

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ

Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання, оформлення та захисту курсової роботи
з навчальної дисципліни Базис даних

для студентів денної та заочної форми навчання за спеціальностями
123 «Комп'ютерна інженерія»,
125 «Кібербезпека»

ЗАТВЕРДЖЕНО
кафедрою кібербезпеки та програмного
забезпечення,
протокол від 2 вересня 2021 року № 2

КРОПИВНИЦЬКИЙ
2021

Методичні рекомендації до виконання, оформлення та захисту курсової роботи з навчальної дисципліни Баз даних для студентів денної та заочної форми навчання за спеціальностями 123 «Комп'ютерна інженерія», 125 «Кібербезпека» / уклад. В.В. Босько, Л.В. Константинова — Кропивницький: ЦНТУ, 2021. — 76 с.

Укладач: Босько В. В., Константинова Л. В.

Рецензенти: Смірнов О. А., д-р техн. наук, професор;
Дреєв О. М., канд. техн. наук.

© Босько В. В., Константинова Л.В., укладання, 2021
© Центральноукраїнський національний
технічний університет, 2021

ЗМІСТ

Вступ	4
Мета і завдання КР	4
Основні етапи КР	5
Рекомендовані теми КР	7
Структура і обсяг КР	11
Завдання та календарний план КР	12
Аналіз інформаційних джерел та складання бібліографії	13
Розробка ПЗ КР	14
Вказівки щодо розробки інформаційно-логічної моделі даних	18
Зміст розділів пояснювальної записки КР.....	32
Вимоги до оформлення текстових документів пояснювальної записки	37
Вимоги до оформлення ілюстрацій	41
Вимоги до оформлення таблиць	43
Оформлення додатків	45
Оформлення лістинга програми	47
Оформлення графічних матеріалів. Умовні позначення	48
Захист КР та критерії оцінювання	60
Список використаних джерел	64
<i>Додаток А.1 – Форма листа-завдання на курсову роботу</i>	<i>66</i>
<i>Додаток А.2 – Календарний план курсового проектування</i>	<i>67</i>
<i>Додаток Б – Титульний аркуш курсової роботи</i>	<i>68</i>
<i>Додаток В – Основний напис відповідно ГОСТ 2.104-2006</i>	<i>69</i>
<i>Додаток Г – Приклад інформаційно-логічної схеми</i>	<i>70</i>
<i>Додаток Д – Приклад виконання блок-схеми алгоритма</i>	<i>71</i>
<i>Додаток Е – Приклад оформлення лістинга програми</i>	<i>72</i>
<i>Додаток Є – Приклад функціональної схеми системи</i>	<i>73</i>
<i>Додаток Ж.1 – Приклад форм БД</i>	<i>74</i>
<i>Додаток Ж.2 – Приклад таблиць БД</i>	<i>75</i>
<i>Додаток Ж.3 – Приклад запитів БД</i>	<i>76</i>

Вступ

Для полегшення виконання, оформлення та захисту курсової роботи (КР) студентам денної та заочної форми навчання необхідно ознайомитись з представленими методичними рекомендаціями.

В методичних рекомендаціях висвітлено організаційні аспекти виконання КР з навчальної дисципліни «Бази даних» для студентів, які навчаються за спеціальностями 123 «Комп'ютерна інженерія», 125 «Кібербезпека», а також поради та вимоги щодо підбору, опрацювання та викладення теоретичного матеріалу, розробки КР та її програмної реалізації, порядку оформлення пояснювальної записки курсової роботи, її підготовки та представлення до захисту.

Мета і завдання КР

Метою виконання КР є закріплення та поглиблення знань, отриманих в процесі вивчення курсу «Бази даних», набуття практичних навичок та вмінь подальшого їх використання для проектування й розробки програмного забезпечення. При виконанні курсових робіт відбувається активний процес систематизації та закріплення знань, поповнення їх із спеціальної літератури, довідкових посібників та інших джерел, зокрема Internet.

Курсова робота виконується здобувачем вищої освіти самостійно під керівництвом викладача протягом терміну встановленого навчальним планом.

КР носить творчий характер, кінцевим результатом якого може стати новий продукт (технологічний процес, механізм, технічні та програмні засоби, тощо) або пропозиції щодо вирішення сформульованої у курсовій роботі проблеми.

Основними завданнями КР є:

- узагальнення теоретичних знань, отриманих під час вивчення дисципліни «Бази даних», за допомогою поглибленого вивчення додаткових фахових інформаційних джерел;
- набуття навичок практичного застосування теоретичних знань, проведення дослідження й аналізу існуючих програмних систем та побудова інформаційної системи предметної області.
- набуття практичних навичок та вмінь постановки інженерних задач, проектування складних систем та їх реалізація; розробка супровідної технічної документації до розробленої роботи.

Кваліфікаційна функція КР – сприяння розширенню і поглибленню теоретичних знань, розвитку досвіду їх практичного використання, самостійного розв’язання конкретних завдань.

Форма реалізації – робота, виконана відповідно до Державних стандартів та вимог, визначених специфікою кафедри Кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету.

Затвердження теми та контроль роботи студента над курсовою роботою здійснює керівник КР.

Основні етапи КР

1. Отримати тему курсової роботи або затвердити власну тему у керівника КР (додаток А.1).
2. Скласти графік виконання курсової роботи, затвердити його у керівника КР та внести у календарний план завдання на курсову роботу (додаток А.2).
3. Розпочати збір і обробку інформації за темою курсової роботи. Виконати чітку постановку задачі на виконання курсової роботи.

4. Розробити структурну, функціональну та інформаційно-логічну схеми, блок-схему, алгоритм програмного забезпечення.
5. Розробити інформаційну систему з використанням бази даних та спроектувати предметну область. Побудувати логічну структуру баз даних. На виході інформаційно-логічна модель.
6. Відповідно до інформаційно-логічної моделі побудувати таблиці та схему даних.
7. Програмування запитів до бази даних (QBE та SQL).
8. Побудова інтерфейсу користувача (головна кнопкова форма та меню користувача).
9. Побудова звітів.
10. Оформлення пояснювальної записки курсової роботи.
11. Подати курсову роботу на попередній захист.
12. виправити виявлені на передзахисті помилки та недоліки роботи, врахувати зауваження керівника й підготувати роботу до захисту.
13. Не пізніше встановленого календарним планом терміну подати курсову роботу до захисту.

За прийняті у КР рішення та достовірність даних відповідає автор курсової роботи, здобувач вищої освіти. Студент, який без поважної причини не підготував КР або не захистив у зазначений термін, вважається таким, що має академічну заборгованість.

Рекомендовані теми КР

1. Автоматизована система продажу комп'ютерної техніки та комплектуючих матеріалів.
2. Розробка програми для продажу автомобілів в розстрочку.
3. Автоматизована система «Автосалон».
4. Автоматизація продажу продукції підприємства.
5. Автоматизована система для роботи фірми з продажу та обслуговування офісної техніки.
6. Розробка інформаційної системи «Телефонний довідник».
7. Автоматизована система «Географічний довідник».
8. Автоматизація роботи книжкового магазину.
9. Автоматизація роботи авіа агентства.
10. Автоматизація роботи завідуючого військовим складом.
11. Інформаційна система інституту досліджень паранормальних явищ.
12. Побудова інформаційної системи «Туристичне агентство».
13. Розробка автоматизованої системи обліку успішності в школі.
14. Розробка програми для автоматизації навчального процесу.
15. Тестуюча система з дисципліни «Бази даних».
16. Автоматизована система психологічного тестування людини.
17. Інформаційна система «Розклад занять у навчальних закладах».
18. Автоматизована система «Обласна стоматологічна поліклініка».
19. Інформаційна система для обліку досліджень патологоанатомічного відділення лікарні.
20. Автоматизована система роботи музею.
21. Інформаційна система для роботи цеху на заводі.
22. Каталогізатор відеоінформації.
23. Розробка інформаційної системи «Адресне бюро».
24. Інформаційна система «Автовокзал».
25. Автоматизована система оптового складу комп'ютерної техніки.

26. Автоматизована система обліку будівельних матеріалів на складі.
27. Розробка інформаційної системи для бібліотеки.
28. Інформаційно-обчислювальна система «Інтернет-провайдер».
29. Розробка програми для автоматизації роботи куратора у ВУЗах.
30. Автоматизована система обліку продукції на складі мототехніки.
31. Автоматизована система обліку матеріалів на аптечних складах.
32. Автоматизована система обліку продукції м'ясокомбінату.
33. Інформаційна система для роботи деканату.
34. Розробка автоматизованої системи розрахунку електроенергії, яка споживається населенням.
35. Автоматизована система для роботи пенсійного фонду.
36. Автоматизована система обліку заробітної плати для малого підприємства.
37. Автоматизована система розрахунку за послуги станції технічного обслуговування автомобілів.
38. Розробка автоматизованої системи для СТО.
39. Інформаційна система «Розклад руху потягів залізничного вокзалу».
40. Автоматизована система роботи мережі магазинів.
41. Автоматизована система «Відділ кадрів».
42. Автоматизована система роботи спортивного клубу.
43. Інформаційна система для професійної футбольної ліги.
44. Автоматизована система «Радіостанція».
45. Автоматизована система організації фестивалів.
46. Розробка програми для автоматизації робочого місця робітника кредитного відділу банку.
47. Розробка автоматизованої системи розрахунку населення за послуги газопостачання.
48. Автоматизована система для сплати комунальних платежів.
49. Автоматизована система для реєстрації абітурієнтів у ВУЗах.
50. База даних для інтернет магазину одягу.

51. База даних для магазину квітів.
52. База даних для продуктового магазину.
53. Інформаційна система косметичної фірми.
54. Розробка бази даних для агентства нерухомості.
55. Інформаційна система для піцерії.
56. Автоматизація роботи ветеринарної клініки.
57. База даних для салону краси.
58. База даних для сайту з продажу меблів.
59. Автоматизація роботи фірми з пошиття та ремонту взуття.
60. Інформаційна система фірми з продажу та ремонту меблів.
61. Інформаційна система дитячого розважального центру.
62. Розробка бази даних для сайту знайомств.
63. База даних для сайту з продажу канцелярських товарів.
64. База даних для сайту з продажу програмного забезпечення.
65. База даних для сайту кінотеатру.
66. База даних для дитячого дошкільного закладу.
67. База даних для танцювального колективу.
68. Розробка програмного забезпечення для навчання з дисципліни «Бази даних».
69. База даних для населеного пункту.
70. База даних для музичної школи.
71. База даних типографій міста Кропивницький.
72. База даних для госпіталя.
73. База даних для сайту волонтерської організації.
74. Інформаційна система для Державної податкової адміністрації.
75. База даних фірми з організації свят.
76. Електронний кабінет платника для Державної фіскальної служби України.
77. База даних для іподрому.
78. База даних для сайту краєзнавчого музею.

79. База даних магазину зоотоварів.
80. Інформаційна система орнітологічного клубу.
81. База даних для роботи рибного господарства.
82. Автоматизована система для роботи фермерського господарства.
83. База даних для обліку підприємницької діяльності.
84. Інформаційна система «Зоопарк».
85. База даних для Державної служби охорони.
86. Інформаційна система для телефонної компанії.
87. Автоматизація розрахунку навантаження викладачів учбового закладу.
88. База даних для сайту учбового закладу.
89. Інформаційна система з продажу комп'ютерних ігор.
90. Автоматизація продажу для магазину дитячих іграшок.
91. База даних для компанії з розробки програмного забезпечення.
92. База даних для виставки агропродукції.
93. Інформаційна система для школи мистецтв.
94. База даних для КРЕП.
95. База даних для магазину ювелірних виробів.
96. База даних для роботи юридичної компанії.
97. База даних для магазину побутової техніки.
98. Програмне забезпечення для автоматизація роботи бухгалтера.
99. База даних для Автошколи.
100. База даних для агентства працевлаштування.

Структура і обсяг КР

Документ курсової роботи повинен мати наступну структуру:

1 – Титульний аркуш.

2 – Завдання на курсову роботу та календарний план

3 – Пояснювальна записка (зміст і розділи):

- Перелік умовних позначень, символів, одиниць і термінів

- **Вступ** (1-2 сторінки).

- **Розділ 1 Призначення та область використання** (10-18 сторінок).

- **Розділ 2 Перегляд аналогічних існуючих систем** (10-18 сторінок).

- **Розділ 3 Опис і обґрунтування проектних рішень** (15-20 сторінок).

- **Розділ 4 Реалізація роботи. Розрахунки та експериментальні дані, що підтверджують вірність проектних і програмних рішень** (12-20 сторінок),

- **Розділ 5 Впровадження системи в промислову експлуатацію** (2-5 сторінок).

- **Розділ 6 Основні висновки** (1-2 сторінки).

- Список використаних джерел

4 Додатки:

4.1 **Додаток А** – Інформаційно-логічна модель предметної області.

4.2 **Додаток Б** – ER-схеми. Семантичне моделювання предметної області.

4.3 **Додаток В** – Схема даних.

4.4 **Додаток Г** – Приклад побудованих таблиць бази даних.

4.5 **Додаток Д** – Побудова форм, презентація інформаційної системи та побудова інтерфейсу користувача.

4.6 **Додаток Е** – Основні запити SQL.

4.7 **Додаток Ж** – Лістинг програми.

В залежності від особливостей теми курсової роботи, за погодженням з керівником КР можливе уточнення змісту розділів (текстові, графічні

документи: функціональну схему системи та блок-схему алгоритму програми, тощо), введення нових або об'єднання деяких розділів пояснювальної записки КР.

Пояснювальна записка КР повинна мати не менше 30 сторінок машинописного тексту, який виконано виключно чорним кольором і чітким шрифтом. Рекомендований шрифт – Times New Roman, розмір шрифту – 14 пт, міжрядковий інтервал – 1.5 пт, абзацний відступ – 15 мм. **Основна частина** складається з розділів, підрозділів, пунктів та підпунктів. Кожний розділ починають з нової сторінки. Структура основної частини роботи повинна розкривати реалізацію завдань кваліфікаційної бакалаврської роботи.

Завдання та календарний план КР

Завдання на курсову роботу видається студентові перед початком роботи над КР і затверджується керівником курсової роботи (додаток А.1, А.2).

Календарний план (додаток А.2) складається студентом та затверджується керівником КР.

Рекомендовано наступний план курсового проектування:

1. Вивчення предметної області інформаційної системи та побудова інформаційно-логічної моделі.
2. Побудова структури таблиць відповідно об'єктам предметної області.
3. Побудова схеми даних відповідно до інформаційно-логічної моделі.
4. Побудова інтерфейсу користувача, функціональної схеми системи.
5. Побудова запитів SQL.
6. Розробка програмного забезпечення.
7. Побудова презентації інформаційної системи.
8. Попередній захист курсової роботи.
9. Захист курсової роботи.

Термін виконання етапів курсового проектування визначається студентом, погоджується з керівником, вноситься до календарного плану листа-завдання (додаток А.2) та підписується студентом і керівником.

Під час виконання КР кожен етап, відповідно до календарного плану, узгоджується з керівником, який робить відповідний запис у графі «Примітки» календарного плану.

Аналіз інформаційних джерел та складання бібліографії

Під час виконання КР необхідно підібрати актуальну літературу та інші інформаційні джерела, які стосуються теми КР. Після їх вивчення та аналізу для реалізації роботи, складається бібліографія, яка подається у розділі “Список використаних джерел” пояснювальної записки КР.

При самостійному пошуку фахової літератури особливу увагу слід звернути на нові публікації та періодичні фахові видання - газети, вісники, журнали -, в яких можна знайти найновіші розробки в галузі інформаційних технологій, результати досліджень спеціалістів тощо. Також важливо опрацювати джерела сучасної зарубіжної літератури.

Окрім друкованих видань допускається використання електронних інформаційних джерел Internet. Проте перевагу слід надавати офіційним (!) web-сайтам і базам даних. Найвідоміший серед них - електронний фонд Національної бібліотеки України імені В.І. Вернадського (НБУВ). Він доступний за адресою: <http://nbuv.gov.ua/>.

Спочатку необхідно ознайомитися з основною літературою (підручниками, журналами, збірниками наукових праць тощо), потім вивчаються електронні джерела інформації, розміщені в мережі або розповсюджуються на електронних носіях.

Бібліографічний опис у списку літератури повинен містити прізвище та ініціали автора, назву, видавництво, рік видання, кількість сторінок. Нариклад,

Гайна Г.А. Основи проектування баз даних: Навчальний посібник. – К.: КНУБА, 2005. – 204 с.

У тексті пояснювальної записки КР рекомендовано вказувати в кінці речення (абзацу) посилання на використану літературу за допомогою квадратних дужок “[]”, в яких вказується номер джерела зі списку літератури. Наприклад, [3] - посилання на джерело зі списку літератури під номером 3; [2-4] - посилання на джерела під номерами 2, 3 та 4; [1-3, 6] - посилання на джерела під номерами 1, 2, 3, 6.

Розробка ПЗ КР

Розробка програмного забезпечення КР здійснюється після аналізу літератури, розробки структурних, функціональних схем та блок-схеми алгоритму програми, інформаційно-логічної моделі даних.

Першим етапом є вибір СКБД, що задовільняє вимогам даної інформаційної системи.

Другий етап - розробка інтерфейса користувача. Інтерфейс повинен мати такі властивості: природність, узгодженість, дружність, принцип “зворотнього зв’язку”, простота, гнучкість, естетична привабливість, чіткість.

Ефективність інтерфейса полягає у швидкому розвитку в користувачів простої концептуальної моделі взаємодії. Це досягається через узгодженість. Концепція узгодженості полягає в тому, що під час роботи з ПК у користувача формується система очікування однакових реакцій на однакові дії, що постійно підкріплює призначену для користувача модель інтерфейсу.

Інтерфейс може бути узгоджений в трьох аспектах: фізичному, синтаксичному і семантичному.

Фізична узгодженість відноситься до технічних засобів.

Синтаксична узгодженість відноситься до послідовності й порядку появи елементів на екрані (мова зображень), послідовності запитів (мова дій).

Наприклад, синтаксична узгодженість матиме місце, якщо заголовок панелі завжди розміщується в центрі та вгорі панелі.

Семантична узгодженість відноситься до значення елементів, складових інтерфейсу. Наприклад, що означає “Вихід”? Де користувачі роблять запит на “Вихід” і що потім відбувається?

Природність інтерфейса - найважливіша його характеристика, оскільки він не змушує користувача істотно змінювати звичні для нього способи розв’язку задачі. Це, зокрема, означає, що повідомлення й результати, які видаються програмним продуктом, не повинні вимагати додаткових пояснень. Доцільно також зберегти систему позначень і термінологію, які застосовувались в даній предметній області.

Інтуїтивно зрозумілий інтерфейс забезпечує використання знайомих користувачеві понять і образів (метафор). Користувачі зазвичай вивчають особливості роботи з новим програмним продуктом методом спроб і помилок. Ефективний інтерфейс повинен враховувати такий підхід. На кожному етапі роботи він повинен виконувати тільки відповідний набір дій і попереджати користувачів про ті ситуації, де вони можуть зашкодити системі або даним; ще краще, якщо у користувача є можливість відмінити або виправити виконані дії. Навіть за наявності добре спроектованого інтерфейсу користувачі можуть робити ті або інші помилки. Ці помилки можуть бути як “фізичного” типу (випадковий вибір неправильної команди або даних) так і “логічного” (ухвалення неправильного рішення на вибір команди або даних). Ефективний інтерфейс повинен дозволяти запобігати ситуаціям, які, ймовірно, закінчаться помилками. Він також повинен уміти адаптуватися до потенційних помилок користувача і полегшувати йому процес усунення наслідків таких помилок.

Завжди забезпечуйте зворотний зв’язок для дій користувача. Кожна дія користувача повинна отримувати візуальне, а іноді і звукове підтвердження того, що програмне забезпечення сприйняло введену команду; при цьому вид реакції, за можливістю, повинен враховувати природу виконаної дії.

Інтерфейс повинен бути простим. При цьому мають на увазі забезпечення легкості в його вивченні й у використанні. Крім того, він повинен надавати доступ до всього набору функціональних можливостей, передбачених даною програмою. Реалізація доступу до широких функціональних можливостей і забезпечення простоти роботи суперечать один одному. Розробка ефективного інтерфейса покликана збалансувати ці цілі.

Один з можливих шляхів підтримки простоти - зображення на екрані інформації, мінімально необхідної для виконання користувачем чергового кроку завдання.

Інший шлях до створення простого, але ефективного інтерфейсу - розміщення і представлення елементів на екрані з урахуванням їх смислового значення й логічного взаємозв'язку. Це дозволяє використовувати в процесі роботи асоціативне мислення користувача.

Під гнучкістю інтерфейсу розуміють здатність враховувати рівень підготовки та продуктивність праці користувача. Властивість гнучкості припускає можливість зміни структури діалогу і/або вхідних даних.

Проектування візуальних компонентів є найважливішою складовою розробки програмного інтерфейсу. Коректне візуальне представлення об'єктів забезпечує передачу дуже важливої додаткової інформації про поведінку і взаємодію різних об'єктів. В той же час слід пам'ятати, що кожен візуальний елемент, який з'являється на екрані, потенційно вимагає уваги користувача, яка, як відомо, не безмежна. Слід забезпечити формування на екрані такого середовища, яке не тільки сприяло б розумінню користувачем представленої інформації, але і дозволяло б зосередитися на найбільш важливих її аспектах.

До візуальних атрибутів інформації, що відображається, відносяться:

- взаємне розташування і розмір об'єктів, що відображаються;
- палітра;
- засоби привертання уваги користувача.

Проектування розміщення даних на екрані передбачає виконання наступних дій:

- визначення складу інформації (елементів, компонентів, об'єктів тощо), яка повинна з'являтися на екрані;

- вибір формату представлення цієї інформації;
- визначення взаємного розташування даних (чи об'єктів) на екрані;
- вибір засобів привертання уваги користувача;
- розробка макету розміщення даних на екрані;
- оцінка ефективності розміщення інформації.

Загальні принципи розташування інформації на екрані повинні забезпечувати для користувача:

- можливість переглядання екрану в логічній послідовності;
- простоту вибору потрібної інформації;
- можливість ідентифікації зв'язаних груп інформації;
- помітність виключних ситуацій (повідомлень про помилки або попередження);

- можливість визначити, яка дія з боку користувача потрібна (і чи потрібна взагалі) для продовження виконання завдання.

Питання про те, яка інформація підлягає відображенню, вирішується залежно від специфіки теми курсової роботи. В даному випадку істотну роль грає правильне розбиття завдання на операції (етапи), які не вимагають одночасної присутності великого об'єму даних на екрані.

Ця умова витікає з такої психофізіологічної особливості людини, як обмеженість його короткочасної пам'яті, здатної зберігати одночасно не більше п'яти-дев'яти об'єктів. Якщо вся інформація початкового документа не вміщується на одному екрані, деякі елементи даних можуть повторюватися на інших екранах для збереження цілісності й послідовності обробки. Як правило, повторювана інформація не повинна міняти свого розташування на всіх кроках виконання завдання.

Властивість природності інтерфейсу припускає, що інформація відображається на екрані у вигляді, придатному для безпосереднього використання. Виділення інформації - це використання таких атрибутів, які

дозволяють привернути увагу користувача до певної області екрану. В якості подібних атрибутів можуть виступати: колір символів, колір фону, рівень яскравості, мерехтіння і застосування різних шрифтів для символів, що виводяться. Основна рекомендація: слід прагнути використовувати мінімально необхідну кількість атрибутів.

Програмна реалізація є **третім етапом** розробки програмного забезпечення. На цьому етапі за допомогою обраної мови програмування та на основі розробленого алгоритма здійснюється написання програми.

Важливою складовою лістинга програми є *коментарі*. Вони не тільки полегшать “читання” та відлагодження програми, але й є ознакою гарного стилю програмування.

Обов’язковим компонентом програмного забезпечення є меню “Допомога” (“Довідка”, “Help”), в якому розміщується інструкція користувача (з детальним описом принципів роботи з програмою). Крім того, необхідно додати меню “Автор”, що містить довідникову інформацію про розробника програмного забезпечення (прізвище, ім’я, по батькові, назва ВНЗ, факультет, група, рік розробки тощо).

Заключним, **етапом** розробки програмного забезпечення є відлагодження програмного забезпечення, після чого його необхідно погодити та затвердити у керівника КР. Рекомендовано зробити скріни основних вікон програми, описати основні їх компоненти та включити цей матеріал до інструкції користувача ПЗ.

Вказівки щодо розробки інформаційно-логічної моделі даних

Реляційна база даних СКБД складається з взаємозв’язаних таблиць. Реляційна таблиця є найпростішою двовимірною таблицею, яка складається з однотипних рядків (записів). Структура таблиці визначається сукупністю полів, типом і розміром даних кожного стовпця, а також унікальним ключем таблиці.

Логічні зв'язки між таблицями в реляційній базі даних реалізуються за рахунок однакових полів в таблицях, що зв'язуються.

Перед створенням бази даних користувач повинен визначити, з яких таблиць повинна складатися БД, які дані потрібно помістити в кожену таблицю; як зв'язати таблиці. Такі питання вирішуються на етапі проектування бази даних.

В результаті проектування повинна бути визначена *логічна структура бази даних*, тобто склад реляційних таблиць, їх структура й зв'язки між таблицями.

Етапи проектування та побудови бази даних

Перш ніж створювати базу даних необхідно описати предметну область, що повинно охоплювати реальні об'єкти й процеси, мати всю необхідну інформацію для задоволення передбачуваних запитів користувача й визначати потреби в обробці даних. На основі такого опису на етапі проектування бази даних здійснюється визначення складу і структури даних предметної області, які повинні знаходитися в БД та забезпечувати виконання необхідних запитів і задач користувача. Структура даних предметної області може відображатися інформаційно-логічною моделлю. На основі цієї моделі легко створюється реляційна база даних.

Існують два підходи, що можна застосувати для розробки моделі даних. В першому підході спочатку визначаються основні задачі, для рішення яких будується база, і виявляються потреби задач в даних. При другому підході відразу встановлюються типові об'єкти предметної області. Найбільш раціонально сполучення обох підходів. Це обґрунтовується тим, що на початковому етапі, як правило, немає вичерпних відомостей про всі задачі. Використання такої технології тим більш виправдане, що гнучкі засоби створення реляційної бази даних дозволяють на будь-якому етапі розробки внести зміни в базу даних і модифікувати її структуру без збитків для введених раніше даних.

В процесі розробки моделі даних необхідно виділити інформаційні об'єкти, відповідні вимогам нормалізації даних і визначити зв'язки між ними. Ця модель дозволяє створити реляційну базу даних без дублювання, в якій забезпечується одноразове введення даних при початковому завантаженні і корегуванні. В такій БД СКБД забезпечує цілісність даних при внесенні змін. При визначенні логічної структури реляційної бази даних на основі моделі кожний інформаційний об'єкт адекватно відображається реляційною таблицею, а зв'язки між таблицями відповідають зв'язкам між інформаційними об'єктами.

В процесі створення БД спочатку здійснюється конструювання таблиць, відповідних інформаційним об'єктам побудованої моделі даних. Далі може створюватися схема даних, в якій фіксуються існуючі логічні зв'язки між таблицями. Ці зв'язки відповідають зв'язкам інформаційних об'єктів. В схемі даних можуть бути задані параметри забезпечення цілісності БД, якщо модель даних була розроблена в відповідності з вимогами нормалізації. Цілісність даних означає, що в БД встановлені й коректно підтримуються взаємозв'язки між записами різних таблиць при завантаженні, додаванні й вилученні записів у зв'язаних таблицях, а також при зміні значень ключових полів.

Введення даних, що містяться в документах предметної області здійснюється після формування схеми даних.

Схема, що наведена на рисунку 1 ілюструє етапи проектування і створення БД.

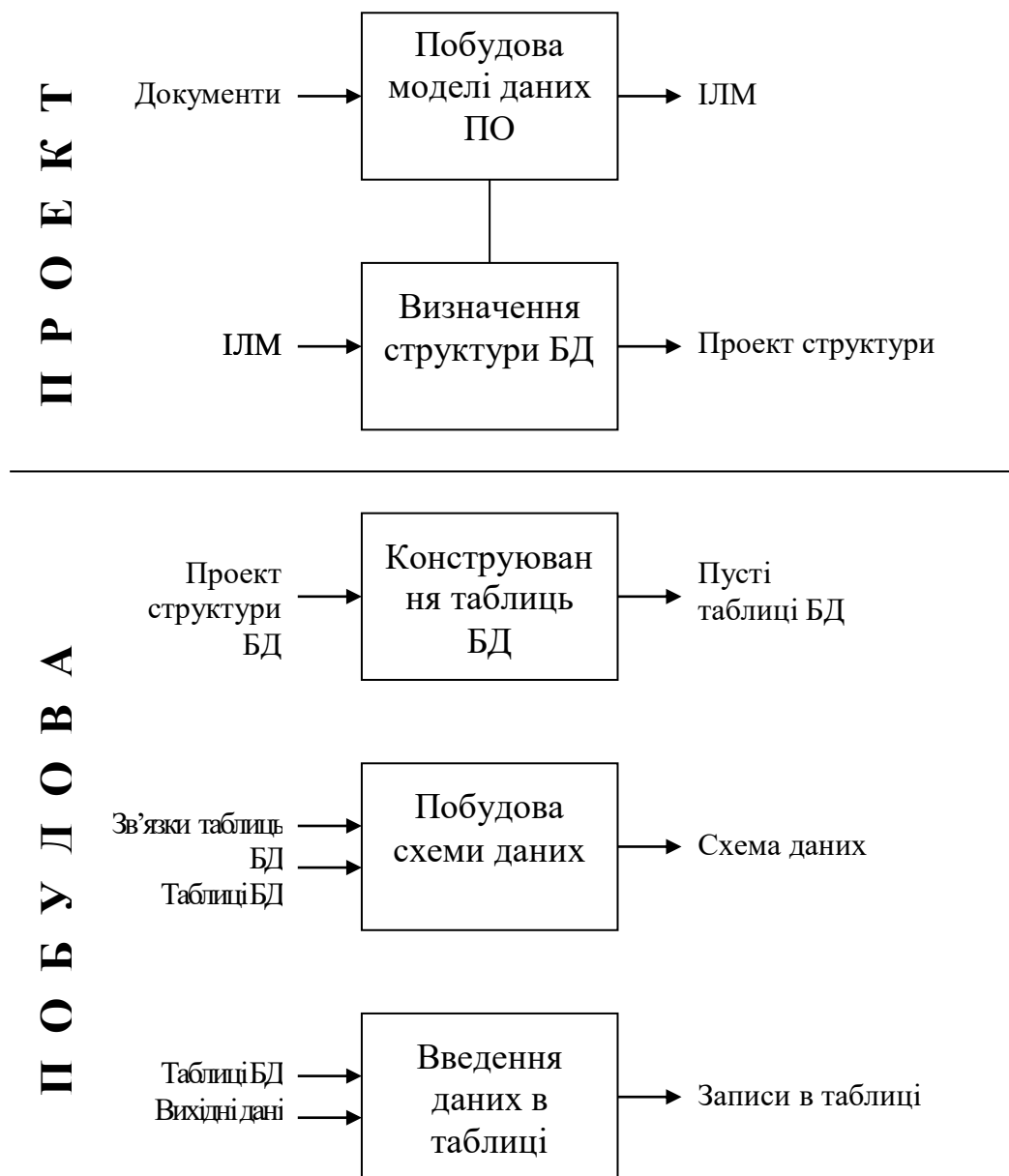


Рисунок 1 - Етапи проектування і створення БД

Побудова інформаційно-логічної моделі даних

Інформаційно-логічна модель (ІЛМ) відображає дані предметної області в вигляді сукупності інформаційних об'єктів і зв'язків між ними. Ця модель представляє дані, що підлягають зберіганню в базі даних.

Інформаційні об'єкти

Інформаційний об'єкт — це інформаційний опис деякої суттєвості реального об'єкту, процесу, явища або події. Інформаційний об'єкт утворюється сукупністю логічно взаємозв'язаних реквізитів, що встановлюють якісні і

кількісні характеристики деякої суттєвості предметної області. Прикладами інформаційних об'єктів можуть бути — ТОВАР, ПОСТАЧАЛЬНИК, ЗАМОВНИК, ПОСТАВКА, ВІДВАНТАЖЕННЯ СПІВРОБІТНИК, ВІДДІЛ, СТУДЕНТ, ВИКЛАДАЧ, КАФЕДРА і т.п.

Інформаційні об'єкти виділяються на основі опису предметної області шляхом визначення функціональних залежностей між реквізитами. Сукупність реквізитів інформаційного об'єкту повинна відповідати вимогам нормалізації.

Кожному інформаційному об'єкту потрібно присвоїти унікальне ім'я, наприклад, СТУДЕНТ, ПРЕДМЕТ, ВИКЛАДАЧ, КАФЕДРА.

Інформаційний об'єкт має безліч реалізацій — примірників. Наприклад, кожний примірник об'єкту СТУДЕНТ представляє конкретного студента. Примірник утворюється сукупністю конкретних значень реквізитів і повинен однозначно визначатися (ідентифікуватися) значенням ключа інформаційного об'єкту, що складається з одного або декількох ключових реквізитів. Таким чином реквізити поділяються на ключові й описові. Останні є функціонально залежними від ключа.

Функціональна залежність реквізитів має місце в тому випадку, якщо одному значенню ключа відповідає тільки одне значення описового (залежного) реквізиту.

Зауваження. При виявленні функціональних залежностей реквізитів не розглядаються арифметичні залежності (наприклад, вартість від кількості), оскільки встановлюється тільки функціональна залежність, яка визначає зв'язки описових і ключових реквізитів, і на основі якої виявляється реквізитний склад кожного інформаційного об'єкту.

При графічному зображенні моделі даних кожний інформаційний об'єкт представляється прямокутником з позначенням його імені й ідентифікатора - ключа. Приклад такого зображення для інформаційних об'єктів ТОВАР і ПОСТАЧАННЯ показаний на Рисунку 2. Тут KODT (код товару) — простий ключ об'єкту ТОВАР, а KODT+KPOST (код постачальника) — складний ключ об'єкту ПОСТАЧАННЯ.

Товар	Постачання
KODT	KODT+KPO

Рисунок 2 - Приклад графічного зображення інформаційних об'єктів з простим і складеним ключем

Реквізити кожного інформаційного об'єкту повинні відповідати вимогам нормалізації:

- 1) інформаційний об'єкт повинен містити унікальний ідентифікатор (ключ); Ключ є *простим*, якщо він складається з одного реквізиту або *складеним*, якщо з декількох;
- 2) всі описові реквізити повинні бути взаємозалежними, тобто між ними не може бути функціональних залежностей;
- 3) всі реквізити, що входять в складений ключ, повинні бути також взаємозалежні;
- 4) кожний описовий реквізит повинен функціонально повно залежати від ключа, тобто кожному значенню ключа відповідає тільки одне значення описового реквізиту;
- 5) при складеному ключі описові реквізити повинні залежати цілком від всієї сукупності реквізитів, що утворюють ключ;
- 6) кожний описовий реквізит не може залежати від ключа транзитивно, тобто через інший проміжний реквізит.

Зауваження. У випадку транзитивної залежності між реквізитами можна виконати розщеплення сукупності реквізитів з утворенням двох інформаційних об'єктів замість одного.

Виділення інформаційних об'єктів предметної області

Процес виділення інформаційних об'єктів предметної області, що відповідають вимогам нормалізації, може проводитися на основі інтуїтивного або формального підходу.

При інтуїтивному підході легко можуть бути виявлені інформаційні об'єкти, відповідні реальним об'єктам. Але отримана при цьому інформаційно-логічна модель, як правило, вимагає подальших перетворень, наприклад, перетворення багато-багатозначних (М: М) зв'язків між об'єктами. При такому підході можливі істотні помилки, якщо буде відсутній достатній досвід. Наступна перевірка виконання вимог нормалізації звичайно призводить до необхідності уточнення інформаційних об'єктів.

Розглянемо формальні правила, що можуть бути використані для виділення інформаційних об'єктів, що відповідають вимогам нормалізації:

1. На основі опису предметної області виявити документи і їхні реквізити, що підлягають зберіганню в БД.

2. Визначити функціональні залежності між реквізитами.

Зауваження. Функціональну залежність реквізитів можна зобразити графічно в вигляді ліній зі стрілками, що ідуть від ключового реквізиту до описового (залежного). Ці залежності доцільно відобразити безпосередньо в таблиці, де подано склад реквізитів, згуртованих за документами.

3. Вибрати всі залежні реквізити і вказати для них ключові реквізити (один або декілька).

Зауваження. В випадку транзитивної залежності деякі реквізити є водночас залежними й ключовими та відповідно представлені в групі залежних і ключових.

4. Згрупувати реквізити, залежні від однакових ключових реквізитів. Отримані групи залежних реквізитів разом з ключовими реквізитами утворюють інформаційні об'єкти.

Після виділення інформаційних об'єктів треба дати їх заключний опис. В такому описі може бути їх змістове визначення.

Приклади виділення інформаційних об'єктів

Розглянемо виділення інформаційних об'єктів на прикладі предметної області "Облік успішності та відвідування студентів".

Опис предметної області

Нехай необхідно побудувати БД, що містить інформацію про відвідування на успішність учнів:

- Список факультетів закладу;
- Список кафедр;
- Список груп;
- Список існуючих спеціальностей;
- Списки студентів груп;
- Перелік предметів, що вивчаються;
- Викладацький склад кафедр, що забезпечують навчальний процес;
- Дані про відвідування студентами занять;
- Результати здачі іспитів (заліків) по кожному з проведених занять.

В результаті аналізу предметної області виявляються документи-джерела даних для створення БД.

Зв'язки інформаційних об'єктів

Наступним кроком проектування після виявлення інформаційних об'єктів є визначення зв'язків між ними. Зв'язок встановлюється між двома інформаційними об'єктами. Зв'язки, як правило, визначаються природою реальних об'єктів, процесів або явищ, що відображаються цими інформаційними об'єктами.

Зв'язки між об'єктами існують, якщо логічно взаємозв'язані примірники цих інформаційних об'єктів

Типи зв'язків інформаційних об'єктів

Зв'язки інформаційних об'єктів можуть бути різного типу:

- Одно-однозначні(1:1),
- Одно-багатозначні (1: M),
- Багато-багатозначні (M:M).

Одно-однозначні зв'язки (1: 1) мають місце, коли кожному примірнику першого об'єкту (А) відповідає тільки один примірник другого об'єкту (В) і

навпаки, кожному примірнику другого об'єкту (В) відповідає тільки один примірник першого об'єкту (А). Слідє помітити, що такі об'єкти легко можуть бути об'єднані в один, структура якого утворюється об'єднанням реквізитів обох вхідних об'єктів, а ключовим реквізитом може бути вибраний будь-який з альтернативних ключів, тобто ключів вхідних об'єктів. Графічне зображення одно-однозначного зв'язку наведено на рисунку 3.

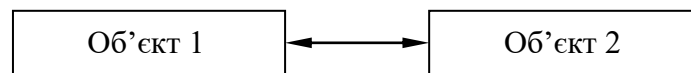
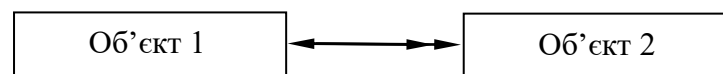


Рисунок 3 - Графічне зображення одно-однозначного зв'язку

Одно-багатозначні зв'язки (1: М) - це такі зв'язки, коли кожному примірнику одного об'єкту (А) може відповідати декілька примірників іншого об'єкту (В), а кожному примірнику другого об'єкту (В) може відповідати тільки один примірник першого об'єкту (А). Графічне зображення одно-багатозначного зв'язку наведено на рисунку 4



Головний об'єкт

Підлеглий об'єкт

Рисунок 4 - Графічне зображення одно-багатозначного зв'язку

В такому зв'язку А є головним об'єктом, а об'єкт В — підлеглим, тобто має місце ієрархічна підпорядкованість об'єкта В об'єкту А.

Багато-багатозначні зв'язки (М: М) — це такі зв'язки, коли кожному примірнику одного об'єкту (А) можуть відповідати декільком примірників другого об'єкту (В) і навпаки, кожному примірнику другого об'єкту (В) може відповідати теж декількох примірників першого об'єкту (А).

Графічне зображення зв'язку типу М : М показане на рисунку 5.

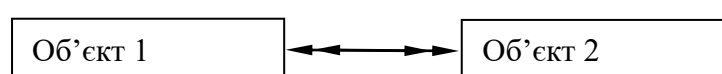


Рисунок 5. - Графічне зображення багато-багатозначного зв'язку

Приклади визначення зв'язків між інформаційними об'єктами

Навчальний заклад містить декілька факультетів. Кожному факультетові належить декілька кафедр. Тому між об'єктами ФАКУЛЬТЕТ та КАФЕДРА можна встановити зв'язок “один до багатьох”.

Саме такий одно-багатозначний зв'язок буде між об'єктами ГРУПА та СТУДЕНТ, КАФЕДРА та ВИКЛАДАЧ, КАФЕДРА та СПЕЦІАЛЬНІСТЬ та ін.

На кожному факультеті є декан. Декан є викладачем, який викладає на цьому факультеті. Тому між об'єктами ВИКЛАДАЧ та ФАКУЛЬТЕТ можна встановити зв'язок “один до одного”. Саме такий зв'язок буде між об'єктами КАФЕДРА та ВИКЛАДАЧ (Зав. кафедрою), ГРУПА та СТУДЕНТ (Староста групи) або ГРУПА та ВИКЛАДАЧ (Куратор групи).

Зв'язок “багато до багатьох” буде встановлено, наприклад, між об'єктами ПРЕДМЕТ_ДЛЯ_СПЕЦІАЛЬНОСТІ та УСПІШНІСТЬ. Кожному ПРЕДМЕТУ буде відповідати багато записів в об'єкті УСПІШНІСТЬ (студентів багато).

Також такий зв'язок буде між об'єктами ПРЕДМЕТ_ДЛЯ_СПЕЦІАЛЬНОСТІ та ПРЕДМЕТ. Об'єкт ПРЕДМЕТ описує всі предмети, що читаються на даній кафедрі, а об'єкт ПРЕДМЕТ_ДЛЯ_СПЕЦІАЛЬНОСТІ описує предмети, що читаються для конкретної спеціальності у конкретному семестрі.

У наведеній нижче таблиці 1 представлено головні та підлеглі об'єкти в зв'язках між ними.

Таблиця 1 - Головні та підлеглі об'єкти та зв'язки між ними

№	Головний об'єкт	Підлеглий об'єкт	Тип зв'язку
1	ФАКУЛЬТЕТ	КАФЕДРА	1:M
2	КАФЕДРА	СПЕЦІАЛЬНІСТЬ	1:M
3	КАФЕДРА	ПРЕДМЕТ	1:M
4	КАФЕДРА	ВИКЛАДАЧ	1:M
5	СПЕЦІАЛЬНІСТЬ	ГРУПА	1:M

6	СПЕЦІАЛЬНІСТЬ	ПРЕД_ДЛЯ_СПЕЦ	1:M
7	ПРЕДМЕТ	ПРЕД_ДЛЯ_СПЕЦ	1:M
8	ВИКЛАДАЧ	ПРЕД_ДЛЯ_СПЕЦ	1:M
9	ГРУПА	СТУДЕНТ	1:M
10	СТУДЕНТ	ВІДВІДУВАННЯ	1:M
11	СТУДЕНТ	УСПІШНІСТЬ	1:M
12	ПРЕД_ДЛЯ_СПЕЦ	УСПІШНІСТЬ	N:M

Інформаційно-логічна модель предметної області “Система обліку відвідування та успішності студентів”

Інформаційно-логічна модель описаної предметної області, побудована в відповідності з виявленими інформаційними об'єктами і зв'язками, представлена на рисунку 6.

Інформаційно-логічна модель наведена в канонічному вигляді, з причини, що об'єкти розміщені за рівнями. На нульовому рівні містяться об'єкти, не підлеглі жодним іншим об'єктам. Рівень інших об'єктів визначається найбільш довгим шляхом до об'єкту від нульового рівня. Таке розміщення об'єктів дає уявлення про їхню ієрархічну підлеглість, робить модель більш наочною та полегшує розуміння зв'язків між об'єктами.

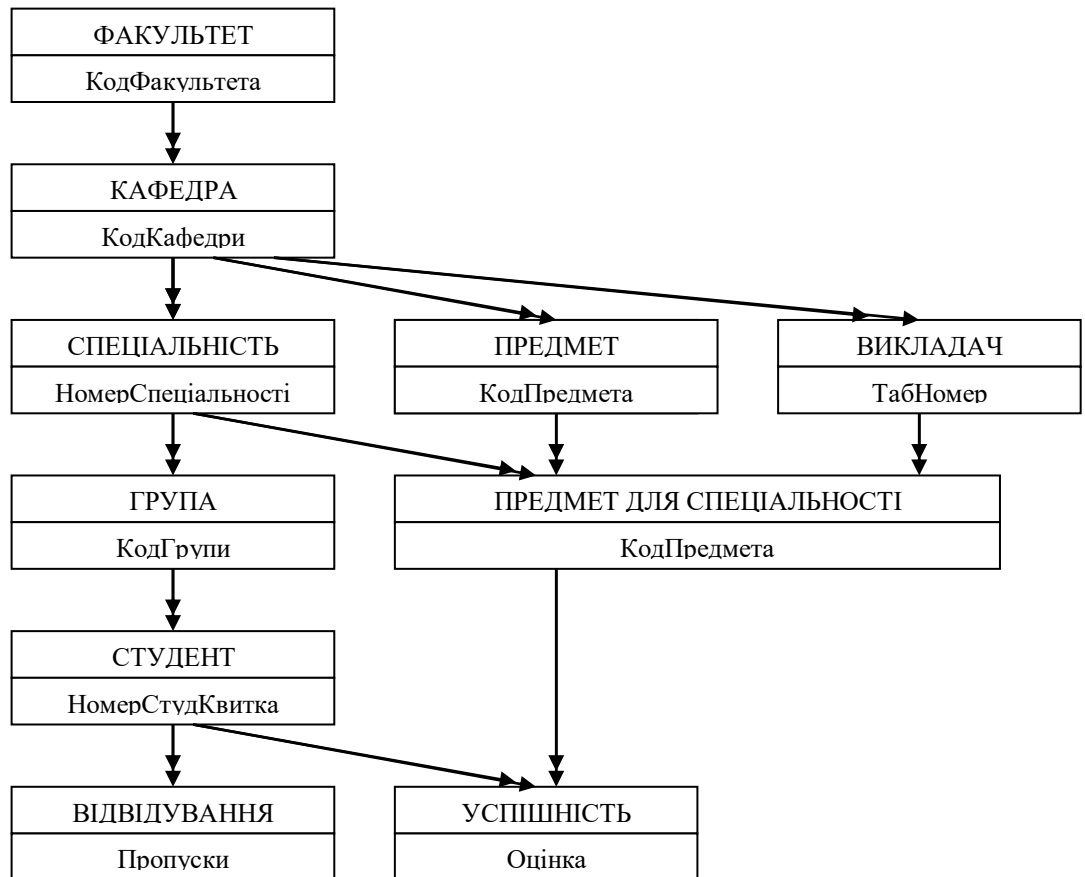


Рисунок 6 - Інформаційно-логічна модель предметної області
“Система обліку відвідування та успішності студентів”

Логічна структура реляційної бази даних

Логічна структура реляційної бази даних Access є адекватним відображенням отриманої інформаційно - логічної моделі, що вимагають перетворень. Кожний інформаційний об'єкт моделі даних відображається відповідною реляційною таблицею. Структура реляційної таблиці визначається реквізитним складом об'єкту, де кожний стовпець (поле) відповідає одному з реквізитів. Ключові реквізити об'єкту утворюють унікальний ключ реляційної таблиці. Для кожного стовпця задається формат і розмір даних. Рядки (записи) таблиці відповідають примірникам об'єкту і формуються при завантаженні таблиці.

Зв'язки, між об'єктами моделі даних реалізуються однаковими реквізитами - ключами зв'язку - у відповідних таблицях при цьому ключем зв'язку завжди повинен бути ідентифікатор головного об'єкту в зв'язку.

Відзначимо, що наявність ключа зв'язку в підлеглий таблиці є засобом опису зв'язку при одно-багатозначних зв'язках.

Визначення одно-багатозначних зв'язків в схемі даних повинно здійснюватися згідно з побудованою моделлю даних. Зовнішній вигляд схеми даних практично співпадає з графічним поданням інформаційно-логічної моделі. Для моделі даних, побудованої в розглянутому прикладі, логічна структура бази даних у вигляді схеми даних наведена на рисунку 7.

На цій схемі прямокутники відображають таблиці бази даних з повним списком полів, а зв'язки показують, по яким полям здійснюються взаємозв'язки таблиць. Імена ключових полів для наочності виділені і знаходяться в верхніх частинах повного списку полів кожної таблиці.

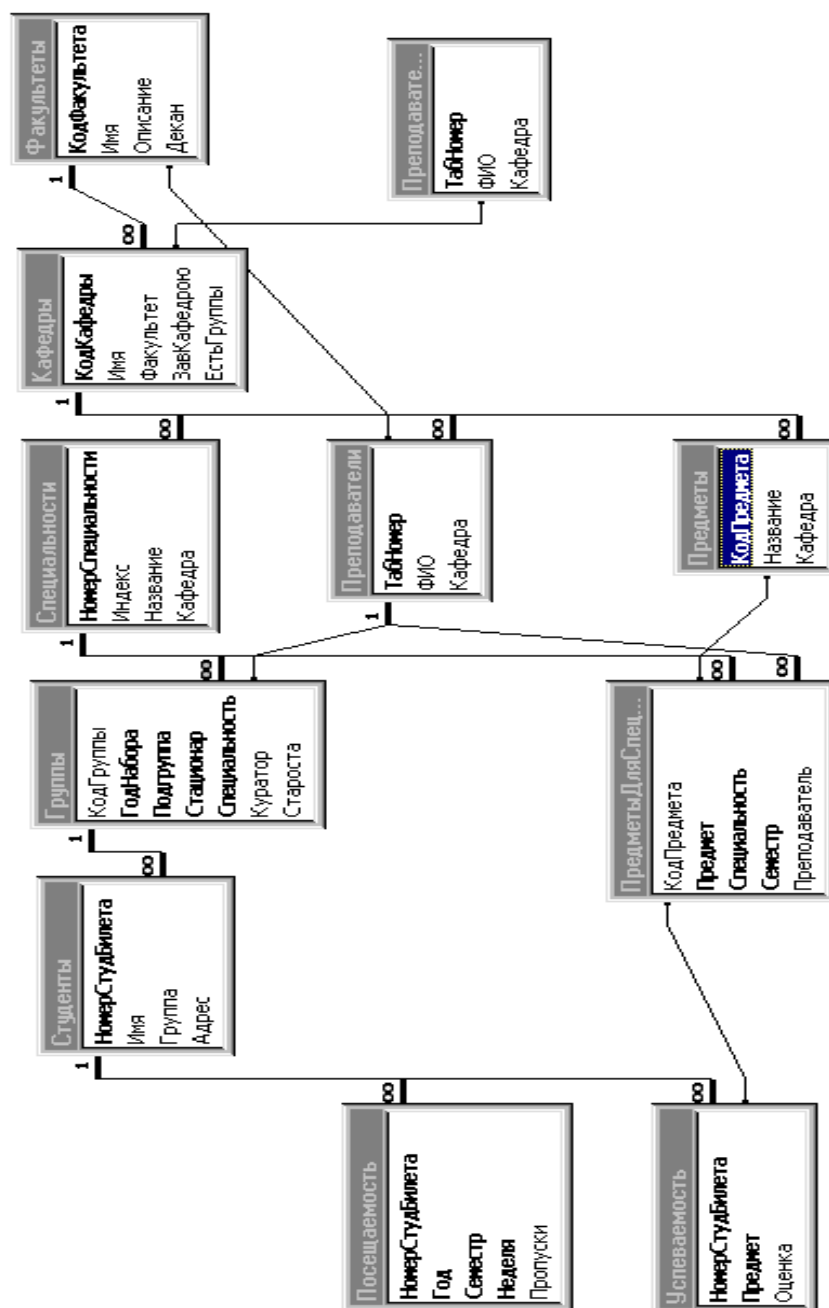


Рисунок 7 - Логічна структура реляційної бази даних предметної області
“Система обліку відвідування та успішності студентів”

Зміст розділів пояснювальної записки

Титульний аркуш, завдання

Приклад оформлення титульного листа наведений у додатку Б, а оформлення завдання у додатку А.1 та А.2.

Пояснювальна записка. Зміст

Зміст подають на початку роботи. Він містить найменування та номери початкових сторінок вступу, усіх розділів, підрозділів та пунктів.

Всі аркуші пояснювальної записки, оформлені в рамці з основним написом (додаток В) згідно ГОСТ 2.104-68. У великому полі основного напису (додаток В) вказується шифр розробника роботи (рисунок 8).

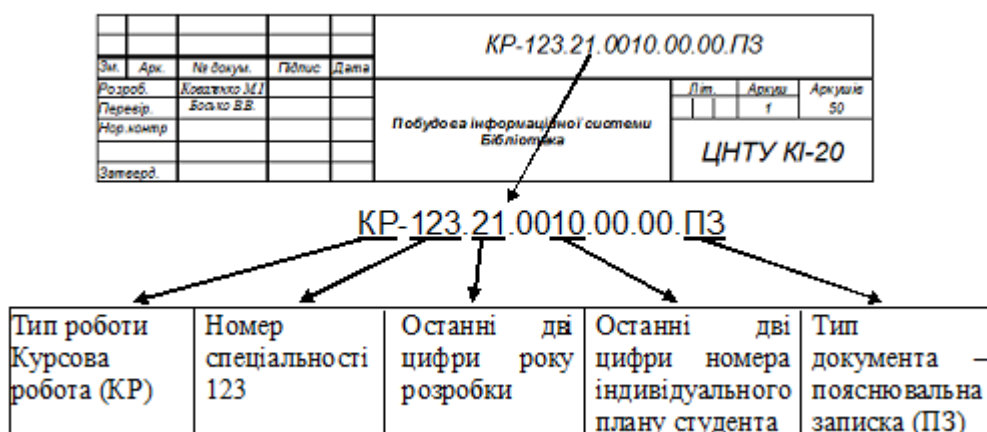


Рисунок 8 – Шифр розробника курсової роботи

Перелік умовних позначень, символів, одиниць і термінів вводиться, якщо в роботі вжита специфічна термінологія, маловідомі скорочення, нові символи тощо. Перелік друкується двома колонками, у яких, ліворуч за абеткою наводять скорочення, праворуч – їх детальну розшифровку. Якщо спеціальні терміни, символи, позначення і таке інше повторюється менше трьох разів, перелік не складають, а їх розшифрування наводять у тексті при першому згадуванні.

Вступ

У вступі розкривається сутність задачі та її значущість, підстави і вихідні дані для розробки теми. Вступ включає актуальність теми, мету і завдання роботи, практичне значення отриманих результатів.

Висвітлення актуальності не повинно бути багатослівним. Досить кількома реченнями висловити головне – сутність проблеми або науково-практичної задачі. та стислий висновок щодо розробки програмного забезпечення.

1 Призначення та область використання

Наводяться основні параметри (характеристики) системи, що розробляється, призначення системи і можливі області застосування.

Структура розділу:

1.1 Призначення системи

1.2 Область застосування

2 Перегляд аналогічних існуючих систем

Здійснюється огляд існуючого програмного забезпечення, систем, приладів, основних напрямків розвитку баз даних та існуючих СКБД. Наводиться аналіз їх характеристик, властивостей, недоліків і переваг, на основі чого робиться висновок про доцільність (необхідність) проектування системи (розробки програмного забезпечення) відповідно до теми КР. Здійснюється короткий опис обраного інструментарію. Обґрунтовуються основні принципи розробки програмного забезпечення, його переваги над іншими для вирішення обраної задачі. Здійснюється чітка постановка задачі.

Структура розділу:

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми КР

Аналіз переваг та недоліків існуючих рішень. Обґрунтування необхідності розробки системи за темою КР з урахуванням проведеного аналізу.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Коротка характеристика методів, апаратних засобів, середовища розробки та СКБД. Обґрунтування вибору по декільком параметрам. Обґрунтування вибору мови програмування.

2.3 Розгорнута постановка завдання

Перерахування основних пунктів розробки, які визначають тему та склад КР (обсяг 1 сторінка).

3 Опис і обґрунтування проектних рішень

Обґрунтовуються основні принципи проектування системи, методика проектування.

Описується хід теоретичної побудови моделі проекту, приводиться її обґрунтування. Робляться стислі висновки.

При необхідності наводиться математична модель розробленої системи.

В цьому розділі наводяться наступні підрозділи, що містять відповідні схеми та їх детальний опис:

3.1 Опис функціонування системи

3.2 Розробка структурної схеми

Розробка структурної схеми пристрою, або системи (повна). Опис побудови логічної структури бази даних. На виході інформаційно-логічна модель, схема даних.

3.3 Розробка функціональної схеми

Розробка функціональної схеми системи. Побудова інтерфейсу, форми, звіти.

3.4 Розробка діаграми процесів

Діаграма процесів, які відбуваються в системі. Принципова схема пристрою (якщо розроблюється конкретний пристрій або інші схеми, якщо такі є).

4 Реалізація роботи. Розрахунки та експериментальні дані, що підтверджують вірність проектних і програмних рішень.

Наводяться розрахунки й експериментальні матеріали, які підтверджують вірність рішень, наведених у кваліфікаційній бакалаврській роботі.

Наводяться алгоритми, які реалізують функціональність системи.

Розробка класів, функцій, таблиць баз даних, взаємодії компонентів системи. Реалізація зв'язних, системних та програмних інтерфейсів. Компонівка частин системи.

В цьому розділі наводяться наступні підрозділи:

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

– Опис блок-схем алгоритмів роботи пристрою, системи. Запити (QBE, SQL), макроси та ін.

4.2 Захист розробленого програмного забезпечення

В підрозділі «Захист розробленого програмного забезпечення» розглядаються механізми або алгоритми захисту розробленого програмного забезпечення (коротко наводиться конкретний алгоритм або метод).

5 Впровадження системи в промислову експлуатацію

Розробляється і описується методика інтеграції компонентів розробленого програмного забезпечення в існуючу апаратну систему. Описується інструкція користувача.

В разі необхідності наводяться скріншоти розробленого програмного забезпечення.

6 Основні висновки.

Дається стисла характеристика розробленої системи за всіма основними параметрами. Містить найбільш важливі результати, одержані в ході виконання КР.

Список використаних джерел

Наводиться нумерований список літератури та інформаційних джерел, які використано для реалізації курсової роботи.

Кількість використаних джерел (літературних джерел, патентів, нормативно-технічних документів, адреси сайтів Інтернету), повинно бути не

менше 30. Список використаних джерел оформляється згідно Національного стандарту України ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання».

Додатки

За необхідності до додатків доцільно включити допоміжний матеріал, необхідний для повноти сприйняття роботи (інструкції, методики, опис алгоритмів дій); ілюстрації допоміжного характеру (таблиці, діаграми, схеми, звіти, запити тощо); формули і розрахунки, зразки анкет, тестів та ін. На кожний додаток повинно бути посилання в тексті.

Вимоги до оформлення текстових документів пояснювальної

записки

Текст пояснювальної записки оформляється у відповідності з вимогами ДСТУ 3008-95 та ДСТ 2.105-95.

Форми і правила виконання текстових документів (відомостей, пояснювальних записок) встановлені ДСТУ 3008-95. У відповідності з листом Міністра освіти України №1/9-73 від 01.03.99р. кваліфікаційні бакалаврські роботи повинні виконуватися лише державною мовою. Скорочення слів у тексті ПЗ здійснюється у відповідності з ДСТУ 3582–97. Інформація та документація. Скорочення слів в українській мові в бібліографічному описі. Загальні вимоги та правила.

ПЗ виконується на аркушах формату А4 (210х297 мм) ДСТ 2.301-68.

Титульний аркуш оформлюється по формі (Додаток Б).

Заголовний та наступні аркуші документа. Зміст оформлюються як заголовні аркуші. Оформлення рамок заголовних та наступних після них аркушів див. в Додатку В. Форма, розміри, зміст і порядок заповнення основного напису й додаткових граф до нього наведені в міжнародному стандарті ДСТ 2.104-2006:

- форма 2 (40 x 185 мм) – для заголовних аркушів текстових документів;
- форма 2а (15 x 185 мм) – для наступних аркушів текстових документів.

Вимоги до оформлення тексту

Текст пояснювальної записки друкується шрифтом **Times New Roman 14** розміру, відстань між рядками – 1.5 інтервал, вирівнювання тексту за шириною. Виключенням є приклади частин програмного коду, для них необхідно застосовувати шрифт **Courier New 10** розміру, полуторний інтервал та вирівнювання тексту по лівому краю.

Абзацний відступ повинен бути однаковим впродовж усього тексту і дорівнювати 1,5 см.

Відступи тексту від рамки: зверху і знизу не менше 10 мм; зліва і справа не менше 3-5 мм. Відступи повинні бути однаковими впродовж всього тексту.

Заголовки розділів ПЗ розміщуються по центру, друкуються великими буквами та виділяють жирним шрифтом. Структурні елементи "ЗМІСТ", "ВСТУП", "ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ ТА СПЕЦІАЛЬНИХ ТЕРМІНІВ" та "СПИСОК ЛІТЕРАТУРИ" не нумерують.

Заголовки підрозділів, пунктів та підпунктів розміщуються з абзацу малими буквами, крім першої великої, виділяються жирним шрифтом, та вирівнюються по ширині.

Відстань між заголовком розділу та подальшим і/або попереднім текстом (а також відстань між заголовком розділу та підрозділу) дорівнює двом рядкам (що також можна виставити інтервалом рівним 50 пт (пунктів)).

Відстань між заголовком підрозділу та подальшим і/або попереднім текстом дорівнює одному рядку (або ж інтервалу рівному 25 пт).

Всі інші заголовки, що знаходяться всередині підрозділів і не відображаються у змісті, розміщуються з абзацу малими буквами, крім першої великої, виділяються жирним шрифтом, та вирівнюються по ширині і не містять відступів між подальшим та попереднім текстом.

Заголовки слід друкувати без крапки в кінці. Якщо заголовок складається з двох і більше речень, їх розділяють крапкою. Перенесення слів у заголовку розділів не допускається.

Розділи і підрозділи повинні мати заголовки. Пункти і підпункти можуть мати заголовки.

Не допускається розміщувати назву розділу, підрозділу, а також пункту й підпункту в нижній частині сторінки, якщо після неї розміщено тільки один рядок тексту.

Нумерація розділів, підрозділів, пунктів, підпунктів. Розділи, підрозділи, пункти, підпункти звіту слід нумерувати арабськими цифрами.

Розділи звіту повинні мати порядкову нумерацію і позначатися арабськими цифрами без крапки, наприклад, 1, 2, 3 і т. д.

Наприклад: **1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ**

Підрозділи звіту повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, відокремлених крапкою. Після номера підрозділу крапку не ставлять, наприклад 1.1, 1.2 і т. д.

Наприклад: **2.3 Розгорнута постановка завдання**

Пункти повинні мати порядкову нумерацію в межах кожного розділу або підрозділу. Номер пункту складається з номера розділу і порядкового номера пункту або з номера розділу, порядкового номера підрозділу та порядкового номера пункту, відокремлених крапкою. Після номера пункту крапку не ставлять, наприклад, 1.1, 1.2, або 1.1.1, 1.1.2 і т.д.

Переліки. Переліки, за потреби, можуть бути наведені всередині пунктів або підпунктів. Перед переліком ставлять двокрапку.

Перед кожною позицією переліку слід ставити малу літеру української абетки з дужкою, або, не нумеруючи – дефіс (перший рівень деталізації).

Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації).

Приклад:

- а) _____;
- б) _____;
- 1) _____;
- 2) _____;
- в) _____;

Переліки першого рівня деталізації друкують малими літерами з абзацного відступу, другого рівня – з відступом відносно місця розташування переліків першого рівня.

В тексті документу, за виключенням формул, таблиць та рисунків, не допускається:

– застосовувати математичний знак мінус(-) перед від’ємним значенням величин (слід писати слово “мінус”);

– застосовувати знак “ \varnothing ” для позначення діаметру (слід писати слово “діаметр”). При зазначенні розміру діаметру на кресленнях, які розташовані в тексті документу, перед розмірним числом слід писати знак “ \varnothing ”;

– застосовувати без числових значень математичні знаки, наприклад $>$ (більше), $<$ (менше), $=$ (дорівнює), \geq (більше або дорівнює), \leq (менше або дорівнює), а також знаки № (номер), % (процент);

– застосовувати індекси стандартів, технічних вимог та інших документів без реєстраційного номеру.

Написання чисел в тексті виконують в відповідності зі стандартом СТ РЕВ 543-73 “Числа. Правила запису та округлення”.

Числові значення величин в тексті слід вказувати зі ступенем точності, яка необхідна для забезпечення необхідних властивостей виробу, при цьому в ряді величин здійснюється вирівнювання числа знаків після коми. Округлення числових значень величин до першого, другого, третього і т.д. десяткового знаку для різних типорозмірів, марок і т.п. виробів одного найменування повинно бути однаковим. Дробові числа необхідно наводити у вигляді десяткових дробів, за виключенням розмірів в дюймах, які слід записувати $\frac{1}{4}$; $\frac{1}{2}$ ” (але не « $\frac{1}{4}$ », « $\frac{1}{2}$ »).

Якщо неможливо виразити числове значення у вигляді десяткового дробу, допускається записувати в вигляді простого дробу в один рядок через похилу риску, наприклад, $5/32$; $(50A-4C)/(40B+20)$.

Якість друку. Під час виконання ПЗ необхідно дотримуватись рівномірної щільності, контрастності й чіткості зображення впродовж усієї ПЗ. У звіті мають бути чіткі, не розпливчасті лінії, літери, цифри та інші знаки. **Помилки, описки та графічні неточності** допускається виправляти підчищенням або зафарбовуванням білою фарбою і нанесенням на тому ж місці

або між рядками виправленого зображення машинописним способом або від руки. Виправлення повинні здійснюватися ручкою чорного кольору.

Прізвища, назви установ, організацій, фірм та інші власні назви у звіті наводять мовою оригіналу. Допускається транслітерувати власні назви і наводити назви організацій у перекладі на мову звіту, додаючи (при першій згадці) оригінальну назву.

Скорочення слів і словосполучень – відповідно до чинних стандартів з бібліотечної та видавничої справи ДСТУ 3582-97.

Вимоги до оформлення ілюстрацій

Усі графічні матеріали (ескізи, діаграми, графіки, схеми, малюнки, креслення тощо) повинні мати однаковий підпис: «Рисунок».

Ілюстрації слід розміщувати у звіті безпосередньо після тексту, де вони згадуються вперше, або на наступній сторінці. На всі ілюстрації мають бути посилання у звіті.

Креслення, рисунки, графіки, схеми, діаграми, розміщені у звіті, мають відповідати вимогам стандартів "Единой системы конструкторской документации" та "Единой системы программной документации".

Ілюстрації можуть мати назву, яку розміщують під ілюстрацією. За необхідності під ілюстрацією розміщують пояснювальні дані (підрисунковий текст).

Ілюстрація позначається словом "Рисунок ____", яке разом з назвою ілюстрації розміщують після пояснювальних даних, наприклад:

Рисунок 3.1 – Схема розміщення

Між назвою ілюстрації та подальшим текстом повинен бути один порожній рядок.

Ілюстрації слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, відокремлених крапкою, наприклад, «рисунок

3.2» – другий рисунок третього розділу. Крапка в кінці назви рисунку не ставиться.

Якщо ілюстрація не вміщується на одній сторінці, можна переносити її на інші сторінки, вміщуючи назву ілюстрації на першій сторінці, а на всіх наступних напис – "Рисунок __, аркуш __", наприклад:

Рисунок 4.2, аркуш 2

Рисунки розміщують так, щоб їх можна було розглядати без повороту документа. При неможливості виконання цієї вимоги рисунки розміщують так, щоб для їхнього перегляду документ можна було повернути за годинниковою стрілкою, при цьому розміщення назви рисунку не повертається разом з рисунком і розміщується нижче рисунку, паралельно тексту документу.

Вимоги до оформлення таблиць

Цифровий матеріал, як правило, оформлюють у вигляді таблиць відповідно до рисунку 9.

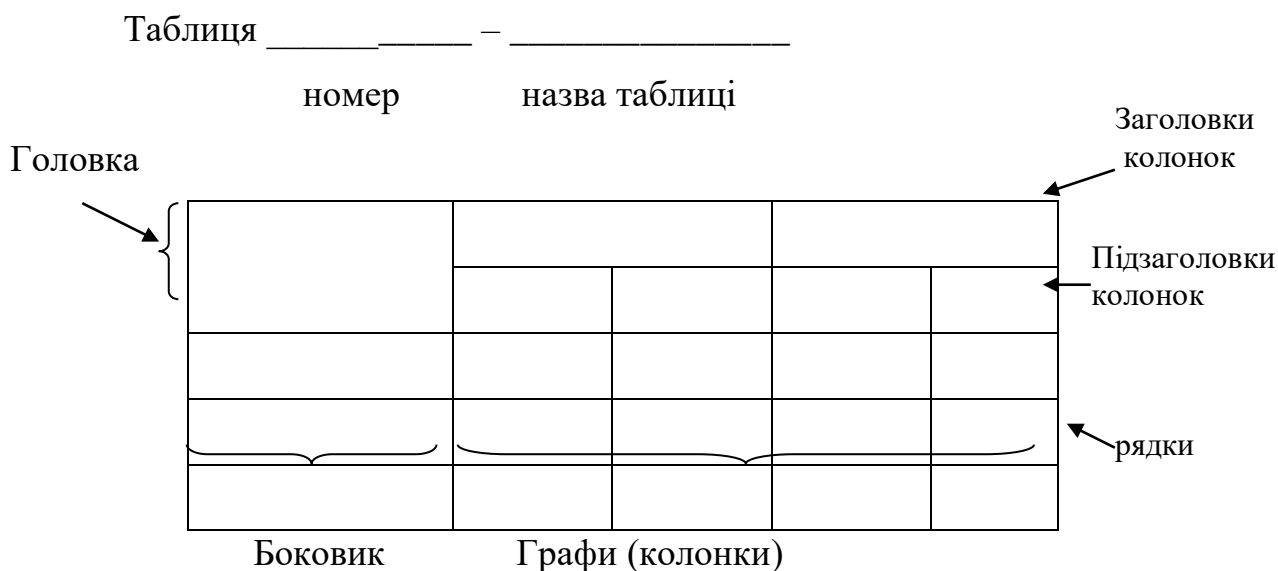


Рисунок 9

Таблицю слід розташовувати безпосередньо після тексту, у якому вона згадується вперше, або на наступній сторінці.

На всі таблиці мають бути посилання в тексті звіту.

Таблиці слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу. Номер таблиці складається з номера розділу і порядкового номера таблиці, відокремлених крапкою, наприклад, таблиця 2.1 – перша таблиця другого розділу. Таблиця може мати назву, яку друкують малими літерами (крім першої великої) і вміщують над таблицею. Назва має бути стислою і відображати зміст таблиці.

Висота рядків таблиці повинна бути не менше 8 мм.

Вимоги до оформлення формул та рівнянь

Формули та рівняння розташовують безпосередньо після тексту, в якому вони згадуються, посередині сторінки.

Вище і нижче кожної формули або рівняння повинно бути залишено не менше одного вільного рядка.

Формули і рівняння у звіті (за винятком формул і рівнянь, наведених у додатках) слід нумерувати порядковою нумерацією в межах розділу.

Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, відокремлених крапкою, наприклад, формула (1.3) – третя формула першого розділу.

Номер формули або рівняння зазначають на рівні формули або рівняння в дужках у крайньому правому положенні на рядку.

Пояснення значень символів і числових коефіцієнтів, що входять до формули чи рівняння, слід наводити безпосередньо під формулою у тій послідовності, в якій вони наведені у формулі чи рівнянні.

Пояснення значення кожного символу та числового коефіцієнта слід давати з нового рядка. Перший рядок пояснення починають з абзацу словом "де" без двокрапки.

Приклад

"Відомо, що:

$$S = F(T), \quad (3.1)$$

де T – базовий код вірусу;

S – зашифровані коди вірусу;

F – функція шифрування вірусу, що довільно вибирається з деякої множини перетворень $\{F\}$ ".

Переносити формули чи рівняння на наступний рядок допускається тільки на знаках виконуваних операцій, повторюючи знак операції на початку наступного рядка. Коли переносять формули чи рівняння на знакові операції множення, застосовують знак "x".

Формули, що йдуть одна за одною й не розділені текстом, відокремлюють комою.

Посилання в тексті пояснювальної записки на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, "у роботах [1 – 7] ...".

При посиланнях на розділи, підрозділи, пункти, підпункти, ілюстрації, таблиці, формули, рівняння, додатки зазначають їх номери.

При посиланнях слід писати: "... у розділі 4 ...", "... дивись 2.1...", "... за 3.3.4 ...", "... відповідно до 2.3.4.1...", "... на рис. 1.3 ...", або "...на рисунку 1.3 ...", "... у таблиці 3.2 ...", "... (див. 3.2) ...", "...за формулою (3.1)...", "... у рівняннях (1.23) – (1.-25) ...", "... у додатку Б...".

Оформлення додатків

КР може містити наступні додатки:

Додаток А – Інформаційно-логічна модель предметної області.

Додаток Б – ER-схеми. Семантичне моделювання предметної області.

Додаток В– Схема даних.

Додаток Г – Приклад побудованих таблиць бази даних.

Додаток Д – Побудова форм, презентація інформаційної системи та побудова інтерфейсу користувача.

Додаток Е – Основні запити SQL.

Додаток Ж – Лістинг програми.

Текст програми (без рамки, приклад титульного аркуша див. у Додатку Е).

Кожен додаток слід починати з нової сторінки з вказівкою зверху посередині сторінки слова “Додаток” та його позначення.

Додаток повинен мати заголовок, який записують симетрично відносно тексту з прописної літери окремим рядком.

Додатки позначають великими літерами українського алфавіту, починаючи з А, окрім літер Г, Є, З, І, Ї, Й, О, Ч, Ь. Після слова “Додаток” йде літера, яка позначає його послідовність.

Ілюстрації в додатках позначають окремою нумерацією арабськими цифрами з додаванням перед цифрою позначення додатка (рисунок 10).

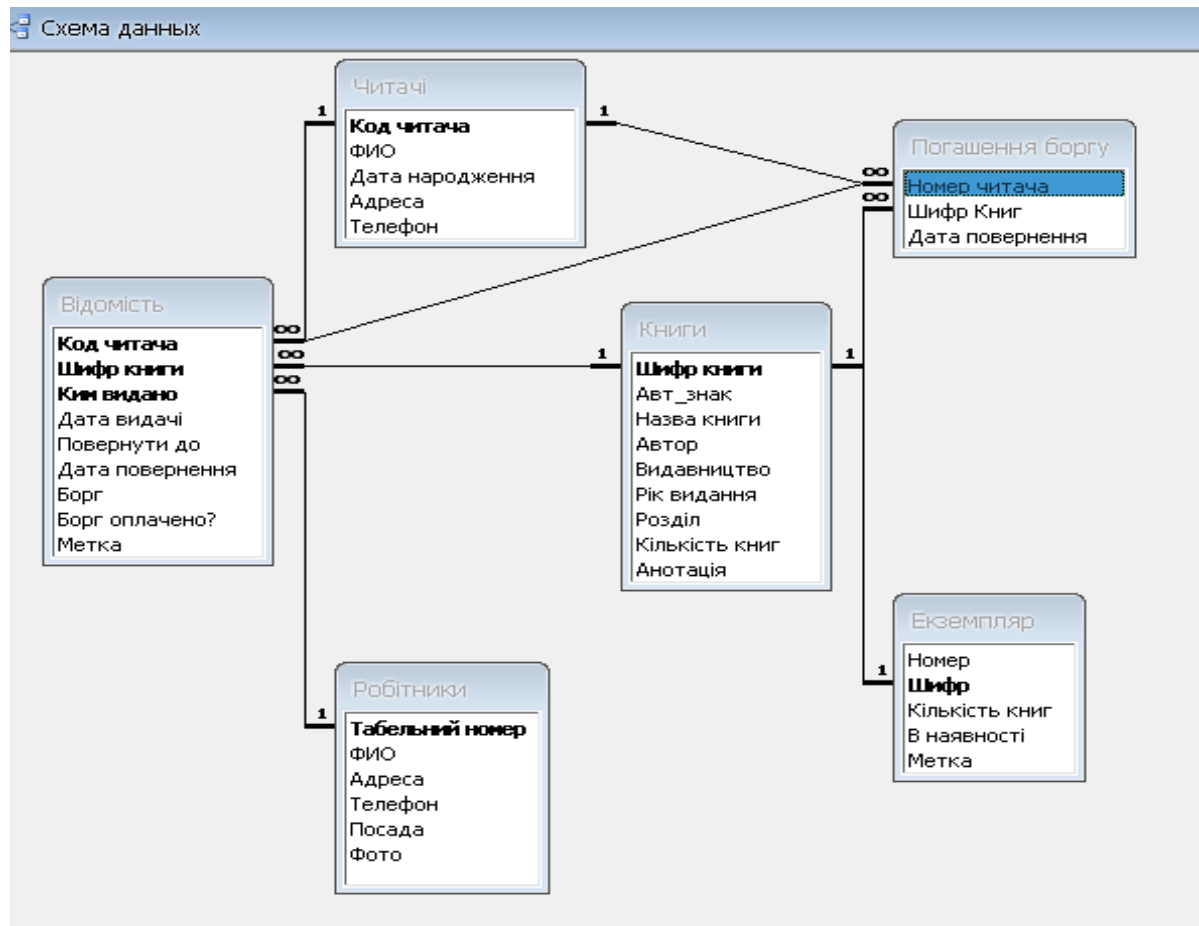


Рисунок А.3 - Схема даних

Рисунок 10 – Приклад рисунків в додатках

Оформлення лістинга програми

Формат тексту лістинга програми: шрифт Courier New, розмір шрифту 10 пт, міжрядковий інтервал 1,0 пт, вирівнювання - ліворуч.

В тексті програми необхідно у вигляді коментарів вказати прізвище та ініціали автора розробки, назву програмного забезпечення (тема курсової роботи) та ін. інформацію для забезпечення авторського права.

Якщо програмне забезпечення складається з декількох модулів, перед лістингом посередині сторінки жирним шрифтом Times New Roman, розмір 14 пт, необхідно вказати назву вихідного файлу (джерело програмного кода). Наприклад, “main.cpp”, “i_timer.h” тощо.

Коментарі є невід’ємною складовою тексту програмного забезпечення. Рекомендовано включати їх до лістингу безпосередньо під час розробки програмного забезпечення (а не після). Вони повинні коротко описувати кожен етап виконання програми (функції, процедури, модулі тощо). Мова коментарів - українська.

Оформлення графічних матеріалів. Умовні позначення

Структурна схема – це сукупність елементарних ланок об'єкта і зв'язків між ними. Під елементарною ланкою розуміють частину об'єкта, системи керування тощо, яка реалізовує елементарну функцію.

Елементарні ланки зображаються прямокутниками, а зв'язки між ними – суцільними лініями зі стрілками, що показують напрям дії ланки.

Функціональна схема – це схема, яка показує логіку роботи системи. Представляє собою схему пристрою, системи, апарату в якій основні вузли (блоки), що утворюють її, зображено прямокутниками та іншими фігурами, а зв'язок між ними показано лініями зі стрілками.

Функціональні схеми можуть виконуватися в менш деталізованому і в більш деталізованому вигляді. У першому випадку на схемі зображають найбільш важливі блоки системи і зв'язки між ними. У другому варіанті схема зображується більш детально, що полегшує її читання та більш повно ілюструє принцип роботи системи.

Діаграма процесів – візуальне представлення графу процесів. Граф процесів є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Процес (дія) є фундаментальною одиницею визначення поведінки системи. Процес отримує множину вхідних сигналів та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Кожен процес може виконуватись один, два, або більше разів під час одного запуску системи. Деякі процеси можуть вимагати певної послідовності.

Процеси зображаються овалами, а зв'язки між ними – вигнутими лініями зі стрілками.

Блок-схема – опис алгоритму у вигляді блоків рішення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних фігур), які позначають такі елементи, як операції, потік, дані тощо.

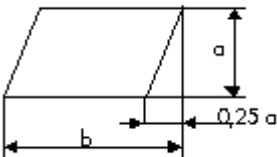
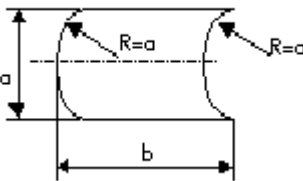
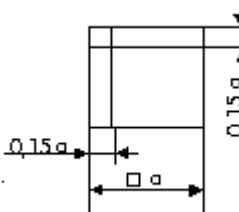
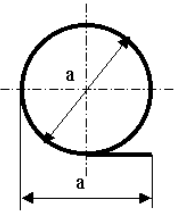
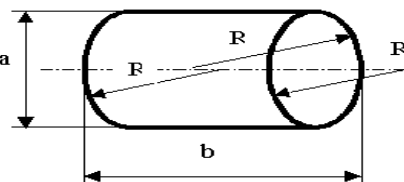
Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

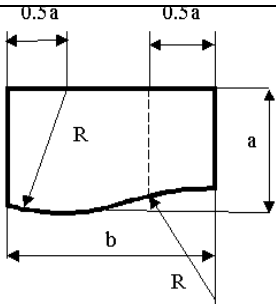
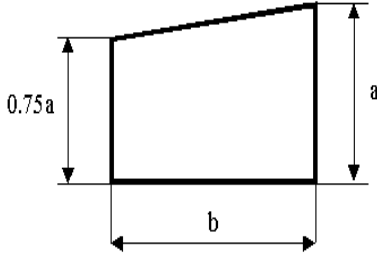
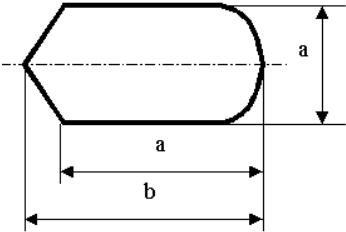
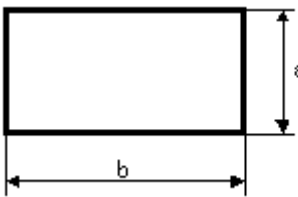
Приклад виконання блок-схеми наведено в Додатку Д.

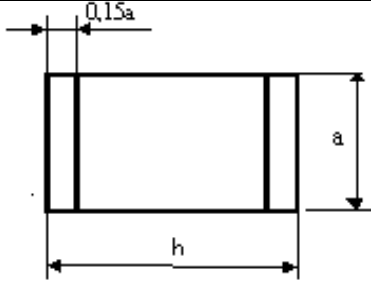
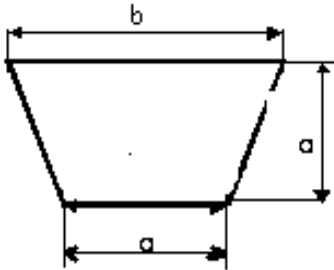
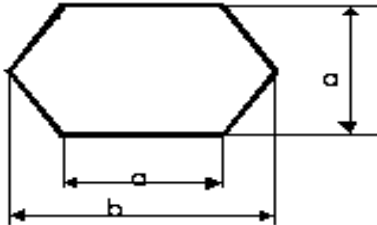
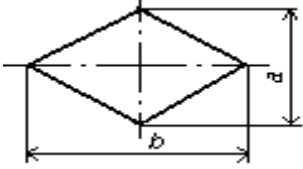
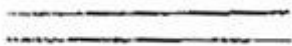
Правила виконання схем алгоритмів і програм

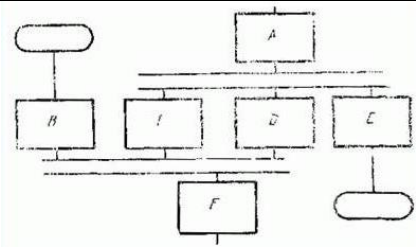
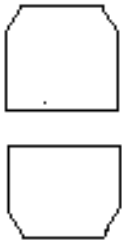

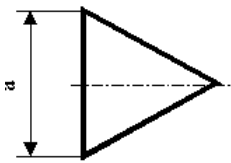
В таблиці 2 наведено умовні позначення регламентовані ДСТ 18.701-90.

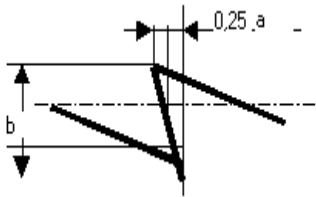
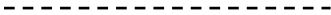
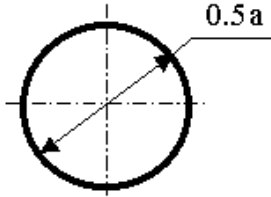
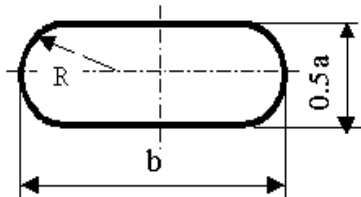
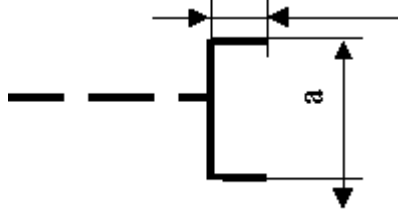
Таблиця 2 – Умовні позначення встановлені ДСТ 18.701-90

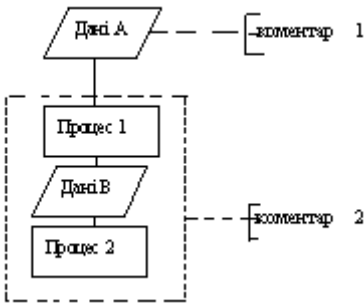

Назва	Позначення	Функції
<i>Символи даних</i>		
<i>Основні символи даних</i>		
Дані		Символ відображає дані, носій даних невизначений. (тут і далі $b = 2a$)
Дані, які запам'ятовуються		Символ відображає дані, що зберігаються у вигляді, придатному для обробки. Носій даних невизначений
<i>Специфічні символи даних</i>		
Оперативний запам'ятовуючий пристрій		Символ відображає дані, які зберігаються в оперативному запам'ятовуючому пристрої
Запам'ятовуючий пристрій з послідовним доступом		Символ відображає дані, які зберігаються в запам'ятовуючому пристрої з послідовним доступом (магнітна стрічка, касета з магнітною стрічкою, магнітофонна касета)
Запам'ятовуючий пристрій з прямим доступом		Символ відображає дані, які зберігаються в запам'ятовуючому пристрої з прямим доступом (магнітний диск, магнітний барабан,

		гнучкий магнітний диск)
Документ		Символ відображає дані, які представлені на носії в формі, що зручно читається (машинограма, документ для оптичного або магнітного зчитування, мікрофільм, рулон стрічки, бланки вводу даних).
Ручне введення		Символ відображає дані, які вводяться вручну під час обробки з пристроїв будь-якого типу (клавіатура, перемикачі, кнопки, світлове перо, смуги з штриховим кодом).
Дисплей		Символ відображає дані, які представлені в зручній для людини формі на носії в вигляді пристрою, що відображається (екран для візуального спостереження, індикатори введення інформації).
<p align="center"><i>Символи процесу</i></p> <p align="center"><i>Основні символи процесу</i></p>		
Процес		Символ відображає функцію обробки даних будь-якого виду (виконання визначеної операції або групи операцій, які приводять до зміни значення, форми або розміщення інформації або до визначення, за яким з декількох напрямків потоку слід рухатися).
<p align="center"><i>Специфічні символи процесу</i></p>		

Визначений процес		Символ відображає визначений процес, який складається з однієї або декількох операцій або кроків програми, які визначені в іншому місці (в підпрограмі, модулі).
Ручна операція		Символ відображає будь-який процес, що виконує людина
Підготовка		Символ відображає модифікацію команди або групи команд з метою впливу на деяку наступну функцію (встановлення перемикача, модифікація індексного регістру або ініціалізація програми)
Рішення		Символ відображає рішення або функцію перемикаючого типу, який має один вхід та ряд альтернативних виходів, один і тільки один з яких може бути активізований після обчислення вимог, визначених всередині цього символу. Відповідні результати обчислення можуть бути записані по сусідству з лініями, які відображають ці шляхи.
Паралельні дії	 Приклад.	Символ відображає синхронізацію двох або більше паралельних процесів

		
Межа циклу		Символ, складений з двох частин, відображає початок та кінець циклу. Обидві частини символу мають один і той же ідентифікатор. Вимоги для ініціалізації, приросту, закінчення і т.ін. розміщуються всередині символу на початку або в кінці в залежності від розміщення операції, яка перевіряє вимогу.
<i>Символи ліній</i> <i>Основний символ ліній</i>		
Лінія		Символ відображає потік даних або керування. При необхідності або для підвищення читання можуть бути додані стрілки-показники.
<i>Специфічні символи ліній</i>		
Передача керування		Символ відображає безпосередню передачу керування від одного процесу до іншого, іноді з можливістю прямого повернення до ініційованого процесу після того, як ініційований процес завершить свої функції. Тип передачі керування

		необхідно називати всередині символу (наприклад, запит, виклик, подія).
Канал зв'язку		Символ відображає передачу даних по каналу зв'язку
Пунктирна лінія		Символ відображає альтернативний зв'язок між двома або більше символами. Крім того, символ використовують для обведення анотованої ділянки.
<i>Спеціальні символи</i>		
З'єднувач		Символ відображає вихід в частину схеми з іншої частини цієї схеми і використовується для обриву лінії та продовження її в іншому місці. Відповідні символи-з'єднувачі повинні вміщувати одне і те ж унікальне позначення.
Термінатор		Символ відображає вихід в зовнішнє середовище та вхід із зовнішнього середовища (початок або кінець схеми програми, зовнішнє використання та джерело або пункт призначення даних).
Коментар		Символ використовують для описових коментарів або пояснювальних записів з метою пояснення або приміток. Пунктирні лінії в символі

		коментарю зв'язані з відповідним символом або можуть
	<p>Приклад.</p> 	обводити групу символів. Текст коментарів або приміток повинен бути розміщений біля обмежуючої фігури
Пропуск		Символ (три крапки) використовують в схемах для відображення пропуску символу або групи символів, в яких не визначені ні тип, ні число символів. Символ використовують тільки в символах лінії або між ними. Він застосовується головним чином в схемах, зображуючих загальне рішення з невідомим числом повторювань

Правила застосування символів та виконання схем

Символи в схемі повинні бути розміщені рівномірно. Слід дотримуватися розумної довжини з'єднувань і мінімальної кількості довгих ліній.

Символи повинні бути одного розміру. Не повинні змінюватися кути та інші параметри символів.

Мінімальну кількість тексту, необхідного для розуміння функції даного символу, слід розміщувати всередині даного символу. Текст для читання повинен записуватися зліва направо чи зверху донизу незалежно від напрямку потоку у відповідності з рисунком 11.

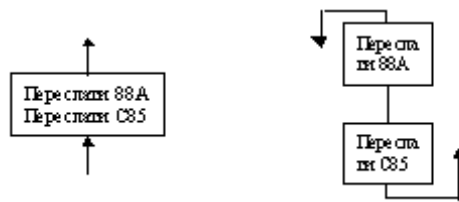


Рисунок 11

Якщо об'єм тексту, який розміщується всередині символу, перевищує його розміри, слід використовувати символ коментарю. Якщо використання символів коментарю може заплутати або зруйнувати хід схеми, текст слід розмістити на окремому аркуші та давати перехресне посилання на символ.

В схемах може використовуватися ідентифікатор символів. Це зв'язаний з даним символом ідентифікатор, котрий визначає символ для використання в довідникових цілях в інших елементах документації (наприклад, в лістингу програми). Ідентифікатор символу повинен розміщуватися з лівого боку над символом у відповідності з рисунком 12.



Рисунок 12

В схемах може використовуватися опис символів – будь-яка інша інформація, наприклад, для покращення розуміння функції як частини схеми. Опис символу повинен бути розміщений з правого боку над символом у відповідності з рисунком 13.



Рисунок 13

Для використання в якості посилання на документацію текст на схемі для символів, що відображають способи виведення, повинен розміщуватися з

правого боку над символом, а текст для символів, що відображають способи введення – з правого боку під символом у відповідності з рисунком 14.

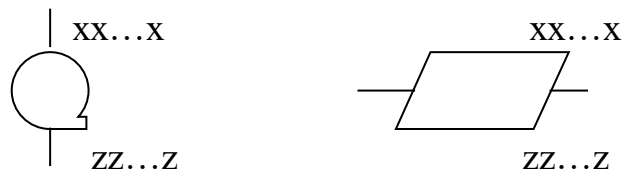


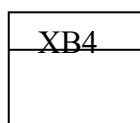
Рисунок 14

В схемах може використовуватися докладний опис, який позначається за допомогою символу з смужкою для процесу або даних. Символ зі смугою вказує, що в цьому ж комплекті документації в іншому місці є більш докладний опис відміченого елемента блок-схеми.

Символ зі смугою являє собою будь-який символ, всередині якого в верхній частині проведена горизонтальна лінія. Між цією лінією та верхньою лінією символу розміщено ідентифікатор, що вказує на докладний опис даного символу у відповідності з рисунком 15.

В якості першого та останнього символу докладного опису повинен бути використаний символ вказівника кінця. Перший символ вказівника кінця повинен містити посилання, яке міститься також в символі зі смугою.

Символ зі смугою



Докладний опис

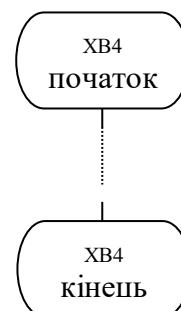


Рисунок 15

Правила виконання з'єднань

Потоки даних або потоки керування в схемах показуються лініями. Напрямок потоку зліва направо та зверху донизу вважається стандартом. В

випадках, коли необхідно ввести більшу ясність в схему (наприклад, при з'єднаннях), на лініях використовуються стрілки. Якщо потік має напрямок, відмінний від стандартного, стрілки повинні вказувати цей напрямок.

У схемах слід уникати перетину ліній. Лінії, які перетинаються, не мають логічного зв'язку між собою, тому зміна напрямку в точках перетину не допускається у відповідності з рисунком 16.

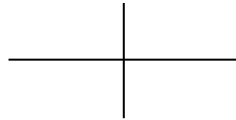


Рисунок 16

Дві або більше вхідних ліній можуть об'єднуватися в одну вихідну лінію. Якщо дві або більше лінії об'єднуються в одну лінію, місце об'єднання повинно бути зсунуто у відповідності з рисунком 17.

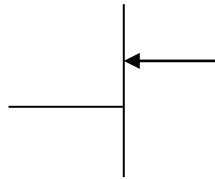


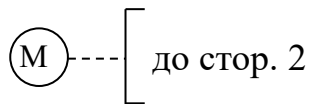
Рисунок 17

Лінії в схемах повинні входити до символу або з лівого боку, або зверху, а виходити або з правого боку, або знизу. Лінії повинні бути спрямовані до центру символу.

При необхідності лінії в схемах слід розривати для уникнення зайвих перетинів або дуже довгих ліній, а також, якщо схема складається з декількох сторінок. З'єднувач на початку розриву називається зовнішнім з'єднувачем, а з'єднувач в кінці – внутрішнім.

Посилання до сторінок може бути приведені спільно з символом коментарю для їх з'єднувачів у відповідності з рисунком 18.

Зовнішній з'єднувач



Внутрішній з'єднувач

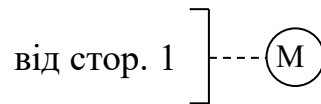


Рисунок 18

Спеціальні умовні позначення

Декілька виходів

Декілька виходів із символу слід зображувати:

- 1) декількома лініями від даного символу до інших символів;
- 2) однією лінією від даного символу, яка потім розгалужується в відповідне число ліній у відповідності з рисунком 19.



Рисунок 19

Кожен вихід із символу повинен супроводжуватися відповідними значеннями умов, щоб показати логічний шлях, який він представляє, з тим, щоб ці умови та відповідні посилання були ідентифіковані у відповідності з рисунком 20.

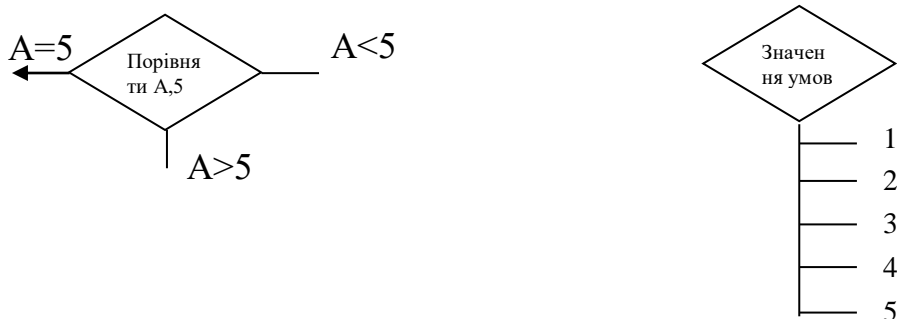


Рисунок 20

Зображення, які повторюються

Замість одного символу з відповідним текстом може бути використано декілька символів з перекриттям зображення, кожен з яких містить описовий текст (використання або формування декількох носіїв даних або файлів, виробництво великої кількості копій друкарських звітів або форматів перфокарт) у відповідності з рисунком 21.

Коли декілька символів представляють упорядковану кількість, це упорядкування повинно розміщуватися від першого до останнього.

Лінії можуть входити або виходити з будь-якої точки перекритих символів, однак вищезгадані правила виконання з'єднань повинні виконуватися. Пріоритет або послідовний порядок декількох символів не змінюється за допомогою точки, в якій лінія входить або з якої виходить.



Рисунок 21

Захист курсової роботи та критерії оцінювання

Захист курсової роботи здійснюється після її повного виконання й передзахисту у термін, встановлений календарним планом.

Оцінювання курсової роботи здійснюється за 100-бальною системою. Критерії оцінювання повноти виконання та захисту курсової роботи наведено в таблиці 3, оцінювання етапів (розділів) курсової роботи - таблиця 4.

Таблиця 3 - Критерії оцінювання повноти виконання й захисту КР

Кількість Балів	Вимоги до виконання та захисту курсової роботи
100-90	<p>Етапи курсового проектування виконано вчасно, курсову роботу подано на передзахист і захист у встановлені календарним планом терміни. Вчасно розроблено, погоджено і затверджено алгоритм та програмне забезпечення. Розробка має творчий підхід та оригінальні проектні рішення з елементами наукової новизни. Пояснювальну записку оформлено відповідно до встановлених вимог, теоретико-практичні аспекти розділів розкрито повністю, подано якісний графічний матеріал, структурні, функціональні схеми, використано вітчизняні та зарубіжні фахові видання.</p> <p>Захист курсової роботи передбачає чітко поставлену мету роботи, результати проведених досліджень і аналізу існуючого програмного забезпечення, опис і обґрунтування прийнятих проектних рішень щодо розробки програми, висновки; студент повинен мати ґрунтовні відповіді на всі додаткові питання.</p>
82-89	<p>Етапи курсового проектування виконано вчасно, курсову роботу подано на передзахист і захист у встановлені календарним планом терміни. Вчасно розроблено, погоджено і затверджено алгоритм та програмне забезпечення. Пояснювальна записка оформлена відповідно до вимог, теоретико-практичні аспекти розділів розкрито повністю. Захист курсової роботи передбачає чітко поставлену мету роботи, результати дослідження й аналізу існуючого програмного забезпечення, опис проектних рішень щодо розробки програми, висновки. Під час захисту курсової роботи студент в повному обсязі виклав теоретико-практичний матеріал, на поставлені додаткові питання мав ґрунтовні відповіді.</p>

75-81	<p>Етапи курсового проектування виконано вчасно, курсову роботу подано на передзахист і захист у встановлені календарним планом терміни. Алгоритм та програмне забезпечення є недосконалим та потребує подальшого вдосконалення, але відповідає темі курсової роботи. Пояснювальна записка оформлена відповідно до вимог, теоретико-практичні аспекти розділів розкрито повністю. Під час захисту проекту студент не в повному обсязі викладав матеріал, на поставлені питання мав нечіткі відповіді.</p>
67-74	<p>Курсову роботу подано до захисту з порушенням встановленого календарним планом термін. Алгоритм та програмне забезпечення є недосконалим, потребує подальшого вдосконалення, але відповідає темі курсової роботи. Пояснювальна записка оформлена відповідно до вимог, суть розділів розкрито не повністю. Під час захисту роботи студент відповів на більшість додаткових питань.</p>
60-66	<p>Етапи курсового проектування виконувались з систематичним запізненням, курсову роботу подано до захисту пізніше встановленого календарним планом терміну. Алгоритм та програмне забезпечення має суттєві недоліки, потребує подальшого вдосконалення, але відповідає темі курсової роботи. Пояснювальна записка оформлена з порушенням вимог, суть розділів розкрито не повністю. Під час захисту роботи студент не відповів на більшість додаткових питань.</p>
34-59	<p>Алгоритм, програмне забезпечення розроблено з грубими помилками та потребують суттєвого доопрацювання. Пояснювальна записка КР виконана з порушеннями вимог, матеріал викладено в неповному обсязі. (До захисту студент не допускається, курсову роботу - доопрацьовується.)</p>

Таблиця 4 - Критерії оцінювання курсової роботи

Назва критерії	Максимальна кількість балів
Пояснювальна записка	
Вступ	5
Розділ 1 Призначення та область використання	5
Розділ 2 Перегляд аналогічних існуючих систем	5
Розділ 3 Опис і обґрунтування проектних рішень	15
Розділ 4 Реалізація роботи вірність проектних і програмних рішень.	10
Розділ 5 Впровадження системи в промислову експлуатацію	5
Розділ 6 Основні висновки	5
Додатки	
Додаток А Інформаційно-логічна модель	10
Додаток Б Схема даних	10
Додаток В Функціональна схема системи	10
Приклади (таблиці, форми, запити, звіти, модулі)	20
ВСЬОГО	100

За одержаними балами студент отримує оцінку за національною системою та ECTS відповідно до таблиці 5.

Таблиця 5 - Оцінки ECTS та національною системою відповідно до одержаних балів за КР

Кількість одержаних балів	ECTS	Національна система оцінювання
90-100	A	Відмінно
82-89	B	Добре
75-81	C	
67-74	D	Задовільно
60-66	E	
34-59	FX	незадовільно
1-34	F	

У разі непогодження з одержаною оцінкою за курсову роботу студент має право у п'ятиденний термін доопрацювати його та подати на повторний захист спеціально створеній екзаменаційній комісії.

Список використаних джерел

1. Глушаков С. В., Ломотько Д. В. Базы данных: Учебный курс. - Харьков: Фолио; М ООО «Издательство АСТ», 2001. – 504 с.
2. Сидоренко В.В., Константинова Л.В., Смірнов С.А. Організація баз даних Навчальний посібник. – Кропивницький: ЦНТУ, 2018. – 274 с. [Електронний ресурс] – Режим доступу: <http://dspace.kntu.kr.ua/jspui/bitstream/123456789/10527/1/NavPosOBD.pdf>
3. Проектування інформаційних систем: Загальні питання теорії проектування ІС(конспект лекцій)[Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки»/ КПП ім. Ігоря Сікорського; уклад.: О.С.Коваленко, Л.М.Добровська. – Електронні текстові дані (1 файл: 2,02Мбайт).–Київ : КПП ім. Ігоря Сікорського, 2020. –192с.
4. Бекоревіч Ю. Б., Пушкіна Н. В. Microsoft Access 2000. СПб.: БХВ - Санкт-Петербург, 1999.
5. Харітонова І., Міхеев М. Microsoft Access 2000 в оригіналі. СПб.: БХВ - Санкт-Петербург, 1999.
6. А.Ю. Берко О.Н. Верес Організація баз даних Практичний курс. Навчальний посібник. Львів, 2003.
7. Горін З. В., Тандосєв А. Ю. CASE засіб S-Designor4.2 для розробки структури бази даних // Системи Управління Базами Даних № 1, 1996. З. 79-86.
8. Сахаров А. А. Принципы проектування і використання багатовимірних баз даних (на прикладі Oracle Express Server) // Системи Управління Базами Даних № 3, 1996. З. 44-59.
9. Системи управління базами даних і знань: Справ, изд. / Наумов А. Н., Вендров А. М., Иванов В. До. і др.; Під ред. Наумова А. Н. М.: Фінанси і статистика, 1991.
10. Методичні вказівки до виконання й захисту бакалаврської дипломної роботи для студентів спеціальності 6.050102 та 123 «Комп'ютерна інженерія» / уклад. Смірнов О.А., Гермак В.С., Мелешко Є.В., Коваленко О.В., Якименко Н.М., Доренський О.П.; Кропивницький: ЦНТУ – 2018. – 55 с.
11. Організація баз даних: Методичні вказівки до курсового проектування для студентів денної та заочної форми навчання за спеціальностями 6.050102/123 «Комп'ютерна інженерія», 125 «Кібербезпека»/ уклад. В.В. Сидоренко, Л.В. Константинова— Кропивницький: ЦНТУ, 2017. —67с.
12. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни «Бази даних» (1 частина) : для студ. денної та заочної форми навч. за спец. : 123 «Комп'ютерна інженерія», 125 «Кібербезпека» / [уклад. : В. В. Босько, Л. В. Константинова] ;

М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програмного забезпечення. - Кропивницький : ЦНТУ, 2020. - 77 с.

13. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни «Організація баз даних» (2 частина) для студентів денної та заочної форми навчання за спеціальностями 123 «Комп'ютерна інженерія», 125 «Кібербезпека» / уклад. В.В. Босько, Л.В. Константинова — Кропивницький: ЦНТУ, 2020. — 60 с.

Додаток А.2

КАЛЕНДАРНИЙ ПЛАН

Назва етапів курсової роботи	Строк виконання етапів роботи	Примітки
Аналіз існуючих систем	12.03.2022 р.	
Вивчення предметної області, постановка задачі	26.03.2022 р.	
Розробка інформаційно-логічної моделі системи	2.04.2022 р.	
Розробка структур даних	13.04.2022 р.	
Побудова схеми даних	20.04.2022 р.	
Побудова інтерфейсу користувача, функціональної схеми системи	27.04.2022 р.	
Побудова запитів	11.05.2022 р.	
Програмування алгоритмів, макросів	18.05.2022 р.	
Оформлення ПЗ	21.05.2022 р.	
Попередній захист курсової роботи	25.05.2022 р.	

Студент

(підпис)

Керівник

(підпис)

Додаток Б
Форма титульного аркушу пояснювальної записки для курсової роботи

ЗАТВЕРДЖЕНО

Наказ Міністерства освіти і науки,
молоді та спорту України
29 березня 2012 року № 384

Форма № Н-6.01

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Кафедра кібербезпеки та програмного забезпечення

КУРСОВА РОБОТА

з дисципліни: «Бази даних»

на тему: _____

Студента (ки) _____ курсу _____ групи
спеціальності _____

(прізвище та ініціали)

Керівник _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS

Члени комісії _____
(підпис) (прізвище та ініціали)

_____ (підпис) (прізвище та ініціали)

_____ (підпис) (прізвище та ініціали)

м. Кропивницький 20 __рік

Додаток В
(довідниковий)

Основний напис для заголовних аркушів текстових документів
за ГОСТ 2.104-2006

									15
Вим.	Арк.	№ докум	Підп.	Дата					
Розроб.						Літ	Аркуш	Аркушів	5 5 15
Перев.									
Н. контр.									
Затв.							15	20	

Основний напис для наступних аркушів текстових документів
за ГОСТ 2.104-2006

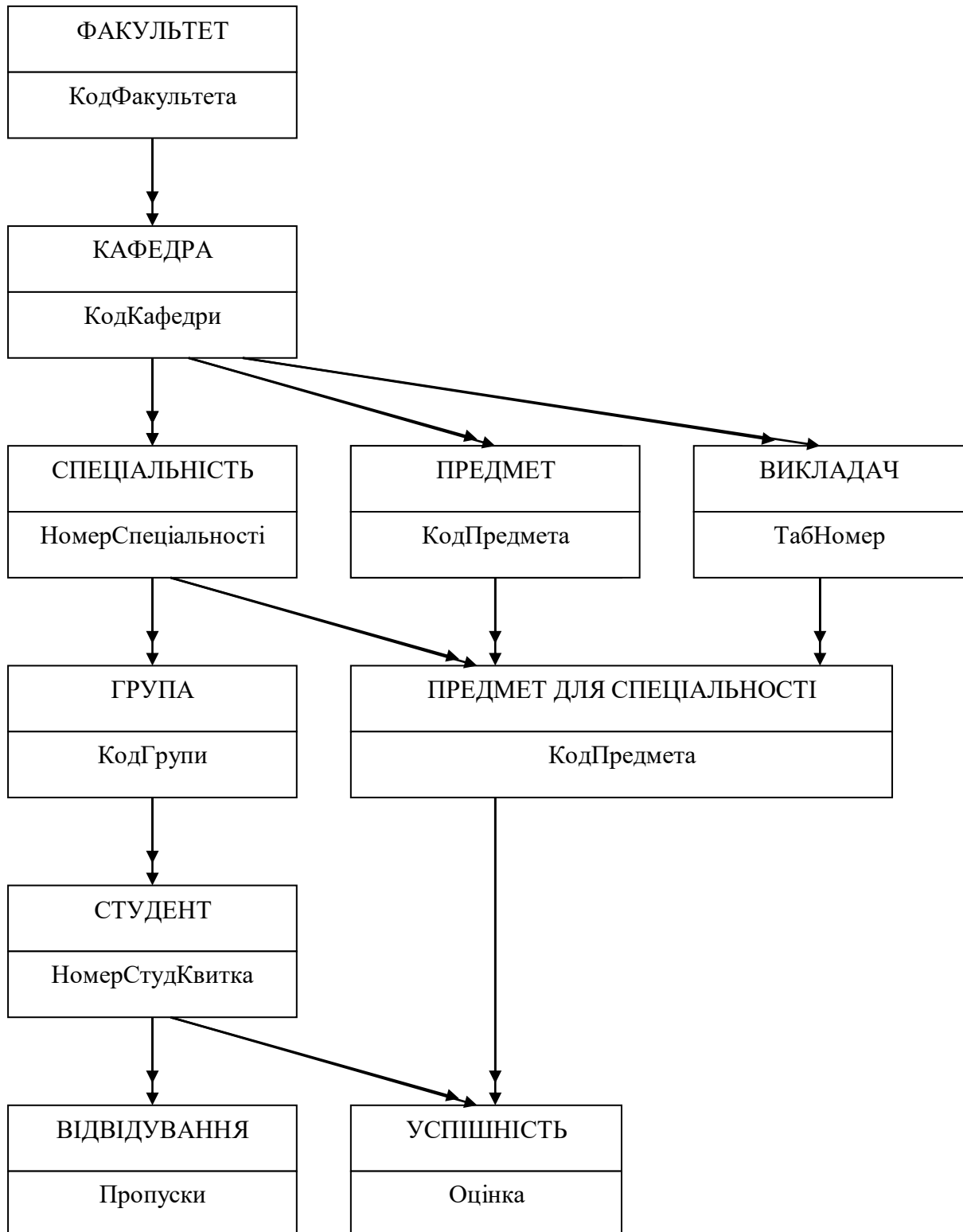
								А р к.	∞
Вим.	А р к.	№ докум	Підп.	Дата					

10

185

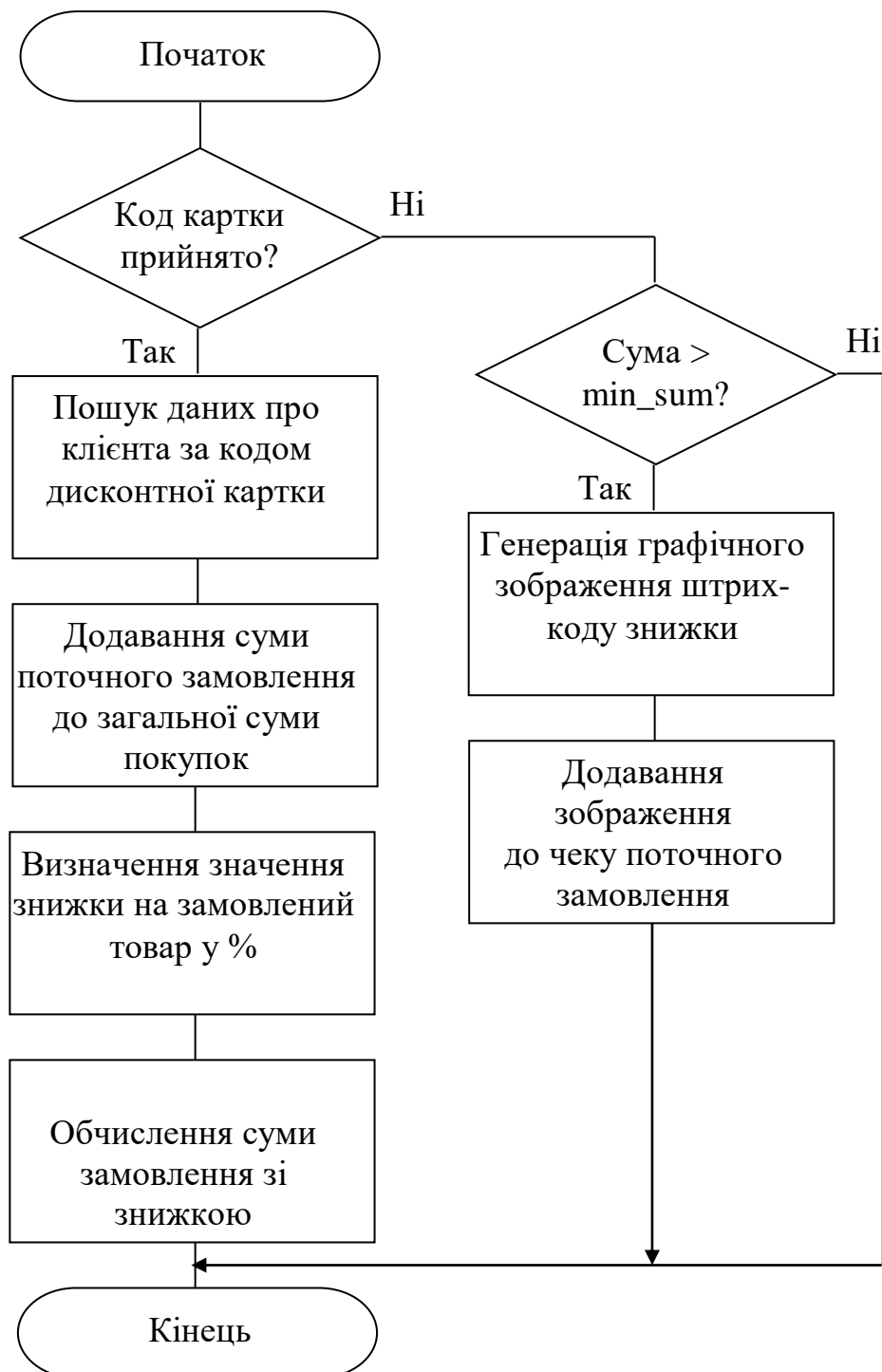
Додаток Г
Приклад інформаційно-логічної схеми
Додаток А

Інформаційно-логічна схема системи



Додаток Д
(рекомендований)

Блок-схема алгоритму роботи(приклад)



Додаток Е
Форма титульного аркушу програмного документу
Додаток Ж

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник курсової роботи
_____ Босько В.В.

***Автоматизована система продажу комп'ютерної техніки та
комплектуючих матеріалів***

Лістинг програми

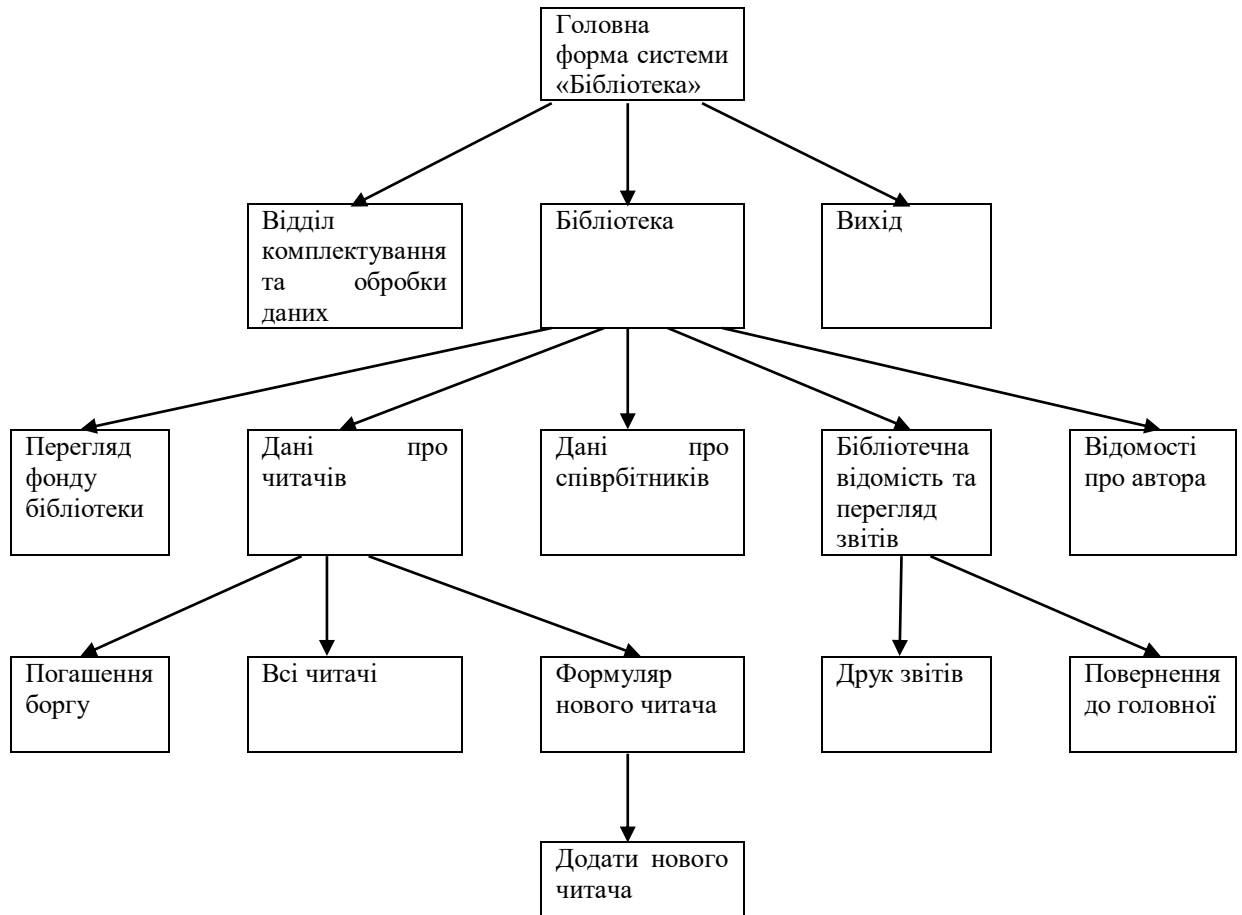
Код документу 12
Носій: CD/BMB-диск

Загальна кількість аркушів: 20

Літера: РП

Кропивницький 20__

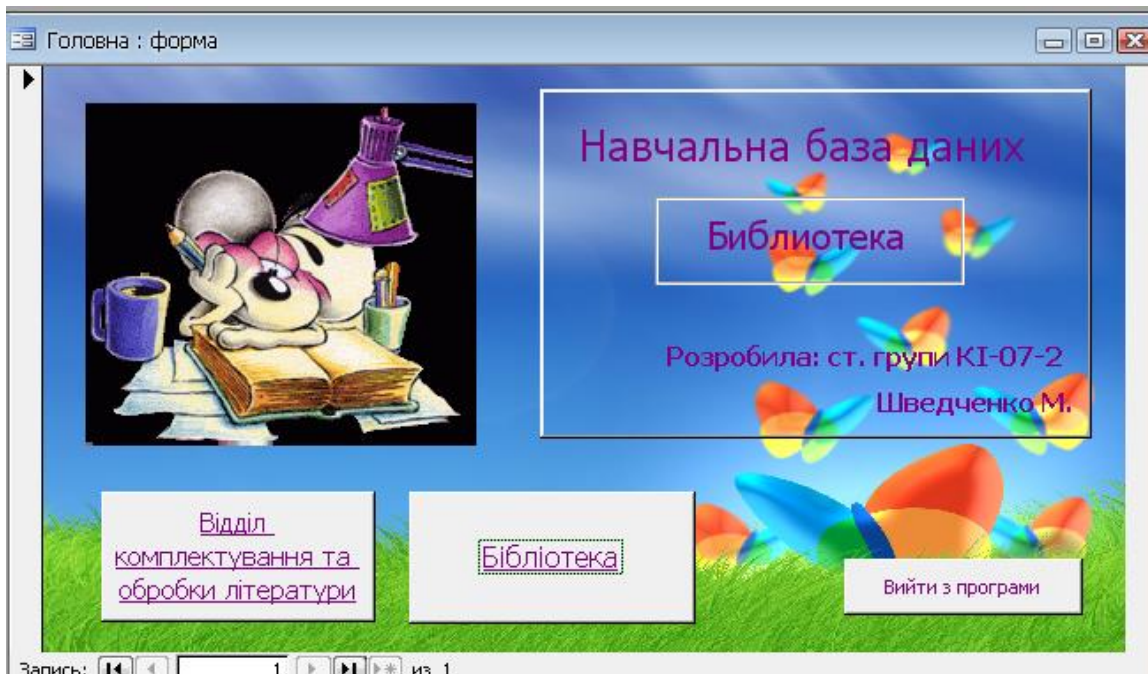
Додаток Є
Приклад функціональної схеми



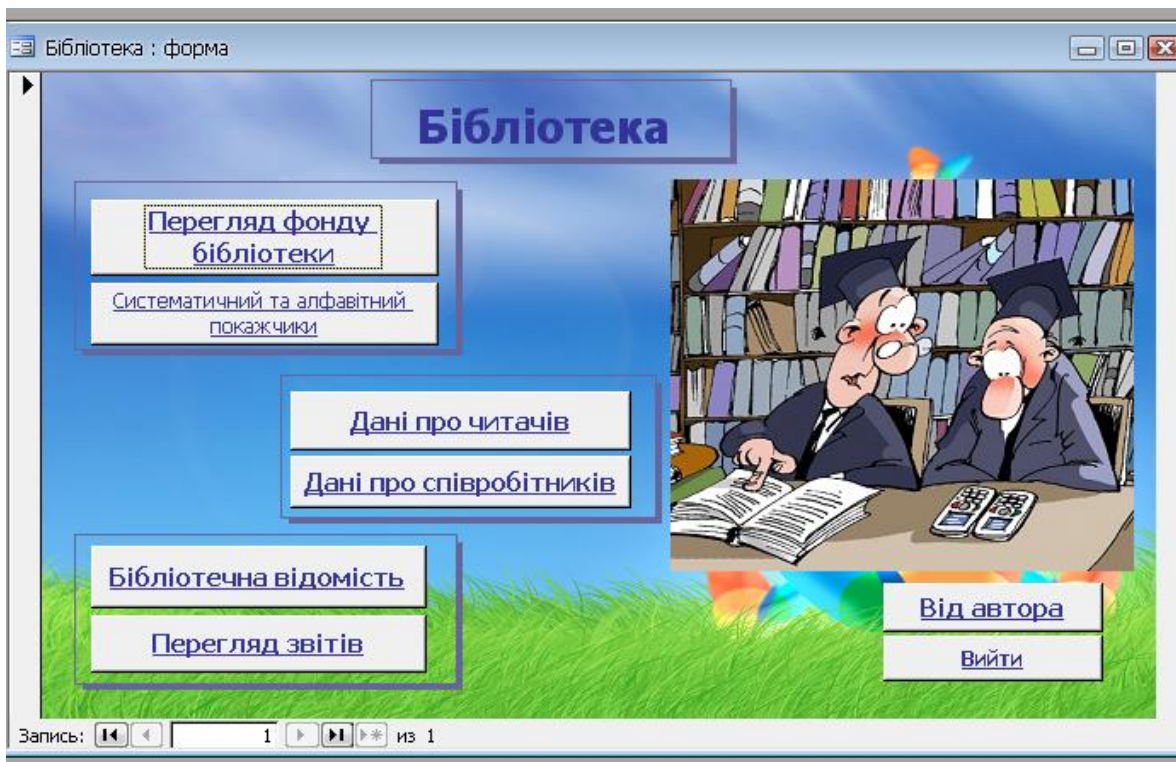
Додаток Ж.1

Приклади форм

Головна форма



Форма Бібліотека



Додаток Ж.2

Приклади таблиць

Таблиця Книги

Книги : таблиця									
	Шифр книги	Авт. знак	Назва книги	Автор	Видавництво	Рік видання	Розділ	Кількість	Анотація
+	978-0-596-10206-7	Ш 33	Perf: изучаем глубже, 1-	Шварц Р.Л.	Символ-Плюс	2006	3	10	Информация о книге
+	978-0-596-10206-8	Ш 33	Perf: изучаем глубже, 2-	Шварц Р.Л.	Символ-Плюс	2007	3	10	Информация о книге
+	978-0-7514-0417-3	С 44	Производство сыра: нау	Скотт Р., Р. Робинсон	Профессия	2005	36	5	Информация о книге
+	978-0-75253-352-0	Б 83	Новая энциклопедия шк	Бородина Т.	Махаон	2005	1	4	Рассчитана на учащи
+	978-0-87814-798-4	Б 24	Нефтехимия - 3 изд.	Бардик Д.Л., Леффле	Олимп-Бизнес	2005	35	2	В книге Д.Л.Бардика
+	978-3-8085-7047-0	И 26	Химическая техника. Пр	Игнатович Э.	Техносфера	2007	35	6	Информация о книге
+	978-3-98-34555-2	Д 70	Братья Карамазовы	Достоевский Федор М	АСТ	2007	80/84	2	Самый сложный, сам
+	978-5-02-011197-4	Б 82	Шампавер	Борель Петрюс	Наука	1993	80/84	7	Издание 1978 года. С
+	978-5-02-011652-8	Г 64	Одиссея	Гомер	Наука	2000	80/84	6	Эпическая поэма "Од
+	978-5-02-028067-0	С 50	Отвергнутые послания	Смит Джеймс и Горал	Наука	1995	80/84	3	Издание 1995 года. С
+	978-5-03-003569-7	Г 86	Болезни и вредители пч	Гробов О.Ф.	Мир	2003	4	15	Информация о книге
+	978-5-06-004414-0	Р 36	Электрооборудование п	Рекус, Григорий Гавр	Школа	2007	31	7	Информация о книге
+	978-5-06-005793-5	Г 68	Мировая экономика и пр	Гордеев В. В.	Москва	2008	65	10	Информация о книге
+	978-5-09-016012-4	Ш 83	Изобразительное искусс	Шпикалова Т.Я.	Просвещение	2010	85	15	Учебник "Изобразите
+	978-5-17-007679-6	И 49	Книга для внеклассного	Иляшенко Л.А.	Харвест	2007	6/8	8	Уважаемые родители
+	978-5-17-012238-7	К 88	Справочник по высшей	Выгодский М.Я.	АСТ	2006	22	10	Информация о книге
+	978-5-17-028192-3	Г 65	Химия. Билеты и ответы.	Гончарук О.Ю.	АСТ	2005	24	15	В сборнике приводят
+	978-5-17-032818-5	Т 30	Все предметы школьной	Текучева И.В.	АСТ	2006	1	15	Книга представляет с
+	978-5-17-044077-1	Р 83	Химия для 10 класса об	Рудзитис Г.Е., Фельд	ХРАНИТЕЛЬ	2008	24	8	Новый учебник автор
+	978-5-17-044312-3	Я 47	Компьютер для школьн	Яковенко, Екатерина	ВКТ	2007	3	20	Вместе с этой книгой
+	978-5-17-049290-9	Ж 91	Социальная компетентн	Журкова В.Н.	Прайм-Еврозна	2008	88	35	Комплексный учебнь
+	978-5-17-058864-0	И 25	Окружающий мир. Учеб	Ивченкова Г.Г.	АСТ	2010	2	35	Планета знаний" - но
+	978-5-222-14359-9	Ш 17	Методика обучения эле	Шайхутдинова Д.И.	Феникс	2010	85	15	Информация о книге
+	978-5-225-02711-7	Б 86	Клиническая генетика	Бочков Н. П.	Медицина	1997	28	2	Информация о книге
+	978-5-225-03450-4	П 56	Основы физиотерапии	Пономаренко Г.Н.	Медицина	2008	5	6	Информация о книге
+	978-5-238-01284-1	Г 68	Экономика предприятия	Горфинкель В. Я., Шв	Юнити	2008	65	8	Информация о книге
+	978-5-248-01352-4	К 90	Наука и общество на ру	Кулькин А.М.	Центрполиграф	2000	70/79	10	В сборнике приведен
+	978-5-27-114422-6	Г 85	Звезды без грима. О ку	Гринин, Леонид Ефим	Харвест	2007	9	10	Информация о книге
+	978-5-271-16704-1	П 37	Повести и рассказы	Платонов, Андрей Пл	АСТ	2007	80/84	1	В книгу вошли расск
+	978-5-271-22157-6	В 92	Справочник по элемент	Выгодский М. Я.	ВКТ	2010	22	35	Справочник включает
+	978-5-271-23455-2	Г 71	Английский язык. 4 клас	Н. Ю. Горячева, С. В.	Астрель	2008	6/8	13	"Планета знаний" - нс
+	978-5-282-02382-4	Я 47	Эпохальные инновации	Яковец Ю.В.	Экономика	2004	87	3	Периодическое инно
+	978-5-282-02865-2	Р 64	Экономическая теория	Розанова Н.М.	Экономика	2008	65	15	Экономическая теори

Таблиця Відомість

Відомість : таблиця								
	Код читача	Шифр книги	Ким видано	Дата видачі	Повернути до	Дата повернення	Борг	Борг оплачено?
▶	БК-08-002	251-1-345-76589-5	НО-78-456	04.04.2010	11.04.2010	11.04.2010		<input checked="" type="checkbox"/>
	БК-08-002	876-0-765-46523-8	НЕ-13-658	10.04.2010	17.04.2010	17.04.2010		<input checked="" type="checkbox"/>
	БК-08-003	251-1-345-76589-5	ЕК-12-012	04.04.2010	11.04.2010	11.04.2010		<input checked="" type="checkbox"/>
	БК-08-003	5-17-028335-0	ЕК-12-012	10.04.2010	17.04.2010		6	<input type="checkbox"/>
	БК-08-004	687-3-234-89564-6	ЕК-12-012	10.04.2010	17.04.2010	22.04.2010		<input checked="" type="checkbox"/>
	БК-08-005	756-3-789-34621-7	ЕК-12-012	09.04.2010	16.04.2010	17.04.2010		<input checked="" type="checkbox"/>
	БК-08-005	978-0-596-10206-7	НО-78-456	05.04.2010	12.04.2010			<input checked="" type="checkbox"/>
	БК-08-006	756-3-789-34621-7	ЕК-12-012	09.04.2010	16.04.2010		7	<input type="checkbox"/>
	БК-08-006	78-5-17-051214-0	НЕ-13-658	09.04.2010	16.04.2010		7	<input type="checkbox"/>
	БК-09-001	78-5-17-054436-3	ЕК-12-012	09.04.2010	16.04.2010	16.04.2010		<input checked="" type="checkbox"/>
	БК-09-002	834-6-123-63456-8	НО-78-456	09.04.2010	16.04.2010		7	<input type="checkbox"/>
	БК-09-006	978-3-98-34555-2	ТО-33-123	10.04.2010	17.04.2010	17.04.2010		<input checked="" type="checkbox"/>
	БК-09-006	978-5-9775-0487-4	НЕ-13-658	05.04.2010	12.04.2010	22.04.2010		<input checked="" type="checkbox"/>
	БК-09-012	687-4-543-87693-4	НО-78-456	17.04.2010	24.04.2010			<input type="checkbox"/>
	БК-10-007	978-5-248-01352-4	ЕК-12-012	06.04.2010	13.04.2010		10	<input type="checkbox"/>
	БК-10-008	876-0-765-46523-8	НЕ-13-658	07.04.2010	14.04.2010	14.04.2010		<input checked="" type="checkbox"/>
*							0	<input type="checkbox"/>

Додаток Ж.3

Приклади запитів

Анотація

```
PARAMETERS [Введіть шифр] Text ( 255 );  
SELECT Книги.Анотація, Книги.[Шифр книги]  
FROM Книги  
WHERE (((Книги.[Шифр книги])=[Введіть шифр]));
```

Пошук книги за жанром

```
PARAMETERS Розділ Text ( 255 );  
SELECT Книги.[Шифр книги], Книги.[Назва книги], Книги.Автор, Книги.Видавництво,  
Книги.[Рік видання], Книги.Розділ, Книги.[Кількість книг], Книги.Анотація, [ББК  
(жанр)].Фото, [ББК (жанр)].Область  
FROM Книги INNER JOIN [ББК (жанр)] ON Книги.Розділ = [ББК (жанр)].Розділ  
WHERE (((Книги.Розділ)=[Розділ]));
```

Пошук книги за видавництвом

```
PARAMETERS Видавництво Text ( 255 );  
SELECT Книги.[Шифр книги], Книги.[Назва книги], Книги.Автор, Книги.Видавництво,  
Книги.[Рік видання], Книги.Розділ, Книги.[Кількість книг], Книги.Анотація  
FROM Книги  
WHERE (((Книги.Видавництво) Like [Видавництво] & "*"));
```

Пошук книги за роком видання

```
PARAMETERS [Рік видання] Text ( 255 );  
SELECT Книги.[Шифр книги], Книги.[Назва книги], Книги.Автор, Книги.Видавництво,  
Книги.[Рік видання], Книги.Розділ, Книги.[Кількість книг], Книги.Анотація  
FROM Книги  
WHERE (((Книги.[Рік видання])=[Рік видання]));
```