про те, які кадри є ключовими, а які генерує Flash. За допомогою часової шкали можна зрозуміти, які кадри містять дії чи мітки. Вона дозволяє переміщувати ключові кадри і цілі шматки анімації. Елементами часової шкали є:

- · маркер, що вказує на поточний кадр, який відображається у вікні;
- · шари – ліворуч знаходиться перелік шарів, під яким існують кнопки, що дозволяють додавати і видаляти шари. Кожен шар можна зробити невидимим і заборонити його для редагування;

- · шкала кадрів – поле, де ви можете додавати і видаляти прості і ключові кадри. Якщо викликати контекстне меню (натиснути на праву клавішу миші) на якому-небудь кадрі, ви побачите перелік дій, які можна зробити. На шкалі відображається інформація про кадри, що є ключовими (такі кадри позначаються чорними кружечками), що містять дії (буква “а” над кружечком) чи мітку (червоний прапорець, після якого йде назва мітки). Колір теж інформує про тип кадрів. Сірий колір – це кадри, які у точності повторюють ключовий кадр (keyframe). Синювате чи зеленувате підсвічування говорить про те, що кадри згенеровані Flash. І, нарешті, білий чи “порожній” смугастий простір говорять про те, що на цих кадрах нічого немає.

- · кнопки керування тінями – це кнопки, що дозволяють відображати сусідні кадри як би через кальку, щоб бачити різницю між попередніми і наступними кадрами. Можна задавати глибину такого відображення з обох боків від маркера.

У Flash є два особливих типи шарів: шари, що містять траєкторії руху і шари-маски. Є досить велика кількість прийомів, у яких використовуються шари, але в Flash без них просто не можна обійтися по одній важливій причині: в один момент часу для кожного об’єкта анімації потрібний окремий шар. Об’єктом анімації вважається фігура чи символ.

Анімація складається з послідовності кадрів. Кадр одного шару може бути як створеним вручну, так і згенерованим Flash. В зв’язку з тим, що сцени Flash складаються звичайно з декількох шарів, підсумкові “багатошарові” кадри можуть містити як згенеровані, так і “саморобні” шари.

Ключові кадри (keyframes) Flash не може змінювати в процесі створення анімації. Проміжні кадри між ключовими вибудовує Flash. Існує два типи проміжних кадрів – кадри, побудовані на основі зміни геометрії (shape tweening) і кадри, побудовані на зміні символів (motion tweening). Кадри також можуть бути порожніми, тобто нічого не містити.

Символи – одне з ключових понять у Flash. Символом може бути як найпростіша фігура чи їхнє об’єднання, так і ціла анімація (movie). Це дозволяє використовувати символи як могутній механізм створення абстракцій у Flash. Наприклад, символами можна зробити елементи якогось механізму, об’єднати їх в один символ, а потім створити сцену, де цей механізм буде рухатись. Існує три види символів: анімація або відеокліп (movie clip), кнопка (button) і зображення (graphic):

- · зображення являє собою символ, що складається з єдиного кадру, звідки впливає його статична назва. Якщо символ дійсно являє собою статичний об’єкт, краще зробити його зображенням;

- · кнопка в Flash є спеціально пристосованим під функції кнопки вид символу. У ньому міститься 4 кадри: Up, Over, Down, Hit, що відповідають таким станам кнопки: Up – звичайний стан кнопки; Over – коли курсор мишки знаходиться над кнопкою; Down – коли курсор знаходиться над кнопкою і натиснута клавіша миші; Hit – активна зона кнопки (не відображається).

- · анімація – це самий “повноцінний” тип символу. У ньому може бути будь-яка кількість кадрів. Символ цього типу може сприйматися як об’єкт типу Movie у ActionScript.

Символи можуть бути вкладеними поза залежністю від типу. Це є самим головним їхнім достоїнством. Наприклад, можна зробити кнопку, що почне рухатися, коли над нею буде “пролітати” курсор миші, просто помістивши в кадр Over символ-анімацію. Для керування символами використовується бібліотека (Library).

Фільм Flash може складатись з довільної кількості сцен, які можуть бути організовані в послідовність. Кожна сцена може містити необмежену кількість шарів. Порядкове розміщення шарів відображено на часовій шкалі: самий верхній шар розміщується на передньому плані робочого поля, а самий нижній – на задньому плані. Кожний шар також може бути пов’язаним з порядковим розміщенням об’єктів на ньому. На нижньому рівні шару завжди розміщуються не згруповані векторні лінії і фігури. Над ними, на верхньому рівні, знаходяться растрові зображення, текст, групи і екземпляри символів. Група являє собою один або декілька об’єктів, які були виділені і “згруповані”. Екземпляри символів можуть являти собою одну або декілька копій елемента, який знаходиться в бібліотеці.

В залежності від закладених в проект функціональних можливостей та призначення фільми Flash можуть бути дуже різноманітними і являти собою:

- · анімацію для Web-сторінок;
- · інтерактивні форми для Web-сторінок;
- · ігри в реальному часі для багатьох користувачів;
- · Web-вузли, повністю реалізовані в програмі Flash;
- · інтерактивні художні презентації;
- · Flash-фільми, що підтримують зворотний зв’язок з глядачем;
- · мультиплікації для Web;
- · мультиплікації для телевізійних передач;
- · музикальні автомати, що працюють в діалоговому режимі;
- · різноманітні екранні заставки, автономні презентації та ін.

Перед здійсненням проекту потрібно ретельно спланувати його та ознайомитись з середовищем розробки.

GIF-анімація

GIF-анімація використовує можливість GIF-формату зберігати у файлі декілька зображень. Якщо в GIF-файлі міститься декілька зображень, то вони будуть показані по черзі, як невеликий фільм. Однак на відміну від звичайного фільму, у якому швидкість відтворення визначається кількістю кадрів за секунду, у GIF-файлі зберігається ряд параметрів, що визначають, яким чином і як довго кожне зображення буде демонструватися. Крім того, зображення GIF-файлу можуть бути різного розміру і розміщені в потрібній позиції екрану незалежно від зображень інших кадрів.

Кожен GIF-файл містить таблицю індексів кольорів, яку називають індексною колірною палітрою. Вона визначає, які кольори використовуються в зображенні і який індекс відповідає кожному кольору. В залежності від способу збереження для індексу потрібно до 4 байтів даних, тому зображення з 256 кольорами має палітру розміром до 1024-х байт.

В анімованих GIF-файлах використовують два типи палітр: глобальну (Global Palette), що визначає кольори кожного зображення анімації за замовчуванням, і локальну (Local Palette), яка є унікальною для кожного окремого кадру анімації. При додаванні зображення в GIF-анімацію можна вказати, яку з палітр варто використовувати. Однак не слід прагнути до зменшення розміру файлу, використовуючи тільки глобальну палітру. Щоб забезпечити високу якість анімації, для кадрів зображень, які значно відрізняються від основних чи хоча б від попереднього кадру, варто використовувати локальну палітру.

Існує велике розмаїття програм для створення анімаційних GIF: Ulead GIF Animator, GIFFY, GIF Construction Set 32, Microsoft GIF Animator, Alchemy GIF Animator, Animated GIF Editor 95 та інші. Більш детально розглянемо особливості програми Ulead GIF Animator 5.

В Ulead GIF Animator 5 кожен кадр іменується шаром (layer), що відрізняє термінологію цієї програми від термінології програмних продуктів інших фірм. Створити послідовність кадрів можна, створюючи зображення вручну, імпортуючи послідовність файлів чи вміст цілої папки, а також використовуючи оцифровану відеопослідовність.

У вікні програми GIF Animator 5 представлені такі основні зони: рядок меню, головна панель інструментів, панель властивостей, робоча зона, панель інструментів малювання, панель менеджера об'єктів, панель кадрів.

Робоча частина містить зону поточного кадру документу, яка розділена на три вкладки:

Edit (Редактирование) – тут можна сформувати і редагувати анімацію, маніпулюючи і переміщуючи об'єкти, а також виділяти певні зони для застосування до них ефектів;

Optimize (Оптимизация) – можна працювати зі стисненою анімацією для оптимізації файлів з метою більш швидкої передачі по мережі;

Preview (Предварительный просмотр) – дозволяє попередньо переглядати результати анімації.

GIF Animator надає можливість редагування об'єктів безпосередньо в програмі. Різні інструменти дозволяють обирати специфічні зони в межах кадру або навіть в межах об'єкту, щоб керувати їх установками або добавляти нові зміни до об'єктів за допомогою інструментів Paintbrush (Кисть), Text (Текст), і Fill Tools (Заливка).

Іноколи потрібно відредагувати об'єкти в зовнішньому редакторі, наприклад, Ulead PhotoImpact, или Adobe Photoshop. Для того, щоб це зробити, необхідно зберегти файл і потім відкрити його в іншому редакторі. Замість цього можна вказати образ редактора, який буде об'єднаний з GIF-аніматором, скориставшись командою: Edit Favorite Image Editor Organize Favorite Image Editor. Потім потрібно вибрати програму з стандартної панелі, відредагувати файл в іншому редакторі і зберегти його, щоб повернутися до GIF-аніматора.

Ulead GIF Animator підтримує імпорт більше 30 розповсюджених форматів графічних файлів, зокрема таких, як GIF, JPEG, PCX, PSD, PCT, PNG, TGA, EPS, IFF, IMG, MAC, MSP, PCD і BMP, та забезпечує широку підтримку різних відеоформатів; прямиий імпорт і перетворення AVI-файлів, файлів AutoDesk animation з розширеннями .FLI, .FLC, .FLX, файлів Quick Time Movie (.QT, .MOV).

Команда File Add Video (Файл Додати відео) відкриває однойменне вікно, у якому можна вказати файл-джерело, виконати його перегляд і одержати повну інформацію про імпортований файл, для чого варто клацнути на кнопці Info (Інформація), після чого відкривається вікно властивостей відеофайлу. При імпорті можна вибрати ділянку відеопослідовності і задати початковий і кінцевий кадри.

Вибір кнопки Animation Wizard (Мастер анімації) у вікні Startup Wizard (Мастер запуску) чи команди File Animation Wizard (Файл Мастер анімації) запускає Майстер анімації, у трьох послідовних вікнах якого вводиться необхідна інформація. У першому вікні за допомогою кнопок Add Image (Додати картинку) і Add Video (Додати відео) можна додати в анімацію файли, що містять малюнки чи відеопослідовності. В другому вікні вказується тип джерела, що містить текст і графіку чи фотографічне зображення. У третьому вікні задається число кадрів у секунду і відповідно час затримки в сотих частках секунди. Результатом роботи Майстра буде створення анімації з послідовності кадрів з локальними палітрами для кожного з них.

Існує кілька способів зберегти анімацію. Найбільш очевидним способом є створення GIF-файлу, однак при цьому варто враховувати можливості його оптимізації. Крім того, анімацію можна зберегти у файлі, що виконується, для якого не потрібно програми-програвача, а також в одному з форматів відео.

GIF Animator дозволяє також експортувати частину кадрів анімації зі створенням нового анімованого GIF-файлу або групи окремих GIF-файлів, що містять по одному кадру.

GIF Animator дозволяє легко і просто створити HTML-код для GIF-файлу, який потім може бути розміщений на відповідній Web-сторінці. При цьому потрібно, щоб GIF-файл знаходився в тій же папці, що і Web-сторінка.

GIF-анімація, що вставляється, може містити також гіперпосилання на задану URL-адресу.

Існують три основних способи істотного скорочення розміру анімованого GIF-файлу: скорочення колірної палітри, зменшення кількості кадрів (шарів), оптимізація окремих кадрів анімації. При цьому варто пам'ятати, що головною метою є одержання якісної анімації для Web-сторінки, а не файлу мінімального розміру. Тому варто шукати компроміс між якістю анімації і розміром файлу для його швидкої передачі по мережі. Найкращим способом для скорочення колірної палітри є використання глобальної палітри Global Palette для завдання кольору окремих кадрів. Якщо якісь кадри незначно розрізняються за колірною гамою, то немає необхідності створювати для них локальну палітру Local Palette, а, використовуючи вкладку General (Общие) вікна Preferences (Установки), додати ці кольори в Global Palette. Для кадрів, що мають істотні розходження в кольорі, створюють свої локальні палітри Local Palette, але видаляють з них кольори, спільні з кольорами глобальної палітри Global Palette.

Іншим способом скоротити розмір файлу є видалення окремих кадрів, які не є необхідними, якщо це не приводить до значного спотворення анімації при її

відтворенні. Також варто звернути увагу на розмір кадрів. Якщо анімований об'єкт займає малу частину вікна, то немає потреби зберігати для нього кадри того ж розміру, що і фон. Тому для кадрів з об'єктом можна задати менший розмір, вказавши його положення на логічному екрані, що істотно скоротить загальний розмір GIF-файлу.

Найбільш ефективним способом скорочення розміру файлу є оптимізація.

При оптимізації GIF-анімації здійснюється видалення надлишкових кольорів з палітри, які не використовуються в зображеннях, з'являються тільки один раз або близькі до кольорів, які часто використовуються. Також здійснюється зміна Global Palette додаванням до неї кольорів локальних палітр, що дозволяє скоротити кількість використовуваних локальних палітр, та видалення повторюваних частин зображень у кадрах.

Команди меню Video F/X (Видеоэффекты) дозволяють легко додати до анімації різні ефекти, але при цьому варто врахувати, що їхнє використання також збільшує розмір GIF-файлу.

Процес створення анімації з послідовності файлів включає:

- · запуск GIF-аніматора і у вікні Startup Wizard (Мастер запуску) вибір самостійного створення анімації (Blank animation);
- · вибір командою File\Add Images (Файл\Добавить картинку) відразу кілька файлів (це можна здійснити також шляхом перетягування файлів з вікна Провідника Windows на робочу область вікна GIF-аніматора);
- · перехід на вкладку Preview (Предварительный просмотр) для перегляду анімації;
- · перехід на вкладку Optimize (Оптимизация) і здійснення оптимізації файлу.
- Процес створення анімації з відеофайлу включає:
 - · запуск GIF-аніматора і у вікні Startup Wizard (Мастер запуску) вибір Open an existing video file (Создание анимации из видеофайла);
 - · для виділеного файлу у вікні Add video (Добавить видео) у полях Subject (Тема) і Description (Описание) вказуються короткі відомості про файл;
 - · відкриття вікна Duration (Продолжительность) і при послідовному перегляді позначення мітками початку і кінця відповідних кадрів (якщо вставляється відеофайл цілком, то вікно Duration (Продолжительность) можна не відкривати);
 - · клацання на кнопці Open (Открыть) у вікні Add video (Добавить видео), при цьому всі виділені кадри відеофайлу послідовно розташовуються на панелі кадрів;
 - · додавання до анімації ефектів;
 - · збереження результату.

Висока якість і малий розмір вихідного файлу зробили в даний час програму Ulead GIF Animator 5.0 однією з самих популярних у Web- дизайнерів.

Java

Мова програмування *Java* зародилася в 1991р. в лабораторіях компанії Sun Microsystems inc. Період її становлення співпав за часом з розквітом міжнародної інформаційної служби World Wide Web. Ця служба висунула Java на передній край програмування, і Java, в свою чергу, сильно вплинула і навіть змінила обличчя Internet, розширивши спектр об'єктів, які можуть розповсюджуватись у кіберпросторі.

Програми нової форми, *аплети*, - завантажуються з віддаленого сервера і можуть запускатися динамічно, тобто без участі користувача. До появи Java такий підхід був неприпустимий з міркувань безпеки та переносимості. В архітектурі апаратів зроблено ряд штучних обмежень, які роблять їх цілком безпечними. Перш за все, Java є інтерпретованою мовою і простір ресурсів Java-програми обмежений так званою віртуальною Java-машиною (*VJM*), яка може контролювати поведінку програми і захищати систему від побічних ефектів, які можуть виникати з вини аплета. Крім того, в мові Java є додаткові обмеження, які не дозволяють аpletу стати «троянським конем». Зокрема, Java-апет не може отримати доступ до локального жорсткого диску. При такій спробі генерується виключна ситуація.

Оскільки аплети Java інтерпретуються, а не компілюються, то їх виконання на різних платформах значно полегшується. В цьому випадку достатньо створити для кожної платформи виконуючу Java-систему. Якщо існує така система для даної операційної системи, то будь-яка Java-програма може виконуватись в даному середовищі без додаткової компіляції на цій платформі. Проте Java не є інтерпретованою мовою в чистому розумінні. Програма на Java компілюється. Результатом роботи компілятора Java є байткод (*bytecode*). *Байткод* - це оптимізований набір команд, призначений для виконання уявним пристроєм - віртуальною Java-машиною. В такий спосіб витрати на інтерпретацію зводяться до мінімуму, оскільки байткод вже є оптимізованим, і досягається досить висока продуктивність Java-програм.

Наведені вище особливості дають підставу розглядати Java не як ще одну мову програмування, а як окрему інформаційну технологію. Таким чином, *інтерпретація* - це найлегший шлях до перенесення програм, реалізований в Java-технології. Незважаючи на те, що мова Java була розроблена в розрахунок на інтерпретацію, технічно немає нічого такого, що б перешкоджало компіляції байткоду в виконуваний код. До байткоду, який пересилається по мережі, застосовується динамічна компіляція, але це ніяк не впливає на переносимість та безпеку, оскільки роботу програми все ще контролює виконуюча система. Такий підхід застосовано в багатьох виконуючих системах Java, що забезпечує продуктивність на рівні оптимізованого коду C++.

Мова Java є однією з наймолодших в сімействі мов програмування і була розроблена з розрахунок на те, щоб професійний програміст міг легко її опанувати та ефективно використовувати. За основу Java взятий синтаксис C++ - безсумнівно однієї з найбільш популярних мов програмування сучасності. Проте, Java - це цілком самостійна мова програмування, і при її створенні не йшлося про будь-яку сумісність з C++. Тому деякі механізми реалізовані в Java інакше, а деякі зовсім відсутні. Ідеологічно ж Java побудована дещо інакше ніж C++. Розробники Java ґрунтувалися на досвіді розробки програм на C++ і прагнули позбутися можливостей, які зарекомендували себе непевними. Так, в Java відсутня перегрузка операторів а також автоматичне приведення несумісних типів - конструкції, які при неуважному використанні є джерелом важких для виявлення помилок. Взагалі, інтерфейси Java більш прості та прозорі для розуміння. Написати на Java-програму з графічним інтерфейсом значно легше. Звичайно, простота інтерфейсів компенсується меншою гнучкістю, бібліотека Java не така багата, як стандартні бібліотеки C/C++. Але згадаймо, що Java задуманий для використання на різних платформах і тому реалізує в собі найбільш стандартні можливості задля легшої адаптації під конкретне середовище.

Від C++ Java успадкувала потужний механізм об'єктно-орієнтованого програмування. Оскільки Java розроблювалась «на пустому місці», тобто не було потреби забезпечувати сумісність з попередніми версіями, розробники мали повну свободу мислення. В результаті був сформований ясний і прагматичний підхід до об'єктів. Вільно переймаючи ідеї, які реалізовувалися протягом останніх десятиріч, мові Java вдалося знайти рівновагу між парадигмою «все є об'єктом» і прагматичним підходом. Об'єктна модель Java проста і легко розширюється, в той же час прості типи, такі, як цілі, зберігаються як дані, що не є об'єктами, і це дозволяє значно підвищити швидкість при їх обробці.

Багатоплатформеність середовища Web висуває надзвичайно високі вимоги до надійності програм. Як наслідок, при розробці Java пріоритет був відданий можливості створення стійких до помилок програм. Java звільняє програміста від хвилювань з приводу багатьох поширених причин, які викликають помилки програмування. Як вже згадувалося, Java є строго типізованою мовою програмування. Крім того, виконуюча система Java бере на себе «прибирання сміття», тобто автоматично звільняє пам'ять, яка була розподілена динамічно. Зазвичай, це дещо знижує ефективність коду, але запобігає типових помилок, коли програміст забуває звільнити виділену пам'ять, або, навпаки, звільняє пам'ять, яка ще використовується. Java підтримує об'єктно-орієнтовану обробку виключних ситуацій подібно до C++. Але на відміну від C++ в Java обробка виключних ситуацій є обов'язковою. Тобто неможливо скомпілювати програму, яка відкриває файл, не обробивши можливі помилки типу «файл не знайдено», які виникають при цьому. Добре написана Java-програма може сама обробляти всі помилки часу виконання.

Java розроблялася з орієнтацією на вимоги до створення інтерактивних програм, які працюють з мережею. З цією метою Java підтримує багатопоточне програмування, яке дозволяє легко розробляти програми, що виконують багато процесів одночасно. Виконання Java-програми засноване на елегантному, але в той же час високоорганізованому рішенні багатопроесової синхронізації, що дозволяє створювати високоефективні інтерактивні системи.

Основним питанням для розробників Java стало питання довготривалості та можливості перенесення. Одна з головних проблем, із якою зустрілися програмісти, полягала в відсутності гарантій того, що написана сьогодні програма завтра працюватиме з тим же успіхом, причому на тій самій машині. Оновлення операційної системи, модернізація процесора та зміна об'єму оперативної пам'яті можуть призвести до збою програми. Розробники Java, прагнули змінити цю ситуацію і прийняли декілька важких рішень відносно мови Java та процесу виконання Java-програми. Їх мета полягала в тому, щоб «одного разу написане працювало всюди, в любий час і завжди». Внаслідок цього Java є системою, яка легко розширюється за рахунок створення нових стандартних класів та бібліотек.

Як вже згадувалось, Java дозволяє створювати незалежні від платформи програми шляхом компіляції в проміжне представлення, яке називається байткодом. Незважаючи на те, що Java є інтерпретованою мовою, генерація байткодів була ретельно оптимізована в такий спосіб, щоб одержуваний байткод можна було легко перекладати в машинний код, який працює з дуже високою продуктивністю. Виконуючі системи такого роду не втрачають жодних переваг коду, що переноситься.

Мова Java призначена для створення програм, які працюють в розподіленому середовищі Internet на базі протоколів TCP/IP. Насправді доступ до ресурсів за

допомогою URL відрізняється від доступу до файлу. Крім того, в Java наявний засіб передачі повідомлень у межах внутрішнього адресного простору. Це дозволяє забезпечити віддалене виконання процедур. Ці інтерфейси включені у пакет RMI (remote method invocation). Цей засіб привносить високий рівень абстракції в програмування для середовища клієнт/сервер.

Java-програми несуть у собі значний обсяг інформації про типи часу виконання (run-time type information), яка використовується для дозволу доступу до об'єктів під час роботи програми. Це дозволяє забезпечити безпечну та оптимальну динамічну компоновку. У такий спосіб досягається захищеність середовища виконання аплетів.

Зазначена вище простота програмування на Java є причиною того, що розробки на Java коштують дешевше аналогічних на більш потужних мовах програмування. Цьому ж сприяє і можливість перенесення програм на Java, оскільки ліквідуються витрати пов'язані з адаптацією програми на конкретній платформі. До того ж інтегровані програми-оболонки для розробки Java програм коштують набагато дешевше (70-100\$) ніж аналогічні продукти C++, Delphi (~1000\$). А набір інструментарію для пакетної компіляції Java-програм JDK (Java Development Kit) є взагалі freeware. Тому платформу Java можна рекомендувати як ідеальну для створення некомерційних програмних продуктів, зокрема для галузі освіти.

Програми на Java можуть знайти різне застосування в навчальному процесі: інтерактивні навчаючі програми (HTML в поєднанні з Java), програми-тести і особливо ділові ігри. Додаткові переваги можна отримати, якщо писати ці програми у вигляді аплетів, які ініціалізуються з Web-сервера внутрішньої мережі Intranet. У такий спосіб можна уникнути інсталяції програми на багатьох комп'ютерах - користувач просто запускає Web-браузер і завантажує потрібну сторінку. Для тестових програм, написаних на Java з використанням архітектури клієнт/сервер можна підвищити ступінь конфіденційності. База даних тестових запитань знаходиться на сервері в каталозі з обмеженим доступом. Коли користувач завантажує аплет, він автоматично підключається до програми-сервера, яка виконується на сервері і може видавати запитання з бази даних у відповідь на запит користувача. У такий спосіб унеможливується викрадення бази даних, за умови відсутності фізичного доступу до серверу у користувачів.

Окреме питання - навчальні ділові ігри. Під такою грою розумітимемо гру, за участю кількох користувачів, в якій мається на увазі динамічний обмін інформацією між ними. Система безпеки Java накладає обмеження, внаслідок якого аплет може встановлювати з'єднання лише з хостом, з якого він був завантажений і ні з яким більше. Але це обмеження легко обходиться: на сервері виконується програма-сервер, з якою з'єднуються усі клієнти і через яку здійснюється обмін інформацією. Таким чином така програма повинна мати архітектуру клієнт/сервер.

В якості прикладу наводимо спрощену chat-програму. В ній реалізовані базові засоби для створення ділової гри. Можливе легке доопрацювання цієї програми в бік розширення кількості протокольних команд.

Програма працює наступним чином: користувач запускає аплет із сервера, вводить свій ідентифікатор і бачить перелік ідентифікаторів користувачів, які приєднані до сервера. Після чого користувач має змогу обмінюватись повідомленнями з іншими. Користувач одержує повідомлення в разі під'єднання або від'єднання користувачів.

Програма складатися з 4 класів: `Server`, `ClientConnection`, `Client` та `ServerConnection`.
Перші 2 класи відносяться до серверної частини, другі 2 - до клієнтської.

Клас `Server`:

```
import java.net.*;
import java.io.*;
import java.util.*;

public class Server implements Runnable
{ private int port = 6564;
  private Hashtable idcon = new Hashtable();
  private int id = 0;
  static final String CRLF = "\r\n";
  synchronized void addConnection(Socket s) {
    ClientConnection con = new ClientConnection(this, s, id);
    id++;
  }
  synchronized void set(String the_id, ClientConnection con)
  { idcon.remove(the_id) ;
    con.setBusy(false);
    Enumeration e = idcon.keys();
    while (e.hasMoreElements())
    { String id = (String)e.nextElement();
      ClientConnection other = (ClientConnection) idcon.get(id);
      if (!other.isBusy())
        con.write("add " + other + CRLF);
    }
    idcon.put(the_id, con);
    broadcast(the_id, "add " + con);
  }
  synchronized void sendto(String dest, String body)
  { ClientConnection con = (ClientConnection)idcon.get(dest);
    if (con != null)
    { con.write(body + CRLF);
    }
  }
  synchronized void broadcast(String exclude, String body)
  { Enumeration e = idcon.keys();
    while (e.hasMoreElements())
    { String id = (String)e.nextElement();
      if (!exclude.equals(id))
      { ClientConnection con = (ClientConnection) idcon.get(id);
        con.write(body + CRLF);
      }
    }
  }
  synchronized void delete(String the_id)
  { broadcast(the_id, "delete " + the_id);
  }
  synchronized void kill(ClientConnection c)
  { if (idcon.remove(c.getId()) == c)
    { delete(c.getId());
    }
  }
  public void run()
  { try
    { ServerSocket acceptSocket = new ServerSocket(port);
      System.out.println("Server listening on port " + port);
      while (true)
```

```

        { Socket s = acceptSocket.accept();
          addConnection(s);
        }
    }
    catch (IOException e)
    { System.out.println("accept loop IOException: " + e);
    }
}
public static void main(String args[])
{ new Thread(new Server()).start();
  try
  { Thread.currentThread().join();
  }
  catch (InterruptedException e)
  { }
}
}
}

```

Цей невеликий клас реалізує програму-сервер. Точка входу програми - функція *main*. В програмі створюється головний потік, в якому створюється об'єкт стандартного класу *ServerSocket*. Цей об'єкт приєднується до певного порту і в циклі здійснюється перевірка на підключення клієнта до порта. В разі такого підключення створюється об'єкт типу *ClientConnection*, в якому реалізований потік по опитуванню сокета, до якого підключений клієнт, та обробка протокольних команд. Клас *Server* є універсальним і нічого не знає про повідомлення, якими обмінюються гравці (клієнти). В ньому реалізоване лише приєднання/від'єднання клієнтів, надсилання строки певному клієнту та функція *broadcast* яка, як зрозуміло з назви, надсилає повідомлення усім зареєстрованим клієнтам.

```

Клас ClientConnection:
import java.net.*;
import java.io.*;
import java.util.*;

class ClientConnection implements Runnable
{ private Socket sock;
  private DataInputStream in;
  private OutputStream out;
  private String host;
  private Server server;
  private static final int bufsize = 8192;
  private byte buffer[] = new byte[bufsize];
  private static final String CRLF = "\r\n";
  private String name = null;
  private String id;
  private boolean busy = false;
  public ClientConnection(Server srv, Socket s, int i)
  { try
  { server = srv;
    sock = s;
    in = new DataInputStream(s.getInputStream());
    out = s.getOutputStream();
    host = s.getInetAddress().getHostName();
    id = "" + i;
    // tell the new one who it is...
    write("id " + id + CRLF);
    new Thread(this).start();
  }
  }
}

```

```

    }
    catch (IOException e)
    { System.out.println("failed ClientConnection " + e);
    }
}

public String toString()
{ return id + " " + host + " " + name;
}

public String getHost()
{ return host;
}

public String getId()
{ return id;
}

public boolean isBusy()
{ return busy;
}

public void setBusy(boolean b)
{ busy = b;
}

public void close()
{ server.kill(this);
  try
  { sock.close(); // closes in and out too.
  }
  catch (IOException e)
  { }
}

public void write(String s)
{ byte buf[] = new byte[s.length()];
  s.getBytes(0, buf.length, buf, 0);
  try
  { out.write(buf, 0, buf.length);
  }
  catch (IOException e)
  {close();}
}

private String readline()
{ try
  { return in.readLine();}
  catch (IOException e)
  { return null;
  }
}

static private final int NAME = 1;
static private final int QUIT = 2;
static private final int TO = 3;
static private final int DELETE = 4;
static private Hashtable keys = new Hashtable();
static private String keystrings[] = {"", "name", "quit", "to",
                                       "delete"};

static
{ for (int i = 0; i < keystrings.length; i++)
  keys.put(keystrings[i], new Integer(i));
}

```

```

private int lookup(String s)
{   Integer i = (Integer) keys.get(s);
    return i == null ? -1 : i.intValue();
}

public void run()
{   String s;
    StringTokenizer st;
    out:
    while ((s = readline()) != null)
    {   st = new StringTokenizer(s);
        String keyword = st.nextToken();
        switch (lookup(keyword)) {
        default:
            System.out.println("bogus keyword: " + keyword + "\r");
            break;
        case NAME:
            name = st.nextToken() +
                (st.hasMoreTokens() ? " " + st.nextToken(CRLF) : "");
            System.out.println("[ " + new Date() + " ] " + this + "\r");
            server.set(id, this);
            break;
        case QUIT:
            break out;
        case TO:
            String dest = st.nextToken();
            String body = st.nextToken(CRLF);
            server.sendto(dest, body);
            break;
        case DELETE:
            busy = true;
            server.delete(id);
            break;
        }
    }
    close();
}
}

```

Клас *ClientConnection* реалізує обмін інформацією з конкретним клієнтом. В ньому зберігається об'єкт класу *Socket*, до якого приєднаний цей клієнт, та створюється потік, в якому здійснюється періодична спроба читати з сокета. Якщо ця спроба завершується успіхом, тобто надійшло повідомлення від клієнта, це повідомлення аналізується і здійснюється відповідна реакція. Для здійснення цієї реакції викликаються методи класу *Server*, об'єкт якого передається класу *ClientConnection* в якості параметра конструктора. Крім того в цьому класі є метод, який здійснює безпосереднє надсилання інформації до клієнта. Коли серверу потрібно надіслати інформацію конкретному клієнтові, він викликає цей метод.

Клас *Client*:

```

import java.util.*;
import java.io.*;
import java.net.*;
import java.awt.*;
import java.applet.*;
public
class Client extends Applet

```

```

{ private ServerConnection server;
  private String serverName;
  private boolean single = false;
  private boolean seen_pass = false;
  private boolean name_set = false;
  private String name;
  private String others_name;
  private Panel topPanel;
  private Label prompt;
  private TextField namefield;
  private Button done;
  private TextField chatfield;
  private List idList;
  private TextArea dialogArea;
  public
  void init()
  { setLayout( new BorderLayout() );
    serverName = getCodeBase().getHost();
    if (serverName.equals(""))
      serverName = "localhost";
    prompt = new Label("Enter id:");
    namefield = new TextField(30);
    topPanel = new Panel();
    topPanel.setBackground(new Color(255, 255, 200));
    topPanel.add(prompt);
    topPanel.add(namefield);
    add("North", topPanel);
    idList = new List(10, false);
    add("West", idList );
    dialogArea = new TextArea();
    dialogArea.setEditable( false );
    add("Center", dialogArea );
  }
  public void start()
  { try
    { showStatus("Connecting to " + serverName);
      server = new ServerConnection(this,serverName);
      server.start();
      showStatus("Connected: " + serverName);
    }
    catch (Exception e)
    { single = true;}
  }
  public void stop()
  { if (!single)
    server.quit();
  }
  void add(String id, String hostname, String name)
  { delete(id); // in case it is already there.
    idList.addItem("(" + id + ") " + name + "@" + hostname);
  }
  void delete(String id)
  { for (int i = 0; i < idList.countItems(); i++)
    { String s = idList.getItem(i);
      s = s.substring(s.indexOf("(") + 1, s.indexOf(")"));
      if (s.equals(id))
        { idList.delItem(i);

```

```

        break;
    }
}
if (idList.countItems() == 0)
    showStatus("Wait for other players to arrive.");
}
private String getName(String id)
{ for (int i = 0; i < idList.countItems(); i++)
  { String s = idList.getItem(i);
    String id1 = s.substring(s.indexOf("(") + 1, s.indexOf(")"));
    if (id1.equals(id))
    { return s.substring(s.indexOf(" ") + 3, s.indexOf("@"));
    }
  }
  return null;
}
void chat(String id, String s)
{ dialogArea.appendText(id + ": " + s + "\n");//append
  showStatus(id + ": " + s);
}
void quit(String id)
{ showStatus(id + " just quit.");
  delete(id);
}
private void nameEntered(String s)
{ if (s.equals(""))
  return;
  System.out.println(s);
  name = s;
  if( !single )
  { server.setName(name);
    showStatus("Wait for other players to arrive.");
  }
  prompt.setText("You say:");
  name_set = true;
}
public boolean action(Event evt, Object arg)
{ System.out.println("a");
  if(evt.id == Event.ACTION_EVENT)
  if (evt.target == namefield)
  { if( name_set )
    { dialogArea.appendText(name + ": " + namefield.getText()+"\n");
      if(!single)
        server.sendTo((String)arg, idList.getSelectedItem());
    }
    else nameEntered((String)arg);
    namefield.setText("");
  }
  return true;
}
}
}

```

Цей клас є розширенням класу *Applet*, отже є аплетом. В цьому класі створюється об'єкт класу *ServerConnection*, через який здійснюється обмін інформацією з сервером. Графічний інтерфейс цього класу представляє три області: поле вводу, в яке спочатку вводиться ім'я користувача, а потім -

повідомлення, адресовані іншим клієнтам, список клієнтів, в якому треба обирати адресу, та текстова область в якій відбивається текст діалога. Метод `action` реалізує обробку події вводу тексту в область вводу. Методи `add`, `delete`, `chat`, `nameEntered`, `quit` є реакцією на наступні події: підключення нового клієнта, зникнення клієнта, надходження повідомлення від клієнта, ідентифікація та вихід. Ці методи визиваються класом `ServerConnection` при надходженні відповідних повідомлень, або ж всередині аплету, якщо джерелом події є сам клієнт.

```

Клас ServerConnection:
import java.io.*;
import java.net.*;
import java.util.*;
class ServerConnection implements Runnable
{
    void sendTo(String s, String id)
    { if( id!= null )
      out.println("to "+id+" "+s);
    }
    private static final int port = 6564;
    private static final String CRLF = "\r\n";
    private DataInputStream in;
    private PrintStream out;
    private String id, toid = null;
    private Client client;
    public ServerConnection(Scrabblet sc, String site) throws IOException
    { Socket server = new Socket(site, port);
      in = new DataInputStream(server.getInputStream());
      out = new PrintStream(server.getOutputStream(), true);
    }
    public ServerConnection( Client c,String site) throws IOException
    { client = c;
      Socket server = new Socket(site, port);
      in = new DataInputStream(server.getInputStream());
      out = new PrintStream(server.getOutputStream(), true);
    }
    private String readline()
    { try
      {return in.readLine();}
      catch (IOException e)
      { return null;
      }
    }
    void setName(String s)
    { out.println("name " + s);}
    void delete()
    { out.println("delete " + id);}
    void setTo(String to)
    { toid = to;}
    void send(String s)
    { if (toid != null)
      out.println("to " + toid + " " + s);
    }
    void chat(String s)
    { send("chat " + id + " " + s);}
    void quit()

```

```

{ send("quit " + id); // tell other player
  out.println("quit");
  stop();
}
private Thread t;
void start()
{ t = new Thread(this);
  t.start();
}
void stop()
{ t.stop();}
private static final int ID = 1;
private static final int ADD = 2;
private static final int DELETE = 3;
private static final int CHAT = 4;
private static final int QUIT = 5;
private static Hashtable keys = new Hashtable();
private static String keystrings[] = {"", "id", "add", "delete", "chat",
                                       "quit" };

static
{ for (int i = 0; i < keystrings.length; i++)
  keys.put(keystrings[i], new Integer(i));
}
private int lookup(String s)
{ Integer i = (Integer) keys.get(s);
  return i == null ? -1 : i.intValue();
}
public void run()
{ String s;
  StringTokenizer st;
  while ((s = readline()) != null)
  { st = new StringTokenizer(s);
    String keyword = st.nextToken();
    switch (lookup(keyword))
    { default:
      System.out.println("bogus keyword: " + keyword + "\r");
      break;
      case ID:
        id = st.nextToken();
        break;
      case ADD:
        { String id = st.nextToken();
          String hostname = st.nextToken();
          String name = st.nextToken(CRLF);
          client.add(id, hostname, name);}
        break;
      { case CHAT:
        String from = st.nextToken();
        client.chat(from, st.nextToken(CRLF));}
        break;
      case QUIT:
        { String from = st.nextToken();
          client.quit(from);}
        break;
    }
  }
}
}
}

```

Цей клас є дзеркальним відображенням класу `ClientConnection`. Але він більш тісно переплетений з класом `Client`. В ньому реалізований потік, що опитує сокет, через який клієнта приєднано до сервера, здійснює аналіз повідомлень, що надходять та викликає потрібні методи класу `Client`. Крім того в ньому є методи, які викликаються класом `Client` для відсилання інформації на сервер.

Ця програма легко може бути модифікована в напрямку розширення кількості протокольних команд та ускладнення клієнта за рахунок додавання специфічних реакцій на нові команди. В класі `ClientConnection` може бути ускладнена реакція на під'єднання клієнта. Наприклад, клієнту може видаватися інформація про початкові умови гри. Таким чином наведена програма може служити скелетом для створення більш складних. Можна дещо урізати класи `Client` та `ServerConnection`, залишивши в них лише базові функції з тим, щоб потім розширювати їх шляхом механізму об'єктного успадкування.

Java технологія є дуже перспективною для застосування в розробках програм як комерційного, так і не комерційного спрямування. Обмеженість інструментарію Java не проявляється в проектах невеликого обсягу і з лихвою компенсується простотою програмування розподілених програм, які працюють з мережею Internet/Intranet .

РОЗДІЛ 11

ЕКСПЕРТНІ НАВЧАЛЬНІ СИСТЕМИ

Експертна система - це програма, що поводиться подібно експерту в деякій, звичайно вузькій прикладній області. Типові застосування експертних систем містять у собі такі задачі, як діагностика, локалізація похибок, інтерпретація результатів аналізу, оцінка тих чи інших характеристик.

Експертні системи повинні вирішувати задачі, що вимагають для свого рішення експертних знань у деякій конкретній області. У тій чи іншій формі експертні системи повинні мати ці знання. Тому їх також називають системами, заснованими на знаннях (однак не всяку систему, засновану на знаннях, можна розглядати як експертну).

Експертна система повинна також уміти певним чином пояснювати свою поведінку і свої рішення користувачу, так само, як це робить експерт - людина. Це особливо необхідно в областях, для яких характерна невизначеність та неточність інформації. У цих випадках здатність до пояснення потрібна для того, щоб підвищити ступінь довіри користувача до порад системи, а також для того, щоб дати можливість користувачу знайти можливий дефект у "міркуваннях" системи. У зв'язку з цим в експертних системах як правило передбачається дружня взаємодія з користувачем, що робить для користувача процес "міркування" системи "прозорим".

Призначення та основні властивості експертних систем

Експертна система (ЕС) - це програмний засіб, що використовує експертні знання для забезпечення високоєфективного рішення неформалізованих задач у вузькій предметній області.

Неформалізованим задачам властиво те, що вони:

- не можуть бути задані в числовій формі;
- їх висновки не можна виразити в термінах точно визначеної цільової функції;
- для них не існує алгоритмічного розв'язку задачі;
- якщо алгоритмічний розв'язок є, то його не можна використовувати через обмеженість ресурсів (за часом, за обсягом пам'яті тощо).

Основа ЕС складає база знань (БЗ) про предметну область, що накопичується в процесі побудови й експлуатації ЕС. Нагромадження й організація знань - найважливіша властивість усіх ЕС.

Знання є явними і доступними, що відрізняє ЕС від традиційних програм, і визначає їхні основні властивості (рис. 30), такі, як:

1) *Застосування* для рішення проблем *високоякісного досвіду*, що представляє рівень мислення найбільш кваліфікованих експертів у даній області, що веде до рішень творчих, точних і ефективних.

2) *Наявність прогностичних можливостей*, при яких ЕС видає відповіді не тільки для конкретної ситуації, але і показує, як змінюються ці відповіді в нових ситуаціях, з можливістю докладного пояснення яким чином нова ситуація привела до змін.

3) *Забезпечення такої нової якості, як інституціональна пам'ять*, за рахунок бази знань, що входить до складу ЕС, розробленої в ході взаємодій з фахівцями організації, яка являє собою поточну політику цієї групи людей. Цей набір знань стає зводом кваліфікованих думок і постійно оновлюваним довідником найкращих стратегій і методів, використовуваних персоналом. Провідні спеціалісти ідуть, але їхній досвід залишається.

4) *Можливість* використання ЕС для навчання і тренування керівників, забезпечуючи нових службовців великим багажем досвіду і стратегій, по яких можна вивчати політику, що рекомендується, і методи.

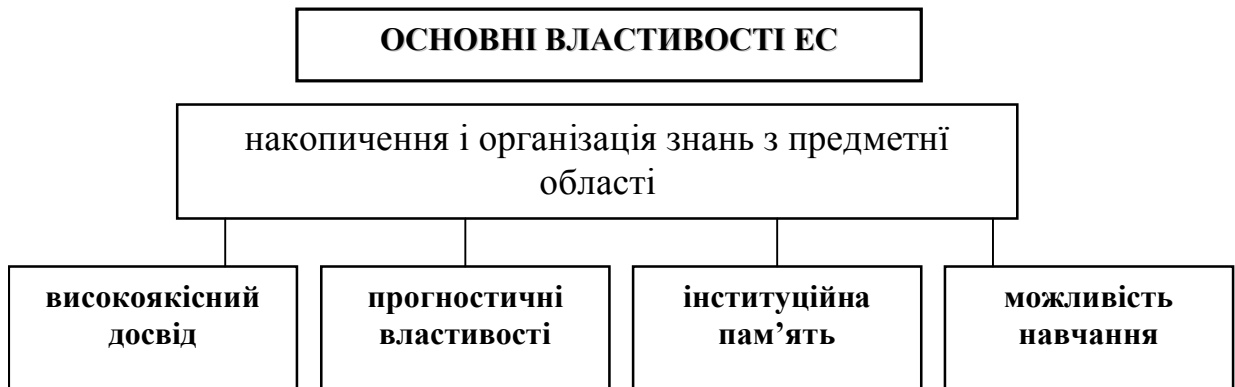


Рис.30

Склад і взаємодія учасників побудови й експлуатації експертних систем

До числа основних учасників варто віднести саму *експертну систему*, *експертів*, *інженерів знань*, *засоби побудови ЕС* і *користувачів*.

Експертна система – це програмний засіб. Але це не просто програмний засіб, а система, що містить базу знань, ролізовач проблеми і *компонент підтримки*. Останній з них допомагає користувачеві взаємодіяти з основною програмою.

Експерт – це людина, здатна ясно виражати свої думки, яка користується репутацією фахівця, що вміє знаходити правильні рішення проблем у конкретній предметній області. Експерт використовує свої прийоми і “хитрощі”, щоб зробити пошук рішення більш ефективним, а ЕС моделює всі його стратегії.

Інженер знань – людина, як правило, що має пізнання в інформатиці і штучному інтелекті і яка знає, як треба будувати ЕС та користуватися всіма її можливостями. Інженер знань опитує експертів, організує знання, вирішує, яким чином вони повинні бути представлені в ЕС, і може допомогти програмісту в написанні програм.

Засіб побудови ЕС – це програмний засіб, який використовується інженером знань або програмістом для побудови ЕС. Цей інструмент відрізняється від звичайних мов програмування тим, що забезпечує зручні способи представлення складних високорівневих понять.

Користувач – це людина, що використовує вже побудовану ЕС. Так, користувачем може бути економіст, що використовує її для кваліфікації конкретного випадку, студент, якому ЕС допомагає вивчати інформатику і т.д. Термін користувач трохи неоднозначний. Зазвичай він позначає *кінцевого користувача*. Однак, користувачем може бути:

- творець інструмента, який налагоджує засіб побудови ЕС;
- інженер знань, що уточнює існуючі в ЕС знання;
- експерт, що додає в систему нові знання;
- клерк, що заносить у систему поточну інформацію.

Важливо розрізнити інструмент, що використовується для побудови ЕС, і саму ЕС. Інструмент побудови ЕС включає як мову, яка використовується для доступу до

знань, що утримується в системі, так і підтримуючі засоби – програми, що допомагають користувачам взаємодіяти з компонентом експертної системи, що вирішує проблему.

Переваги використання експертних систем

Перевагами і позитивними якостями штучної (електронної) компетенції є:

1) *Її сталість*. Людська компетенція слабшає згодом. Перерва в діяльності людини-експерта може серйозно відбитися на його професійних якостях.

2) *Легкість передачі або відтворення*. Передача знань від однієї людини іншій – довгий і дорогий процес. Передача штучної інформації – це простий процес копіювання програми або файлу даних.

3) *Стійкість і відтворюваність результатів*. Експерт-людина може приймати в тотожних ситуаціях різні рішення через емоційні фактори. Результати ЕС – стабільні.

4) *Вартість*. Експерти, особливо висококваліфіковані обходяться дуже дорого. ЕС, навпаки, порівняно недорого. Їхня розробка дорога, але вони дешеві в експлуатації.

Разом з тим, розробка ЕС не дозволяє цілком відмовитися від експерта-людини. Хоча ЕС добре справляється зі своєю роботою, проте у певних областях людська компетенція явно перевершує штучну. Однак і в цих випадках ЕС може дозволити відмовитися від послуг висококваліфікованого експерта, залишивши експерта середньої кваліфікації, використовуючи при цьому ЕС для посилення і розширення його професійних можливостей.

Особливості побудови й організації експертних систем

Основою будь-якої ЕС є сукупність знань, структурована з метою спрощення процесу ухвалення рішення. Для фахівців в області штучного інтелекту термін *знання* означає інформацію, що необхідна програмі, щоб вона поведилася "інтелектуально". Ця інформація приймає форму *фактів і правил*. Факти і правила в ЕС не завжди або правдиві, або помилкові. Іноді існує деякий ступінь непевності у вірогідності факту або точності правила. Якщо цей сумнів виражений явно, то використовують поняття "*коефіцієнт довіри*".

Коефіцієнт довіри – це число, що означає імовірність або ступінь упевненості, з яким можна вважати даний факт або правило достовірним або справедливим.

Багато правил відносяться до евристичних, спрощених, тобто таких, що обмежують пошук рішень, звужують їх, адже значна кількість задач відноситься до класу важких, не до кінця зрозумілих, таких, що не піддаються детальному формалізованому аналізу та застосуванню відомих алгоритмів, які б гарантувати оптимальність, точність, надійність. На відміну від останнього евристика допомагає в багатьох випадках отримувати прийнятне рішення.

Знання в ЕС організовані так, щоб знання про предметну область відокремити від інших типів знань системи, таких як загальні знання про те, як вирішувати задачі або знання про те, як взаємодіяти з користувачем. Виділені знання про предметну область називаються *базою знань*, тоді як загальні знання про перебування рішень задач називаються *механізмом виведення*. Програмні засоби, що працюють зі знаннями, організованими таким чином, називаються *системами, заснованими на знаннях*.

База знань (БЗ) містить факти (дані) і правила (або інші представлення знань), що використовують ці факти як основу для прийняття рішень.

Механізм виведення містить:

- інтерпретатор, що визначає як застосовувати правила для виведення нових знань на основі інформації, що зберігається в БЗ;
- диспетчер, що встановлює порядок застосування цих правил.

Такі ЕС одержали назву *статичних ЕС*. Ці ЕС використовуються в тих додатках, де можна не враховувати зміни навколишнього світу за час рішення задачі. Однак існує більш високий клас додатків, де потрібно враховувати динаміку зміни навколишнього світу за час виконання додатку. Такі експертні системи одержали назву *динамічних ЕС*.

У порівнянні зі статичною ЕС у динамічну вводиться ще два компоненти:

- підсистема моделювання зовнішнього світу;
- підсистема сполучення з зовнішнім світом.

Динамічні ЕС здійснюють зв'язок з зовнішнім світом через систему контролерів і датчиків. Крім того компонента БЗ і механізму виведення істотно змінюються, щоб відбити тимчасову логіку подій, що відбуваються в реальному світі.

До розряду таких динамічних середовищ розробки ЕС відноситься сімейство програмних продуктів фірми Gensym Corp. (США). Один з таких продуктів система G2 – базовий програмний продукт, що представляє собою графічне, об'єктно-орієнтоване середовище для побудови і супроводу експертних систем реального часу, призначених для моніторингу, діагностики, оптимізації, планування і керування динамічним процесом.

Основні режими роботи експертних систем

У роботі ЕС можна виділити два основних режими: *режим придбання знань* і *режим рішення задачі* (режим консультації або режим використання). У *режимі придбання знань* спілкування з ЕС здійснює експерт (за допомогою інженера знань).

Використовуючи компонент придбання знань, експерт описує проблемну область у виді сукупності фактів і правил. Іншими словами, "наповнює" ЕС знаннями, що дозволяють їй самостійно вирішувати задачі з проблемної області.

Відзначимо, що цьому режимові, при традиційному підході до програмування, відповідають етапи алгоритмізації, програмування і налагодження, виконувани *програмістом*. Таким чином, на відміну від традиційного підходу, у випадку ЕС розробку програм здійснює не програміст, а *експерт*, що не володіє програмуванням.

У режимі *консультації* спілкування з ЕС здійснює кінцевий користувач, якого цікавить результат і (або) спосіб його одержання.

Користувач може:

- не бути фахівцем у даній предметній області, і в цьому випадку він звертається до ЕС за результатом, що не вміє одержати сам;
- бути фахівцем, і в цьому випадку він звертається до ЕС з метою прискорення одержання результату, покладаючи на ЕС рутинну роботу.

Слід зазначити, що, на відміну від традиційних програм ЕС, при рішенні задачі не тільки виконують запропоновану алгоритмом послідовність операцій, але і самі попередньо формують її.

Добре побудована ЕС має можливість самонавчатися на розв'язуваних задачах, поповнюючи автоматично свою БЗ результатами отриманих висновків і рішень.

Відмінність експертних систем від традиційних програм

Особливості ЕС, що відрізняють їх від звичайних програм, полягають у тім, що вони повинні володіти:

1. *Компетентністю*, а саме:

- досягати експертного рівня рішень (тобто в конкретній предметній області мати той же рівень професіоналізму, що й експерти-люди);
- мати активну працездатність (тобто застосовувати знання ефективно і швидко, уникаючи, як і люди, непотрібних обчислень);
- мати адекватну працездатність (тобто здатність лише поступово знижувати якість роботи з міри наближення до границь діапазону компетентності або припустимої надійності даних).

2. *Можливістю до символічних міркувань*, а саме:

- представляти знання в символічному виді;
- переформулювати символічні знання. На жаргоні штучного інтелекту символ – це рядок знаків, що відповідає змісту деякого поняття. Символи поєднують, щоб виразити відносини між ними. Коли відносини представлені в ЕС, вони називаються символічними структурами.

3. *Глибиною*, а саме:

- працювати в предметній області, що містить важкі задачі;
- використовувати складні правила (тобто використовувати або складні конструкції правил, або велику їхню кількість);

4. *Самосвідомістю*, а саме:

- досліджувати свої міркування (тобто перевіряти їхню правильність);
- пояснювати свої дії.

Існує ще одна важлива відмінність ЕС. Якщо звичайні програми розробляються так, щоб щораз породжувати правильний результат, то ЕС розроблені для того, щоб поводитися як експерти. Вони, як правило, дають правильні відповіді, але іноді, як і люди, здатні помилятися.

Традиційні програми для рішення складних задач теж можуть робити помилки, але їх дуже важко виправити, оскільки алгоритми, що лежать у їхній основі, явно в них не сформульовані. Отже, помилки нелегко знайти і виправити. ЕС, подібно людям, мають потенційну можливість учитися на своїх помилках.

Технологія розробки експертних систем

Технологія розробки ЕС, містить у собі шість етапів: *ідентифікації, концептуалізації, формалізації, виконання, тестування, досвідченої експлуатації*.

1) На етапі *ідентифікації* необхідно виконати наступні дії:

- визначення задач, що підлягають рішенню і меті розробки,
- визначення експертів і типу користувачів.

2) На етапі *концептуалізації*:

- проводиться змістовний аналіз предметної області,
- виділяються основні поняття і їхні взаємозв'язки,
- визначаються методи рішення задач.

3) На етапі *формалізації*:

- вибираються програмні засоби розробки ЕС,
- визначаються способи представлення усіх видів знань,

– формалізуються основні поняття.

4) На етапі **виконання** (найбільш важливе і трудомісткому) здійснюється наповнення експертом БЗ, при якому процес придбання знань розділяють:

- на "витяг" знань з експерта,
- на організацію знань, що забезпечує ефективну роботу ЕС,
- на представлення знань у виді, зрозумілому для ЕС.

Процес придбання знань здійснюється інженером по знаннях на основі діяльності експерта.

5) На етапі **тестування** експерт і інженер по знаннях з використанням діалогових і пояснювальних засобів перевіряють компетентність ЕС. Процес тестування продовжується доти, поки експерт не вирішить, що система досягла необхідного рівня компетентності.

6) На етапі **досвідченої експлуатації** перевіряється придатність ЕС для кінцевих користувачів. За результатами цього етапу можлива істотна модернізація ЕС.

Процес створення ЕС не зводиться до строгої послідовності цих етапів, тому що в ході розробки приходиться неодноразово повертатися на більш ранні етапи і переглядати прийняті там рішення.

У самому загальному випадку для того, щоб побудувати експертну систему, необхідно розробити механізми виконання наступних функцій системи: *рішення задач з використанням знань про конкретну предметну область* (можливо, при цьому виникне необхідності мати справу з невизначеністю); *взаємодія з користувачем*, включаючи пояснення намірів і рішень системи під час і після закінчення процесу рішення задачі.

Кожна з цих функцій може виявитися дуже складною і залежить від прикладної області, а також від різних практичних вимог. У процесі розробки і реалізації можуть виникати різноманітні важкі проблеми.

При розробці експертної системи прийнято поділяти її на три основних модулі:

1. База знань;
2. Машина логічного висновку;
3. Інтерфейс із користувачем.

База знань повинна містити все, що до них відноситься: факти, правила, методи, евристики, ідеї, що відносяться до рішення задач у певній прикладній області.

Машина логічного висновку повинна вміти активно використовувати інформацію, що міститься в базі знань.

Інтерфейс із користувачем повинен відповідати за безперебійний обмін інформацією між користувачем і системою; він також дає користувачу можливість спостерігати за процесом рішення задач, що протікають у машині логічного висновку.

Для створення оболонки, за допомогою якої можна проілюструвати основні ідеї і методи в області експертних систем, можна дотримувати наступного плану:

- вибрати формальний апарат для представлення знань;
- розробити механізм логічного висновку, що відповідає цьому формалізму;
- додати засоби взаємодії з користувачем;
- забезпечити можливість роботи в умовах невизначеності.

Знання, типи представлення знань в експертних системах

Експертна система містить три типи знань:

- *структуровані знання про предметну область* (після того, як ці знання виявлені, вони не змінюються);
- *структуровані динамічні знання* (змінні знання з предметної області, які обновляються по мірі виявлення нової інформації);
- *робочі знання*, які використовуються для розв'язування конкретної задачі або проведення консультації.

Всі перераховані знання зберігаються в базі знань. Для її побудови потрібно провести аналіз досвіду спеціалістів, які є експертами в конкретній предметній ділянці, а потім систематизувати, організувати та індексувати отриману інформацію для простоти її використання.

Існує багато способів представлення знань в сучасних експертних системах. Найчастіше використовується такі три методи представлення знань: *правила, семантичні сітки та фрейми*.

Термін "*фрейм*" у 1975 році ввів М.Мінський, як визначення структури даних для представлення стереотипних ситуацій. В цьому випадку модель даних представляється комбінацією трьох компонентів:

- *множини структур даних*, об'єкти яких складають вмістище баз даних;
- *множини операцій*, які використовуються для пошуку та модифікації даних;
- *множини обмежень цілісності*, які явно чи неявно визначають множину допустимих станів елементів баз даних.

Представлення знань, що базується на правилах, побудовано на використанні виразу вигляду - "*якщо*" (*умова*) - "*тоді*" (*дія*). Якщо ситуація (факти) в задачі задовольняє правило "*якщо*", тоді використовується дія, що визначається частиною "*тоді*". Співставлення частин "*якщо*" (правил з фактами) може утворити так званий ланцюжок виводу. Правила забезпечують природній спосіб опису процесів, що керуються складним і швидкозмінним середовищем. За допомогою правил можна визначити, як експертна система буде реагувати на зміну даних і при цьому не потрібно заздалегідь вказувати блок-схему управління обробкою даних.

В програмі традиційного типу передача управління і використання ресурсів здійснюються послідовними кроками, а розгалуження має місце тільки в заздалегідь вибраних точках. Цей спосіб добре діє для проблем, які допускають алгоритмічні рішення. Для задач, хід розв'язування яких керується самими даними і де розгалуження швидше норма, ніж виняток, цей спосіб малоефективний. Використання правил спрощує пояснення дій експертної системи і дозволяє людині прийти до певного результату.

Можна розглядати *фрейми* і *семантичні сітки*, як методи представлення знань, що базуються на фреймах. Таке представлення знань використовує сітку вузлів, що пов'язуються відношеннями і організуються ієрархічно. Кожен вузол представляє собою концепцію, яка може бути представлена атрибутами і значеннями, пов'язаними з цим вузлом. Вузли, які знаходяться на нижніх рівнях в ієрархії, автоматично наслідують властивості вузлів, що займають вище становище. Ці методи зазвичай забезпечують ефективний шлях класифікації того чи іншого об'єкту (події).

Багато правил експертної системи є евристичними, тобто емпіричними правилами, або спрощеннями, які ефективно обмежують пошук рішення. Евристика - це сукупність логічних прийомів і методологічних правил теоретичного дослідження і пошуку істини, методика пошуку доведення.

Евристичні правила - неформальні правила, які використовуються з метою підвищення ефективності пошуку в даній предметній ділянці. Такі підходи до розв'язування проблем швидше властиві людському мисленню "взагалі", для якого властива поява "догадки" про шлях їх вирішення з наступною перевіркою отриманого рішення. Евристичному методу протиставлявся алгоритмічний (процедурний) метод, більше характерний для комп'ютера, який інтерпретувався як механічне здійснення заданої послідовності кроків, яка однозначно приводила до розв'язку. Експертні системи використовують евристики через те, що поставлені задачі важкі і до кінця незрозумілі. Ці задачі не підлягають чіткому математичному аналізу або алгоритмічному рішенню. Алгоритмічний метод гарантує визначене коротке або оптимальне рішення задачі, тоді як евристичний метод дає прийнятне або раціональне рішення. Знання в експертних системах організовані таким чином, щоб знання про предметну ділянку відокремити від загальних (наприклад, як вирішувати задачу, або знання про те, як взаємодіяти з користувачем).

У системах, заснованих на концепції банку знань, реалізуються функції дедуктивного висновку - від узагальнених знань, що подаються в базі знань, здійснюється перехід до конкретних знань, що формуються для вирішення заданої практичної задачі.

Переваги та слабкі місця експертних систем

Експертні системи відзначаються певними перевагами при використанні. Зокрема, експертна система:

- перевищує можливості людини при вирішенні громіздких проблем;
- не має упереджених думок, тоді як експерт користується побічними знаннями і легко піддається впливу зовнішніх факторів;
- не робить поспішних висновків, нехтуючи певними етапами виводу;
- забезпечує діалоговий режим роботи;
- дозволяє роботу з інформацією, що містить символічні змінні;
- забезпечує коректну роботу з інформацією, яка містить помилки, за рахунок використання імовірнісних методів досліджень;
- дозволяє проводити одночасну обробку альтернативних версій;
- по вимозі пояснює хід кроків реалізації програми;
- забезпечує можливість обґрунтування рішення та відтворення шляху його прийняття.

Але навіть найкращі з існуючих експертних систем мають певні обмеження у порівнянні з людиною-експертом, які зводяться до наступного:

- Більшість експертних систем не цілком придатні для широкого використання. Якщо користувач не має деякого досвіду роботи з цими системами, у нього можуть виникнути серйозні труднощі. Багато експертних систем доступні лише тим експертам, які створювали їх бази знань. Тому потрібно паралельно розробляти відповідний користувацький інтерфейс, який би забезпечив кінцевому користувачу властивий йому режим роботи.

- Навики системи не завжди зростають після сеансу експертизи, навіть коли проявляються нові знання.

– Все ще залишається проблемою приведення знань, отриманих від експерта, до вигляду, який забезпечував би їх ефективно використання.

– Експертні системи, як правило, не можуть набувати якісно нових знань, не передбачених під час розробки, і тим більше не володіють здоровим глуздом. Людина-експерт при розв'язанні задач зазвичай звертається до своєї інтуїції або здорового глузду, якщо відсутні формальні методи рішення або аналоги розв'язування даної проблеми.

Сфера застосування та перспективи розвитку

Експертні системи можуть використовуватися для:

- інтерпретації;
- діагностики;
- моніторингу;
- передбачення;
- планування;
- проектування.

Експертні системи можна використовувати в прогнозуванні, плануванні, контролі, управлінні та навчанні. Наприклад, експертні системи вже застосовуються в банківській справі у таких напрямках:

- програмах аналізу інвестиційних проектів;
- програмах аналізу стану валютного, грошового та фондового ринку;
- програмах аналізу кредитоспроможності чи фінансового стану підприємств і банків.

Процес створення експертних систем значно змінився за останні роки. Завдяки появі спеціальних інструментальних засобів побудови експертних систем значно скоротились терміни та зменшилась трудомісткість їх розробки. Інструментальні засоби, що використовуються при створенні експертних систем, можна розбити на три класи:

- *мови програмування*, орієнтовані на створення експертних систем (Ліпс, Пролог, Smalltalk, FRL, Interlisp та такі загальноновживані, як: Сі, Асемблер, Паскаль, Фортран, Бейсик);
- *середовища програмування* (Delphi, Java);
- *пусті експертні системи* (оболонка EXSYS Professional for Windows).

На американському і західноєвропейських ринках систем штучного інтелекту організаціям, які бажають створити експертну систему, фірми-розробники пропонують сотні інструментальних засобів для їх побудови. Нараховуються тисячі розроблених вузькоспеціалізованих експертних систем. Це свідчить про те, що експертні системи складають дуже вагомий частину програмних засобів.

РОЗДІЛ 12

ПЕРСПЕКТИВИ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Згідно звітів про глобальний розвиток інформаційних технологій, представлених Всесвітнім Економічним Форумом, більше ста країн світу щорічно оцінюються за параметрами – сприйнятливості країни до технічних нововведень, рівень проникнення високих технологій у бізнес і державний сектор, розвиток мережевої інфраструктури країни, а також кількість і якість ІКТ-сервісів, які надаються індивідуальним і комерційним користувачам. У останньому звіті рейтинг за цими показниками очолила Данія, за нею в списку йдуть Швеція, Сінгапур і Фінляндія. США займає 7-му позицію. Найнижчий рівень розвитку інформаційних технологій спостерігається в африканських країнах. У цілому всі Європейські країни розташувалися в першій півсотні (замикає список Греція з 48-м місцем). Набирають темпу розвитку Азіатські країни: Гонконг, Тайвань, Японія, Австралія і Корея зайняли 12, 13, 14, 15 і 19-те місця. Україна посіла 75-те місце.

Наявність такого звіту – переконливе підтвердження важливості розвитку індустрії інформаційних технологій як для окремих країн, які піклуються про соціально-економічне майбутнє, так і людської цивілізації в цілому.

Перспективи розвитку інформатики, комп'ютерної техніки, інформаційних систем в цілому багатоаспектні. Вони стосуються апаратної частини систем автоматизованої обробки інформації, програмних засобів, організації електронного документообігу, середовища та сфери застосування всього комплексу задіяних для цього засобів. Зупинимось лише на деяких із них – на елементній базі (пам'яті), сфери майбутнього застосування ("нової економіки") та електронної комунікації (систем глобального електронного зв'язку).

Перспективи розвитку електронної пам'яті

Поява у швидкому майбутньому задач, що вимагають дуже великих обчислювальних потужностей, змушує вже зараз здійснювати пошук нових технічних рішень не тільки в плані удосконалення самих інформаційних процесорів, але й апаратних компонентів ПК. Незалежно від того, яка технологія використовується для виготовлення процесора, кількість даних, що поставляються на обробку, визначається можливостями комп'ютера. Ємності сучасних пристроїв масової пам'яті відбивають цю тенденцію. Диски CD-ROM дозволяють зберігати до 800 MB інформації, технологія DVD-ROM, - до 17 GB. Технологія магнітного запису також розвивається дуже швидко - типова ємність твердого диска в настільних комп'ютерах зросла до 10 TB.

Однак, у майбутньому комп'ютерам прийдеться обробляти сотні терабайт інформації. Обслуговування таких обсягів даних і переміщення їх для обробки зверху швидкими процесорами вимагають радикально нових підходів при створенні пристроїв збереження інформації.

Широкі перспективи відкриває технологія оптичного запису, відома як голографія: вона дозволяє забезпечити дуже високу щільність запису при збереженні максимальної швидкості доступу до даних. Це досягається за рахунок того, що голографічний образ (голограма) кодується в один великий блок даних, що записується усього за одне звертання. А коли відбувається читання, цей блок цілком "витягується" з пам'яті. Для читання чи запису блоків голографічно збережених на світлочутливому матеріалі (за основний матеріал прийнятий ніобат літію, LiNb₃) даних ("сторінок") використовуються лазери.

Технологічно це виглядає так (рис.31): на світлочутливий матеріал (SLM — SpatialLightModulator), який є просто рідкокристалічною панеллю, записуються дані — так званий голографічний образ. Після запису ці дані прочитуються лазером. Промінь лазера ділиться на два, перший із яких — сигнальний — проходить крізь світлочутливий матеріал, що відображає необхідну інформацію в бінарному виді. Таким чином промінь поглинає (прочитує) інформацію, після чого попадає на кристал, де «зустрічається» із другим променем. В результаті «зустрічі» з'являється так звана інтерференційна картина, а дані зберігаються у відповідній частині кристала у вигляді голограми.

Теоретично, тисячі таких цифрових сторінок, кожна з яких містить до мільйона біт, можна помістити в пристрій розміром зі шматочок цукру. Причому теоретично очікується щільність даних у 1 ТБ на кубічний сантиметр (ТБ/sm³).

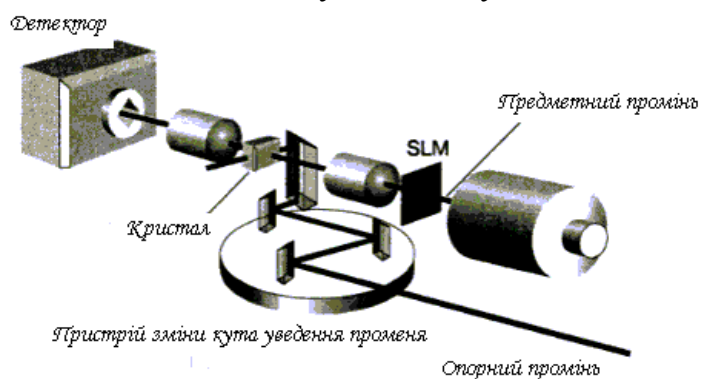


Рис.31

Для здійснення голографічного запису потрібно було розробити особливий тип носія, що поєднує велику світлочутливість, міцність, дешевизну і стабільність. Важливою вимогою були і допустимі лінійні розміри носія. Всім цим критеріям, на думку розробників, відповідають фотополімерні диски. Діаметр їх не на багато перевищує діаметр сучасних дисків і складає 130 мм. Вони поміщені в картриджі на зразок перших моделей DVD-носіїв, оскільки попадання світла на поверхню фотополімеру викличе хімічну реакцію, яка незворотно зруйнує записані дані. Спочатку технологія була розрахована для використання в архівах, оскільки вартість носіїв варіює від 120 до 180 дол. Сам же привід коштує близько 15 тис. дол.

Сьогодні ряд компаній, наприклад NTT і Optware в Японії, InPhaseTechnology в США, мають закінчені розробки із голографічними дисками (HolographicVersatileDisc - HVD) і картами (HolographicVersatileCard - HVC), і вже приступили до продажу своїх перших голографічних приладів.

Одна з розробок фірми-виготовлювача Ikegami (Японія) - голографічний диск HDS300K, який може вміщати до 300 Гб інформації, що по місткості еквівалентно 64 DVD дискам при однакових розмірах носія - діаметр його складає ті ж 12 см.

Переваги технології:

– висока щільність запису і велика швидкість зчитування інформації (наприклад, запис фільму в HD-якості розміром 30 Гб на жорсткий диск займає 30–45 хвилин, а у разі використання голографічної пам'яті процес скоротиться приблизно до 10 секунд);

– паралельний запис інформації (не по одному біту, а цілими сторінками);

– високоточне відтворення даних;

– неруйнівне зчитування даних;

– довгострокове зберігання даних на носіїві — 30–50 років;

– інформація записується не поодинці, а цілими сторінками.

Радикально інший підхід у створенні пристроїв збереження даних - молекулярний. Група дослідників центру "W.M. Keck Center for Molecular Electronic" під керівництвом професора Роберта Р. Бирга (Robert R. Birge) уже відносно давно одержала прототип підсистеми пам'яті, що використовує для запам'ятовування цифрові біти молекули. Це молекули протеїну, названі терміном бактеріородопсин (bacteriorhodopsin). Ці молекули містяться у мембрані мікроорганізму, названого *halobacterium halobium*, який "проживає" у соляних болотах, де температура може досягати +150 °С. Коли рівень вмісту кисню у навколишньому середовищі настільки низький, що для одержання енергії неможливо використовувати дихання (процес окислювання), цей організм для фотосинтезу використовує протеїн.

Бактеріородопсин був вибраний тому, що фотоцикл (послідовність структурних змін при реакції молекули зі світлом) робить цю молекулу ідеальним логічним запам'ятовуючим елементом типу "&" чи типу перемикача з одного стану в інший (тригер). Як показали дослідження, bR-стан (логічне значення біта "0") і Q-стан (логічне значення біта "1") є проміжними станами молекули і ці стани можуть залишатися стабільними протягом багатьох років.

Дані, записані на бактеріородопсин як запам'ятовуючий пристрій, повинні зберігатися приблизно п'ять років. Іншою важливою особливістю бактеріородопсина є те, що згадані два стани мають помітно різні спектри поглинання. Останнє дозволяє легко визначити поточний стан молекули за допомогою лазера, налаштованого на відповідну частоту.

У створеному прототипі системи пам'яті бактеріородопсин запам'ятовує дані в тривимірній матриці. Матриця являє собою кювету (прозору судину), заповнену поліакридним гелем, у який поміщений протеїн (рис.32). Кювета має довгасту форму розміром 1x1x2 дюйми. Протеїн, що знаходиться в bR-стані, фіксується в просторі при полімеризації гелю. Кювету оточують батарея лазерів і детекторна матриця, побудована на базі приладу, що використовує принцип зарядової інжекції (CID - Charge Injection Device), що слугають для запису і читання даних.

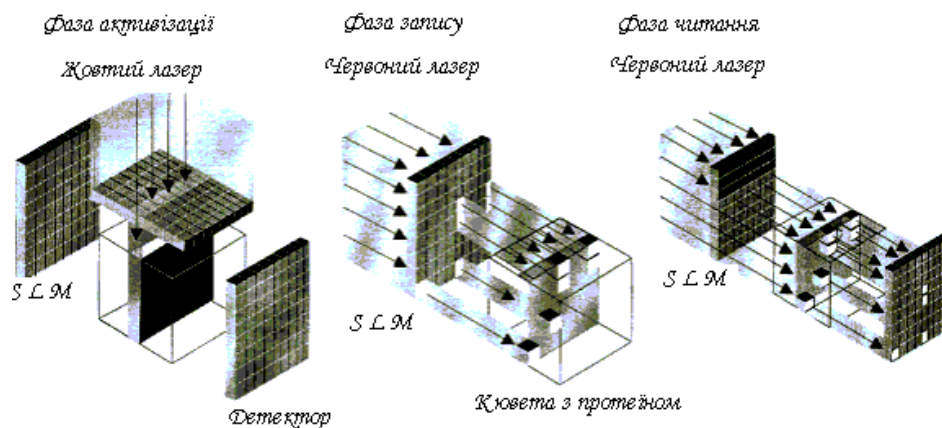


Рис. 32

При записі даних спочатку треба запалити жовтий "сторінковий" лазер - для переведення молекул у Q-стан. Просторовий світловий модулятор (SLM) – матриця, що створює маску на шляху променя, – викликає виникнення активної (збудженої) площини в матеріалі усередині кювети. Ця енергоактивна площина являє собою сторінку даних, що може вміщувати масив 4096x4096 bit. Перед поверненням протеїну в стан спокою (у ньому він може знаходитися досить

тривалий час, зберігаючи інформацію) запалюється червоний лазер (записуючий), розташований під прямим кутом стосовно жовтого. Інший світловий модулятор SLM відображає двоцифрові дані і, таким чином, створює на шляху променя відповідну маску, тому опроміненню піддадуться тільки визначені “плями” (“точки”) сторінки. Молекули в цих місцях перейдуть у Q-стан і будуть представляти двоцифрову одиницю. Частина сторінки, що залишилася, повернеться в первісний bR-стан і буде представляти двоцифрові нулі. Для того, щоб прочитати дані, треба знову запалити сторінковий лазер, що переводить сторінку, що читається, у Q-стан. Це робиться для того, щоб надалі, за допомогою розходження в спектрах поглинання, ідентифікувати двійкові нулі й одиниці. Через 2 ms після цього сторінка “занурюється” у низькоінтенсивний світловий потік червоного лазера. Низька інтенсивність потрібна для того, щоб попередити “перескакування” молекул у Q-стан. Молекули, що представляють двоцифровий нуль, поглинають червоне світло, а ті, що представляють двоцифрову одиницю, пропускають промінь повз себе. Це створює “шаховий” малюнок зі світлих і темних плям на LCD-матриці, що захоплює сторінку цифрової інформації.

Для стирання даних досить короткого імпульсу синього лазера, щоб повернути молекули з Q-стану у вихідний bR-стан. Синє світло не обов'язково повинно йти від лазера (так можна стерти всю кювету за допомогою звичайної ультрафіолетової лампи). Для забезпечення цілісності даних при вибіркового стиранні сторінок застосовується кешування декількох суміжних сторінок. При операціях читання-запису також використовуються два додаткових біти парності, щоб захиститися від помилок. Сторінка даних може бути прочитана без руйнування до 5000 разів. Кожна сторінка відслідковується лічильником, і якщо відбувається 1024 читання, то сторінка “освіжується” (регенерується) за допомогою нової операції запису.

З огляду на те, що молекула змінює свої стани в межах 1 ms, сумарний час для виконання операції чи читання запису складає близько 10 ms. Однак, за аналогією із системою голографічної пам'яті, цей пристрій здійснює рівнобіжний доступ у циклі читання-запису, що дозволяє розраховувати на швидкість до 10 MBps.

Передбачається, що якщо об'єднати по восьми запам'ятовуючих бітових осередків у байт із рівнобіжним доступом, то можна досягти швидкості 80 MBps, але для такого способу необхідна відповідна схемотехнічна реалізація підсистеми пам'яті.

Теоретично, кювета, про яку вже йшла мова, може вмістити 1 ТВ даних. Обмеження на ємність пов'язані, в основному, із проблемами лінзової системи і якістю протеїну.

Перспективи розвитку інтелектуального потенціалу

Перспективних змін слід чекати не лише у компонентах інформаційних систем, але й у сфері їх застосування. Скажімо, у так званій “новій економіці”, яку називають ще “економікою знань”, “економікою розвитку інтелектуального потенціалу” тощо. Хоча після деяких останніх тенденцій в світовій економіці міф про “нову економіку” в значній мірі розвіявся, ринок інформаційних технологій все ще має незаповнені ніші, які, якщо їх зайняти, можуть принести значні прибутки.

До певного часу основну частку ринку інформаційних технологій займали виробництво і продаж комп'ютерного та телекомунікаційного обладнання. Тепер маємо дещо іншу ситуацію – все більшу частку здобуває виробництво програмного забезпечення. Перспективи розвитку інформаційних технологій можуть стосуватися

й інших сфер економіки. Значною мірою суттєвих змін слід чекати від застосування знань при утвердженні інноваційної моделі економічного розвитку в Україні.

Важко заперечити, що Україна має імідж держави з “високим інтелектуальним потенціалом”, і можна стверджувати, що не безпідставно. Сотні тисяч студентів щорічно отримують дипломи про вищу освіту, помітна їх частина отримує знання в сфері інформаційних технологій. Крім того, існує велика кількість спеціалістів, які не отримали вищої освіти в цій сфері, але так чи інакше пов’язані зі сферою інформаційних технологій і, зокрема, софтверною індустрією (індустрією розробки інтелектуальної складової інформаційних технологій – програмних виробів).

Якщо порівняти рівень заробітної платні, що отримують спеціалісти зі сфери індустрії програмування різних країн світу, отримаємо наступні результати:

	Середня заробітна плата програміста
США	\$ 60 000
Індія	\$ 9 600
Україна	\$ 6 000

Отже, маємо значні конкурентні переваги – великий пул достатньо кваліфікованого робочого потенціалу за ціну, набагато нижчу, ніж в інших країнах. Логічно зробити висновок, що Україна має переваги порівняно з іншими учасниками ринку програмного забезпечення, й потужний розвиток індустрії програмного забезпечення в державі мав би бути природним.

Світовий попит на послуги інформаційних технологій складає \$400-500 млрд. в рік, і цей ринок до минулого року стійко зростає: на 10-12% щорічно. Значна частина цього попиту могла б покриватися за рахунок експорту програмних продуктів, однак експорт ПЗ не перевищує \$10 млрд., тобто залишається потужний експортний потенціал, частину з якого має шанси заповнити і Україна. Зважаючи на те, що основними замовниками ПЗ залишаються англосаксонські країни, де рівень заробітної платні спеціаліста - розробника програмного забезпечення перевищує рівень зарплатні українського спеціаліста в декілька разів, то на Україну повинна припадати значна частина “пирога” офшорного програмування.

На підтвердження наявності не використаного потенціалу, звернемося до статистики. За даними Державного комітету статистики, рівень виробництва ПЗ в Україні в останні роки оцінюється в мільйони (максимум – десятки мільйонів гривень), при цьому обсяги ПЗ, що йдуть на експорт обчислюється щонайбільше одиницями мільйонів гривень. Для порівняння – експорт ПЗ з Індії має складати в 2014 р. до 90 млрд. доларів. Отже, розвиток такого напрямку – одна з вагомих перспектив для розвитку економіки України.

Розвиток науково-освітнього інформаційного сегменту

Ще одна із перспектив – вирішення в Україні проблеми, викликані відсутністю комплексної державної програми розвитку науково-освітньої мережі. Світовий досвід розвитку науково-освітніх комп’ютерних мереж досить багатий. Найвідомішими з таких мереж є DFN, B-WIN у Німеччині, Super Janet у Великій Британії, SURFNET у Голландії, Renater-2 у Франції, CANET-2 у Канаді. Навіть Росія, яка ще не входить до числа країн з розвинутою інформаційною інфраструктурою, має кілька потужних науково-освітніх національних і регіональних мереж (RBNET, RUNET, RELARN, RSSI, MSUNET, FREENET, РАН/ORC) і випереджає Україну більш ніж на порядок за кількістю головних

серверів. То ж використання досвіду інших країн мало б сприяти розвитку необхідної для України науково-освітньої мережі.

Інша проблема, вирішення якої можна вважати напрямком розвитку інформаційних технологій в Україні, це приєднання Українського науково-освітнього інформаційного сегменту до Європейського і світового інформаційного простору. Об'єднання національних науково-освітніх мереж є світовою тенденцією. Так, у Європі функціонувала транснаціональна мережа TEN-155 (швидкість передачі даних 155 Мбіт/с), що трансформувалася у науково-освітню мережу GEANT, яка вже переходить до використання оптиковолоконних каналів передачі даних із швидкістю до 1 Терабіт/с. Швидкого розвитку набуває ще одна інформаційна наукова мережа країн Південної та Східної Європи – SINSEE. Використання їх можливостей – один із прерогативних напрямів розвитку інформаційних технологій в нашій країні.

Перспективи розвитку інформаційних технологій економічних застосуваннях

Розвиток інформатики надзвичайно динамічний. При цьому і апаратні і програмні технології розвиваються шляхом, який робить можливим обробку великої кількості інформації в режимі реального часу і доступ до цієї інформації можливий майже з будь-якої точки за допомогою розподілених баз даних. Більш того, збільшення комфортності для користувачів і зростання можливостей удосконалення персональних комп'ютерів і робочих станцій дозволяють працювати з цими системами людям, які формально мають невелику підготовку.

Удосконалення перш за все включають в себе взаємозв'язок комп'ютерів в локальних, регіональних, національних і міжнародних мережах. Крім цього, все важливішою стає сама робота на персональних комп'ютерах, робочих станціях або універсальних комп'ютерах. Має змінитися зміст такої роботи: пріоритетом повинно стати не вміння користуватися комп'ютерною технікою, а вміння ефективно використовувати можливості, надані її апаратним та програмним забезпеченням.

Розвиток ґрунтується на тенденції до відкритих систем, які дозволяють переносити сімейства програмних засобів загального використання з однієї моделі комп'ютера на іншу.

Посилюються спроби встановлення більш ефективних і тісно взаємозв'язаних ділових відносин між підприємствами і організаціями. Акцент робиться на збільшення продуктивності, скорочення витрат і удосконалення товарів і послуг за допомогою наскрізного електронного зв'язку з високим рівнем автоматизації. З іншого боку, існує тенденція до більшої взаємодії всередині організації для підвищення ефективності функціонування і продуктивності.

Організаціям (як і всій економіці в цілому) знадобляться суттєві корективи, щоб пристосуватися до прискорюваних змін буквально у всьому: виробництві, організації послуг тощо, які потребують більшої адаптивності виробничих організаційних структур і розробки засобів їх підтримки.

Вже сьогоднішній економічний розвиток включає в себе проблему інформаційного перевантаження бізнес-менеджерів. Що потрібно менеджерам – так це допомога в оцінці і виборі даних, які містять важливу інформацію для конкретної проблемної ситуації, і перетворення цих даних в знання для покращення процесу прийняття рішення.

В означеному контексті інформаційна технологія (ІТ) – це тільки початок розв'язування проблем, пов'язаних з обробкою інформації з використанням системи технічних та програмних засобів.

Науково-методичне видання

Федунець А.Д., Рибаківа Л.В., Пархоменко Ю.М., Кислун О.А

Економічна інформатика

Підручник

Технічний редактор – Лисенко В.Ф.

Формат 60x84/16. Ум. друк. арк 12,00.
Облік. видав арк. 14,7. Тираж ____ . Зам. ____

Видавець і виготовлювач СПД ФО Лисенко В.Ф.
25029, м. Кіровоград, вул. Пацаєва, 14, к. 1, кв. 101. Тел. (0522) 322-326
Свідоцтво суб'єкта видавничої справи ДК № 3904 від 22.10.2010.