

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи інформаційної
системи підприємства”

КБПЗ – 2023

Виконав здобувач вищої освіти
II курсу, групи КН-22МЗ
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Петленко М.В.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань 12 *“Інформаційні технології”*

Спеціальність 122 *“Комп’ютерні науки”*

Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерні науки”*

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

**ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА
ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

Петленку Максиму Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи інформаційної системи підприємства*

2. Керівник роботи *Лисенко Ірина Анатоліївна, канд. техн. наук*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 37-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту *10.12.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи інформаційної системи підприємства*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Петленко М.В. Дослідження та програмна реалізація системи інформаційної системи підприємства. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інформаційної системи підприємства.

Метою розробки є дослідження та програмна реалізація системи інформаційної системи підприємства.

Об'єктом дослідження є процес інформаційної системи підприємства.

Предметом дослідження є методи інформаційної системи підприємства.

Методи дослідження базуються на методах побудови інформаційних систем, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інформаційної системи підприємства.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Visual C++.

Ключові слова: комп'ютерні науки, інформаційна система

ABSTRACT

Petlenko M.V. Research and software implementation of the enterprise information system. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the information system of the enterprise.

The purpose of the development is the research and software implementation of the information system of the enterprise.

The object of research is the process of the information system of the enterprise.

The subject of the study is the methods of the information system of the enterprise.

Research methods are based on methods of building information systems, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the enterprise's information system.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Visual C++ environment.

Keywords: computer science, information system

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	36
2.3 Розгорнута постановка завдання	38
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	40
3.1 Опис функціонування системи	40
3.2 Розробка структурної схеми.....	48
3.3 Розробка функціональної схеми	52
3.4 Розробка діаграми процесів.....	55
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	57
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	57
4.2 Захист розробленого програмного забезпечення.....	70
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	71
6 НАУКОВА НОВИЗНА	73

					ВКРМ-122.23.0072.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи інформаційної системи підприємства	Літ.	Аркуш	Аркушів
Розроб.	Петленко М.В.					М	1	113
Перев.	Писенко І.А							
Н.контр.	Коваленко А.С.					ЦНТУ КН-22МЗ		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	74
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	74
7.2 Розрахунок трудомісткості розробки програмної продукції.....	76
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	78
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	83
7.5 Визначення собівартості розробки та ціни програмної продукції.....	87
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	90
7.7 Визначення експлуатаційних витрат.....	90
7.8 Визначення економічної ефективності програмної продукції.....	92
7.9 Висновок.....	94
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	95
8.1 Вступ.....	95
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	96
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	97
8.4 Розробка заходів з умов поліпшення охорони праці.....	100
8.5 Розрахункова частина	101
9 ОСНОВНІ ВИСНОВКИ.....	105
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	107

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕА	–	електронний архів
ЕДО	–	електронний документообіг
САДД	–	система автоматизації електронного документообігу й діловодства
СЕА	–	системи електронних архівів
СДЕ	–	системи діловодства електронні
СУЕД	–	система управління електронним документообігом
ТЕ	–	типові елементи
ФФО	–	формат файлових об'єктів
EDMS	–	Electronic Document Management Systems

КБПЗ – 2023

ВСТУП

Актуальність теми. Інформаційна система підприємства (ІСП) – це тип інформаційної системи, яка шляхом інтеграції покращує функції бізнес-процесів підприємства. Зазвичай це означає надання вищої якості послуг, роботу з великими даними та здатність підтримувати деякі великі та складні організації/підприємства. ІСП може бути корисним для підвищення продуктивності бізнесу, скорочення циклів надання послуг, циклу розробки продукту та життєвого циклу маркетингу, і, отже, повинен використовуватися всіма частинами та всіма рівнями підприємства. Його також можна використовувати для об'єднання існуючих програм. Наявність належної ІСП дозволяє підприємствам і організаціям підвищити ефективність роботи та скоротити витрати. Тоді як фінансова цінність не є загалом прямим результатом впровадження ІСП.

ІСП має динамічний характер. ІСП приймає зміни зі змінами в процесі управління бізнесом. Він надає коригувальний метод у системі для ефективного контролю за змінами потреб в інформації шляхом постійної взаємодії з внутрішнім і зовнішнім середовищем бізнесу. Людські та комп'ютерні ресурси в об'єднаній формі визначають інформаційну систему підприємства. ІСП надає важливу інформацію членам організації від працівників нижчого рівня до керівництва вищого рівня. ІСП забезпечує підтримку збору, зберігання, отримання, передачі та використання даних для ефективного планування, виконання та управління бізнес-операціями. Більшість організацій можуть використовувати ІСП, щоб зробити свої бізнес-процеси ефективними.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інформаційної системи підприємства.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

- Огляд існуючих систем інформаційної системи підприємства.
- Дослідження системи інформаційної системи підприємства.
- Програмна реалізація системи інформаційної системи підприємства.

Об'єктом дослідження є процес інформаційної системи підприємства.

Предметом дослідження є методи інформаційної системи підприємства.

Методи дослідження базуються на методах побудови інформаційних систем, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод інформаційної системи підприємства.
- Розроблено вітчизняний продукт інформаційної системи підприємства,

який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інформаційної системи підприємства.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інформаційної системи підприємства, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

ІСП є інтегрованою системою і дозволяє підтримувати спільну базу даних.

ІСП має такі важливі характеристики:

– Інтегрована – ІСП поєднує життєво важливі елементи підприємства та інформації.

– Гнучкий – ІСП може надавати результати відповідно до запитів користувачів.

– Ефективний – ІСП дозволяє організації робити ефективні оцінки та приймати рішення для досягнення короткострокових і довгострокових цілей організації.

– Точний – ІСП надає найновішу та найточнішу інформацію на основі правдивих даних.

– Повна – ІСП може надати абсолютну та актуальну інформацію в узагальненій формі.

– Predictor-ІСП надає доступ до узагальнених звітів про поточну ефективність організації. Особи, які приймають рішення, використовують ці звіти, щоб спостерігати та керувати бізнес-діяльністю та прогнозувати майбутні показники та можливості.

Функції ІСП

Нижче наведені важливі функції ІСП:

– Бухгалтерський облік і фінанси: функція бухгалтерського обліку та фінансів веде облік фінансів організації, включаючи розподіл, планування, облік, контроль надходжень, управління податками тощо.

– Управління взаємовідносинами з клієнтами (CRM). Функція CRM допомагає підприємствам керувати клієнтськими операціями, такими як

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

маркетинг, продажі та обслуговування клієнтів. Це допомагає співробітникам відстежувати ймовірність продажів і канали клієнтів. CRM також можна використовувати в управлінні маркетинговими операціями, включаючи рекламу та кампанії із залучення потенційних клієнтів.

– Ланцюг постачання: функції ланцюга постачання ІСП контролюють продукти від моменту виробництва до розподілу. Його основні функції включають закупівлю запасів, доставку, відстеження, відшкодування, обробку претензій, планування постачальників тощо.

– Управління запасами: функція управління запасами ІСП дозволяє підприємствам легко відстежувати ресурси та постачання через замовлення на купівлю, автоматичне оформлення замовлень, контроль запасів, і сканування.

– Виробництво: виробники та інші підприємства, орієнтовані на виробництво, можуть здійснювати виробництво, щоб контролювати контроль якості, робочі замовлення, планування, виробничі операції, витрати ресурсів і загальний виробничий цикл організації.

– Людські ресурси: кадрові ресурси пропонують свої руки від самого початкового, тобто процесу найму, до найму через наступність. Послуги кадрових ресурсів, такі як нарахування заробітної плати, управління часом, управління навчанням, управління продуктивністю та відстеження кандидатів.

– Бізнес-аналітика: бізнес-аналітика накопичує та перевіряє дані з різних джерел і допомагає користувачам визначити краще рішення для організації. Деякі з його основних функцій – це заплановані звіти, інструменти візуалізації, налаштуванні інформаційні панелі та доступ до даних у реальному часі.

1.2 Область застосування

Спробуємо позначити деякі завдання, що відрізняють роботу з документами від традиційної роботи з даними й, відповідно, що вимагають розробки власного інструментарію.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Насамперед, мова йде про розробку концепції документа й способів його опису. Поняття структури, змісту, форми подання для документів істотно інші, ніж для «даних», навіть у найпростіших випадках, коли документ може розглядатися як простий текст або як фрагмент бази даних.

Процедура «створення» документа взагалі відсутній у БД, де документ є поняттям зовнішнім. Для СУД це найважливіший процес, що повинен підтримувати компонування цілого із частин, версійність, можливість вибору варіантів, фіксацію й т.п.

Новим по суті є й поняття пересилання документа. Воно пов'язане зі збереженням часу й дати, підтвердженням доставки, перевіркою електронного підпису.

Створення великих архівів, пошук у них, обробка потоку документів, виділення й класифікація – типові завдання при роботі з документами. При цьому можуть використовуватися традиційні для БД методи відбору або пошуку по комбінаціях значень формально виділених реквізитів, елементи лінгвістичної обробки, наприклад, повнотекстовий пошук, а також і «штучно-інтелектуальні» методи визначення «близькості», змістовних понять і т.п.

Сюди ж примикають завдання добування з документів «знань» – змістовної структурованої інформації – для подальшого використання.

Необхідність у нових методах пошуку, а також у зберіганні графічного образу документа поряд з його змістом приводить до специфічних способів організації сховища документів: додатковим індексним довідникам за словами, вираженням, поняттям і особливим методам стиску й упакування самих документів.

По-новому варто підійти й до самої традиційної частини роботи з документами – уведенню й виводу. Коли ми вводимо в комп'ютер «паперовий» документ, у багатьох випадках необхідно зберегти не тільки його зміст, але й графічне подання, щоб його завжди можна було відтворити в «первозданному виді». Укупі з технологіями розпізнавання, це породжує цілу систему відносин

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

між описом документа й різних програмних комплексів: екранного уведення й виводу, сканування – розпізнавання, печатки, передачі в електронній формі. Необхідний єдиний для всіх цих комплексів погляд на відповідність форми подання й змісти. Поки ж усе такого роду програми мають власні описи форм, перевірки цілісності, відображення у внутрішнє подання.

Однак, у зв'язку з поширенням документно-орієнтованого підходу, виникає ще одна проблема загального плану. З одного боку, успішно розвиваються системи уведення, виводу, руху документів, архіви й методи роботи з ними. З іншого боку – величезне число управлінських програм: бухгалтерії й кадри, планування й звітність, замовлення й збут, а також безліч інших – повністю орієнтовані на роботу з базами даних. Виникає розрив: старі системи, націлені на безпосереднє уведення документів у БД і вивід їх з бази, явно недостатні, а більше сучасні, враховуючі процеси створення, передачі, зберігання, виконання, ідентифікації документів, безпосередньо із цими базами даних не зв'язані.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інформаційної системи підприємства, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Сьогодні організаціям потрібен надійний доступ до всіх даних. Це важливо не лише для малого бізнесу, але й для підприємств ще більше потребують придбання найкращих можливих процесів керування даними та інструментів для забезпечення стабільного розвитку. Дивовижний потенціал, який корпоративне програмне забезпечення (ІСП) представило світові завдяки потужності технологій, не лише змінив процес розвитку бізнесу, але й те, як вони тримають під контролем і оновлюють свої операційні та стратегічні ініціативи. Оптимізація великомасштабних даних потребує великомасштабного програмного забезпечення, і в цій статті ви дізнаєтеся все, що вам потрібно знати – від найкращих інструментів ВІ на ринку до програмного забезпечення для керування проектами та веб-аналітики.

Ми почнемо з визначення корпоративного програмного забезпечення, пояснюючи, які переваги може отримати бізнес, використовуючи його, і які найкращі компанії корпоративного програмного забезпечення пропонують ключові функції, щоб конкурувати в нашому жорсткому середовищі. Зрештою, ви побачите, як використання переваг імен у нашому списку може виявитися корисним для вашого підприємства, будь то малого чи великого.

Корпоративне програмне забезпечення – це комп'ютерна програма, яка націлена на допомогу великим компаніям у таких потребах, як аналіз даних, управління продажами та маркетингом, обслуговування клієнтів тощо. Як правило, ці інструменти призначені для обслуговування великої кількості користувачів із високою масштабованістю та можливостями інтеграції.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Таке програмне забезпечення можна знайти у великих галузях промисловості, урядах, роздрібних торговцях, лікарнях і таких відділах, як продажі, маркетинг, обслуговування клієнтів і фінанси, серед іншого. Таке програмне забезпечення, очевидно, потребує міждисциплінарних методів і процесів для керування великими масштабами даних – численні рішення на ринку забезпечують саме це. Наприклад, корпоративна програма, орієнтована на людські ресурси, матиме численні HR-функції, які дозволять професіоналам отримати доступ до єдиного рішення та генерувати інформацію одночасно.

Щоб легко автоматизувати численні процеси та завдання та додати цінність вашим бізнес-операціям, ми детально розглянемо найкращі корпоративні інструменти на 2023 рік, де обговоримо різні сфери та надамо приклади інструментів для кожної. Ми також розглядатимемо рейтинги Capterra та G2Crowd як двох провідних веб-сайтів для перевірених перевірок програмного забезпечення, які можуть надати вам додаткову інформацію. Усі рейтинги, згадані в цій публікації, датуються січнем 2023 року, а вибране програмне забезпечення має щонайменше 50 відгуків між обома платформами.

Як уже згадувалося, професійне програмне забезпечення для керування підприємством може допомогти великим підприємствам ефективно керувати та оптимізувати їх діяльність завдяки передовим технологіям, таким як автоматизація, аналіз даних, співпраця та багато інших. Як ми побачимо в нашому списку найпопулярніших інструментів на ринку, існує широкий їх вибір для різних функцій і сфер організації, таких як фінанси, ІТ, кадри, маркетинг тощо, кожен із яких пропонує різні функції щоб зробити організації більш ефективними за всіма напрямками. Тим не менш, незалежно від того, яку сферу ви намагаєтеся охопити, усі типи корпоративного програмного забезпечення пропонують набір ключових переваг, які ми обговоримо нижче.

Підвищена організаційна ефективність. Інструменти керування підприємством допомагають автоматизувати виснажливі та трудомісткі завдання, на виконання яких у співробітників потрібні години. Згодом це залишає різним

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

командам достатньо часу для роботи над більш цінними завданнями, що призведе до підвищення операційної ефективності та кращих бізнес-результатів за менших грошей і зусиль.

Підвищення продуктивності співробітників. Так само, як і в попередньому пункті, цей тип програмного забезпечення робить роботу співробітників набагато ефективнішою та приємнішою, оскільки надає набір функцій для полегшення їхніх щоденних операцій на багатьох рівнях. Завдяки цьому підвищується загальна ефективність і продуктивність співробітників, що також полегшує співпрацю з іншими командами.

Удосконалені продукти та послуги: корпоративне програмне забезпечення дає підприємствам необхідні знання для прийняття обґрунтованих рішень, коли йдеться про розробку продукту та керування ним. Можливість доступу до всіх важливих деталей, пов'язаних із продуктивністю продукту, може допомогти оптимізувати будь-які вузькі місця та запропонувати найкращі продукти чи послуги відповідно до потреб клієнтів.

Покращення задоволеності клієнтів: природно, те, як організація працює всередині, безпосередньо впливатиме на те, як клієнти сприймуть це пізніше. ІСП допомагає компаніям керувати робочим процесом, щоб клієнти були задоволені та поверталися за новими. Програмне забезпечення, таке як CRM або автоматизація маркетингу, забезпечує централізований перегляд даних клієнтів, щоб зрозуміти, що працює, а що ні. Поговоримо про це докладніше нижче.

Як ви бачили вище, ІСП може покращити загальне функціонування організації від співробітників до продуктів і клієнтів. Тим не менш, існує широкий спектр категорій і бізнес-сфер, у яких ці типи програмного забезпечення виявляються надзвичайно цінними, і ми швидко розглянемо кожен з них.

Продажі. Коли мова йде про підприємства, продажі – це складний процес, який включає багато кроків і ключових гравців. Щоб тримати все під контролем і бути максимально ефективним, організації покладаються на програмне забезпечення для продажів, щоб автоматизувати ключові процеси, покращити

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

співпрацю між командами, відстежувати їх ефективність за допомогою аналітики та звітів про продажі та отримувати 360-градусний огляд усіх операцій продажів. Інструменти CRM часто використовуються для цієї мети, об'єднуючи маркетинг і продажі для отримання кращих бізнес-результатів.

Маркетинг: ІСП широко використовується у відділі маркетингу, щоб допомогти визначити вподобання клієнтів і проблемні точки та оптимізувати всі точки дотику, щоб переконатися, що вони залишаються залученими та задоволеними компанією. Як і в сфері продажів, відділи маркетингу також покладаються на корпоративні системи CRM, щоб сегментувати клієнтів, проводити рекламні кампанії та оцінювати їхню ефективність для залучення потенційних клієнтів, виконувати аналітику в реальному часі та автоматизувати керування соціальними мережами, серед інших завдань.

Людські ресурси: чим більше підприємство, тим важче відділу кадрів ефективно виконувати всі завдання та процеси, пов'язані з працівниками. З цієї причини особи, які приймають рішення, впроваджують програмне забезпечення для допомоги в управлінні та автоматизації різних кадрових процесів, таких як підбір персоналу, нарахування заробітної плати, навчання співробітників, управління продуктивністю, кадрова звітність тощо. Насправді очікується, що ринок кадрових технологій буде експоненціально зростати в найближчі роки, досягнувши значення в 39,90 мільярдів доларів до 2029 року.

Фінанси: можливо, це одна з найважливіших сфер бізнесу, фінансовий відділ також може отримати значну користь від використання цього програмного забезпечення. Наприклад, для автоматизації податкових звітів і дотримання закону, а також для інших відповідних завдань, таких як бюджетування та прогнозування, бухгалтерські звіти, відстеження витрат, управління доходами, планування виробництва та будь-які інші процеси фінансової звітності та аналізу. Враховуючи, що фінансовий відділ має бути пов'язаний з усіма іншими бізнес-сферами, це програмне забезпечення зазвичай забезпечує легку інтеграцію з іншими інструментами.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

4. Відгуки клієнтів і користувачів

Перегляд думок ваших користувачів – це прониклива діяльність, яка може сприяти покращенню взаємодії з клієнтами, розуміючи, чого хочуть ваші клієнти, використовуючи дані й аналітику для створення екосистеми зворотного зв'язку, яка дозволить вам завчасно вдосконалювати свій продукт і ділитися своїми висновками з усією компанією.. Одним із таких рішень є Morinion.

Мопіньон

Ключові особливості:

- Налаштовуванні форми зворотного зв'язку для веб-сайтів, рідних програм і електронних листів.
- Легка інтеграція з іншими програмами CRM, управління проектами або веб-аналітики.
- Відгуки клієнтів і статистика в реальному часі, а також аналіз відкритих текстів і значень.

Універсальна платформа зворотнього зв'язку з користувачами Morinion допомагає цифровим підприємствам керувати всіма цифровими точками взаємодії (веб, мобільний та електронна пошта), пов'язаними з розумінням поведінки клієнтів. Корпоративна програма дозволяє своїм користувачам створювати налаштовуванні форми зворотного зв'язку, включаючи візуальний зворотний зв'язок із автоматичними знімками екрана. Запуск форм на основі поведінки користувачів, типу пристрою та демографічних показників може надати вам корисну інформацію про відвідувачів вашого сайту та їхню позицію у вашій конверсії. Він також включає візуалізацію за допомогою налаштовуваних інформаційних панелей, а також простий імпорт ваших даних через Excel і CSV. Рейтинг 4,1 зірки в Capterra та 4,1 в G2Crowd робить Morinion програмним забезпеченням, яке варто розглянути.

5. Презентації / контент-дизайн

Наш список корпоративних програмних програм був би неповним без інструменту дизайну вмісту. Дизайн контенту відіграє вирішальну роль у

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

загальній маркетинговій стратегії. Це пояснюється тим, що візуальні ефекти, кольори та форми, які ви представляєте, є елементами ідентичності, за якими ваші клієнти ідентифікуватимуть вас. Замість того, щоб ви бажали рекламувати продукт чи послугу чи зробити презентацію для відповідних зацікавлених сторін, професійне рішення для дизайну вмісту може допомогти вам створити візуально привабливий вміст, який привабливо описує ваше повідомлення. Тепер давайте розглянемо приклад рішення дизайну контенту.

Visme

Ключові особливості:

- Це універсальний набір інструментів для розробки вмісту, який пропонує десятки функцій, які допомагають створювати візуальний вміст.
- Багата база даних різних професійно створених шаблонів у різних категоріях.
- Він пропонує безкоштовну пробну версію та безкоштовний базовий план.

Visme – це інструмент для розробки контенту, який може використовувати кожен. Якщо ви професійний дизайнер чи звичайний користувач без технічних знань, ви можете використовувати інструмент для створення кількох презентацій і цікавої інфографіки в інтерактивний, але простий спосіб. Visme особливо корисний для тих людей, які працюють у сфері освіти та маркетингу. Його також можна використовувати для представлення складної інформації в інтуїтивно зрозумілий спосіб за допомогою кількох шаблонів діаграм. Найкраща частина цього інструменту – його простота. Ви можете створити обліковий запис за лічені хвилини та почати працювати над своїм проектом відразу. Рецензенти Capterra дали Visme дуже солідну оцінку 4,5 зірки, а рецензенти G2Crowd – 4,5.

6. Онлайн-платежі

Індустрія обробки платежів і фінтех значною мірою покладається на програмне забезпечення для онлайн-платежів. Світ став більш зв'язаним, ніж будь-коли раніше, тому підприємствам необхідно піклуватися про свої платежі,

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

вибираючи рішення, якими легко керувати та доступними як продавцям, так і клієнтам у будь-якій точці земної кулі. Давайте розглянемо приклад цієї категорії корпоративних інструментів.

Stripe

Ключові особливості:

- Проста модель ціноутворення на основі використання без налаштування, щомісячних витрат чи прихованих комісій.
- Інтелектуальне керування виставленням рахунків і передплатою, а також складні платіжні потоки для ринків.
- Понад 100 функцій охоплюють усі важливі сфери всієї процедури оплати.

Stripe, світовий лідер у сфері онлайн-платежів, зруйнувала галузь, і її технологія стала еталоном у плавних платежах, виявленні шахрайства, управлінні підписками тощо. Вона досить проста у використанні та пропонує регулярні платежі, які можуть бути корисними деяким компаніям. Stripe стягує невелику комісію за кожен платіж, тому вам не доведеться вкладати великі суми в різні способи оплати. Він пропонує різні варіанти партнерства з іншими сайтами, які, наприклад, спрощують впровадження Stripe у програмне забезпечення для виставлення рахунків. Рецензенти G2Crowd дали цьому інструменту 4,4 зірки, тоді як рецензенти Capterra дали йому надзвичайно високу оцінку – 4,7 зірки, отже Stripe варто протестувати, щоб дослідити всі функції самостійно.

7. Рекрутинг кадрів

Рекрутинг став цифровим як ніколи. З огляду на постійно мінливі вимоги до людських ресурсів для перевірки тисяч кандидатів і обробки всіх даних, які збираються на цьому шляху, корпоративне програмне забезпечення для управління персоналом потрібне, щоб допомогти в наборі, підключенні та адаптації потенційних співробітників. Саме тут таке рішення, як iCIMS, може справити сильний вплив із сильним рейтингом 4,3 на Capterra та 4,1 на G2Crowd.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Icims

Ключові особливості:

- Надійний пакет, орієнтований на маркетинг, комунікації та наймання персоналу.
- Управління, аналіз та оптимізація всіх відповідних етапів життєвого циклу підбору персоналу.
- Інтелектуальні функції попередньої підготовки, щоб полегшити кандидату початок роботи в новій компанії.

Крім того, iCIMS, одне з хмарних програм у цьому списку, обіцяє «безпроблемні подорожі кандидатів, від заявки до пропозиції до попередньої реєстрації». Це програмне забезпечення, яке є корпоративною платформою найму персоналу, пропонує низку автоматизованих робочих процесів, аналітику та функції відстеження, щоб остаточно скоротити цикл найму та оптимізувати весь відділ кадрів. Вони надають повний пакет для управління відносинами з кандидатами, розширений пакет комунікацій і найму, а також партнерську екосистему, яка дозволяє інтегрувати інші рішення для підбору персоналу через їхній ринок.

8. Програмне забезпечення для управління заробітною платою

Управління заробітною платою є, мабуть, одним із найважливіших завдань відділу кадрів. У процесі нарахування заробітної плати підприємствам необхідно забезпечити дотримання податкового законодавства та інших фінансових норм, завдання, яке ускладнюється зі збільшенням кількості працівників. Маючи це на увазі, корпоративне програмне забезпечення для розрахунку заробітної плати допомагає в управлінні та автоматизації процесу розрахунку заробітної плати, забезпечуючи при цьому повну відповідність, залишаючи працівникам відділу кадрів більше часу, щоб зосередитися на інших важливих завданнях. Як приклад для цього розділу ми вибрали Rippling, одну з провідних платформ для нарахування зарплати на ринку.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

вони надають точну документацію, забезпечують швидку реакцію та групові чати за певними темами чи проектами, що зрештою призводить до підвищення продуктивності та кращих бізнес-рішень. Давайте тепер подивимося на обраний нами приклад.

Slack

Ключові особливості:

- Організовував бесіди в канали та прямі повідомлення.
- Історія з можливістю пошуку, яка полегшує пошук інформації з попередніх чатів.
- Нескладна інтеграція з іншим корпоративним програмним забезпеченням (наприклад, Jira, Gmail, Outlook, Zendesk, Monday тощо).

Slack використовується підприємствами в усьому світі, і він отримав стабільний рейтинг 4,7 зірки на Capterra та надійні 4,5 на G2Crowd. Це корпоративне програмне забезпечення підтримує комунікаційну співпрацю, інтеграцію з різними зовнішніми інструментами та файлами під час організації повідомлень у прямих і чистих каналах. Існують різні плани передплати, щоб ваше підприємство могло масштабувати Slack на основі різних бізнес-потреб, забезпечуючи при цьому ефективну командну роботу, яка економить час. Їх вбудована безпека та відповідність гарантують, що компанії матимуть 11 різних функцій безпеки корпоративного рівня, а можливість підключення до інших платформ, зокрема Salesforce, Zendesk, OneDrive або Google Drive, забезпечить командам доступ до кількох інструментів з одного місця.

10. Інтернет-маркетинг

Маркетинг розвинувся в такій кількості різних сфер, що без програмного забезпечення компанії не змогли б керувати всіма точками даних і розробляти успішні стратегії. Від веб-аналітики, контент-маркетингу, реклами та соціальних медіа, серед іншого, правильне корпоративне маркетингове програмне забезпечення дозволяє користувачам повністю контролювати свої дані, відстежувати, контролювати, аналізувати та розвивати рекламні зусилля. Тут ми

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

зосередимося на Moz як повному рішенні для вхідного маркетингу, яке отримало рейтинг Capterra 4,5 зірки та 4,4 у G2Crowd.

Moz

Ключові особливості:

– Універсальний інструмент для відстеження, моніторингу та аналізу вхідних заходів підприємства.

– Численні функції включають відстеження ключових слів, аудит сайту, аналіз зворотних посилань і дослідження ключових слів.

– Підтримує різноманітні пошукові системи Google, Yahoo та Bing для певних країн.

Незалежно від того, чи потрібне вашому підприємству відстеження ключових слів, моніторинг чи аналітика посилань, Moz створено для того, щоб об'єднати всі SEO та вхідні дані під одним дахом. Завдяки таким функціям, як аудит сайту, відстеження рейтингу, аналіз зворотних посилань і дослідження ключових слів, це корпоративне прикладне програмне забезпечення подбає про всі ваші корпоративні пошукові функції та вхідні зусилля. Онлайн-даними для сотень місць легко керувати за допомогою корпоративного рішення, включаючи списки, огляди та засоби пошуку магазинів. Крім того, вони пропонують безкоштовні навчальні матеріали, які можуть покращити будь-який проект і забезпечити рейтинг ваших веб-сайтів.

11. Автоматизація маркетингу

Програмне забезпечення для автоматизації маркетингу полегшує маркетинговим групам досягнення найкращих можливих результатів з мінімальними зусиллями завдяки широкій автоматизації різних процесів, таких як міжканальне відстеження та аналіз поведінки відвідувачів. Вони дозволяють спілкуватися з потенційними клієнтами, клієнтами та партнерами через усі доступні канали, щоб збільшити охоплення та, зрештою, максимізувати цінність клієнтів. Давайте розглянемо цю корпоративну програмну програму на прикладі.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Sendinblue

Ключові особливості:

- Великі можливості автоматизації завдяки індивідуальним правилам і умовам.
- Синхронізація з численними сторонніми постачальниками, такими як WordPress, Shopify або Salesforce.
- Різні робочі процеси можна порівнювати та оптимізувати за допомогою A/B-тестів.

Sendinblue – це комплексне цифрове програмне рішення, яке поєднує численні інструменти маркетингу та продажів на одній платформі. Функції автоматизації маркетингу дозволяють компаніям надавати кожному окремому потенційному клієнту цільову інформацію. Це робиться шляхом створення індивідуальних правил і умов, які запускають надсилання електронних листів або SMS у повністю автоматизований та персоналізований спосіб. Окрім різноманітних платних пакетів, Sendinblue також пропонує безкоштовну версію програмного забезпечення з обмеженим набором функцій. Маючи 4,6 зірки на Capterra та 4,5 зірки на G2Crowd, користувачі оцінюють програмне забезпечення за всіма напрямками.

12. Відеоконференція

Онлайн-конференції та заходи, такі як вебінари, стали невід’ємним елементом повсякденного ділового життя в останні роки, особливо після COVID-19, коли цифрові комунікації значно розвинулися. Ефективний віддалений зв’язок має вирішальне значення для таких подій, і вам потрібне правильне програмне забезпечення, яке може легко обслуговувати велику кількість учасників. Щоб надати учасникам найкращий досвід під час онлайн-заходів, необхідні широкі можливості, які нічим не поступаються класичним офлайн-заходам. Давайте подивимося, як таке корпоративне програмне забезпечення працює в реальному житті.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Gotowebinar

Ключові особливості:

- Численні шаблони вебінарів і гнучкі автоматичні параметри планування.
- Високий рівень конфіденційності та безпеки конфіденційної інформації про учасників.
- Вбудована аналітика у формі аналітичних інформаційних панелей для відстеження взаємодій і показників ефективності.

GoToWebinar є одним із найпопулярніших постачальників корпоративного програмного забезпечення для онлайн-подій. Він пропонує багато функцій, які допомагають користувачам створювати безперебійне онлайн-середовище та проводити великі віддалені події. За даними компанії, понад 50 000 клієнтів у всьому світі покладаються на це рішення, і воно проводить близько 3 мільйонів вебінарів щорічно. GoToWebinar надає комплексне рішення для створення та керування онлайн-подіями, включаючи бібліотеку відео та інтерактивні функції, такі як опитування та запитання та відповіді, щоб помічники могли взаємодіяти так само, як вони робили б це під час особистого заходу. Управління подіями можна легко інтегрувати з існуючим рішенням автоматизації маркетингу або CRM. Крім того, програмне забезпечення має численні позитивні відгуки клієнтів, які виражаються в середньому рейтингу 4,5 зірки на Capterra та 4,2 зірки на G2crowd. Програмне забезпечення можна тестувати безкоштовно протягом 7 днів.

13. Системи управління контентом (cms)

Створення інтуїтивно зрозумілого веб-сайту з різноманітним цифровим вмістом, таким як текст, зображення, відео та розділи, у яких користувачі можуть взаємодіяти з вмістом, є важливим компонентом успішної маркетингової стратегії для будь-якої компанії. Як правило, так звані системи керування контентом (CMS) використовуються для створення, публікації та керування контентом, щоб забезпечити його найкращим чином. Вибираючи правильну CMS, ви спочатку повинні зрозуміти свої конкретні потреби та порівняти

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

доступні рішення на ринку, щоб знайти те, яке ідеально відповідає вашим потребам. Це важливий крок, оскільки пропозиція систем керування вмістом широка, включаючи безкоштовні платформи з відкритим кодом, а також дорогі, але більш повні корпоративні інструменти. Давайте обговоримо це далі через одного з найбільших гравців CMS, WordPress.

Wordpricns

Ключові особливості:

- Безкоштовна платформа з відкритим кодом, яка дозволяє підприємствам створювати веб-сайти, блоги та/або програми.
- Більше 7000 тем, які можна використовувати як стиль для свого веб-сайту та додатково налаштовувати.
- Тисячі простих у встановленні плагінів і доповнень для будь-якого сценарію застосування.

WordPress – це безкоштовна система керування вмістом із відкритим кодом, яка зараз використовується на понад 40% усіх веб-сайтів у всьому світі. Він заснований на PHP і MySQL і порівняно простий у впровадженні та управлінні. Доступні тисячі безкоштовних і платних плагінів, які можна використовувати для налаштування CMS у таких аспектах, як дизайн, збір даних, взаємодія та багато іншого. Незважаючи на те, що WordPress є повноцінним інструментом, який полегшує створення веб-сайтів для багатьох компаній, порівняно з набагато дорожчими комерційними рішеннями CMS, він має деякі обмеження. Тому його не завжди рекомендують дуже великим компаніям із унікальними вимогами, але це дуже хороше бізнес-програмне забезпечення для малого бізнесу. Дуже хороші оцінки на Capterra (4,6 зірки) і G2Crowd (4,4 зірки) підкріплюють цю рекомендацію.

14. Управління проектами (pm)

Програмне забезпечення для керування корпоративними проектами є ідеальним рішенням для команд, які хочуть завершувати проекти вчасно та в рамках бюджету. Незалежно від того, чи потрібне узгодження проекту, гнучке

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

управління в центрі уваги чи просто краща співпраця, є кілька прикладів корпоративного програмного забезпечення, які надають численні функції керування проектами, важливі для досягнення найкращих результатів. Давайте подивимося на це до понеділка. Інструмент, який отримав оцінку 4,6 зірки на Capterra та 4,7 на G2Crowd.

Monday

Ключові особливості:

- Налаштовуванні дошки та доступні шаблони для керування всіма аспектами проекту.
- Автоматизація робочого процесу без необхідності розробки чи впровадження будь-якого коду.
- Інтеграція з більш ніж 50 іншими інструментами, такими як Zoom, Slack, Zendesk тощо.

Один із інструментів управління проектами, який також зосереджується на візуальному представленні, Monday допомагає корпоративним командам керувати робочими процесами, повсякденними завданнями та проектами за допомогою налаштовуваних дошок, які візуалізують робоче навантаження та графік усіх учасників. Візуальне планування роботи команди, використання інформаційних панелей і діаграм для консолідації всіх частин проекту та легкого перегляду результатів робить Monday одним із великих гравців у галузі. Інтеграція з іншими інструментами, такими як Slack, Zendesk, Mailchimp, Hubspot і понад 50 іншими інструментами, дозволяє підприємствам об'єднувати різноманітні дані в одному центральному центрі.

15. Веб-аналітика

Різні типи корпоративного програмного забезпечення були б неповними без веб-аналітики. Веб-сайт є потужним інструментом у сучасному цифровому середовищі, і без нього підприємства були б майже непомітними. Але всі ці дані веб-сайту потрібно проаналізувати, щоб успішно оптимізувати присутність компанії, і тому програмне забезпечення веб-аналітики та інструменти, такі як

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Google Analytics, потрібні як ніколи. Завдяки стабільному рейтингу 4,7 зірки на Capterra та 4,5 на G2Crowd, Google Analytics є найкращим рішенням для аналізу веб-сайтів.

Google analytics

Ключові особливості:

– Найпопулярніша у світі безкоштовна платформа веб-аналітики для моніторингу й аналізу продуктивності веб-сайтів.

– Швидке впровадження, автоматичний збір даних і налаштування звітів.

– Легка інтеграція з іншими інструментами Google, такими як Google Ads або Google Search Console.

Одне з найпопулярніших програм на ринку, Google Analytics, дозволяє відстежувати всю цифрову аналітику, на якій зосереджено ваші маркетингові зусилля в Інтернеті, незалежно від того, платні, звичайні, соціальні чи реферальні тощо. Численні основні та розширені функції дозволяють вам знати, звідки надходить ваш трафік, відстежувати процеси залучення потенційних клієнтів і перевіряти демографічну інформацію, необхідну для оптимізації та перевищення цифрових показників вашого підприємства в пошукових системах. Цей інструмент не тільки призначений для малого бізнесу, але й є одним із програмних програм для бізнесу, які компаніям не слід ігнорувати.

16. Планування ресурсів підприємства (erp)

Програмне забезпечення для планування ресурсів підприємства (ERP) допомагає підприємствам інтегрувати всі аспекти управління запасами, бухгалтерським обліком, CRM, людськими ресурсами, рекламою тощо, щоб обмінюватися інформацією через єдину базу даних, яка також дозволяє користувачам отримувати доступ до даних з різних бізнес-підрозділів. як свої власні. Sage є одним із провідних постачальників у галузі з рейтингом Capterra 4,1 зірки та 4,3 на G2Crowd.

						ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			29

Sage

Ключові особливості:

- Основні можливості продукту включають управління фінансами та виробництвом, ланцюг постачання тощо..
- Вбудований торговий майданчик для розширення продукту відповідно до ваших потреб.
- Додаткові рішення, розроблені безпосередньо компанією Sage або її партнерами.

Sage ERP працює як хмарне рішення, розділене на 3 основні продукти: Sage 100cloud, Sage 300cloud і Sage X3. Використовуючи їх рішення, кожне підприємство може контролювати всі аспекти бізнесу, від фінансів, ланцюга постачання, управління виробництвом до продажів. Їх мета – надати гнучке рішення для компаній, які прагнуть розвиватися, і керувати своїми даними за допомогою єдиного корпоративного програмного рішення. Вони також пропонують ринок, де ви можете підключитися до інших програм і ще більше масштабувати свою ERP-систему.

17. Програмне забезпечення для керування замовленнями (oms)

Як випливає з назви, програмне забезпечення для керування замовленнями, також відоме як OMS, є рішеннями, які допомагають компаніям відстежувати, обробляти та керувати всіма аспектами, пов'язаними з виконанням замовлень, від прийняття замовлення до обробки платежів, параметрів доставки та відстеження доставки, серед інших важливих кроків.. Ці платформи часто включають або можуть бути легко інтегровані з функціями управління запасами, щоб легко уніфікувати весь процес постачання та ефективно керувати замовленнями. Давайте зануримося в наш приклад для цієї категорії.

Brightpearl

Ключові особливості:

- Єдина платформа для багатоканальної обробки замовлень для онлайн- і офлайн-продажів.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Автоматизовані робочі процеси для розподілу запасів, маршрутизації замовлень, доставки та виставлення рахунків тощо.

– Оновлення запасів у режимі реального часу, щоб уникнути перепродажу або інших помилок, пов'язаних із запасами.

Brightpearl – це чудове рішення OMS для великих компаній, які продають свої продукти чи послуги через кілька каналів і потребують уніфікації та організації всіх операцій в одному інтерфейсі. Інструмент можна інтегрувати з великими платформами електронної комерції, включаючи Amazon, Magento та Shopify, для онлайн-продажів або з системою Point of Sales (PoS) для фізичних продажів. Інструмент пропонує функції автоматизованих процесів замовлення, виконання, доставки та виставлення рахунків, а також аналітику та звітність для відстеження ефективності кожного каналу продажів на основі важливих показників, таких як вартість замовлення та дохід. Користувачі Brightpearl оцінили платформу в 4,4 зірки як на Capterra, так і на G2Crowd.

18. Обслуговування клієнтів

Тісно пов'язане з CRM, комплексне програмне забезпечення для обслуговування клієнтів (CS) надасть широкий спектр можливостей для ефективного керування даними клієнтів і забезпечення найкращого досвіду клієнтів. Підприємствам необхідно включити належне рішення CS, щоб стежити за очікуваннями та вимогами клієнтів, що постійно змінюються. Одним із корпоративних додатків, який може належним чином керувати всіма даними, пов'язаними з клієнтами, є Zendesk, який отримав солідні 4,4 зірки на Capterra та 4,3 зірки на G2Crowd.

Zendicnk

Ключові особливості:

– Різноманітні корпоративні пакети програмного забезпечення для підтримки клієнтів, продажів і залучення клієнтів.

– Надає професійні інструменти обслуговування клієнтів, довідковий центр і живі чати.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– Дозволяє зменшити витрати на підтримку та підвищити продуктивність і задоволеність клієнтів.

Допомагаючи підприємствам в управлінні глобально розосередженою командою обслуговування клієнтів, Zendesk пропонує індивідуальні функції для процесів підтримки та конфігурації робочого процесу в одному місці. Вони пропонують багатоканальні рішення для підтримки електронної пошти, чату, телефону або самообслуговування, забезпечуючи платформу для безперервних розмов і системи підтримки. Це чудовий приклад того, що пропонують корпоративні технології, керуючи підтримкою клієнтів незалежно від розташування команди. Завдяки 3 основним планам, починаючи з 5 євро за агента, продукти та платформи Zendesk означають кращий досвід роботи з клієнтами та, зрештою, прибуток.

19. Відстеження проблем

Продовжуючи приклади корпоративного програмного забезпечення, ми пропонуємо інструменти відстеження проблем. Команди програмного забезпечення повинні мати чітке уявлення про процес розробки, усі помилки та їхній статус. Програмне забезпечення для відстеження проблем і звітів про помилки допомагає їм відстежувати, записувати та знаходити всі помилки, які потребують уваги, і надає платформу, яка допоможе керівникам команд призначити проблему людині, яка має її вирішити. Таким чином команди можуть легко визначати пріоритети та ефективніше спілкуватися, а будь-які потенційні проблеми з програмним забезпеченням можна вирішувати негайно.

Jira

Ключові особливості:

– Можливість передавати квитки та налаштовувати їх логіку на основі конкретних потреб (наприклад, передавати їх).

– Детальні параметри фільтрації та аналізу заявок, а також створення звітів і інформаційних панелей.

– Просте підключення до сотень інших інструментів розробника, таких як

						ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			32

GitHub або Bitbucket.

Jira є одним із наших прикладів корпоративних програм, які досить прості у використанні з точки зору функціональності, і тому дуже популярні серед ІТ-спільноти. Команди можуть легко створювати власні дошки залежно від потреб, а також різних точок зору, як-от для розробників, технічних директорів, відстеження тощо. Підключення до інших інструментів розробника, таких як GitHub і Bitbucket, і перегляд коду, який стоїть за заявкою, також можливе з Jira, серед багато інших безцінних ІТ-функцій. Рецензент Capterra дав цьому рішення середню оцінку 4,4 зірки, а рецензенти G2Crowd – 4,2 зірки.

20. Опитування

Підприємства використовують програмне забезпечення для проведення опитувань, щоб підтримувати будь-який аспект, який пов'язує їх із аудиторією, незалежно від того, чи йдеться про клієнтів чи замовників, чи ділових партнерів. Опитування можуть надати інформацію про маркетингові кампанії, службу підтримки, рівень задоволеності співробітників або загальні відгуки щодо конкретних питань бізнесу. SurveyMonkey лідирує, ми бачимо його з сильним рейтингом 4,6 Capterra та 4,4 у G2Crowd.

Surveymonkey

Ключові особливості:

- Одна з найпопулярніших платформ опитувань у всьому світі для будь-якої цільової групи.
- Опитування можна проводити через веб-посилання, електронну пошту чи чат, а також у соціальних мережах.
- Потужні функції аналізу та швидкий експорт результатів у різні формати.

З багатьох типів корпоративного прикладного програмного забезпечення SurveyMonkey стверджує, що йому довіряють 98% компаній зі списку Fortune 500 і є одним із найпопулярніших рішень на ринку. Він пропонує централізовану інформаційну панель адміністратора та журнали аудиту для керування та

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

відстеження активності користувачів, а також легку адаптацію та безпеку на рівні підприємства. Ви також можете інтегрувати дані свого опитування в понад 80 зовнішніх програм, таких як Salesforce, Marketo або Oracle Marketing Cloud.

21. Ідея

Мозковий штурм, інновації та генерування ідей часто об'єднують людей у всьому бізнесі. Зворотній зв'язок, коментарі, системи голосування та збір інформації від співробітників, важливих зацікавлених сторін і зовнішніх партнерів керуються технологіями, які сприяють безперервному вдосконаленню та зберігають повну прозорість – бази даних можуть отримати доступ будь-хто в компанії. Одним із найпопулярніших програм є IdeaScale з рейтингом 4,6 як на Capterra, так і на G2Crowd.

Ideascale

Ключові особливості:

- Комплексна платформа управління інноваціями з трьома основними принципами інновацій: співпраця, оцінка та розвиток.
- Пропонує численні інструменти для інтеграції краудсорсингу в інноваційний процес.
- Кілька варіантів звітності та аналітики за допомогою інформаційних панелей IdeaScale.

Це програмне забезпечення системи управління підприємством можна використовувати для залучення співробітників, що дозволяє командам ставити важливі запитання в одному централізованому місці. Це надзвичайно корисно для підприємств, які мають розрізнені та великі команди, оскільки процеси спілкування часто можуть здаватися складними. Залучення громадськості також є функцією, яка робить це рішення ідеальним для дослідження нереалізованих ідей, які можуть створити незамінну цінність. Орієнтований на збір ідей і надання платформи для голосування, IdeaScale також підтримує інші засоби інновацій, зокрема автоматизацію робочого процесу, формування команди, співпрацю над ідеями та багато іншого.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

22. Платформи хмарних обчислень

Хмарні сервіси використовуються підприємствами, які потребують великої обчислювальної потужності та шукають рішення, які дозволять їм уникнути створення власної ІТ-інфраструктури, зазвичай досить дорогої. Деякі приклади наших корпоративних додатків мають хмару як опцію доступу до ваших даних у режимі онлайн незалежно від місця чи часу. Веб-служба Amazon (AWS) є однією з найпопулярніших хмарних платформ у світі, і її середня оцінка 4,6 зірки для різних продуктів на Capterra та 4,4 на G2Crowd робить це рішення високою позицією у списку вибору хмарної платформи.

Веб-сервіси amazon

Ключові особливості:

- Провідний міжнародний постачальник із широким портфоліо продуктів, що включає понад 160 хмарних сервісів.
- Послуги включають бази даних, обробку даних, інструменти розробника та машинне навчання.
- Прозорі ціни, що залежать від використання, зі знижками за обсяг і короткими періодами сповіщення.

AWS пропонує широкий спектр платформ хмарних обчислень, таких як зберігання даних, послуги обчислень за запитом і мережеві зв'язки. Маючи вражаючу частку ринку в 35%, Amazon пропонує свої послуги підприємствам по всьому світу. Абоненти можуть використовувати конфігурацію віртуального апаратного забезпечення (процесори, графічні процесори для обробки, SSD для зберігання та RAM для пам'яті), операційну систему та програми, такі як CRM.

Ми пояснили визначення корпоративного програмного забезпечення, показали переваги, які вони можуть принести вашій організації, і надали вичерпний перелік корпоративних інструментів, які допоможуть вашій компанії об'єднати команди в межах вашого бізнесу. Кожне рішення має свої особливості, і, як і будь-яке підприємство, ви можете розглянути лише одне або декілька. Звичайно, існують інші сфери застосування сучасного корпоративного

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

програмного забезпечення, які ми тут не обговорювали. Також існує широкий спектр провайдерів для кожної області, ми вибрали приклад для кожного з них на основі рейтингів вище середнього на Capterra та G2crowd. Тому цей список не претендує на повноту. Швидше, воно покликане служити натхненням. Якщо ви хочете глибше дослідити будь-яке зі згаданих програм, майже всі вони пропонують безкоштовну пробну версію для цієї мети. Якщо ви хочете почати з потужного, але простого у використанні програмного забезпечення бізнес-аналітики, ви можете протестувати test datapine 14 днів безкоштовно !

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для реалізації програми мною була використана мова програмування Visual C++. У зв'язку з тим, що сьогодні рівень складності програмного забезпечення дуже високий, розробка додатків Windows з використанням тільки якої-небудь мови програмування значно утрудняється. Програміст повинен затратити масу часу на рішення стандартних завдань по створенню багатовіконного інтерфейсу. Реалізація технології зв'язування й вбудовування об'єктів – OLE – зажадає від програміста ще більш складної роботи. Щоб полегшити роботу програміста практично всі сучасні компілятори з мови C++ містять спеціальні бібліотеки класів. Такі бібліотеки містять у собі практично весь програмний інтерфейс Windows і дозволяють користуватися при програмуванні засобами більш високого рівня, ніж звичайні виклики функцій. За рахунок цього значно спрощується розробка додатків, що мають складний інтерфейс користувача, полегшується підтримка технології OLE і взаємодія з базами даних. Сучасні інтегровані засоби розробки додатків Windows дозволяють автоматизувати процес створення додатка. Для цього використовуються генератори додатків. Програміст відповідає на питання генератора додатків і визначає властивості додатка – чи підтримує воно багатовіконний режим, технологію OLE, тривимірні органи управління, довідкову систему. Генератор додатків, створить додаток, що відповідає вимогам, і надасть вихідні тексти.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Користуючись їм як шаблоном, програміст зможе швидко розробляти свої додатки. Подібні засоби автоматизованого створення додатків включені в компілятор Microsoft Visual C++ і називаються MFC AppWizard. Заповнивши кілька діалогових панелей, можна вказати характеристики додатка й одержати його тексти, постачені великими коментарями. MFC AppWizard дозволяє створювати одновіконні й багатовіконні додатки, а також додатки, що не мають головного вікна, – замість нього використовується діалогова панель. Можна також включити підтримку технології OLE, баз даних, довідкової системи. Звичайно, MFC AppWizard не всесильний. Прикладну частину додатка програмістові прийдеться розробляти самостійно. Вихідний текст додатка, створений MFC AppWizard, стане тільки основою, до якої потрібно підключити інше. Але працюючий шаблон додатка – це вже половина всієї роботи. Вихідні тексти додатків, автоматично отриманих від MFC AppWizard, можуть становити сотні рядків тексту. Набір його вручну був би дуже стомлюючий. Потрібно відзначити, що MFC AppWizard створює тексти додатків тільки з використанням бібліотеки класів MFC (Microsoft Foundation Class library). Тому тільки вивчивши мову C++ і бібліотеку MFC, можна користуватися засобами автоматизованої розробки й створювати свої додатки в найкоротший термін. Як уже згадувався, MFC – це базовий набір (бібліотека) класів, написаних мовою C++ і призначених для спрощення й прискорення процесу програмування для Windows. Бібліотека містить багаторівневу ієрархію класів, що нараховує близько 200 членів. Вони дають можливість створювати Windows-додатки на базі об'єктно-орієнтованого підходу. З погляду програміста, MFC являє собою каркас, на основі якого можна писати програми для Windows. Бібліотека MFC розроблялася для спрощення завдань, що стоять перед програмістом. Як відомо, традиційний метод програмування під Windows вимагає написання досить довгих і складних програм, що мають ряд специфічних особливостей. Зокрема, для створення тільки каркаса програми таким методом знадобиться близько 75 рядків коду. У міру ж збільшення складності програми її код може досягати воістину неймовірних розмірів. Однак та ж сама програма, написана з використанням

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

MFC, буде приблизно в три рази менше, оскільки більшість приватних деталей приховано від програміста.

Одною з основних переваг роботи з MFC є можливість багаторазового використання того самого коду. В зв'язку з тим, що бібліотека містить багато елементів, загальних для всіх Windows-додатків, немає необхідності щораз писати їх заново. Замість цього їх можна просто успадковувати (говорячи мовою об'єктно-орієнтованого програмування). Крім того, інтерфейс, забезпечуваний бібліотекою, практично незалежний від конкретних деталей, його що реалізують. Тому програми, написані на основі MFC, можуть бути легко адаптовані до нових версій Windows (на відміну від більшості програм, написаних звичайними методами). Ще однією істотною перевагою MFC є спрощення взаємодії із прикладним програмним інтерфейсом (API) Windows. Будь-який додаток взаємодіє з Windows через API, що містить кілька сотень функцій. Значний розмір API утрудняє спроби зрозуміти й вивчити його цілком. Найчастіше навіть складно простежити, як окремі частини API зв'язані один з одним! Але оскільки бібліотека MFC поєднує (шляхом інкапсуляції) функції API у логічно організовану безліч класів, інтерфейсом стає значно легше управляти.

Оскільки MFC являє собою набір класів, написаних мовою C++, тому програми, написані з використанням MFC, повинна бути в той же час програмами на C++. Для цього необхідно володіти відповідними знаннями. Для початку необхідно вміти створювати власні класи, розуміти принципи спадкування й вміти перевизначати віртуальні функції. Хоча програми, що використовують бібліотеку MFC, звичайно не містять занадто специфічних елементів з арсеналу C++, для їхнього написання проте потрібні солідні знання в даній області.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інформаційної системи підприємства.

В процесі розробки випускної кваліфікаційної роботи за другим

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

(магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методика побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Структура ІСП представляє загальний дизайн потоку інформації для прийняття корпоративних рішень.

Концептуальна структура

Описує існуючу або плановану інформаційну систему. Кожна функціональна підсистема розділена на чотири частини обробки інформації: обробка транзакцій, оперативний контроль, управлінський контроль і стратегічне планування. Кожна функціональна підсистема зберігає та використовує деякі унікальні файли даних. Там також є деякі загальні файли, якими можуть спільно користуватися підсистеми. Для структурування цих файлів використовується загальна система керування базами даних. Для кожної функціональної підсистеми пишуться спеціальні прикладні програми. Функціональні підсистеми мають посилання на деякі загальні програми, які забезпечують багато функцій. Багато аналітичних моделей і моделей прийняття рішень можуть використовуватися багатьма додатками. Це є основою для інформаційної системи.

Фізична структура

Фізична структура корпоративної інформаційної системи являє собою інтегровану обробку та використання спільних модулів. Інтегрована обробка відноситься до процесу проектування численних взаємопов'язаних програм як окремої системи, так що взаємозв'язки в системі спрощуються, а дублювання вхідних даних зводиться до мінімуму. Інтегрована система завжди долає функціональні обмеження. Конструкцію інформаційної системи часто називають модульною, оскільки вона складається з різних типів модулів. Модуль являє собою невеликий набір інструкцій обробки. Кілька модулів використовуються

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

один раз у певній програмі, тоді як деякі використовуються в багатьох програмах. Модулі, що використовуються в інформаційній системі, можна писати та тестувати незалежно. Це забезпечує більш ефективний підхід до обслуговування шляхом розпізнавання функціональних меж модуля, що змінюється. На фізичну структуру ІСП впливає використання загальних модулів кожного разу для кількох операцій обробки, оскільки модуль перевірки вхідних даних може використовуватися для кожної програми. Нижче наведено проблеми, пов'язані зі структурою ІСП:

– Організації зосереджуються на використанні формальних публічних систем. Офіційні публічні системи добре задокументовані, відомі відповідним особам і доступні всім уповноваженим особам в організації. Оскільки системи є публічними, витрати на розробку розподіляються між багатьма користувачами. Якщо нова особа на посаді приєднується до організації, офіційні публічні системи забезпечують необхідну інформаційну підтримку для належного функціонування на цій посаді.

– Непрактично розробити повністю інтегровану систему, а також важко підтримувати таку систему. Таким чином, Enterprise Information System розроблено за модульним принципом з інтеграцією лише там, де це необхідно.

– Структура корпоративної інформаційної системи не дає детальної інформації про онлайн-взаємодію користувача з системою. Він являє собою єдину підтримку для різних дій, пов'язаних з роботою та управлінням системою.

Роль ІСП

ІСП покращує загальне функціонування підприємства шляхом інтеграції та координації їхніх бізнес-процесів на міцній основі. ІСП може використовуватися всіма частинами та всіма рівнями підприємства. Він надає підприємству технологічну платформу для реалізації різноманітних процесів. Корпоративна інформаційна система забезпечує єдину систему, яка є центральною для організації, яка забезпечує обмін інформацією між усіма функціональними рівнями та ієрархіями управління. Його можна

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

використовувати для об'єднання існуючих програм. ІСП найкраще підходить для великого бізнесу з великою кількістю рухомих частин, тоді як ефективність його роботи перевищує витрати на впровадження. Більшість підприємств у сфері виробництва, дистрибуції, будівництва та інших промислових послуг використовують певний вид ІСП для керування виробництвом, фінансами, ланцюгом постачання та продажами. Але це не означає, що це погано для малого бізнесу. Крім того, малий бізнес зобов'язаний попередньо ретельно провести аналіз витрат і вигод. Правильне та ретельне впровадження ІСП може значно покращити внутрішню логістику компанії. Але слід пам'ятати, що це не гарантує миттєвого успіху. Ретельне планування та належна оцінка необхідні, щоб отримати найкращу віддачу. Якщо робота може бути виконана відповідно до функцій, які пропонує система, потреб користувачів і загальних цілей компанії, переваги цієї системи можна буде побачити в прибутку компанії протягом багатьох років.

Інформаційна система

Інформаційна система визначається як добре структурована група людей, апаратного забезпечення, програмного забезпечення, комунікаційних мереж і ресурсів даних. Ці ресурси відповідають за збір, перетворення та розповсюдження інформації в організації.

Інформаційні системи використовують різні типи інформаційних технологій для надання допомоги бізнес-експертам, наприклад комп'ютерне обладнання, програмне забезпечення, мережі зв'язку та керування ресурсами даних. Технології комп'ютерного обладнання включають мікрокомп'ютери, мейнфрейми, сервери, а також пристрої введення, виведення та зберігання. Програмні технології включають операційну систему, різні веб-браузери та інше програмне забезпечення на основі програм. Технології комунікаційних мереж включають канали зв'язку, програмне забезпечення, необхідне для забезпечення дротового та бездротового доступу, а також Інтернет, інтранет та екстранет. Технології керування ресурсами даних включають програмне забезпечення

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

керування базами даних для розробки, доступу та підтримки організаційних баз даних.

Типи інформаційних систем

Організації використовують різні типи інформаційних систем для задоволення своїх потреб. Класифікацію різних типів інформаційних систем для організації можна здійснити за такими категоріями:

- Система автоматизації офісу.
- Система обробки транзакцій.
- Система підтримки прийняття рішень.
- Система виконавчої інформації.
- Система бізнес-експертів.

Система автоматизації офісу

Система автоматизації офісу допомагає автоматизувати загальні офіційні роботи, такі як керування та організація документів у різних формах, таких як текст, зображення, презентація PowerPoint, файл Word тощо. Це не відіграє важливої ролі в процесі прийняття рішень. Зазвичай люди канцелярського рівня використовують системи автоматизації офісу.

Система обробки транзакцій

Система обробки транзакцій записує транзакції, які виконуються щодня. Наприклад, система обробки транзакцій банку містить операції депозитів і зняття коштів. Системи обробки транзакцій відіграють важливу роль у кожній організації, оскільки вони надають вхідні дані, необхідні для різних інших систем. Наприклад, це допомагає створювати щомісячні звіти про продажі та надавати важливу маркетингову інформацію керівництву. Організація стикається з катастрофічною ситуацією в разі збою в системі обробки транзакцій.

Інформаційна система підприємства

Система підтримки прийняття рішень Системи підтримки прийняття рішень допомагають людям на рівні керівництва організації аналізувати дані та приймати важливі рішення. Вони використовують інформацію з внутрішніх

джерел, таких як системи обробки транзакцій та інформаційні системи управління, і зовнішні джерела, такі як поточні ціни на продукцію конкурентів. Вони зручні та інтерактивні.

Виконавча інформаційна система

Виконавча інформаційна система забезпечує підтримку керівництва вищого рівня для прийняття рішень. Виконавча інформаційна система робить судження та оцінює нестандартні рішення, оскільки не існує попередньо визначеної процедури для прийняття таких рішень. Керівна інформаційна система скорочує час і зусилля, необхідні для отримання цінної для керівників інформації, фільтруючи, стискаючи та ефективно відстежуючи важливі дані. Виконавча інформаційна система менш аналітична, ніж система підтримки прийняття рішень. Він забезпечує узагальнене обчислювальне середовище, а не спеціальне програмне забезпечення.

Бізнес-експертна система

Бізнес-експертна система – це система, заснована на знаннях. Він працює як експерт-консультант і надає експертні поради користувачам. Прикладами є системи моніторингу й керування процесом, а також системи діагностичного обслуговування.

Декомпозиція

Важко зрозуміти складну систему в цілому. Таким чином, необхідно розкласти його на окремі функціональні системи. Виробнича організація може розкласти інформаційні системи на наступні функціональні системи:

- Система продажів і маркетингу.
- Система виробництва.
- Система логістики.
- Система фінансів і бухгалтерського обліку.
- Система людських ресурсів.
- Система вищого менеджменту.

Система продажів і маркетингу

Система продажів і маркетингу забезпечує підтримку виконання оперативні дії, такі як працевлаштування та навчання торгового персоналу, регулярне планування продажів і рекламних завдань, а також регулярний аналіз продажів за регіоном, типом продукту, клієнтом тощо. Транзакції, що обробляються, є замовленнями на продаж замовлення на просування тощо. На рівні управління він порівнює загальну ефективність продукту з маркетинговим планом. Система продажів і маркетингу допомагає стратегічно розробити план нових маркетингових стратегій і дослідити нові ринкові можливості шляхом аналізу клієнтів і конкурентів, а також інших факторів.

Виробнича система

Виробнича система обробляє транзакції, такі як виробничі замовлення, замовлення на складання тощо. Вона відповідає за оперативну діяльність, як-от планування та розклад завдань, пов'язаних із виробництвом, найм персоналу, необхідного для виробництва, перевірку якості та технічне обслуговування, а також звіти про ефективність покоління. На управлінському рівні він допомагає менеджерам ефективно створювати зведені звіти про ефективність, щоб менеджери могли приймати правильні рішення. Це дозволяє менеджерам розробляти альтернативні стратегії, такі як розміщення нових заводів або використання нових технологій для досягнення довгострокової мети.

Логістична система

Транзакції логістичної системи пов'язані із закупівлею, інвентаризацією, виробництвом, доставкою та виставленням рахунків. На операційному рівні логістична система виконує такі дії, як закупівля, управління запасами та розподіл, а також підтримує звіти, що містять таку інформацію, як минулі неоплачені закупівлі та відвантаження, список товарів, яких немає в наявності та надлишок, доходи від запасів та звіти про ефективність постачальника та постачальника. Логістична система допомагає організації правильно використовувати запаси. Він використовується для аналізу та розробки нових

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

планів розповсюдження та нової політики для постачальників, а також стратегій «робити проти покупки».

Система фінансів і бухгалтерського обліку

Відповідальність фінансової функції полягає в управлінні фінансовими активами організації, такими як готівка, запаси та надання позик клієнтам для отримання максимального прибутку. Бухгалтерський облік охоплює діяльність, пов'язану з підтримкою фінансових звітів організації, таких як звіти про доходи та баланси, і керування ними, а також підготовку бюджету та аналіз витрат. Транзакції, що підлягають обробці, – це продажі, виставлення рахунків, чеки, переміщення запасів тощо. Система фінансів і бухгалтерського обліку створює звіти, що містять інформацію про помилки та винятки, переривання під час обробки та необроблені транзакції. Управлінський контроль системи фінансів і бухгалтерського обліку передбачає порівняння бюджетної вартості з реальною вартістю фінансових активів плюс вартість обробки даних бухгалтерського обліку. Ця система також відповідає за планування довгострокової фінансової стратегії та політики щодо зменшення впливу податків і бюджету.

Система людських ресурсів

Відповідальність системи людських ресурсів полягає в тому, щоб визначити потенційних працівників, найняти та навчати нового персоналу, організувати програми розвитку особистості та навичок та вести записи кожного працівника. Система людських ресурсів стосується оперативних заходів, таких як зайнятість, навчання, припинення інформаційної системи підприємства, зміни зарплати та надання пільг. Ця система допомагає керівництву у створенні звітів про ефективність та аналізу відхилень між запланованою та реальною ефективністю щодо кількості найманого персоналу, витрат на працевлаштування та навчання, виплату зарплати та інших витрат. Система людських ресурсів стратегічно стосується оцінки інших стратегій щодо працевлаштування, навчання, оплати та засобів, що надаються працівникам, щоб утримати цінних працівників, необхідних для досягнення організаційних цілей.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Система вищого управління

Ця система реагує на запити та приймає рішення, отримуючи доступ до бази даних або отримуючи інформацію від інших систем організації. Він охоплює оперативні заходи, такі як підготовка графіків зустрічей, підтримка листування та контактних файлів. Ця система допомагає вищому керівництву для оцінки ефективності функцій інших систем. На стратегічному рівні він забезпечує силу для бізнесу шляхом розробки та виконання планів гарантії основних ресурсів.

Елементи інформаційної системи

Кожна інформаційна система вимагає деяких ресурсів. Ці ресурси необхідні для зйомки та зберігання вхідних даних, обробки їх та проведення інших контрольних заходів для перенесення даних у корисні інформаційні продукти. Основними елементами інформаційної системи є:

– Апаратне забезпечення: це стосується комп'ютерних систем та периферійних пристроїв.

– Програмне забезпечення: воно включає системне програмне забезпечення, програмне забезпечення та процедури. Прикладами є програмне забезпечення для обробки текстових систем операційних систем та процедура введення даних.

– Дані: Для організації, зберігання та доступу до даних інформаційних систем використовуються різні технології, такі як бази даних, склади даних та дані MART.

– Люди: Особи зобов'язані обробляти та контролювати операції інформаційних систем. Людські ресурси включають кінцевих користувачів та фахівців. Кінцеві користувачі використовують інформаційну систему. Наприклад, клієнти, службовці, менеджери тощо. Фахівці розробляють інформаційну систему. Наприклад, системні аналітики, розробники програмного забезпечення тощо.

– Мережа: мережеві ресурси відносяться до середовища зв'язку, такі як двожилий провід, коаксіальний кабель, волоконно-оптичний кабель та

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

бездротові медіа та мережі, такі як Інтернет, інтранет та екстранет. Для підтримки функціональних можливостей комунікаційної мережі використовуються різні апаратні, програмні та технології даних.

3.2 Розробка структурної схеми

Діяльність будь-якого підприємства або організації супроводжується обробкою й переміщенням великої кількості інформації. І від швидкості обробки й передачі інформації прямо залежить успіх діяльності підприємства (час – гроші).

Прискорити цю роботу можна тільки за рахунок застосування технічних засобів. Але тут керівника підприємства підстерігає одна, досить розповсюджена помилка – придбання й установка дорогого встаткування, витрати на впровадження не дають бажаного результату.

Справа в тому, що встаткування й програмні засоби для інформаційної системи підприємства – це всього лише набір інструментів. Тільки повне подання про тім як, і які завдання будуть вирішуватися з використанням цих інструментів, може дати позитивні результати при впровадженні інформаційної системи.

Типова структура

Інформаційна система підприємства являє собою листковий пиріг, де кожний шар забезпечує функціонування наступних перед ним шарів:

- Виконавчий шар – набір завдань, рішення яких забезпечує ефективну роботу підприємства.
- Прикладний шар – набір прикладного програмного забезпечення, безпосередньо застосовуваного для обробки ділової інформації.
- Системний логічний шар – системне програмне забезпечення, що забезпечує функціонування фізичного шару, механізми обміну інформацією, розмежування прав доступу, і т.п.
- Фізичний шар – устаткування, кабельні мережі, канали передачі даних.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Для шарів 2, 3 і 4 можливе застосування готових типових проектів, зі змінами під конкретні вимоги замовника (локальна мережа підприємства).

Етапи створення системи

Реалізація системи звичайно починається з етапу проектування виконавчого шару, але для цього на підприємстві повинна бути створена модель документообігу й визначені функції кожного виконавця в цій моделі.

Звичайно модель документообігу тісно зв'язана зі структурою підприємства. На основі моделі документообігу проектується виконавчий шар. Визначити, що повинне входити у виконавчий шар системи – це і є основна завдання проектування виконавчого шару. Виконавчий шар може формуватися на підприємстві самотужки, можливо із залученням сторонніх фахівців.

Інші шари проектуються послідовно один за одним, на основі вихідних даних попереднього шару. І якщо на підприємстві немає досвіду проектування й будівництва в даній області, то для економії засобів і часу має сенс звернутися до сторонніх спеціалізованих організацій, що зарекомендували себе в проведенні подібних робіт.

Типовий проект системного шару системи

Основна проблема проектування цього шару системи полягає в тому, яку платформу вибрати за основу. Будемо розглядати тільки платформи на процесорі Intel, тому що інші системи не знайшли в нас широкого поширення. У цей час із відомих мережних платформ на ринку представлені наступні:

- Microsoft Windows.
- Open Enterprise Server.
- Різновиди Unix.

Кожна з перерахованих систем має свої достоїнства й недоліки. Зупинимося як приклад на платформі Microsoft Windows, як найпоширенішої. Серія, що просувається на ринок, Windows Server 2022 має у своєму складі повний перелік продуктів, необхідних для побудови мережі будь-якої складності, і легко адаптується з мережами на інших платформах.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

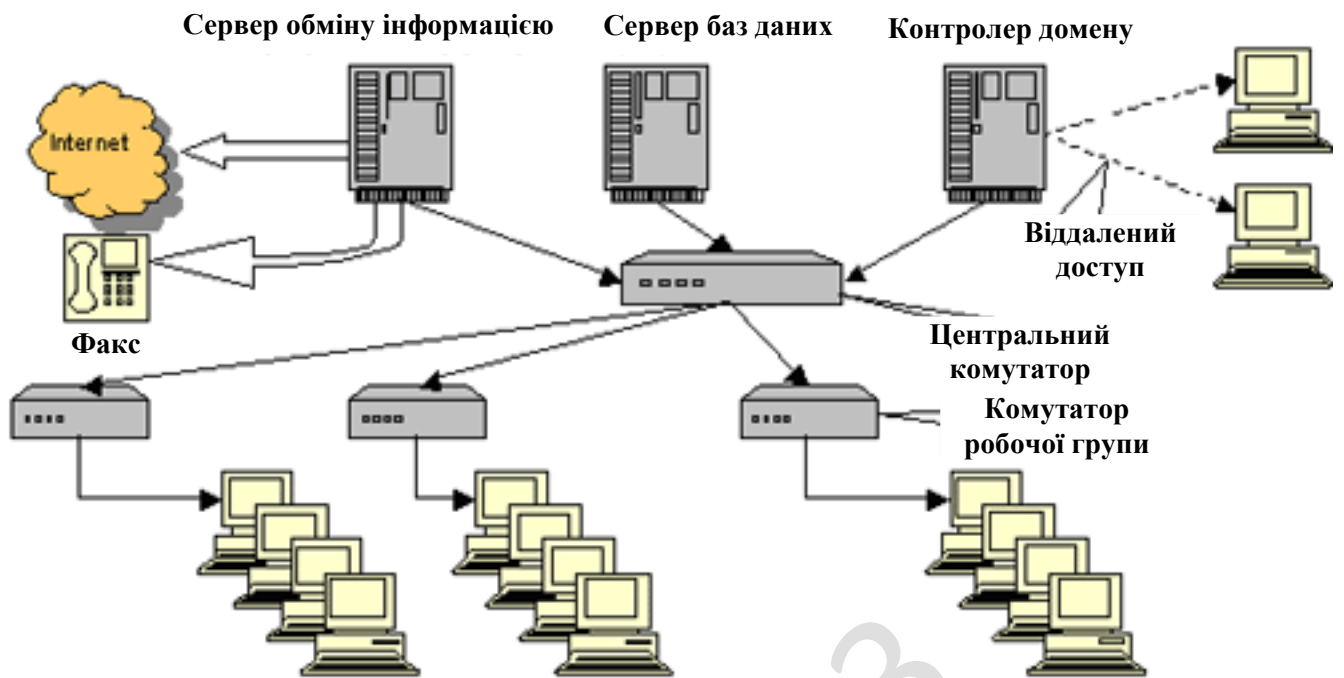


Рисунок 3.1 – Структурна схема системи

Даний типовий проект легко адаптується під будь-які вимоги, і перекриває практично всі потреби в інформаційному забезпеченні, і дозволяє взаємодіяти з іншими аналогічними системами.

Прикладний шар системи

Фактично цей шар визначає ті прикладні програмні засоби, за допомогою яких вирішуються завдання по обміні й обробці інформації. Звичайно на підприємстві використовуються програми автоматизації бухгалтерського обліку, складського обліку. Рідше застосовуються системи фінансового планування, кадрового обліку, автоматизації роботи підрозділів постачання й збуту. Однак, часто буває, що використовувані програми автоматизації не сумісні між собою, і як наслідок не дозволяють повністю автоматизувати документообіг на підприємстві.

Існують комплексні системи, що покривають всі потреби підприємства. Вартість таких систем значно вище, ніж окремих програм, строки їхнього повного впровадження більше. До того ж вартість впровадження часто буває

більше вартості самої системи. На жаль, часто буває (і тому є приклади) що затративши більші фінансові ресурси, керівництво підприємства так і не одержує очікуваного ефекту через те, що на етапі проектування виконавчого й прикладного шарів не були пророблені всі деталі системи й не враховані всі вимоги, яким система повинна відповідати.

Для успішного вибору й впровадження прикладної системи необхідно чітко знати відповіді на наступні питання:

- Які процеси діяльності підприємства можуть бути автоматизовані.
- Як повинні взаємодіяти різні частини системи.
- Яка інформація й у якому виді може бути надана як вихідні звіти системи.
- Яким образом у систему може вводитися вхідна інформація.
- Чи можливо зміна алгоритмів обробки інформації, і яким образом.
- Чи можливо доповнення й зміна системи у випадку зміни потреб підприємства.
- Які апаратні й програмні засоби необхідні для функціонування системи.
- Як буде здійснюватися підтримка й відновлення системи.
- Як вироблятися навчання персоналу.
- Також бажано знати, де вже використовується обрана система і які про неї відкликання в споживача.

Маючи відповіді на ці питання можна ухвалювати рішення щодо придбання конкретної системи.

Тут навмисно не приводяться приклади конкретного застосування яких або систем автоматизації й управління підприємства, оскільки зовнішнє наповнення таких систем і виконувані функції дуже схоже один на одного, і кожна з них має свої функції й недоліки, і що є важливим для одних, може бути другорядним для інших.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Документи – основа управління

Робота з документами є основною частиною процесів управління й прийняття управлінських рішень. Процес ухвалення управлінського рішення можна розбити на три етапи:

- одержання інформації;
- переробка й аналіз інформації;
- підготовка й ухвалення рішення.

Всі ці етапи самим тісним образом пов'язані з документообігом.

У пропонованій архітектурі системи управління прикладні компоненти опираються на документообіг підприємства, що є документаційним забезпеченням управління. Система документообігу повинна базуватися на програмно-технічних платформах підприємства, включати засоби й правила створення документів, підтримки їхнього руху, електронний архів і т.д. [2].

Всі інші компоненти автоматизованої система управління (АСУ) підприємством повинні опиратися на систему документообігу підприємства. Як провести границі між документообігом і прикладними компонентами АСУ, які інтерфейси повинні бути між цими шарами?

З юридичної точки зору, система управління організацією – це її документообіг. Будь-яка комісія (податкова, екологічна й т.д.) працює тільки з документами, перевіряючи їхню дійсність і відповідність реальності. Тому можна сказати: «управління організацією це її документообіг». З іншого боку, згідно [5]: «документообіг – процес проходження документів усередині системи управління», тобто тільки облік надходження документів, їхнє твердження, виконання, здача в архів і т.п. без аналізу змісту документів. В [9] приводиться набір функцій, необхідний для розширення поняття документообігу убік обліку змісту документів.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Чого не вистачає інформаційним системам для управління підприємством?

Не вистачає багато чого:

- Упорядкування документопотоків.
- Маршрутизації проходження документів.
- Визначення відповідальних виконавців.
- Формування завдань і строків.
- Контролю виконання.
- Аналізу й оцінки вироблених робіт.

Інакше кажучи, їм не вистачає справжньої системи документообігу. У прикладних інформаційних системах (музейних, банківських, патентних і т.д.) необхідно перетворити документи в основну одиницю управлінського «господарства», підтримати основні функції обробки документів (підготовка нормативних актів і договорів, включення робіт у план, видача довідок, звітів, підготовка й узгодження рішень і т.п.).

При цьому повинні бути дотримані наступні принципи:

- Зберігання всіх документів (включаючи версії).
- Можливість побачити документи із всіх робочих місць.
- Можливість пересування документів з будь-якої підсистеми в кожному (подібно паперовим).
- Інформаційна безпека.
- Юридична значимість.

Суть пропонованого підходу:

- документ – основна одиниця управлінського господарства;
- підтримка основних функцій обробки документів (підготовка нормативних актів, договорів, включення в план, узгодження, видача довідок, звітів...);
- прозорість діяльності організації (взаємозв'язку документів і операцій).

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

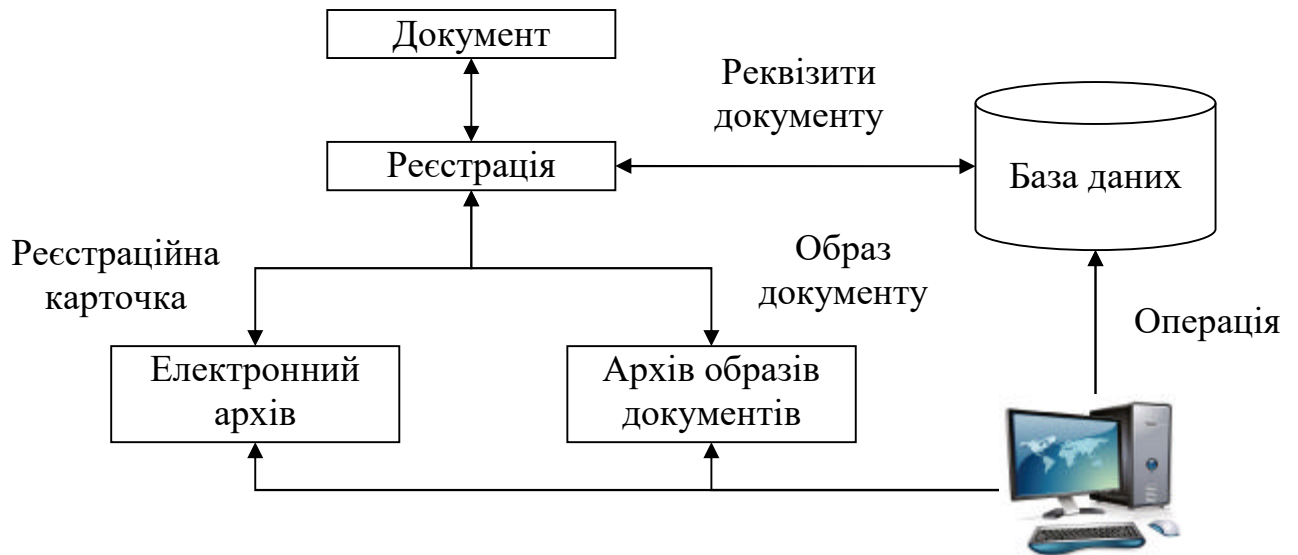


Рисунок 3.2 – Функціональна схема системи

Документообіг удалося розкласти [9, 20] на основні компоненти, їх 16 (з обліком підкомпонент – 88). Для організації інструментарію при складанні АСУ необхідно розкласти на компоненти етап переробки й аналізу інформації. Це трудомістка й не просте завдання. Зокрема, потрібно розкласти на незалежні компоненти з фіксацією їхніх взаємозв'язків (інтерфейсів) системи ERP. Очевидно, що як компоненти повинні бути апарат подій, убудовані процедури переробки інформаційних ресурсів (подоба убудованих процедур у СУБД), генерація внутрішніх електронних документів для взаємодії між підсистемами, ведення протоколів обробки зовнішніх і внутрішніх заявок (документів) між підсистемами й багато чого іншого.

Документообіг уже розкладений на компоненти, і тому на його основі можна будувати АСУ. Якби вдалося також розбити на компоненти системи класу ERP, то з таких складових уже можна було б конструювати АСУ, підключаючи й готові прикладні компоненти (кадри, зарплата й ін.).

У порівнянні з такими монстрами, як ERP, запуск яких на конкретному підприємстві полягає у відсіканні безлічі непотрібних можливостей системи засобами налаштування, що є трудомісткою й які вимагають високої кваліфікації.

У перераховані вище роботах – АСУ організацій, побудовані на основі системи документобігу шляхом доробки порівняно невеликого числа спеціальних блоків.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Інтерфейс ПЗ.
- Налаштування ПЗ.
- Введення документів.

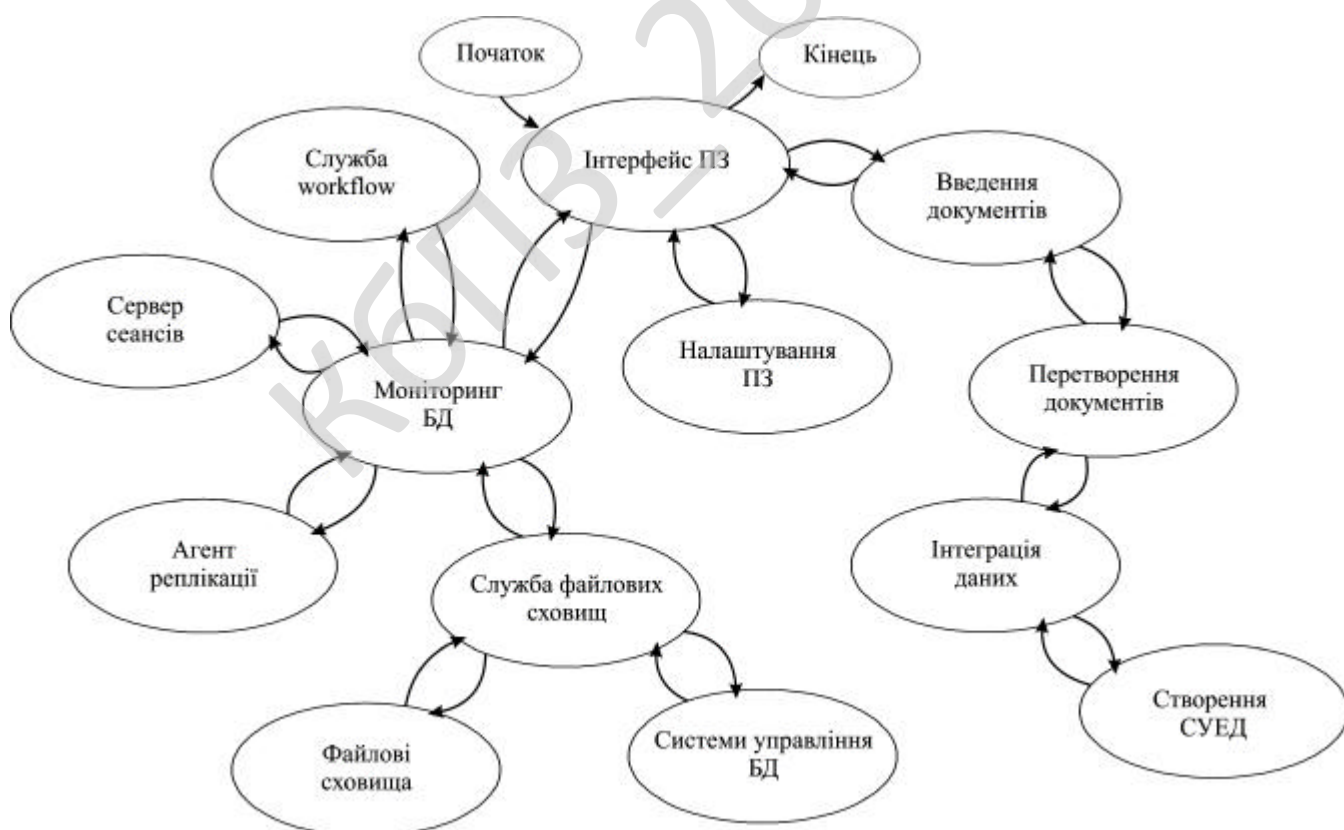


Рисунок 3.3 – Діаграма взаємодії процесів

- Перетворення документів.
- Інтеграція даних.
- Створення СУЕД.
- Моніторинг БД.
- Служба workflow.
- Сервер сеансів.
- Агент реплікації.
- Служба файлових сховищ.
- Файлові сховища.
- Системи управління БД.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ – 2023

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю інформаційної системи підприємства.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я над основною проблематикою.

Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

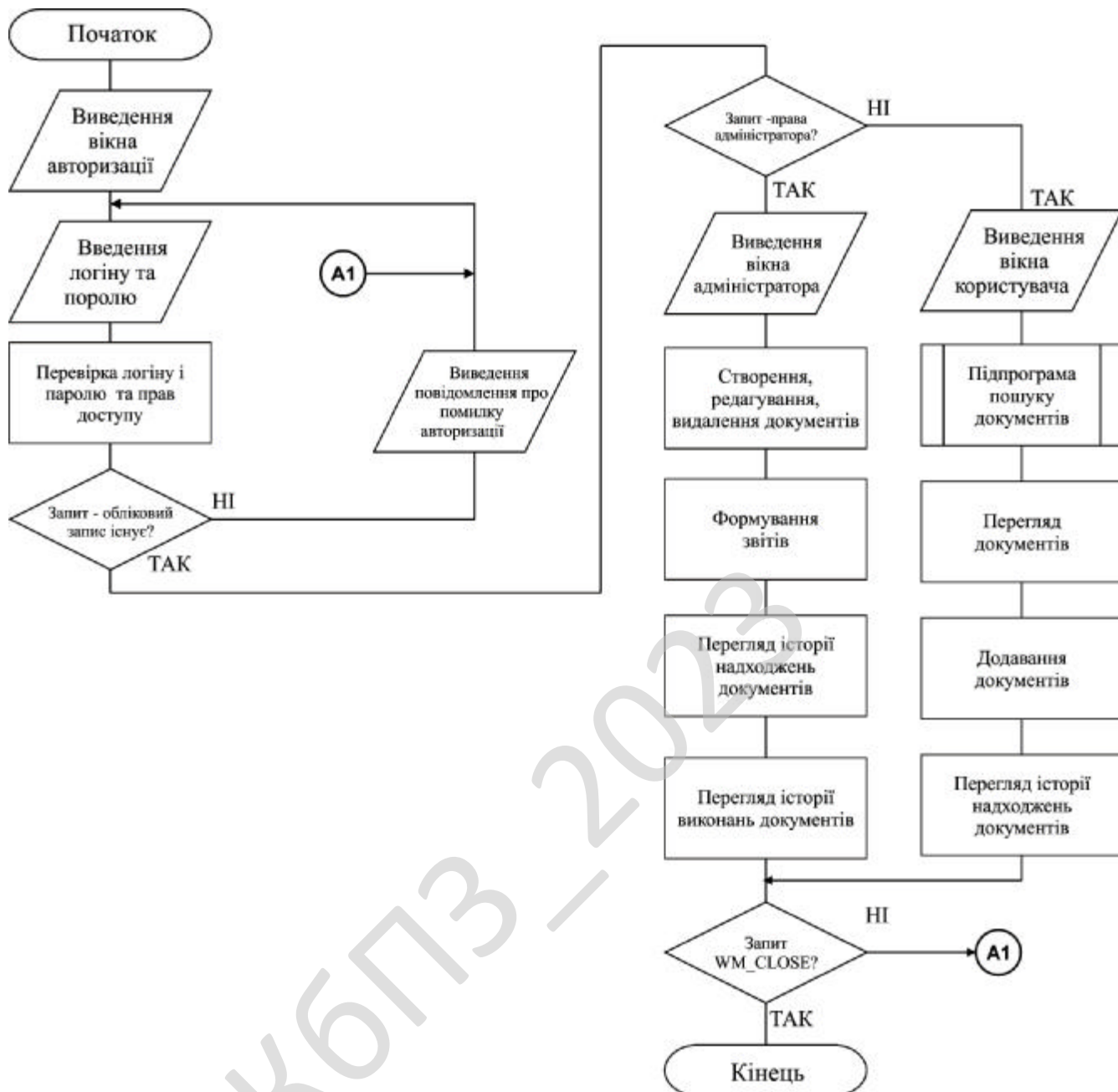


Рисунок 4.1 – Блок схема основної програми

Опис алгоритмів функціонування системи

Проведемо оцінку якості роботи системи управління електронним документообігом. Всім відомі основні переваги застосування теорії нечітких множин для рішення завдання оцінки ефективності роботи СУЕД у порівнянні із традиційними підходами теорії автоматичного управління.

Розроблено критерії оцінки якості роботи системи управління електронним документообігом.

Показано архітектуру нечіткої моделі управління якістю роботи СУЕД. Основним модулем системи оцінки якості роботи СУЕД є блок прийняття рішень, хоча інші блоки не менш важливі для нормального функціонування моделі.

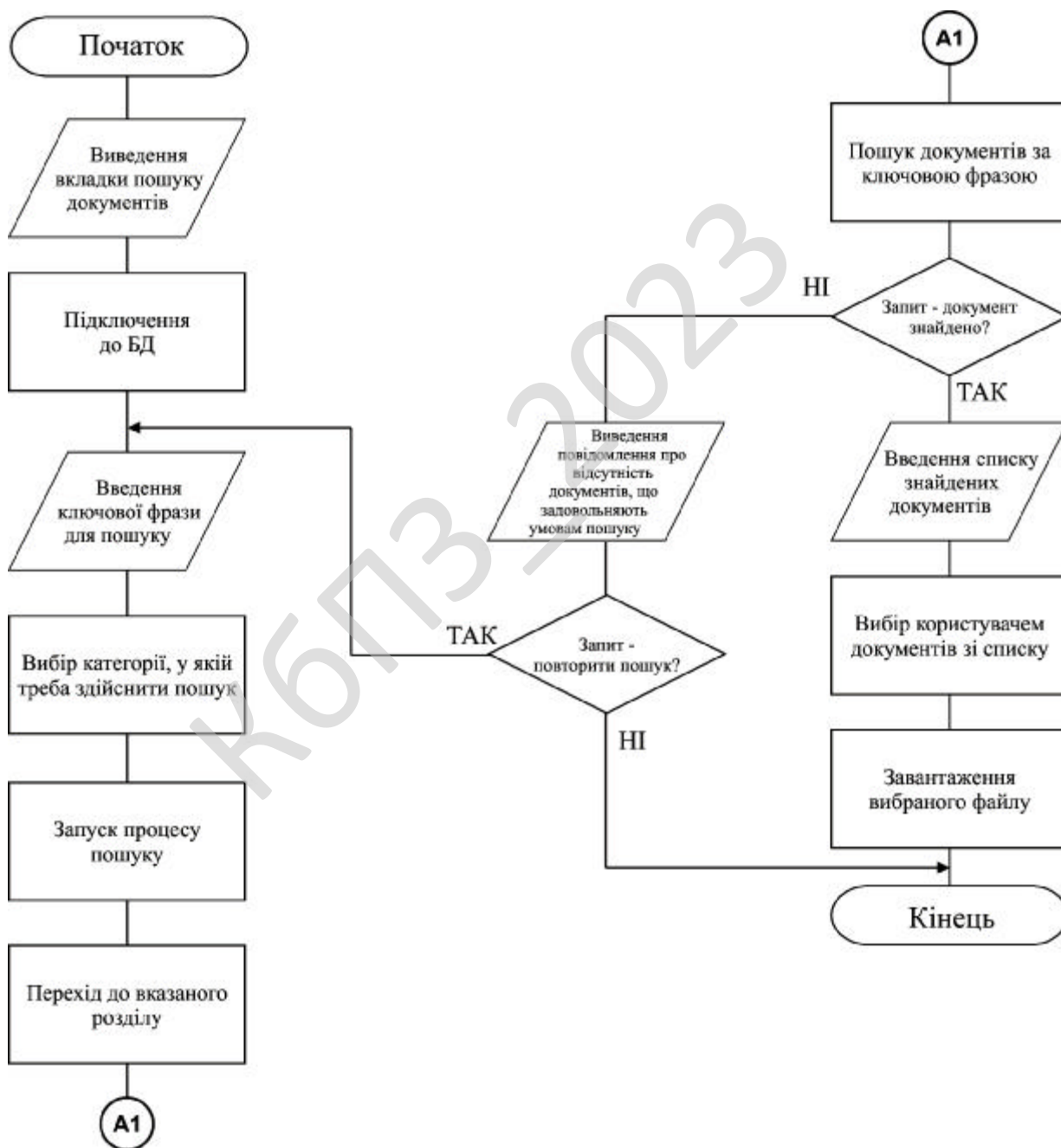


Рисунок 4.2 – Блок схема підпрограми

Блок оцінки станів, на основі вступник на його вхід інформації, буде формалізований опис ситуації, що виникла при роботі СУЕД.

У блоці видачі керуючих впливів здійснюється перехід від внутрішньої форми завдання керуючих рішень до зовнішньої форми.

Для оцінки якості роботи СУЕД скористаємося критерієм, вираженим безліччю M , елементи якого є інтервалами якісної (нечіткої) шкали:

$M = \{\text{«дуже добре»}, \text{«добре»}, \text{«задовільно»}, \text{«незадовільно»}\}$, для цього методом експертних оцінок були визначені конкретні значення процентного вмісту тексту, графіки й таблиць у файлах певного формату.

Виходячи з аналізу експериментальних даних, був отриманий масив $\{W_N(X)\}$, що містить інтервали значень фізичної величини, що характеризує ефективність роботи елемента, при яких забезпечується ефективна робота елементів пристрою (системи) і масив $\{WW_N(X)\}$, що містить інтервали значень фізичної величини, що характеризує функціонування елемента, і значення функції приналежності стану елемента системи до одного з можливих станів. Розроблено вирішальні правила виходячи зі значень функції приналежності стану елемента одному з можливих станів для кожного з інтервалів базових значень.

Для перевірки запропонованих правил була зроблена побудова моделі оцінки якості роботи системи управління електронним документообігом у середовищі MatLab 7.0 для СУЕД фінансової організації.

Модель заснована на методі Мамдани, що має у своїй основі базу знань у вигляді сукупності нечітких предикатних правил виду:

П1: якщо $x \in A1$, тоді $z \in B1$,

П2: якщо $x \in A2$, тоді $z \in B2$,

.....

Пn: якщо $x \in An$, тоді $z \in Bn$,

З метою перевірки даних про те, що використовувані критерії можуть застосовуватися для діючих систем проведений ряд експериментів –

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

інструментальних досліджень на діючій системі, для чого якого були обмірювані й визначені величини:

- кількість переданих у СУЕД файлів;
- формати переданих у СУЕД файлів.

Отримано підтвердження того, що використовувані для оцінки ефективності роботи СУЕД критерії можуть застосовуватися для діючих систем. Розроблено методику оцінки ефективності роботи елементів СУЕД. Зроблено оцінку ефективності й повернення інвестицій від впровадження відкритої системи управління електронним документообігом.

Також при розробці використовувались концепції діаграм діяльності. Тобто в UML, візуальне представлення графу діяльності. Граф діяльності є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Це фундаментальна одиниця визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Опис функціоналу головного вікна

Приведемо частину програмного коду, яка реалізує функції головного вікна.

```
/*-----*/  
// NAME: GiveDoc  
// DESCRIPTION: Функція видачі документів користувачеві
```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

```

// INPUT:                listDoc - список документів користувача (для виконання
                          додавання)
//
//                        ddISBNHash - хеш вхідного номера документа, що потрібно
                          одержати
// OUTPUT:               TRUE - операція пройшла успішно
//                        FALSE - у наявності немає жодного екземпляра заданого
                          документа

/*****/
Boolean BAKALAVR::GiveDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
// Визначити індекс по вхідному номеру документа
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
// Зменшити кількість вільних документів
    if((ddIndex != -1) && ((CDoc^)listDoc[ddIndex])->DecFreeNumber())
    {
// Додати документ у список користувача
        listReader->Add(listDoc[ddIndex]);
// Функція відробила успішно
        return true;
    }
    else
    {
// Такого документа не є в наявності
        return false;
    }
}

/*****/
// NAME:                  TakeDoc
// DESCRIPTION:           Функція прийняття документів від користувача назад у
                          систему

// INPUT:
// listDoc - список документів користувача (для виконання видалення)
//
//                        ddISBNHash - хеш вхідного номера документа, що потрібно
                          повернути

/*****/
void BAKALAVR::TakeDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
// Визначити індекс по вхідному номеру документа
// (документ існує, тому що його вхідний номер документа був повідомлений
// користувачеві)
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
// Видалити документ зі списку користувача

```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

```

listReader->RemoveAt(ddIndex);
// Збільшити кількість вільних документів
((CDoc^)listDoc[ddIndex])->IncFreeNumber();
}
/*****/
// NAME:          LoadDocList
// DESCRIPTION:   Функція завантаження документів системи з файлу
// INPUT:         N/D
/*****/
void BAKALAVR::LoadDocList()
{
// Перевірити існування файлу
if(!File::Exists(DOCS_FILE))
{
// Файл не знайдений, закінчити виконання функції
return;
}
// Відкрити файл
StreamReader^ srDoc = gcnew StreamReader(DOCS_FILE);
// Продовжувати, поки не буде знайдений кінець файлу
while(srDoc->EndOfStream == false)
{
// Створити новий об'єкт "документ"
CDoc^ NewDoc = gcnew CDoc();
// Завантажити з файлу й установити параметри документа
NewDoc->SetName(srDoc->ReadLine());
NewDoc->SetAuthor(srDoc->ReadLine());
NewDoc->SetISBN(srDoc->ReadLine());
NewDoc->SetTheme(srDoc->ReadLine());
NewDoc->SetPages(Convert::ToInt32(srDoc->ReadLine()));
NewDoc->SetTotalNumber(Convert::ToInt32(srDoc->ReadLine()));
NewDoc->SetFreeNumber(Convert::ToInt32(srDoc->ReadLine()));
// Включити документ у загальний список документів
listDoc->Add(NewDoc);
}
// Закрити файл
srDoc->Close();
}
/*****/
// NAME:          SaveDocList
// DESCRIPTION:   Функція збереження документів системи у файл
// INPUT:         N/D

```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

```

/*****/
void BAKALAVR::SaveDocList()
{
// Перевірити існування файлу
if(File::Exists(DOCS_FILE))
{
// Видалити файл
File::Delete(DOCS_FILE);
}
// Створити й відкрити файл
StreamWriter^ swDoc = gcnew StreamWriter(DOCS_FILE);
// Цикл запису по одному документу
for(Int32 i = 0; i < listDoc->Count; i++)
{
// Одержати параметри документа й записати їх у файл
swDoc->WriteLine(((CDoc^)listDoc[i])->GetName());
swDoc->WriteLine(((CDoc^)listDoc[i])->GetAuthor());
swDoc->WriteLine(((CDoc^)listDoc[i])->GetISBN());
swDoc->WriteLine(((CDoc^)listDoc[i])->GetTheme());
swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetPages()));
swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetTotalNumber()));
swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetFreeNumber()));
}
// Закрити файл
swDoc->Close();
}
/*****/
// NAME: FindDoc
// DESCRIPTION: Функція пошуку документа по заданих параметрах
// INPUT: strFindDoc - рядок для пошуку
/*****/
ArrayList^ BAKALAVR::FindDoc(String^ strFindValue)
{
// Список для результату
ArrayList^ listResult = gcnew ArrayList();
// Шукане значення без обліку регістра
String^ strValue = strFindValue->ToLower();
// Задана порожній рядок?
if(String::IsNullOrEmpty(strValue->Trim()))
{
//Вивести весь список
for(Int32 i = 0; i < listDoc->Count; i++)

```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

```

    {
// Додати документ у результуючий список
        listResult->Add(listDoc[i]);
    }
}
else
{
// Вибрати з умовою
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
// Пошук рядка в кожному полі без обліку регістра
        if((((CDoc^)listDoc[i])->GetName()->ToLower()->IndexOf(strValue) != -1 ||
            (((CDoc^)listDoc[i])->GetAuthor()->ToLower()->IndexOf(strValue) != -1 ||
            (((CDoc^)listDoc[i])->GetISBN()->ToLower()->IndexOf(strValue) != -1 ||
            (((CDoc^)listDoc[i])->GetTheme()->ToLower()->IndexOf(strValue) != -1 ||
            Convert::ToString((((CDoc^)listDoc[i])->GetPages()->IndexOf(strValue) != -1)
                {
// Додати документ у результуючий список
                    listResult->Add(listDoc[i]);
                }
            }
        }
// Повернути список збігів
        return listResult;
    }
/*****/
// NAME:          ViewDoc
// DESCRIPTION:   Функція перегляду інформації про заданий документ
// INPUT:         ddISBNHash - хеш вхідного номера документа, інформацію
                 про яку треба переглянути
/*****/
CDoc^ BAKALAVR::ViewDoc(Int32 ddISBNHash)
{
// Визначити індекс по вхідному номеру документа
// (документ існує, тому що його вхідний номер документа
// був повідомлений користувачеві)
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
// Повернути елемент "документ" із загальне списку
    return (CDoc^)listDoc[ddIndex];
}
/*****/
// NAME:          ReadLogs

```

					БКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

```

// DESCRIPTION:      Функція зчитування історії з файлу
// INPUT:           N/D
/*****/
void BAKALAVR::ReadLogs()
{
// Файл із історією надходжень існує?
  if(File::Exists(LOG_INPUT))
  {
// Зчитування історії надходжень
    logInput->strLog = File::ReadAllText(LOG_INPUT);
  }
  else
  {
    logInput->strLog = "";
  }
// Файл із історією списань існує?
  if(File::Exists(LOG_OUTPUT))
  {
// Зчитування історії списань
    logOutput->strLog = File::ReadAllText(LOG_OUTPUT);
  }
  else
  {
    logOutput->strLog = "";
  }
}
/*****/
// NAME:           WriteLogs
// DESCRIPTION:    Функція збереження історії у файл
// INPUT:          N/D
/*****/
void BAKALAVR::WriteLogs()
{
// Запис історії
  File::WriteAllText(LOG_INPUT, logInput->strLog);
  File::WriteAllText(LOG_OUTPUT, logOutput->strLog);
}
/*****/
// NAME:           ClearLogs
// DESCRIPTION:    Функція очищення історії
// INPUT:          N/D
/*****/

```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

```

void BAKALAVR::ClearLogs()
{
    logInput->strLog = "";
    logOutput->strLog = "";
}
/*****/
// NAME:          ViewInputLog
// DESCRIPTION:   Функція перегляду історії надходжень
// INPUT:         N/D
/*****/
String^ BAKALAVR::ViewInputLog()
{
    return logInput->strLog;
}
/*****/
// NAME:          ViewOutputLog
// DESCRIPTION:   Функція перегляду історії списань
// INPUT:         N/D
/*****/
String^ BAKALAVR::ViewOutputLog()
{
    return logOutput->strLog;
}
/*****/
// NAME:          BAKALAVR
// DESCRIPTION:   Конструктор класу
// INPUT:         N/D
/*****/
BAKALAVR::BAKALAVR(void)
{
    listDoc = gcnew ArrayList();
    logInput = gcnew CLogger();
    logOutput = gcnew CLogger();
}
/*****/
// NAME:          GetIndexByISBNHash
// DESCRIPTION:   Функція одержання індексу документа в загальному списку по
                вхідному номеру документа
// INPUT:         ddHashValue - значення хеша вхідного номера документа
/*****/
Int32 BAKALAVR::GetIndexByISBNHash(Int32 HashValue)
{

```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

```

// Цикл пошуку документа із заданим вхідним номером документа
for(Int32 i = 0; i < listDoc->Count; i++)
{
    // Хеш-значення вхідного номера документа заданого документа збігається з
    // хеш- значенням вхідного номера документа знайденого документа?
    if(HashValue == ((CDoc^)listDoc[i])->GetISBNHash())
    {
        // Документ знайдений, повернути індекс
        return i;
    }
}
// Документ не знайдений, повернути -1
return -1;
}

/*****/
// NAME:                AddDoc
// DESCRIPTION:         Функція додавання нового документа
// INPUT:               SourceDoc - об'єкт "документ" для включення в список
// OUTPUT:              TRUE - документ успішно доданий
//                      FALSE - такий документ уже існує в системі
/*****/
Boolean BAKALAVR::AddDoc(CDoc^ SourceDoc)
{
    // Документи з таким вхідним номером документа не існує?
    if(this->GetIndexByISBNHash(SourceDoc->GetISBNHash()) == -1)
    {
        // Додати заданій документ у загальний список
        listDoc->Add(SourceDoc);
        // Записати подія в лог
        logInput->WriteEvent(
            "Новий документ '" + ((CDoc^)SourceDoc)->GetName() +
            "', Автор '" + ((CDoc^)SourceDoc)->GetAuthor() +
            "', Вхідний номер документа " + ((CDoc^)SourceDoc)->GetISBN());
        // Функція відробила успішно
        return true;
    }
    else
    {
        // Документ із таким же вхідним номером документа існує, повернути помилку
        return false;
    }
}

```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

```

/*****/
// NAME:                RemoveDoc
// DESCRIPTION:         Функція видалення документів із системи
// INPUT:               ddISBNHash - значення хеша вхідного номера документа
// OUTPUT:              0 - видалення зроблене
//                      1 - видалення неможливо, не всі екземпляри перебувають у
//                      системі
//                      2 - документ із заданим вхідним номером документа не
//                      знайдений
/*****/
Int32 BAKALAVR::RemoveDoc(Int32 ddISBNHash)
{
// Знайти документ по вхідному номеру документа
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
// Документ із таким вхідним номером документа існує?
    if(ddIndex != -1)
    {
// Перевірити, що жодного документа немає в користувача
        if(((CDoc^)listDoc[ddIndex])->GetTotalNumber() ==
            ((CDoc^)listDoc[ddIndex])->GetFreeNumber())
        {
// Записати подія в лог
            logOutput->WriteEvent(
                "Виконаний документ '" + ((CDoc^)listDoc[ddIndex])->GetName() +
                "', Автор '" + ((CDoc^)listDoc[ddIndex])->GetAuthor() +
                "', Вхідний номер документа '" + ((CDoc^)listDoc[ddIndex])->GetISBN());
// Видалити знайдений документ зі списку
            listDoc->RemoveAt(ddIndex);
// Функція відробила успішно
            return 0;
        }
        else
        {
// Повернути помилку, не всі документи перебувають у системі
            return 1;
        }
    }
    else
    {
// Повернути помилку, документ із таким вхідним номером документа не був знайдений
        return 2;
    }
}

```

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою наступного алгоритму захисту інформації DSA.

Відправник і одержувач електронного документа використовують при обчисленні великі цілі числа: G і P – прості числа, L біт кожне ($512 < L < 1024$); q – просте число довжиною 160 біт (дільник числа $(P-1)$). Числа G , P , q є відкритими й можуть бути загальними для всіх користувачів мережі.

Відправник вибирає випадкове ціле число X , $1 < X < q$. Число X є секретним ключем відправника для формування електронного цифрового підпису. Потім відправник обчислює значення $Y = G^X \text{ mod } P$. Число Y є відкритим ключем для перевірки підпису відправника. Число Y передається всім одержувачам документів.

Цей алгоритм також передбачає використання однобічної функції гешування $h(-)$. У стандарті DSS визначений алгоритм безпечного гешування SHA. Для того щоб підписати документ M , відправник гешує його в ціле геш-значення m : $m = h(M)$, $1 < m < q$, потім генерує випадкове ціле число K , $1 < K < q$, і обчислює число r : $r = (G^K \text{ mod } P) \text{ mod } q$. Потім відправник обчислює за допомогою секретного ключа X ціле число s :

$$s = \frac{m + r * X}{K} \text{ mod } q .$$

Пара чисел r і s утворить цифровий підпис $S = (r, s)$ під документом M . Таким чином, підписане повідомлення являє собою трійку чисел $[M, r, s]$. Одержувач підписаного повідомлення $[M, r, s]$ перевіряє виконання умов $0 < r < q$, $0 < s < q$ і відкидає підпис, якщо хоча б одна із цих умов не виконана. Потім одержувач обчислює значення $w = 1/s \text{ mod } q$, геш-значення $m = h(M)$ і числа $u_1 = (m * w) \text{ mod } q$, $u_2 = (r * w) \text{ mod } q$. Далі одержувач за допомогою відкритого ключа Y обчислює значення $v = ((G^{u_1} * Y^{u_2}) \text{ mod } P) \text{ mod } q$ і перевіряє виконання умови $v = r$. Якщо умова $v = r$ виконується, тоді підпис $S = (r, s)$ під документом M визнається одержувачем справжнім.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні розділи:

- Розділ меню, яке складається з підрозділів: Файл; Довідка.
- Розділ навігаційних вкладок з підвкладками: Документи; Звіти; Історія.
- Розділ функціональних прапорів.
- Розділ функціональних кнопок.
- Розділ інформаційної панелі.

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

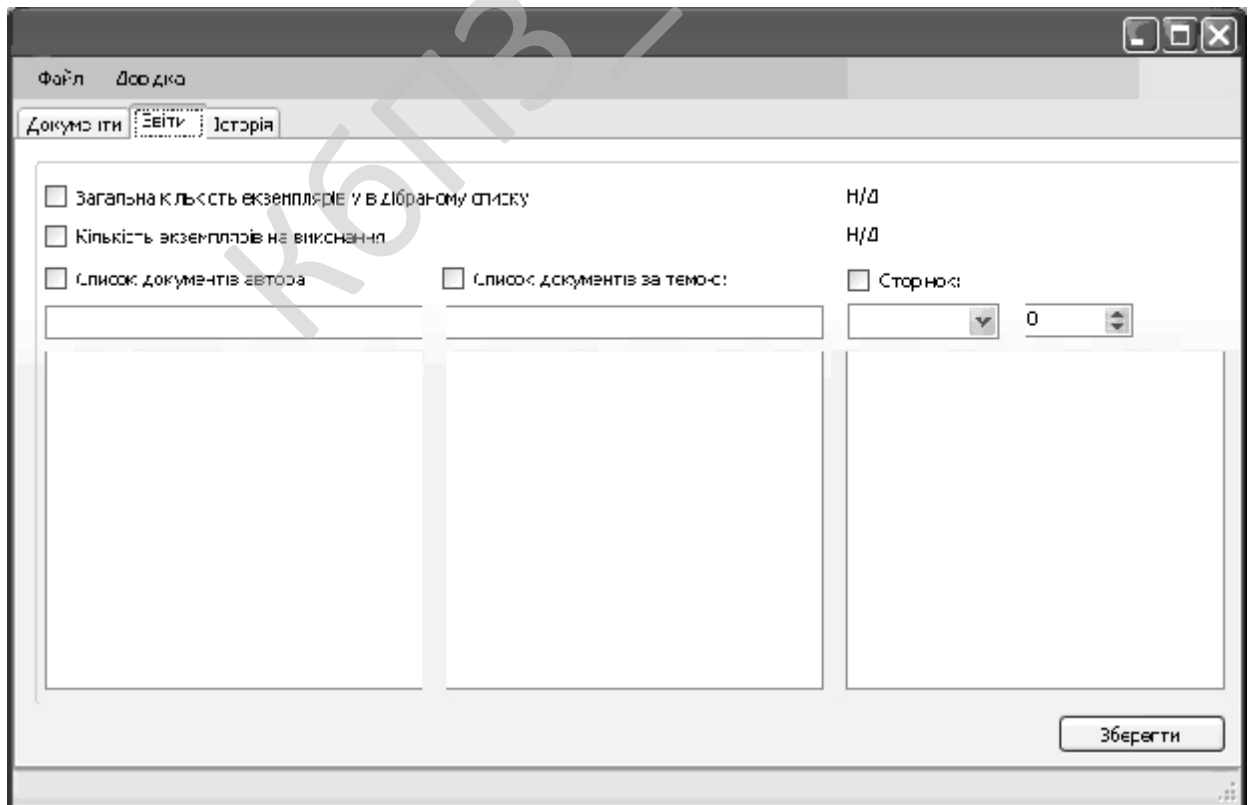


Рисунок 5.1 – Головне вікно ПЗ

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

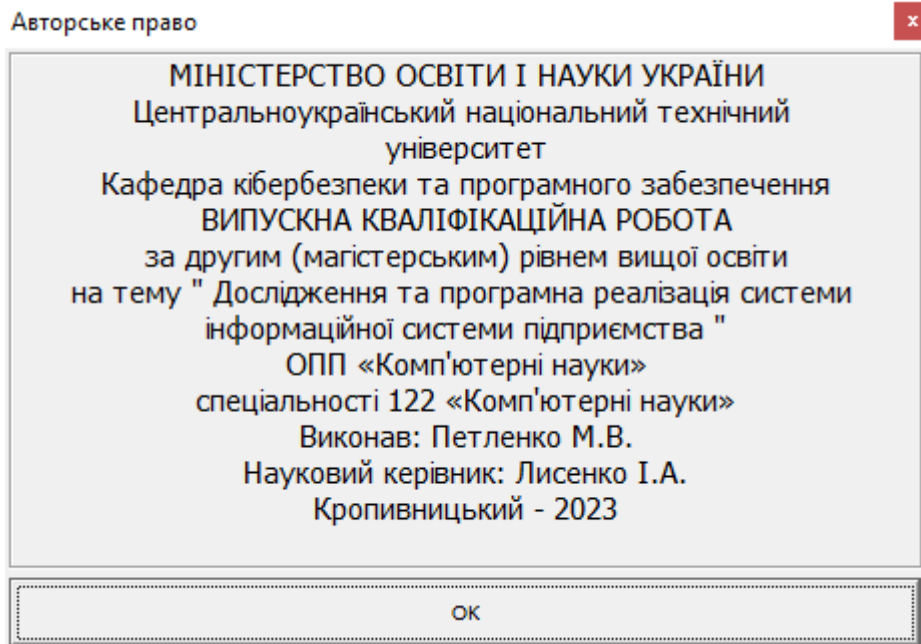


Рисунок 5.2 – Авторське право

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інформаційної системи підприємства.

Метою розробки є дослідження та програмна реалізація системи інформаційної системи підприємства.

Об'єктом дослідження є процес інформаційної системи підприємства.

Предметом дослідження є методи інформаційної системи підприємства.

Методи дослідження базуються на методах побудови інформаційних систем, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод інформаційної системи підприємства.
- Розроблено вітчизняний продукт інформаційної системи підприємства, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Продовження таблиці 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	60000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B \quad (7.1)$$

де А – коефіцієнт Боема, А=2,45;

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \cdot \Pi V_j, \quad (7.3)$$

де ΠV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 100 = 168 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнан ня	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{др}^c = \frac{3_{ч} \cdot n_{mic}}{1,2} \quad (7.6)$$

$$\Phi_{др}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$C_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}} \quad (7.7)$$

$$C_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів–електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	0,8	0,2
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,4	
Всього		1,6	

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,5	0,2
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,3	
	Розміщення графіки і контенту на Інтернет сторінках	0,3	
Всього		1,6	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,2
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		1,6	

Складемо штатний розклад виконавців у таблицю 7.5.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	15000	30000
Продакт-менеджер	0,5	12000	12000
Інженер-програміст	4,8	14000	134400
Інженер-електронщик	1,2	9000	21600
Інженер-системотехнік	0,2	9000	3600
Адміністратор мережі	0,2	11000	4400
Системний програміст	0,2	9000	3600
Дизайнер WEB	0,2	9000	3600
Інженер-верстальник	0,2	9000	3600
Бухгалтер-економіст	0,2	10000	4000
Всього за період розробки	$R_{cn}=8,7$	-	$\Phi_{роб}=220800$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} \cdot F_{pq}}, \quad (7.8)$$

де $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{220800}{8,7 \cdot 48} = 529 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

$$B_{y\partial} = R_{cn}^1 S_y \Pi_{nl}, \quad (7.9)$$

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць.

S_y – питома площа на одне робоче місце, m^2 ,

Π_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн на одне робоче місце. Тобто

$$I_{nv} = R_{cn}^1 \cdot \Pi_m, \quad (7.10)$$

де Π_m – ціна меблів для одного робочого місця, грн.

$$I_{nv} = 8 \cdot 3500 = 28000 \text{ грн}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались за комерційною пропозицією фірми Brain за 14.10.23 – джерело <http://brain.com.ua/>

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771
Системний блок		7771
Процесор	Intel Core™ i3 10105F (BX8070110105F) 1200, 4 ядра, 8 потоків, 3.7 GHz, 4.4 GHz, TDP – 65 Вт, 14nm	-
Системна плата	GIGABYTE H510M S2H V3 сокет – 1200, DDR4, 3200 MHz, LAN – 1 Гбит/с, D-Sub (VGA), DisplayPort, HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Жорсткий диск	SSD 2.5" 256GB Mibrand, 3D TLC NAND	-
Оперативна пам'ять	DDR4 8GB 3200 MHz Dato	-
Відеокарта	Radeon R7 350 2Gb PCI-Express 3.0, 2 ГБ, GDDR5, 128 Bit	-
Корпус	Vinga CS105B-450W Miditower, ATX	-
інше	Клавіатура, мишка	-
Монітор	Монітор BenQ GL2450HM Black	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	11771	9416,8	103584,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	125216,3

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	125216	-	-
Всього по групі	125216	50	62608

Продовження таблиці 7.8

1	2	3	4
Група 5,6			
4. Вимірювальні пристрої	5190	-	-
5. Транспортні засоби	143000	-	
6. Господарський інвентар	28000	-	-
Всього по групі	176190	20	35238
7. Нематеріальні активи	60000	10	6000
Разом	$K_p = 1769406$		$A_p = 174246$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 529 \cdot 209 / 58 = 1906 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де H_q – норматив додаткової зарплати, %

$$Z_d = 1906 \cdot 10 \cdot 0,01 = 191 \text{ грн}$$

Відрахування на соціальні потреби за нормативом $H_c=22\%$ від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(1906+191) = 461 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z=15\%$ від основної зарплати

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де H_z – загальногосподарські витрати, %

$$G_{ocn} = 1906 \cdot 15 \cdot 0,01 = 286 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де Z_{M1} – вартість паперу, грн., Z_{M2} – вартість запам'ятовуючих пристроїв, грн., Z_{M3} – вартість фарби, картриджів, тонеру, грн., N_e – кількість екземплярів програм, шт.

Згідно виданих викладачем норм приймаємо одну пачку паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n=80$ грн., визначаємо вартість паперу за період розробки $N_m=2$ міс:

$$Z_{M1} = Ц_n \cdot N. \quad (7.16)$$

$$Z_{M1} = 80 \cdot 1 = 80 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum Ц_d, \quad (7.17)$$

де $Ц_d$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 2 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 2 грн/шт.

$$Z_{M2} = 58 \cdot 8 + 10 = 474 \text{ грн.}$$

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_{3.}, \quad (7.18)$$

де: $C_{3.}$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (80 + 474 + 1702) / 58 = 39 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де H_n – норматив витрат на освоєння нових мов програмування, %

$$O_n = 1906 \cdot 15 \cdot 0,01 = 286 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 58$ прим.)

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 174246 \cdot 2 / (58 \cdot 12) = 501 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_M + O_n + A_m. \quad (7.21)$$

$$C_n = 1906 + 191 + 461 + 286 + 39 + 286 + 501 = 3670 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_p) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

де P_c – рівень рентабельності, %

$$P_p = 0,01 \cdot 50 \cdot 3670 = 1835 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	Z_o	1906
2. Додаткова зарплата виконавців	Z_d	191
3. Відрахування на соціальні потреби	C_{ou}	461
4. Загальногосподарські витрати	G_{ocn}	286
5. Витрати на матеріали	Z_M	39
6. Освоєння нових операційних систем, мов програмування	O_n	286
7. Амортизація основних фондів	A_m	501
8. Повна собівартість програмного забезпечення	C_n	3670
9. Плановий прибуток	P_p	1835
10. Ціна підприємства $C_n = C_n + P_p$	C_n	5505
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{ов} \cdot C_n$	$ПДВ$	1101
12. Відпускна ціна програмної продукції $Ц = Ц_n + ПДВ$	$Ц$	6606

Витрати на технічне обслуговування:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де T_p – кількість годин обслуговування системи за рік, год.,

Z_z – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 300 годин на рік до 250 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 300 \cdot 100 \cdot 1,1 \cdot 1,22 = 40260 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 250 \cdot 100 \cdot 1,1 \cdot 1,22 = 33550 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням спожитої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$).

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

Витрати на електроенергію при впровадженні нової системи не змінюються.

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	6606	–	3303
Всього відрахувань	-	–	6606	–	3303

$$T_{cn} = \frac{K_n - K_6}{I_6 - I_n} \quad (7.28)$$

$$T_{cn} = \frac{6606}{40260 - 36853} = 1,9 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	58
2. Повна собівартість розробленої програми	Грн.	3670
3. Ціна розробленої програми	Грн.	5505
4. Плановий прибуток від реалізації розробленої програми	Грн.	1835
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1769406
7. Загальний прибуток від реалізації програмної продукції	Грн.	106430
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	77389
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	2,8
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	6606
11. Величина економічного ефекту у користувача програмної продукції	Грн.	104
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	1,9

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

КБПЗ_2023

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – система збереження життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи. Загальні положення державної політики, щодо галузі охорони праці зазначені у Законі України “Про охорону праці”. Цей Закон визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні [20]. Законодавство про працю містить норми і вимоги з техніки безпеки і виробничої санітарії, норми, що регулюють робочий час і час відпочинку, звільнення та переведення на іншу роботу, норми праці щодо жінок, молоді, гігієнічні норми і правила тощо. Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний – на професійні спілки. Контроль за безпекою праці здійснюють також, державні й відомчі спеціалізовані інспекції. Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності робітників розумової діяльності. Їх праця стала більш інтенсивною, напруженою і вимагає значних витрат розумової, емоційної і фізичної енергії. Це призвело до необхідності у знаходженні комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку. Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальні машини (ЕОМ) та інше обладнання є джерелами безпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат;

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 3 м, довжина – 4,6 м, висота – 3 м.

У зазначеному приміщенні працює 1 особа.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F=ESKZ/n,$$

де: F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S – площа освітлюваного приміщення (у нашому випадку $S=3 \times 4,6 = 13,8$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i=S/(h(A+B)),$$

де: S – площа приміщення, $S = 13,8$ м²;

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

h – розрахункова висота підвісу, $h = 3$ м (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 3$ м;

B – довжина приміщення, $B = 4,6$ м.

Підставимо всі значення у формулу та визначемо індекса приміщення:
 $i=0,43$.

Знаючи індекс приміщення, за знаходимо $n = 0,23$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп). Підставимо всі значення у формулу, визначемо світловий потік:
 $F=29700$ Лм.

Для розрахунку думемо використовувати світлодіодні панелі LED панель 42Вт 6000K SUNLED 000000127, світловий потік яких $F_l = 3990$ Лм.

Число ламп визначається по формулі:

$$N=F/F_l$$

де: F – світловий потік,

F_l – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначемо індекса приміщення: $N=$
 $29700/3990=7,4$ шт.

Приймаємо необхідну кількість світлодіодних світильників 8 шт.

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

Список використаних джерел інформації

1. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
4. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
5. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>
6. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>
7. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення 19.09.22).
8. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://spo.stu.cn.ua>
9. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

10. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральнoукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу : <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240>

КБПЗ – 2023

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інформаційної системи підприємства.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інформаційної системи підприємства.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інформаційної системи підприємства.

– Досліджена система інформаційної системи підприємства.

– На основі отриманих результатів досліджень створена програмна реалізація системи інформаційної системи підприємства.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інформаційної системи підприємства.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Visual C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм DSA.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 104 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,9 роки.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Петленко М.В. Дослідження та програмна реалізація системи інформаційної системи підприємства // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
3. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
4. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
5. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
6. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
7. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
8. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
9. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
10. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
11. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
12. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.

13. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

14. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.

15. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

16. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143

17. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

18. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

19. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

20. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

21. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

22. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

23. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

24. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

25. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

26. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

27. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

28. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

29. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

30. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

31. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 2(72), С. 170-178.

32. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.

33. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

34. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

35. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.

36. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

37. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнoукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

38. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

39. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральнoукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

40. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

41. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities. Scientific

& practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Электронный Журнал]. Georgia. Tbilisi: SCSA – 2018.

42. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

43. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

44. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the DOM XSS vulnerability. International Conference «Information technologies, systems and networks ITSN-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.

45. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA – 2017.

46. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). – Полтава: ПолтНТУ. – 2017. – С. 112-115.

47. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів .Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті».

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

48. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.

49. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). – Харків: ХУПС. – 2016. – С.96-100.

50. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". – Випуск 8 (145). – Х.: ХУПС – 2016. – С. 77-80.

51. Смірнов О.А., Лисенко І.А. Розробка впорядкованих каскадних таблиць рішень із використанням матриць слідування. Збірник наукових праць "Системи обробки інформації". – Випуск 6 (143). – Х.: ХУПС – 2016. – С. 216-220.

52. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод кількісної оцінки ризиків розроблення програмного забезпечення. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 128-133.

53. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод якісного аналізу ризиків розроблення програмного забезпечення. Наука і техніка Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.

54. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Проблеми аналізу та оцінки ризиків інформаційної діяльності. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 40-42.

					ВКРМ-122.23.0072.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-122.23.0072.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Петленко М.В.				<i>Дослідження та програмна реалізація системи інформаційної системи підприємства</i>	Літ.	Аркуш	Аркушів
Перевірів	Лисенко І.А.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22МЗ			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи інформаційної системи підприємства.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 37-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи інформаційної системи підприємства.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-122.23.0072.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи інформаційної системи підприємства;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-122.23.0072.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Visual C++.

					ВКРМ-122.23.0072.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-122.23.0072.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 113 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 14.12.2023 р.

					ВКРМ-122.23.0072.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Лисенко І.А

*Дослідження та програмна реалізація
системи інформаційної системи підприємства*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 72

Літера: РП

Кропивницький – 2023 року

```

<?xml version="1.0" encoding=" utf-8" ?>
- <root>
- <xsd:schema id="root" xmlns="" xmlns:xsd=" " xmlns:msdata="urn: schemas-
microsoft-com: xml-msdata">
  <xsd:import namespace=" " />
- <xsd:element name="root" msdata:IsDataSet="true">
- <xsd:complexType>
- <xsd:choice maxOccurs="unbounded">
- <xsd:element name="metadata">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element name="value" type="xsd:string" minOccurs="0" />
</xsd:sequence>
  <xsd:attribute name="name" use="required" type="xsd:string" />
  <xsd:attribute name="type" type="xsd:string" />
  <xsd:attribute name="mimetype" type="xsd:string" />
  <xsd:attribute ref="xml:space" />
</xsd:complexType>
</xsd:element>
- <xsd:element name="assembly">
- <xsd:complexType>
  <xsd:attribute name="alias" type="xsd:string" />
  <xsd:attribute name="name" type="xsd:string" />
</xsd:complexType>
</xsd:element>
- <xsd:element name="data">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>
  <xsd:element name="comment" type="xsd:string" minOccurs="0" msdata:Ordinal="2"
/>
</xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1"
/>
  <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
  <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
  <xsd:attribute ref="xml:space" />
</xsd:complexType>
</xsd:element>
- <xsd:element name="resheader">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1"
/>
</xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required" />
</xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
- <resheader name="resmimetype">
  <value>text/ microsoft-resx</value>
</resheader>
- <resheader name="version">
  <value>2.0</value>
</resheader>
- <resheader name="reader">
  <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
- <resheader name="writer">

```

```
<value>System.Resources.ResXResourceWriter, System.Windows.Forms,  
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
```

```
</resheader>
```

```
- <data name="txtInfo.Text" xml:space="preserve">
```

```
<value>Магістерська робота: «Дослідження та програмна реалізація системи  
інформаційної системи підприємства». Керівник: Лисенко І.А.. Розробив Петленко  
Максим Володимирович, гр. КН-22Мз, ЦНТУ, Кропивницький 2023</value>
```

```
</data>
```

```
</root>
```

КБПЗ_2023

Файл frmAbout.h - бібліотека для файлу frmAbout.resx

```

#pragma once

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

namespace UsrsManager
{
    /// <summary>
    /// Summary for frmAbout
    ///
    /// </summary>
    public ref class frmAbout : public System::Windows::Forms::Form
    {
    /*****
    public:    frmAbout(void)
            {
                InitializeComponent();
                //
                //Додавання цього коду в конструктор
                //
            }
    /*****
    protected: ~frmAbout()
            {
                if (components)
                {
                    delete components;
                }
            }
    /*****
    private: System::Windows::Forms::Button^  btnOk;
    private: System::Windows::Forms::TextBox^  txtInfo;

    private: System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
        /// <summary>
        /// Необхідний метод для підтримки Розроблювача - не модифікувати
        /// зміст цього методу з кодовим редактором.
        /// </summary>
        void InitializeComponent(void)
        {
            System::ComponentModel::ComponentResourceManager^  resources =
(gcnew System::ComponentModel::ComponentResourceManager(frmAbout::typeid));
            this->btnOk = (gcnew System::Windows::Forms::Button());
            this->txtInfo = (gcnew System::Windows::Forms::TextBox());
            this->SuspendLayout();
            //
            // btnOk
            //
            this->btnOk->Location = System::Drawing::Point(205, 216);
            this->btnOk->Name = L"btnOk";
            this->btnOk->Size = System::Drawing::Size(75, 23);
            this->btnOk->TabIndex = 0;
            this->btnOk->Text = L"OK";
            this->btnOk->UseVisualStyleBackColor = true;
            this->btnOk->Click += gcnew System::EventHandler(this,
&frmAbout::btnOk_Click);
            //

```

```

        // txtInfo
        //
        this->txtInfo->Location = System::Drawing::Point(7, 12);
        this->txtInfo->Multiline = true;
        this->txtInfo->Name = L"txtInfo";
        this->txtInfo->ReadOnly = true;
        this->txtInfo->Size = System::Drawing::Size(273, 198);
        this->txtInfo->TabIndex = 3;
        this->txtInfo->Text = resources->GetString(L"txtInfo.Text");
        this->txtInfo->TextAlign =
System::Windows::Forms::HorizontalAlignment::Center;
        //
        // frmAbout
        //
        this->AutoScaleDimensions = System::Drawing::Size(6, 13);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(287, 243);
        this->Controls->Add(this->btnOk);
        this->Controls->Add(this->txtInfo);
        this->MaximizeBox = false;
        this->MinimizeBox = false;
        this->Name = L"frmAbout";
        this->Text = L"Про програму...";
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
/*****/
private: System::Void btnOk_Click(System::Object^ sender, System::EventArgs^
e)
    {
        // Закрити форму
        frmAbout::Close();
    }
/*****/
};
}

```

Файл Worker.cpp - вікно адміністратора інформаційної системи підприємства

```

/*****/
#include "StdAfx.h"
#include "Worker.h"
#include "Doc.h"
#include "Logger.h"
/*****/
using namespace System;
using namespace System::IO;
using namespace System::Collections;
/*****/
#define DOCS_FILE "DATA.LMB"
#define LOG_INPUT "INPUT.LML"
#define LOG_OUTPUT "OUTPUT.LML"
/*****/
// NAME: CWorker
// DESCRIPTION: Конструктор класу
// INPUT: N/D
/*****/
CWorker::CWorker(void)
{
    listDoc = gcnew ArrayList();
    logInput = gcnew CLogger();
    logOutput = gcnew CLogger();
}
/*****/
// NAME: GetIndexByISBNHash
// DESCRIPTION: Функція одержання індексу документа в загальному списку по
вхідному номеру документа
// INPUT: ddHashValue - значення хеша вхідного номера документа
/*****/
Int32 CWorker::GetIndexByISBNHash(Int32 HashValue)
{
    // Цикл пошуку документа із заданим вхідним номером документа
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Хеш-значення вхідного номера документа заданого документа збігається з
        // хеш- значенням вхідного номера документа знайденого документа?
        if(HashValue == ((CDoc^)listDoc[i])->GetISBNHash())
        {
            // Документ знайдений, повернути індекс
            return i;
        }
    }

    // Документ не знайдений, повернути -1
    return -1;
}
/*****/
// NAME: AddDoc
// DESCRIPTION: Функція додавання нового документа
// INPUT: SourceDoc - об'єкт "документ" для включення в список
// OUTPUT: TRUE - документ успішно доданий
// FALSE - такий документ уже існує в системі
/*****/
Boolean CWorker::AddDoc(CDoc^ SourceDoc)
{
    // Документи з таким вхідним номером документа не існує?
    if(this->GetIndexByISBNHash(SourceDoc->GetISBNHash()) == -1)
    {
        // Додати заданній документ у загальний список
        listDoc->Add(SourceDoc);
        // Записати подія в лог
        logInput->WriteEvent(
            "Новий документ '" + ((CDoc^)SourceDoc)->GetName() +
            "', Автор '" + ((CDoc^)SourceDoc)->GetAuthor() +

```

```

        "", Вхідний номер документа " + ((CDoc^)SourceDoc)->GetISBN());
// Функція відробила успішно
return true;
}
else
{
    // Документ із таким же вхідним номером документа існує, повернути помилку
    return false;
}
}
}
/*****/
// NAME:          RemoveDoc
// DESCRIPTION:   Функція видалення документів із системи
// INPUT:         ddISBNHash - значення хеша вхідного номера документа
// OUTPUT:        0 - видалення зроблене
//               1 - видалення неможливо, не всі екземпляри перебувають у
системі
//               2 - документ із заданим вхідним номером документа не
знайдений
/*****/
Int32 CWorker::RemoveDoc(Int32 ddISBNHash)
{
    // Знайти документ по вхідному номеру документа
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);

    // Документ із таким вхідним номером документа існує?
    if(ddIndex != -1)
    {
        // Перевірити, що жодного документа немає в користувача інформаційної
системи підприємства
        if(((CDoc^)listDoc[ddIndex])->GetTotalNumber() ==
            ((CDoc^)listDoc[ddIndex])->GetFreeNumber())
        {
            // Записати подія в лог
            logOutput->WriteEvent(
                "Виконаний документ '" + ((CDoc^)listDoc[ddIndex])->GetName()
+
                "' , Автор '" + ((CDoc^)listDoc[ddIndex])->GetAuthor() +
                "' , Вхідний номер документа " + ((CDoc^)listDoc[ddIndex])-
>GetISBN());
            // Видалити знайдений документ зі списку
            listDoc->RemoveAt(ddIndex);
            // Функція відробила успішно
            return 0;
        }
        else
        {
            // Повернути помилку, не всі документи перебувають у системі
            return 1;
        }
    }
    else
    {
        // Повернути помилку, документ із таким вхідним номером документа не був
знайдений
        return 2;
    }
}
/*****/
// NAME:          GiveDoc
// DESCRIPTION:   Функція видачі документів користувачеві
// INPUT:         listDoc - список документів користувача інформаційної системи
підприємства (для виконання додавання)
//               ddISBNHash - хеш вхідного номера документа, що потрібно
одержати
// OUTPUT:        TRUE - операція пройшла успішно
//               FALSE - у наявності немає жодного екземпляра заданого
документа

```

```

/*****/
Boolean CWorker::GiveDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);

    // Зменшити кількість вільних документів
    if((ddIndex != -1) && ((CDoc^)listDoc[ddIndex])->DecFreeNumber())
    {
        // Додати документ у список користувача інформаційної системи підприємства
        listReader->Add(listDoc[ddIndex]);
        // Функція відобразила успішно
        return true;
    }
    else
    {
        // Такого документа не є в наявності
        return false;
    }
}
/*****/
// NAME:         TakeDoc
// DESCRIPTION:   Функція прийняття документів від користувача інформаційної
системи підприємства назад у систему
// INPUT:        // listDoc - список документів користувача інформаційної
системи підприємства (для виконання видалення)
//               ddISBNHash - хеш вхідного номера документа, що потрібно
повернути
/*****/
void CWorker::TakeDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа (документ існує, тому що його
вхідний номер документа був повідомлений користувачеві)
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);

    // Видалити документ зі списку користувача інформаційної системи підприємства
    // listReader->RemoveAt(ddIndex);

    // Збільшити кількість вільних документів
    ((CDoc^)listDoc[ddIndex])->IncFreeNumber();
}
/*****/
// NAME:         LoadDocList
// DESCRIPTION:   Функція завантаження документів системи з файлу
// INPUT:        N/D
/*****/
void CWorker::LoadDocList()
{
    // Перевірити існування файлу
    if(!File::Exists(DOCS_FILE))
    {
        // Файл не знайдений, закінчити виконання функції
        return;
    }

    // Відкрити файл
    StreamReader^ srDoc = gcnew StreamReader(DOCS_FILE);

    // Продовжувати, поки не буде знайдений кінець файлу
    while(srDoc->EndOfStream == false)
    {
        // Створити новий об'єкт "документ"
        CDoc^ NewDoc = gcnew CDoc();

        // Завантажити з файлу й установити параметри документа
        NewDoc->SetName(srDoc->ReadLine());
        NewDoc->SetAuthor(srDoc->ReadLine());
        NewDoc->SetISBN(srDoc->ReadLine());
        NewDoc->SetTheme(srDoc->ReadLine());
    }
}

```

```

NewDoc->SetPages (Convert::ToInt32 (srDoc->ReadLine ()));
NewDoc->SetTotalNumber (Convert::ToInt32 (srDoc->ReadLine ()));
NewDoc->SetFreeNumber (Convert::ToInt32 (srDoc->ReadLine ()));

// Включити документ у загальний список документів
listDoc->Add (NewDoc);
}

// Закрити файл
srDoc->Close ();
}
/*****
// NAME:          SaveDocList
// DESCRIPTION:   Функція збереження документів системи у файл
// INPUT:         N/D
*****/
void CWorker::SaveDocList ()
{
// Перевірити існування файлу
if (File::Exists (DOCS_FILE))
{
// Видалити файл
File::Delete (DOCS_FILE);
}

// Створити й відкрити файл
StreamWriter^ swDoc = gcnew StreamWriter (DOCS_FILE);

// Цикл запису по одному документу
for (Int32 i = 0; i < listDoc->Count; i++)
{
// Одержати параметри документа й записати їх у файл
swDoc->WriteLine (((CDoc^) listDoc[i])->GetName ());
swDoc->WriteLine (((CDoc^) listDoc[i])->GetAuthor ());
swDoc->WriteLine (((CDoc^) listDoc[i])->GetISBN ());
swDoc->WriteLine (((CDoc^) listDoc[i])->GetTheme ());
swDoc->WriteLine (Convert::ToString (((CDoc^) listDoc[i])->GetPages ());
swDoc->WriteLine (Convert::ToString (((CDoc^) listDoc[i])->GetTotalNumber ());
swDoc->WriteLine (Convert::ToString (((CDoc^) listDoc[i])->GetFreeNumber ());
}

// Закрити файл
swDoc->Close ();
}
/*****
// NAME:          FindDoc
// DESCRIPTION:   Функція пошуку документа по заданих параметрах
// INPUT:         strFindDoc - рядок для пошуку
*****/
ArrayList^ CWorker::FindDoc (String^ strFindValue)
{
// Список для результату
ArrayList^ listResult = gcnew ArrayList ();

// Шукане значення без обліку регістра
String^ strValue = strFindValue->ToLower ();

// Задана порожній рядок?
if (String::IsNullOrEmpty (strValue->Trim ()))
{
//Вивести весь список
for (Int32 i = 0; i < listDoc->Count; i++)
{
// Додати документ у результуючий список
listResult->Add (listDoc[i]);
}
}
else

```

```

{
    // Вибрати з умовою
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Пошук рядка в кожному полі без обліку регістра
        if((((CDoc^)listDoc[i])->GetName()->ToLower()->IndexOf(strValue)
!= -1 ||
            (((CDoc^)listDoc[i])->GetAuthor()->ToLower()-
>IndexOf(strValue) != -1 ||
            (((CDoc^)listDoc[i])->GetISBN()->ToLower()->IndexOf(strValue)
!= -1 ||
            (((CDoc^)listDoc[i])->GetTheme()->ToLower()->IndexOf(strValue)
!= -1 ||
            Convert::ToString((((CDoc^)listDoc[i])->GetPages()))-
>IndexOf(strValue) != -1)
        {
            // Додати документ у результуючий список
            listResult->Add(listDoc[i]);
        }
    }
}

// Повернути список збігів
return listResult;
}
/*****/
// NAME:          ViewDoc
// DESCRIPTION:   Функція перегляду інформації про заданий документ
// INPUT:         ddISBNHash - хеш вхідного номера документа, інформацію про яку
треба переглянути
/*****/
CDoc^ CWorker::ViewDoc(Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа (документ існує, тому що його
вхідний номер документа був повідомлений користувачеві)
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);

    // Повернути елемент "документ" із загальне списку
    return (CDoc^)listDoc[ddIndex];
}
/*****/
// NAME:          ReadLogs
// DESCRIPTION:   Функція зчитування історії з файлу
// INPUT:         N/D
/*****/
void CWorker::ReadLogs()
{
    // Файл із історією надходжень існує?
    if(File::Exists(LOG_INPUT))
    {
        // Зчитування історії надходжень
        logInput->strLog = File::ReadAllText(LOG_INPUT);
    }
    else
    {
        logInput->strLog = "";
    }

    // Файл із історією списань існує?
    if(File::Exists(LOG_OUTPUT))
    {
        // Зчитування історії списань
        logOutput->strLog = File::ReadAllText(LOG_OUTPUT);
    }
    else
    {
        logOutput->strLog = "";
    }
}
}

```

```

/*****/
// NAME:          WriteLogs
// DESCRIPTION:   Функція збереження історії у файл
// INPUT:         N/D
/*****/
void CWorker::WriteLogs()
{
    // Запис історії
    File::WriteAllText(LOG_INPUT, logInput->strLog);
    File::WriteAllText(LOG_OUTPUT, logOutput->strLog);
}
/*****/
// NAME:          ClearLogs
// DESCRIPTION:   Функція очищення історії
// INPUT:         N/D
/*****/
void CWorker::ClearLogs()
{
    logInput->strLog = "";
    logOutput->strLog = "";
}
/*****/
// NAME:          ViewInputLog
// DESCRIPTION:   Функція перегляду історії надходжень
// INPUT:         N/D
/*****/
String^ CWorker::ViewInputLog()
{
    return logInput->strLog;
}
/*****/
// NAME:          ViewOutputLog
// DESCRIPTION:   Функція перегляду історії списань
// INPUT:         N/D
/*****/
String^ CWorker::ViewOutputLog()
{
    return logOutput->strLog;
}
/*****/

```

Файл Worker.resx - xml опис вікна адміністратора інформаційної системи підприємства

```

<?xml version="1.0" encoding=" utf-8"?>
<root>
  <xsd:schema id="root" xmlns="" xmlns:xsd=" " xmlns:msdata="urn: schemas-
microsoft-com: xml-msdata">
    <xsd:import namespace=" " />
    <xsd:element name="root" msdata:IsDataSet="true">
      <xsd:complexType>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element name="metadata">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0" />
              </xsd:sequence>
              <xsd:attribute name="name" use="required" type="xsd:string" />
              <xsd:attribute name="type" type="xsd:string" />
              <xsd:attribute name="mimetype" type="xsd:string" />
              <xsd:attribute ref="xml:space" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="assembly">
            <xsd:complexType>
              <xsd:attribute name="alias" type="xsd:string" />
              <xsd:attribute name="name" type="xsd:string" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="data">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
                <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" use="required"
msdata:Ordinal="1" />
              <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
              <xsd:attribute name="mimetype" type="xsd:string"
msdata:Ordinal="4" />
              <xsd:attribute ref="xml:space" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="resheader">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" use="required" />
            </xsd:complexType>
          </xsd:element>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
  <resheader name="resmimetype">
    <value>text/ microsoft-resx</value>
  </resheader>
  <resheader name="version">
    <value>2.0</value>
  </resheader>
  <resheader name="reader">
    <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
  <resheader name="writer">

```

```
<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<metadata name="menuStrip.TrayLocation" type="System.Drawing.Point,
System.Drawing, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a">
  <value>230, 17</value>
</metadata>
<metadata name="statusStrip.TrayLocation" type="System.Drawing.Point,
System.Drawing, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a">
  <value>332, 17</value>
</metadata>
<metadata name="groupInfo.Locked" type="System.Boolean, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
<metadata name="groupInfo.Locked" type="System.Boolean, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089">
  <value>True</value>
</metadata>
</root>
```

K6П3_2023

Файл Worker.h - бібліотека для файлу frmAbout.cpp

```

#pragma once
/*****
#include "Worker.h"
#include "Doc.h"
#include "Counter.h"
#include "frmAbout.h"
/*****
#define PAGES_LESS           "менше"
#define PAGES_GREATER       "більше"
#define PAGES_EQUAL         "дорівнює"
/*****
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
/*****
namespace UsrsManager
{
    public ref class frmWorker : public System::Windows::Forms::Form
    {
/*****
private:
    CWorker^ Worker;           // Об'єкт "Адміністратор"

    Boolean IsCreateClicked;    // Прапор натискання по кнопці "Створити"
    Boolean IsEditClicked;     // Прапор натискання по кнопці "Редагувати"

    ArrayList^ listView;      // Поточний список документів для
відображення

    Int32 ddCurrentIndex;     // Індекс поточного відображуваного елемента
/*****
private: System::Windows::Forms::Button^ btnClearLogs;
/*****
public:    frmWorker(void)
    {
        // Ініціалізація компонентів форми
        InitializeComponent();

        // Кнопки "Створити" і "Редагувати" не минулого натиснуті жодного
разу
        IsCreateClicked = false;
        IsEditClicked = false;

        // Створити список відображуваних документів
        listView = gcnew ArrayList();

        // Поточний індекс документа для відображення не визначений
        ddCurrentIndex = -1;
    }
/*****
protected: ~frmWorker()
    {
        if (components)
        {
            delete components;
        }
    }
/*****
private: System::Windows::Forms::ListBox^ lstPagesDocList;
private: System::Windows::Forms::NumericUpDown^ nmrPagesNumber;
private: System::Windows::Forms::ComboBox^ cmbPagesDirect;
private: System::Windows::Forms::ListBox^ lstThemeDocList;

```

```

private: System::Windows::Forms::ListBox^ lstAuthorDocList;
private: System::Windows::Forms::CheckBox^ chkPagesFlag;
private: System::Windows::Forms::MenuStrip^ menuStrip;
private: System::Windows::Forms::ToolStripMenuItem^ menuFile;
private: System::Windows::Forms::ToolStripMenuItem^ menuFileSave;
private: System::Windows::Forms::ToolStripMenuItem^ menuFileExit;
private: System::Windows::Forms::ToolStripMenuItem^ menuHelp;

private: System::Windows::Forms::ToolStripMenuItem^ menuHelpAbout;
private: System::Windows::Forms::GroupBox^ groupBox2;
private: System::Windows::Forms::TextBox^ txtThemeFilter;
private: System::Windows::Forms::TextBox^ txtAuthorFilter;
private: System::Windows::Forms::Label^ lblFreeInstanceNumber;
private: System::Windows::Forms::Label^ lblTotalInstanceNumber;
private: System::Windows::Forms::CheckBox^ chkThemeDocFlag;
private: System::Windows::Forms::CheckBox^ chkAuthorDocFlag;
private: System::Windows::Forms::CheckBox^ chkFreeInstanceFlag;
private: System::Windows::Forms::CheckBox^ chkTotalInstanceFlag;
private: System::Windows::Forms::NumericUpDown^ nmrFreeNumber;
private: System::Windows::Forms::NumericUpDown^ nmrTotalNumber;
private: System::Windows::Forms::TabPage^ tabHistory;
private: System::Windows::Forms::GroupBox^ groupOutput;
private: System::Windows::Forms::TextBox^ txtOutputHistory;
private: System::Windows::Forms::GroupBox^ groupInput;
private: System::Windows::Forms::TextBox^ txtInputHistory;
private: System::Windows::Forms::NumericUpDown^ nmrPageNumber;
private: System::Windows::Forms::TextBox^ txtTheme;
private: System::Windows::Forms::TextBox^ txtISBN;
private: System::Windows::Forms::TextBox^ txtAuthor;
private: System::Windows::Forms::TextBox^ txtDocName;
private: System::Windows::Forms::TabPage^ tabReports;
private: System::Windows::Forms::Label^ lblFreeNumber;
private: System::Windows::Forms::Label^ lblTotalNumber;
private: System::Windows::Forms::Label^ lblPagesNumber;
private: System::Windows::Forms::Label^ lblTheme;
private: System::Windows::Forms::StatusStrip^ statusStrip;
private: System::Windows::Forms::ToolStripStatusLabel^ lblStatus;
private: System::Windows::Forms::TextBox^ txtFindString;
private: System::Windows::Forms::TabControl^ tabControl;
private: System::Windows::Forms::TabPage^ tabDocs;
private: System::Windows::Forms::Button^ btnEdit;
private: System::Windows::Forms::Button^ btnDelete;
private: System::Windows::Forms::Button^ btnCreate;
private: System::Windows::Forms::GroupBox^ groupInfo;
private: System::Windows::Forms::Label^ lblISBN;
private: System::Windows::Forms::Label^ lblAuthor;
private: System::Windows::Forms::Label^ lblDocName;
private: System::Windows::Forms::GroupBox^ groupFinding;
private: System::Windows::Forms::Button^ btnFind;
private: System::Windows::Forms::Button^ btnFirst;
private: System::Windows::Forms::Button^ btnPrev;
private: System::Windows::Forms::Button^ btnNext;
private: System::Windows::Forms::Button^ btnLast;
private: System::Windows::Forms::TextBox^ txtPosition;
private: System::ComponentModel::Container ^components;
/*****
#pragma region Windows Form Designer generated code
    /// <summary>
    /// Необхідний метод для підтримки Розроблювача - не модифікувати
    /// зміст цього методу з кодовим редактором.
    /// </summary>
    void InitializeComponent(void)
    {
        this->lstPagesDocList = (gcnew System::Windows::Forms::ListBox());
        this->nmrPagesNumber = (gcnew
System::Windows::Forms::NumericUpDown());
        this->cmbPagesDirect = (gcnew System::Windows::Forms::ComboBox());
        this->lstThemeDocList = (gcnew System::Windows::Forms::ListBox());

```

```

        this->lstAuthorDocList = (gcnew System::Windows::Forms::ListBox());
        this->chkPagesFlag = (gcnew System::Windows::Forms::CheckBox());
        this->menuStrip = (gcnew System::Windows::Forms::MenuStrip());
        this->menuFile = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuFileSave = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuFileExit = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuHelp = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuHelpAbout = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());
        this->txtThemeFilter = (gcnew System::Windows::Forms::TextBox());
        this->txtAuthorFilter = (gcnew System::Windows::Forms::TextBox());
        this->lblFreeInstanceNumber = (gcnew
System::Windows::Forms::Label());
        this->lblTotalInstanceNumber = (gcnew
System::Windows::Forms::Label());
        this->chkThemeDocFlag = (gcnew System::Windows::Forms::CheckBox());
        this->chkAuthorDocFlag = (gcnew System::Windows::Forms::CheckBox());
        this->chkFreeInstanceFlag = (gcnew
System::Windows::Forms::CheckBox());
        this->chkTotalInstaceFlag = (gcnew
System::Windows::Forms::CheckBox());
        this->nmrFreeNumber = (gcnew
System::Windows::Forms::NumericUpDown());
        this->btnRefresh = (gcnew System::Windows::Forms::Button());
        this->nmrTotalNumber = (gcnew
System::Windows::Forms::NumericUpDown());
        this->tabHistory = (gcnew System::Windows::Forms::TabPage());
        this->btnClearLogs = (gcnew System::Windows::Forms::Button());
        this->groupOutput = (gcnew System::Windows::Forms::GroupBox());
        this->txtOutputHistory = (gcnew System::Windows::Forms::TextBox());
        this->groupInput = (gcnew System::Windows::Forms::GroupBox());
        this->txtInputHistory = (gcnew System::Windows::Forms::TextBox());
        this->nmrPageNumber = (gcnew
System::Windows::Forms::NumericUpDown());
        this->txtTheme = (gcnew System::Windows::Forms::TextBox());
        this->txtISBN = (gcnew System::Windows::Forms::TextBox());
        this->txtAuthor = (gcnew System::Windows::Forms::TextBox());
        this->txtDocName = (gcnew System::Windows::Forms::TextBox());
        this->tabReports = (gcnew System::Windows::Forms::TabPage());
        this->lblFreeNumber = (gcnew System::Windows::Forms::Label());
        this->lblTotalNumber = (gcnew System::Windows::Forms::Label());
        this->lblPagesNumber = (gcnew System::Windows::Forms::Label());
        this->lblTheme = (gcnew System::Windows::Forms::Label());
        this->statusStrip = (gcnew System::Windows::Forms::StatusStrip());
        this->lblStatus = (gcnew
System::Windows::Forms::ToolStripStatusLabel());
        this->txtFindString = (gcnew System::Windows::Forms::TextBox());
        this->tabControl = (gcnew System::Windows::Forms::TabControl());
        this->tabDocs = (gcnew System::Windows::Forms::TabPage());
        this->btnEdit = (gcnew System::Windows::Forms::Button());
        this->btnDelete = (gcnew System::Windows::Forms::Button());
        this->btnCreate = (gcnew System::Windows::Forms::Button());
        this->groupInfo = (gcnew System::Windows::Forms::GroupBox());
        this->txtPosition = (gcnew System::Windows::Forms::TextBox());
        this->btnFirst = (gcnew System::Windows::Forms::Button());
        this->btnPrev = (gcnew System::Windows::Forms::Button());
        this->btnNext = (gcnew System::Windows::Forms::Button());
        this->btnLast = (gcnew System::Windows::Forms::Button());
        this->lblISBN = (gcnew System::Windows::Forms::Label());
        this->lblAuthor = (gcnew System::Windows::Forms::Label());
        this->lblDocName = (gcnew System::Windows::Forms::Label());
        this->groupFinding = (gcnew System::Windows::Forms::GroupBox());
        this->btnFind = (gcnew System::Windows::Forms::Button());

```

```

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrPagesNumber))->BeginInit();
        this->menuStrip->SuspendLayout();
        this->groupBox 2-2->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrFreeNumber))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrTotalNumber))->BeginInit();
        this->tabHistory->SuspendLayout();
        this->groupOutput->SuspendLayout();
        this->groupInput->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrPageNumber))->BeginInit();
        this->tabReports->SuspendLayout();
        this->statusStrip->SuspendLayout();
        this->tabControl->SuspendLayout();
        this->tabDocs->SuspendLayout();
        this->groupInfo->SuspendLayout();
        this->groupFinding->SuspendLayout();
        this->SuspendLayout();
        //
        // lstPagesDocList
        //
        this->lstPagesDocList->FormattingEnabled = true;
        this->lstPagesDocList->HorizontalScrollbar = true;
        this->lstPagesDocList->Location = System::Drawing::Point(478, 117);
        this->lstPagesDocList->Name = L"lstPagesDocList";
        this->lstPagesDocList->Size = System::Drawing::Size(223, 199);
        this->lstPagesDocList->TabIndex = 13;
        //
        // nmrPagesNumber
        //
        this->nmrPagesNumber->Location = System::Drawing::Point(583, 89);
        this->nmrPagesNumber->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) {65535, 0, 0, 0});
        this->nmrPagesNumber->Name = L"nmrPagesNumber";
        this->nmrPagesNumber->Size = System::Drawing::Size(64, 20);
        this->nmrPagesNumber->TabIndex = 6;
        //
        // cmbPagesDirect
        //
        this->cmbPagesDirect->FormattingEnabled = true;
        this->cmbPagesDirect->Location = System::Drawing::Point(479, 89);
        this->cmbPagesDirect->Name = L"cmbPagesDirect";
        this->cmbPagesDirect->Size = System::Drawing::Size(89, 21);
        this->cmbPagesDirect->TabIndex = 5;
        //
        // lstThemeDocList
        //
        this->lstThemeDocList->FormattingEnabled = true;
        this->lstThemeDocList->HorizontalScrollbar = true;
        this->lstThemeDocList->Location = System::Drawing::Point(242, 117);
        this->lstThemeDocList->Name = L"lstThemeDocList";
        this->lstThemeDocList->Size = System::Drawing::Size(223, 199);
        this->lstThemeDocList->TabIndex = 9;
        //
        // lstAuthorDocList
        //
        this->lstAuthorDocList->FormattingEnabled = true;
        this->lstAuthorDocList->HorizontalScrollbar = true;
        this->lstAuthorDocList->Location = System::Drawing::Point(6, 117);
        this->lstAuthorDocList->Name = L"lstAuthorDocList";
        this->lstAuthorDocList->Size = System::Drawing::Size(223, 199);
        this->lstAuthorDocList->TabIndex = 8;
        //
        // chkPagesFlag
        //
        this->chkPagesFlag->AutoSize = true;
        this->chkPagesFlag->Location = System::Drawing::Point(479, 68);

```

```

this->chkPagesFlag->Name = L"chkPagesFlag";
this->chkPagesFlag->Size = System::Drawing::Size(75, 17);
this->chkPagesFlag->TabIndex = 4;
this->chkPagesFlag->Text = L"Страницы:";
this->chkPagesFlag->UseVisualStyleBackColor = true;
//
// menuStrip
//
this->menuStrip->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(2) {this->menuFile, this->menuHelp});
this->menuStrip->Location = System::Drawing::Point(0, 0);
this->menuStrip->Name = L"menuStrip";
this->menuStrip->Size = System::Drawing::Size(722, 24);
this->menuStrip->TabIndex = 9;
this->menuStrip->Text = L"menuStrip1";
//
// menuFile
//
this->menuFile->DropDownItems->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(2) {this->menuFileSave,
this->menuFileExit});
this->menuFile->Name = L"menuFile";
this->menuFile->Size = System::Drawing::Size(45, 20);
this->menuFile->Text = L"Файл";
//
// menuFileSave
//
this->menuFileSave->Name = L"menuFileSave";
this->menuFileSave->Size = System::Drawing::Size(152, 22);
this->menuFileSave->Text = L"Зберегти";
this->menuFileSave->Click += gcnew System::EventHandler(this,
&frmWorker::menuFileSave_Click);
//
// menuFileExit
//
this->menuFileExit->Name = L"menuFileExit";
this->menuFileExit->Size = System::Drawing::Size(152, 22);
this->menuFileExit->Text = L"Вихід";
this->menuFileExit->Click += gcnew System::EventHandler(this,
&frmWorker::menuFileExit_Click);
//
// menuHelp
//
this->menuHelp->DropDownItems->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) {this->menuHelpAbout});
this->menuHelp->Name = L"menuHelp";
this->menuHelp->Size = System::Drawing::Size(60, 20);
this->menuHelp->Text = L"Довідка";
//
// menuHelpAbout
//
this->menuHelpAbout->Name = L"menuHelpAbout";
this->menuHelpAbout->Size = System::Drawing::Size(166, 22);
this->menuHelpAbout->Text = L"Про програму...";
this->menuHelpAbout->Click += gcnew System::EventHandler(this,
&frmWorker::menuHelpAbout_Click);
//
// groupBox2
//
this->groupBox 2-2->Controls->Add(this->lstPagesDocList);
this->groupBox 2-2->Controls->Add(this->nmrPagesNumber);
this->groupBox 2-2->Controls->Add(this->cmbPagesDirect);
this->groupBox 2-2->Controls->Add(this->chkPagesFlag);
this->groupBox 2-2->Controls->Add(this->lstThemeDocList);
this->groupBox 2-2->Controls->Add(this->lstAuthorDocList);
this->groupBox 2-2->Controls->Add(this->txtThemeFilter);
this->groupBox 2-2->Controls->Add(this->txtAuthorFilter);
this->groupBox 2-2->Controls->Add(this->lblFreeInstanceNumber);
this->groupBox 2-2->Controls->Add(this->lblTotalInstanceNumber);

```

```

this->groupBox 2-2->Controls->Add(this->chkThemeDocFlag);
this->groupBox 2-2->Controls->Add(this->chkAuthorDocFlag);
this->groupBox 2-2->Controls->Add(this->chkFreeInstanceFlag);
this->groupBox 2-2->Controls->Add(this->chkTotalInstaceFlag);
this->groupBox 2-2->Location = System::Drawing::Point(8, 6);
this->groupBox 2-2->Name = L"groupBox2";
this->groupBox 2-2->Size = System::Drawing::Size(712, 325);
this->groupBox 2-2->TabIndex = 0;
this->groupBox 2-2->TabStop = false;
//
// txtThemeFilter
//
this->txtThemeFilter->Location = System::Drawing::Point(242, 90);
this->txtThemeFilter->Name = L"txtThemeFilter";
this->txtThemeFilter->Size = System::Drawing::Size(223, 20);
this->txtThemeFilter->TabIndex = 7;
//
// txtAuthorFilter
//
this->txtAuthorFilter->Location = System::Drawing::Point(6, 90);
this->txtAuthorFilter->Name = L"txtAuthorFilter";
this->txtAuthorFilter->Size = System::Drawing::Size(223, 20);
this->txtAuthorFilter->TabIndex = 6;
//
// lblFreeInstanceNumber
//
this->lblFreeInstanceNumber->AutoSize = true;
this->lblFreeInstanceNumber->Location = System::Drawing::Point(476,
42);

this->lblFreeInstanceNumber->Name = L"lblFreeInstanceNumber";
this->lblFreeInstanceNumber->Size = System::Drawing::Size(26, 13);
this->lblFreeInstanceNumber->TabIndex = 5;
this->lblFreeInstanceNumber->Text = L"Н/Д";
//
// lblTotalInstanceNumber
//
this->lblTotalInstanceNumber->AutoSize = true;
this->lblTotalInstanceNumber->Location = System::Drawing::Point(476,
19);

this->lblTotalInstanceNumber->Name = L"lblTotalInstanceNumber";
this->lblTotalInstanceNumber->Size = System::Drawing::Size(26, 13);
this->lblTotalInstanceNumber->TabIndex = 4;
this->lblTotalInstanceNumber->Text = L"Н/Д";
//
// chkThemeDocFlag
//
this->chkThemeDocFlag->AutoSize = true;
this->chkThemeDocFlag->Location = System::Drawing::Point(240, 67);
this->chkThemeDocFlag->Name = L"chkThemeDocFlag";
this->chkThemeDocFlag->Size = System::Drawing::Size(176, 17);
this->chkThemeDocFlag->TabIndex = 3;
this->chkThemeDocFlag->Text = L"Список документів за темою:";
this->chkThemeDocFlag->UseVisualStyleBackColor = true;
//
// chkAuthorDocFlag
//
this->chkAuthorDocFlag->AutoSize = true;
this->chkAuthorDocFlag->Location = System::Drawing::Point(6, 67);
this->chkAuthorDocFlag->Name = L"chkAuthorDocFlag";
this->chkAuthorDocFlag->Size = System::Drawing::Size(165, 17);
this->chkAuthorDocFlag->TabIndex = 2;
this->chkAuthorDocFlag->Text = L"Список документів автора:";
this->chkAuthorDocFlag->UseVisualStyleBackColor = true;
//
// chkFreeInstanceFlag
//
this->chkFreeInstanceFlag->AutoSize = true;
this->chkFreeInstanceFlag->Location = System::Drawing::Point(6, 42);
this->chkFreeInstanceFlag->Name = L"chkFreeInstanceFlag";

```

```

this->chkFreeInstanceFlag->Size = System::Drawing::Size(208, 17);
this->chkFreeInstanceFlag->TabIndex = 1;
this->chkFreeInstanceFlag->Text = L"Кількість екземплярів на
виконання";
this->chkFreeInstanceFlag->UseVisualStyleBackColor = true;
//
// chkTotalInstaceFlag
//
this->chkTotalInstaceFlag->AutoSize = true;
this->chkTotalInstaceFlag->Location = System::Drawing::Point(6, 19);
this->chkTotalInstaceFlag->Name = L"chkTotalInstaceFlag";
this->chkTotalInstaceFlag->Size = System::Drawing::Size(293, 17);
this->chkTotalInstaceFlag->TabIndex = 0;
this->chkTotalInstaceFlag->Text = L"Загальна кількість екземплярів у
відібраному списку";
this->chkTotalInstaceFlag->UseVisualStyleBackColor = true;
//
// nmrFreeNumber
//
this->nmrFreeNumber->Location = System::Drawing::Point(372, 196);
this->nmrFreeNumber->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) {65535, 0, 0, 0});
this->nmrFreeNumber->Name = L"nmrFreeNumber";
this->nmrFreeNumber->ReadOnly = true;
this->nmrFreeNumber->Size = System::Drawing::Size(80, 20);
this->nmrFreeNumber->TabIndex = 8;
//
// btnRefresh
//
this->btnRefresh->Location = System::Drawing::Point(611, 337);
this->btnRefresh->Name = L"btnRefresh";
this->btnRefresh->Size = System::Drawing::Size(99, 23);
this->btnRefresh->TabIndex = 7;
this->btnRefresh->Text = L"Зберегти";
this->btnRefresh->UseVisualStyleBackColor = true;
this->btnRefresh->Click += gcnew System::EventHandler(this,
&frmWorker::btnRefresh_Click);
//
// nmrTotalNumber
//
this->nmrTotalNumber->Location = System::Drawing::Point(139, 196);
this->nmrTotalNumber->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) {65535, 0, 0, 0});
this->nmrTotalNumber->Name = L"nmrTotalNumber";
this->nmrTotalNumber->Size = System::Drawing::Size(80, 20);
this->nmrTotalNumber->TabIndex = 7;
this->nmrTotalNumber->ValueChanged += gcnew
System::EventHandler(this, &frmWorker::nmrTotalNumber_ValueChanged);
//
// tabHistory
//
this->tabHistory->Controls->Add(this->btnClearLogs);
this->tabHistory->Controls->Add(this->groupOutput);
this->tabHistory->Controls->Add(this->groupInput);
this->tabHistory->Location = System::Drawing::Point(4, 22);
this->tabHistory->Name = L"tabHistory";
this->tabHistory->Size = System::Drawing::Size(717, 366);
this->tabHistory->TabIndex = 2;
this->tabHistory->Text = L"Історія";
this->tabHistory->UseVisualStyleBackColor = true;
//
// btnClearLogs
//
this->btnClearLogs->Location = System::Drawing::Point(631, 339);
this->btnClearLogs->Name = L"btnClearLogs";
this->btnClearLogs->Size = System::Drawing::Size(75, 23);
this->btnClearLogs->TabIndex = 0;
this->btnClearLogs->Text = L"Очистити";
this->btnClearLogs->UseVisualStyleBackColor = true;

```

```

        this->btnClearLogs->Click += gcnew System::EventHandler(this,
&frmWorker::btnClearLogs_Click);
        //
        // groupOutput
        //
        this->groupOutput->Controls->Add(this->txtOutputHistory);
        this->groupOutput->Location = System::Drawing::Point(6, 166);
        this->groupOutput->Name = L"groupOutput";
        this->groupOutput->Size = System::Drawing::Size(700, 167);
        this->groupOutput->TabIndex = 1;
        this->groupOutput->TabStop = false;
        this->groupOutput->Text = L"Історія виконань";
        //
        // txtOutputHistory
        //
        this->txtOutputHistory->Location = System::Drawing::Point(6, 19);
        this->txtOutputHistory->Multiline = true;
        this->txtOutputHistory->Name = L"txtOutputHistory";
        this->txtOutputHistory->ScrollBars =
System::Windows::Forms::ScrollBars::Both;
        this->txtOutputHistory->Size = System::Drawing::Size(688, 139);
        this->txtOutputHistory->TabIndex = 2;
        //
        // groupInput
        //
        this->groupInput->Controls->Add(this->txtInputHistory);
        this->groupInput->Location = System::Drawing::Point(8, 3);
        this->groupInput->Name = L"groupInput";
        this->groupInput->Size = System::Drawing::Size(700, 157);
        this->groupInput->TabIndex = 0;
        this->groupInput->TabStop = false;
        this->groupInput->Text = L"Історія надходжень";
        //
        // txtInputHistory
        //
        this->txtInputHistory->Location = System::Drawing::Point(6, 19);
        this->txtInputHistory->Multiline = true;
        this->txtInputHistory->Name = L"txtInputHistory";
        this->txtInputHistory->ScrollBars =
System::Windows::Forms::ScrollBars::Both;
        this->txtInputHistory->Size = System::Drawing::Size(688, 130);
        this->txtInputHistory->TabIndex = 1;
        //
        // nmrPageNumber
        //
        this->nmrPageNumber->Location = System::Drawing::Point(139, 162);
        this->nmrPageNumber->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) {65535, 0, 0, 0});
        this->nmrPageNumber->Name = L"nmrPageNumber";
        this->nmrPageNumber->Size = System::Drawing::Size(80, 20);
        this->nmrPageNumber->TabIndex = 6;
        //
        // txtTheme
        //
        this->txtTheme->Location = System::Drawing::Point(139, 128);
        this->txtTheme->Name = L"txtTheme";
        this->txtTheme->Size = System::Drawing::Size(302, 20);
        this->txtTheme->TabIndex = 5;
        //
        // txtISBN
        //
        this->txtISBN->Location = System::Drawing::Point(139, 94);
        this->txtISBN->Name = L"txtISBN";
        this->txtISBN->Size = System::Drawing::Size(163, 20);
        this->txtISBN->TabIndex = 4;
        //
        // txtAuthor
        //
        this->txtAuthor->Location = System::Drawing::Point(139, 60);

```

```

this->txtAuthor->Name = L"txtAuthor";
this->txtAuthor->Size = System::Drawing::Size(505, 20);
this->txtAuthor->TabIndex = 3;
//
// txtDocName
//
this->txtDocName->Location = System::Drawing::Point(139, 26);
this->txtDocName->Name = L"txtDocName";
this->txtDocName->Size = System::Drawing::Size(557, 20);
this->txtDocName->TabIndex = 2;
//
// tabReports
//
this->tabReports->Controls->Add(this->btnRefresh);
this->tabReports->Controls->Add(this->groupBox2);
this->tabReports->Location = System::Drawing::Point(4, 22);
this->tabReports->Name = L"tabReports";
this->tabReports->Padding = System::Windows::Forms::Padding(3);
this->tabReports->Size = System::Drawing::Size(717, 366);
this->tabReports->TabIndex = 1;
this->tabReports->Text = L"Звіт";
this->tabReports->UseVisualStyleBackColor = true;
//
// lblFreeNumber
//
this->lblFreeNumber->AutoSize = true;
this->lblFreeNumber->Location = System::Drawing::Point(263, 196);
this->lblFreeNumber->Name = L"lblFreeNumber";
this->lblFreeNumber->Size = System::Drawing::Size(103, 13);
this->lblFreeNumber->TabIndex = 6;
this->lblFreeNumber->Text = L"Вільно екземплярів";
//
// lblTotalNumber
//
this->lblTotalNumber->AutoSize = true;
this->lblTotalNumber->Location = System::Drawing::Point(34, 196);
this->lblTotalNumber->Name = L"lblTotalNumber";
this->lblTotalNumber->Size = System::Drawing::Size(105, 13);
this->lblTotalNumber->TabIndex = 5;
this->lblTotalNumber->Text = L"Всього екземплярів";
//
// lblPagesNumber
//
this->lblPagesNumber->AutoSize = true;
this->lblPagesNumber->Location = System::Drawing::Point(40, 162);
this->lblPagesNumber->Name = L"lblPagesNumber";
this->lblPagesNumber->Size = System::Drawing::Size(99, 13);
this->lblPagesNumber->TabIndex = 4;
this->lblPagesNumber->Text = L"Кількість сторінок";
//
// lblTheme
//
this->lblTheme->AutoSize = true;
this->lblTheme->Location = System::Drawing::Point(108, 128);
this->lblTheme->Name = L"lblTheme";
this->lblTheme->Size = System::Drawing::Size(31, 13);
this->lblTheme->TabIndex = 3;
this->lblTheme->Text = L"Тема";
//
// statusStrip
//
this->statusStrip->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) {this->lblStatus});
this->statusStrip->Location = System::Drawing::Point(0, 417);
this->statusStrip->Name = L"statusStrip";
this->statusStrip->Size = System::Drawing::Size(722, 22);
this->statusStrip->TabIndex = 10;
this->statusStrip->Text = L"statusStrip1";
//

```

```

// lblStatus
//
this->lblStatus->Name = L"lblStatus";
this->lblStatus->Size = System::Drawing::Size(109, 17);
this->lblStatus->Text = L"toolStripStatusLabel1";
//
// txtFindString
//
this->txtFindString->Location = System::Drawing::Point(6, 19);
this->txtFindString->Name = L"txtFindString";
this->txtFindString->Size = System::Drawing::Size(609, 20);
this->txtFindString->TabIndex = 0;
//
// tabControl
//
this->tabControl->Controls->Add(this->tabDocs);
this->tabControl->Controls->Add(this->tabReports);
this->tabControl->Controls->Add(this->tabHistory);
this->tabControl->Location = System::Drawing::Point(0, 27);
this->tabControl->Name = L"tabControl";
this->tabControl->SelectedIndex = 0;
this->tabControl->Size = System::Drawing::Size(725, 392);
this->tabControl->TabIndex = 11;
//
// tabDocs
//
this->tabDocs->Controls->Add(this->btnEdit);
this->tabDocs->Controls->Add(this->btnDelete);
this->tabDocs->Controls->Add(this->btnCreate);
this->tabDocs->Controls->Add(this->groupInfo);
this->tabDocs->Controls->Add(this->groupFinding);
this->tabDocs->Location = System::Drawing::Point(4, 22);
this->tabDocs->Name = L"tabDocs";
this->tabDocs->Padding = System::Windows::Forms::Padding(3);
this->tabDocs->Size = System::Drawing::Size(717, 366);
this->tabDocs->TabIndex = 0;
this->tabDocs->Text = L"Документи";
this->tabDocs->UseVisualStyleBackColor = true;
//
// btnEdit
//
this->btnEdit->Location = System::Drawing::Point(402, 337);
this->btnEdit->Name = L"btnEdit";
this->btnEdit->Size = System::Drawing::Size(99, 23);
this->btnEdit->TabIndex = 14;
this->btnEdit->Text = L"Редагувати";
this->btnEdit->UseVisualStyleBackColor = true;
this->btnEdit->Click += gcnew System::EventHandler(this,
&frmWorker::btnEdit_Click);
//
// btnDelete
//
this->btnDelete->Location = System::Drawing::Point(507, 337);
this->btnDelete->Name = L"btnDelete";
this->btnDelete->Size = System::Drawing::Size(99, 23);
this->btnDelete->TabIndex = 15;
this->btnDelete->Text = L"Видалити";
this->btnDelete->UseVisualStyleBackColor = true;
this->btnDelete->Click += gcnew System::EventHandler(this,
&frmWorker::btnDelete_Click);
//
// btnCreate
//
this->btnCreate->Location = System::Drawing::Point(612, 337);
this->btnCreate->Name = L"btnCreate";
this->btnCreate->Size = System::Drawing::Size(99, 23);
this->btnCreate->TabIndex = 16;
this->btnCreate->Text = L"Створити";
this->btnCreate->UseVisualStyleBackColor = true;

```

```

        this->btnCreate->Click += gcnew System::EventHandler(this,
&frmWorker::btnCreate_Click);
        //
        // groupInfo
        //
        this->groupInfo->Controls->Add(this->txtPosition);
        this->groupInfo->Controls->Add(this->btnFirst);
        this->groupInfo->Controls->Add(this->btnPrev);
        this->groupInfo->Controls->Add(this->btnNext);
        this->groupInfo->Controls->Add(this->btnLast);
        this->groupInfo->Controls->Add(this->nmrFreeNumber);
        this->groupInfo->Controls->Add(this->nmrTotalNumber);
        this->groupInfo->Controls->Add(this->nmrPageNumber);
        this->groupInfo->Controls->Add(this->txtTheme);
        this->groupInfo->Controls->Add(this->txtISBN);
        this->groupInfo->Controls->Add(this->txtAuthor);
        this->groupInfo->Controls->Add(this->txtDocName);
        this->groupInfo->Controls->Add(this->lblFreeNumber);
        this->groupInfo->Controls->Add(this->lblTotalNumber);
        this->groupInfo->Controls->Add(this->lblPagesNumber);
        this->groupInfo->Controls->Add(this->lblTheme);
        this->groupInfo->Controls->Add(this->lblISBN);
        this->groupInfo->Controls->Add(this->lblAuthor);
        this->groupInfo->Controls->Add(this->lblDocName);
        this->groupInfo->Location = System::Drawing::Point(6, 65);
        this->groupInfo->Name = L"groupInfo";
        this->groupInfo->Size = System::Drawing::Size(702, 266);
        this->groupInfo->TabIndex = 9;
        this->groupInfo->TabStop = false;
        this->groupInfo->Text = L"Інформація про документ";
        //
        // txtPosition
        //
        this->txtPosition->Location = System::Drawing::Point(564, 241);
        this->txtPosition->Name = L"txtPosition";
        this->txtPosition->Size = System::Drawing::Size(58, 20);
        this->txtPosition->TabIndex = 11;
        this->txtPosition->TextAlign =
System::Windows::Forms::HorizontalAlignment::Center;
        this->txtPosition->KeyDown += gcnew
System::Windows::Forms::KeyEventHandler(this, &frmWorker::txtPosition_KeyDown);
        //
        // btnFirst
        //
        this->btnFirst->Location = System::Drawing::Point(490, 241);
        this->btnFirst->Name = L"btnFirst";
        this->btnFirst->Size = System::Drawing::Size(31, 20);
        this->btnFirst->TabIndex = 9;
        this->btnFirst->Text = L"|<<";
        this->btnFirst->UseVisualStyleBackColor = true;
        this->btnFirst->Click += gcnew System::EventHandler(this,
&frmWorker::btnFirst_Click);
        //
        // btnPrev
        //
        this->btnPrev->Location = System::Drawing::Point(527, 241);
        this->btnPrev->Name = L"btnPrev";
        this->btnPrev->Size = System::Drawing::Size(31, 20);
        this->btnPrev->TabIndex = 10;
        this->btnPrev->Text = L"<<";
        this->btnPrev->UseVisualStyleBackColor = true;
        this->btnPrev->Click += gcnew System::EventHandler(this,
&frmWorker::btnPrev_Click);
        //
        // btnNext
        //
        this->btnNext->Location = System::Drawing::Point(628, 240);
        this->btnNext->Name = L"btnNext";
        this->btnNext->Size = System::Drawing::Size(31, 20);

```

```

this->btnNext->TabIndex = 12;
this->btnNext->Text = L">>";
this->btnNext->UseVisualStyleBackColor = true;
this->btnNext->Click += gcnew System::EventHandler(this,
&frmWorker::btnNext_Click);
//
// btnLast
//
this->btnLast->Location = System::Drawing::Point(665, 240);
this->btnLast->Name = L"btnLast";
this->btnLast->Size = System::Drawing::Size(31, 20);
this->btnLast->TabIndex = 13;
this->btnLast->Text = L">>|";
this->btnLast->UseVisualStyleBackColor = true;
this->btnLast->Click += gcnew System::EventHandler(this,
&frmWorker::btnLast_Click);
//
// lblISBN
//
this->lblISBN->AutoSize = true;
this->lblISBN->Location = System::Drawing::Point(2, 94);
this->lblISBN->Name = L"lblISBN";
this->lblISBN->Size = System::Drawing::Size(137, 13);
this->lblISBN->TabIndex = 2;
this->lblISBN->Text = L"Вхідний номер документу";
//
// lblAuthor
//
this->lblAuthor->AutoSize = true;
this->lblAuthor->Location = System::Drawing::Point(101, 60);
this->lblAuthor->Name = L"lblAuthor";
this->lblAuthor->Size = System::Drawing::Size(38, 13);
this->lblAuthor->TabIndex = 1;
this->lblAuthor->Text = L"Автор";
//
// lblDocName
//
this->lblDocName->AutoSize = true;
this->lblDocName->Location = System::Drawing::Point(44, 26);
this->lblDocName->Name = L"lblDocName";
this->lblDocName->Size = System::Drawing::Size(95, 13);
this->lblDocName->TabIndex = 0;
this->lblDocName->Text = L"Назва документу";
//
// groupFinding
//
this->groupFinding->Controls->Add(this->btnFind);
this->groupFinding->Controls->Add(this->txtFindString);
this->groupFinding->Location = System::Drawing::Point(6, 6);
this->groupFinding->Name = L"groupFinding";
this->groupFinding->Size = System::Drawing::Size(702, 53);
this->groupFinding->TabIndex = 8;
this->groupFinding->TabStop = false;
this->groupFinding->Text = L"Ключові рядки для пошуку";
//
// btnFind
//
this->btnFind->Location = System::Drawing::Point(621, 16);
this->btnFind->Name = L"btnFind";
this->btnFind->Size = System::Drawing::Size(75, 23);
this->btnFind->TabIndex = 1;
this->btnFind->Text = L"Знайти";
this->btnFind->UseVisualStyleBackColor = true;
this->btnFind->Click += gcnew System::EventHandler(this,
&frmWorker::btnFind_Click);
//
// frmWorker
//
this->AutoScaleDimensions = System::Drawing::Size(6, 13);

```

```

        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(722, 439);
        this->Controls->Add(this->menuStrip);
        this->Controls->Add(this->statusStrip);
        this->Controls->Add(this->tabControl);
        this->Name = L"frmWorker";
        this->Text = L"Режим адміністратора інформаційної системи
підприємства - Система Управління Електронним Документообігом ";
        this->Load += gcnew System::EventHandler(this,
&frmWorker::frmWorker_Load);
        this->FormClosed += gcnew
System::Windows::Forms::FormClosedEventHandler(this,
&frmWorker::frmWorker_FormClosed);
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrPagesNumber))->EndInit();
        this->menuStrip->ResumeLayout(false);
        this->menuStrip->PerformLayout();
        this->groupBox 2-2->ResumeLayout(false);
        this->groupBox 2-2->PerformLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrFreeNumber))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrTotalNumber))->EndInit();
        this->tabHistory->ResumeLayout(false);
        this->groupOutput->ResumeLayout(false);
        this->groupOutput->PerformLayout();
        this->groupInput->ResumeLayout(false);
        this->groupInput->PerformLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nmrPageNumber))->EndInit();
        this->tabReports->ResumeLayout(false);
        this->statusStrip->ResumeLayout(false);
        this->statusStrip->PerformLayout();
        this->tabControl->ResumeLayout(false);
        this->tabDocs->ResumeLayout(false);
        this->groupInfo->ResumeLayout(false);
        this->groupInfo->PerformLayout();
        this->groupFinding->ResumeLayout(false);
        this->groupFinding->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
/*****/
private: System::Void LockingForm()
    {
        // Блокування всіх елементів керування
        btnFind->Enabled = false;
        btnEdit->Enabled = false;
        btnDelete->Enabled = false;
        btnCreate->Enabled = false;
        btnFirst->Enabled = false;
        btnLast->Enabled = false;
        btnNext->Enabled = false;
        btnPrev->Enabled = false;
        txtPosition->Enabled = false;
        menuStrip->Enabled = false;
        tabControl->TabPage[1]->Enabled = false;
        tabControl->TabPage[2]->Enabled = false;
    }
/*****/
private: System::Void UnlockingForm()
    {
        // Розблокування всіх елементів керування
        btnFind->Enabled = true;
        btnEdit->Enabled = true;
        btnDelete->Enabled = true;
        btnCreate->Enabled = true;
    }

```

```

        btnFirst->Enabled = true;
        btnLast->Enabled = true;
        btnNext->Enabled = true;
        btnPrev->Enabled = true;
        txtPosition->Enabled = true;
        menuStrip->Enabled = true;
        tabControl->TabPage[1]->Enabled = true;
        tabControl->TabPage[2]->Enabled = true;
    }
    /*****
private: System::Void UpdateView()
    {
        // Вивести історію
        txtInputHistory->Text = Worker->ViewInputLog();
        txtOutputHistory->Text = Worker->ViewOutputLog();

        // Список не порожній?
        if(listView->Count != 0)
        {
            // Перевірити індекс поточного елемента
            if(ddCurrentIndex >= listView->Count)
            {
                // Установити покажчик на останній елемент
                ddCurrentIndex = listView->Count - 1;
            }

            // Значення індексу негативно?
            if(ddCurrentIndex < 0)
            {
                // Установити індекс на перший елемент
                ddCurrentIndex = 0;
            }

            // Одержати характеристики документа й заповнити поля форми
            txtDocName->Text = ((CDoc^)listView[ddCurrentIndex])-
>GetName();
            txtAuthor->Text = ((CDoc^)listView[ddCurrentIndex])-
>GetAuthor();
            txtISBN->Text = ((CDoc^)listView[ddCurrentIndex])->GetISBN();
            txtTheme->Text = ((CDoc^)listView[ddCurrentIndex])-
>GetTheme();
            nmrPageNumber->Value = ((CDoc^)listView[ddCurrentIndex])-
>GetPages();
            nmrTotalNumber->Value = ((CDoc^)listView[ddCurrentIndex])-
>GetTotalNumber();
            nmrFreeNumber->Value = ((CDoc^)listView[ddCurrentIndex])-
>GetFreeNumber();

            // Установити поточну позицію
            txtPosition->Text = Convert::ToString(ddCurrentIndex + 1) + "
/ " +
                Convert::ToString(listView->Count);

            // Розблокувати кнопки редагування й видалення
            btnDelete->Enabled = true;
            btnEdit->Enabled = true;
        }
        else
        {
            // Очистити всі поля
            txtDocName->Text = "";
            txtAuthor->Text = "";
            txtISBN->Text = "";
            txtTheme->Text = "";
            nmrPageNumber->Value = 0;
            nmrTotalNumber->Value = 0;
            nmrFreeNumber->Value = 0;
            // Поточна позиція - 0
            txtPosition->Text = "0/0";
        }
    }
}

```

```

        // Установити індекс поточного елемента
        ddCurrentIndex = 0;
        // Заблокувати кнопки редагування й видалення
        btnDelete->Enabled = false;
        btnEdit->Enabled = false;
        // Вивести стан
        lblStatus->Text = "Немає записів для відображення";
    }
}
/*****/
private: System::Void frmWorker_Load(System::Object^ sender, System::EventArgs^ e)
{
    // Створити об'єкт "Адміністратор"
    Worker = gcnew CWorker();

    // Завантажити список документів
    Worker->LoadDocList();

    // Завантажити історію
    Worker->ReadLogs();

    // Обновити форму
    UpdateView();

    // Вивести стан
    lblStatus->Text = "Для відображення існуючого списку скористайтеся
пошуком";

    // Заповнення випадаючого списку вибору документів по кількості
сторінок
    cmbPagesDirect->Items->Add(PAGES_LESS);
    cmbPagesDirect->Items->Add(PAGES_GREATER);
    cmbPagesDirect->Items->Add(PAGES_EQUAL);
}
/*****/
private: System::Void btnCreate_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Кнопка "Створити" була натиснута раніше?
    if(IsCreateClicked)
    {
        // Забрати зайві пробіли у всіх полях
        txtDocName->Text = txtDocName->Text->Trim();
        txtAuthor->Text = txtAuthor->Text->Trim();
        txtISBN->Text = txtISBN->Text->Trim();
        txtTheme->Text = txtTheme->Text->Trim();

        // Перевірити значимі поля на коректність
        if(txtDocName->Text == "" || txtISBN->Text == "")
        {
            // Обов'язкові поля не заповнені
            MessageBox::Show("Поля 'Назва документа' і 'вхідний
номер документа' обов'язкові для заповнення!",
                "Usrs Manager", MessageBoxButtons::OK,
                MessageBoxIcon::Error);
            return;
        }

        // Перевірити кількість екземплярів
        if(Convert::ToInt32(nmrTotalNumber->Value) <= 0)
        {
            // Вивести повідомлення про помилку
            MessageBox::Show("Некоректне значення кількості
екземплярів",
                "Usrs Manager", MessageBoxButtons::OK,
                MessageBoxIcon::Error);
            return;
        }
    }
}

```



```

nmrFreeNumber->Value = 1;

// Вивести стан
lblStatus->Text = "Введіть значення полів";

// Змінити значення прапора натискання по кнопці "Створити"
IsCreateClicked = true;
    }
}
/*****/
private: System::Void btnDelete_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Видалити документ із загального списку
    Int32 ddResult = Worker->RemoveDoc(
        ((CDoc^)listView[ddCurrentIndex])->GetISBNHash());

    // Перевірити код повернення
    switch(ddResult)
    {
    case 0:
        {
            // Документ вилучений успішно
            // Видалити документ зі списку для виводу
            listView->RemoveAt(ddCurrentIndex);
            // Оновити форму
            UpdateView();

            break;
        }
    case 1:
        {
            // Документ не був вилучений, не всі екземпляри в
системі
            MessageBox::Show("Видалення неможливо, тому що не всі
документи перебувають у системі",
                "Usrs Manager", MessageBoxButtons::OK,
MessageBoxIcon::Error);
            break;
        }
    case 2:
        {
            // Документ не знайдений
            MessageBox::Show("Видалення неможливо, документ у
системі відсутній",
                "Usrs Manager", MessageBoxButtons::OK,
MessageBoxIcon::Error);
            // Видалити документ зі списку для виводу
            listView->RemoveAt(ddCurrentIndex);
            // Оновити форму
            UpdateView();

            break;
        }
    }
}
/*****/
private: System::Void btnFind_Click(System::Object^ sender, System::EventArgs^
e)
{
    // Забрати в рядку пошуку зайві пробіли й перетворити до нижнього
регістра
    String^ strValue = txtFindString->Text->Trim()->ToLower();

    // Перевірити рядок
    if(String::IsNullOrEmpty(strValue))
    {
        // Рядок порожня, запропонувати вивести весь список

```

```

        if (MessageBox::Show("Рядок пошуку не заданий, вивести весь
        список?",
        "Usrs Manager",
        MessageBoxButtons::YesNo,
        MessageBoxIcon::Question) ==
        ::DialogResult::Yes)
        {
            // Перейти на вивід списку
            goto OUT_LIST;
        }
        // Завершити роботу функції
        return;
    }

OUT_LIST: // Здійснити пошук документа по заданих параметрах
    listView = Worker->FindDoc(strValue);

    // Вивести стан
    lblStatus->Text = "Пошук завершений";
    // Обновити форму
    UpdateView();
}
/*****/
private: System::Void btnFirst_Click(System::Object^ sender, System::EventArgs^
e)
    {
        // Установити індекс на перший елемент
        ddCurrentIndex = 0;
        // Обновити форму
        UpdateView();
    }
/*****/
private: System::Void btnPrev_Click(System::Object^ sender, System::EventArgs^
e)
    {
        // Установити індекс на попередній елемент
        ddCurrentIndex ---i;
        // Обновити форму
        UpdateView();
    }
/*****/
private: System::Void btnNext_Click(System::Object^ sender, System::EventArgs^
e)
    {
        // Установити індекс на наступний елемент
        ddCurrentIndex ++;
        // Обновити форму
        UpdateView();
    }
/*****/
private: System::Void btnLast_Click(System::Object^ sender, System::EventArgs^
e)
    {
        // Установити індекс на останній елемент
        ddCurrentIndex = listView->Count - 1;
        // Обновити форму
        UpdateView();
    }
/*****/
private: System::Void btnEdit_Click(System::Object^ sender, System::EventArgs^
e)
    {
        // Кнопка "Редагувати" була натиснута раніше?
        if(IsEditClicked)
        {
            // Забрати зайві пробіли
            txtDocName->Text = txtDocName->Text->Trim();
            txtAuthor->Text = txtAuthor->Text->Trim();
            txtTheme->Text = txtTheme->Text->Trim();
        }
    }
}

```

```

// Перевірити значимі поля на коректність
if(txtDocName->Text == "")
{
    // Обов'язкові поля не заповнені
    MessageBox::Show("Поле 'Назва документа' обов'язково
для заповнення!",
                    "Usrs Manager", MessageBoxButtons::OK,
MessageBoxIcon::Error);
    return;
}

// Відредагувати об'єкт у списках
((CDoc^)listView[ddCurrentIndex])->SetName(txtDocName->Text);
((CDoc^)listView[ddCurrentIndex])->SetAuthor(txtAuthor-
>Text);
((CDoc^)listView[ddCurrentIndex])->SetTheme(txtTheme->Text);
((CDoc^)listView[ddCurrentIndex])->
>SetPages(Convert::ToInt32(nmrPageNumber->Value));

// (???) Видалити документ зі списку перегляду
//Worker->RemoveDoc(((CDoc^)listView[ddCurrentIndex])-
>GetISBNHash());
// (???) Додати документ у список перегляду
//Worker->AddDoc((CDoc^)listView[ddCurrentIndex]);

// Обновити форму
UpdateView();

// Розблокувати елементи керування
UnlockingForm();

// Розблокувати некоректируємі поля
txtISBN->Enabled = true;
nmrTotalNumber->Enabled = true;
nmrFreeNumber->Enabled = true;

// Перемінити напис на кнопці
btnEdit->Text = "Редагувати";

// Змінити значення прапора натискання кнопки "Редагувати"
IsEditClicked = false;
}
else
{
    // Перемінити напис на кнопці
    btnEdit->Text = "Зберегти";

    // Заблокувати елементи керування
    LockingForm();

    // Розблокувати кнопку збереження
    btnEdit->Enabled = true;

    // Заблокувати некоректируємі поля
    txtISBN->Enabled = false;
    nmrTotalNumber->Enabled = false;
    nmrFreeNumber->Enabled = false;

    // Змінити значення прапора натискання кнопки "Редагувати"
    IsEditClicked = true;
}
}
}
/*****
private: System::Void menuFileSave_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Зберегти список документів у файл
    Worker->SaveDocList();
}
}

```

```

// Зберегти історію
Worker->WriteLogs();

// Вивести стан
lblStatus->Text = "Збереження виконане";
}
/*****/
private: System::Void menuFileExit_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Вийти з додатка
    Application::Exit();
}
/*****/
private: System::Void nmrTotalNumber_ValueChanged(System::Object^ sender,
System::EventArgs^ e)
{
    // Можливе натискання тільки при створенні документа
    // Кількість вільних екземплярів дорівнює загальній кількості
екземплярів
    nmrFreeNumber->Value = nmrTotalNumber->Value;
}
/*****/
private: System::Void frmWorker_FormClosed(System::Object^ sender,
System::Windows::Forms::FormClosedEventArgs^ e)
{
    // Вийти з додатка
    Application::Exit();
}
/*****/
private: System::Void btnClearLogs_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Очистити історію
    Worker->ClearLogs();

    // Оновити форму
    UpdateView();

    // Вивести стан
    lblStatus->Text = "Очищення виконане";
}
/*****/
private: System::Void btnRefresh_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Створити об'єкт "Лічильник"
    CCounter^ Counter = gcnew CCounter();

    // Установлений прапор підрахунку сумарної кількості екземплярів?
    if(chkTotalInstaceFlag->Checked)
    {
        // Обчислити
        lblTotalInstanceNumber->Text =
            Convert::ToString(Counter-
>GetTotalInstanceNumber(listView));
    }

    // Установлений прапор підрахунку сумарної кількості вільних
екземплярів?
    if(chkFreeInstanceFlag->Checked)
    {
        // Обчислити
        lblFreeInstanceNumber->Text =
            Convert::ToString(Counter-
>GetFreeInstanceNumber(listView));
    }
}

```

```

// Встановлений прапор підрахунку документів заданого автора?
if(chkAuthorDocFlag->Checked)
{
    // Створити тимчасовий список
    ArrayList^ listTemp = gcnew ArrayList();

    // Одержати список документів заданого автора
    listTemp = Counter->GetDocListOfAuthor(listView,
txtAuthorFilter->Text);

    // Очистити список на формі
    lstAuthorDocList->Items->Clear();
    // Вивести список знайдених документів
    for(Int32 i = 0; i < listTemp->Count; i++)
    {
        // Зчитати параметри документа й додати в список на
формі
        lstAuthorDocList->Items->Add(
            Convert::ToString(i + 1) + ". " +
            ((CDoc^)listTemp[i])->GetName() + ", " +
            ((CDoc^)listTemp[i])->GetAuthor() + ", що входить
номер документа " +
            ((CDoc^)listTemp[i])->GetISBN());
    }

    // Очистити тимчасовий список
    listTemp->Clear();
}

// Установлений прапор виводу списку документів по заданій темі?
if(chkThemeDocFlag->Checked)
{
    // Створити тимчасовий список
    ArrayList^ listTemp = gcnew ArrayList();

    // Одержати список документів по заданій темі
    listTemp = Counter->GetDocListOnTheme(listView,
txtThemeFilter->Text);

    // Очистити список для виводу на формі
    lstThemeDocList->Items->Clear();
    // Вивести список знайдених документів
    for(Int32 i = 0; i < listTemp->Count; i++)
    {
        // Завантажити параметри документа й додати в список на
формі
        lstThemeDocList->Items->Add(
            Convert::ToString(i + 1) + ". " +
            ((CDoc^)listTemp[i])->GetName() + ", " +
            ((CDoc^)listTemp[i])->GetTheme() + ", що входить
номер документа " +
            ((CDoc^)listTemp[i])->GetISBN());
    }

    // Очистити тимчасовий список
    listTemp->Clear();
}

// Установлений прапор виводу списку документів із заданою
кількістю сторінок?
if(chkPagesFlag->Checked)
{
    // Створити тимчасовий список
    ArrayList^ listTemp = gcnew ArrayList();

    // Одержати список документів, у яких кількість сторінок
задовольняє
    // заданій умові
    listTemp = Counter->GetDocListByPages(listView,
        cmbPagesDirect->Text, Convert::ToInt32(nmrPagesNumber-
>Value));
}

```

```

// Очистити список для виводу на формі
lstPagesDocList->Items->Clear();
// Вивести список знайдених документів
for(Int32 i = 0; i < listTemp->Count; i++)
{
    // Завантажити параметри документа й додати в список на
    формі
    lstPagesDocList->Items->Add(
        Convert::ToString(i + 1) + ". " +
        ((CDoc^)listTemp[i])->GetName() + ", " +
        Convert::ToString(((CDoc^)listTemp[i])->GetPages())
        + " стор., що входить номер документа " +
        ((CDoc^)listTemp[i])->GetISBN());
    }
    // Очистити тимчасовий список
    listTemp->Clear();
}
// Вивести стан
lblStatus->Text = "Відновлення необхідних звітів закінчене";
}
/*****
private: System::Void txtPosition_KeyDown(System::Object^ sender,
System::Windows::Forms::KeyEventEventArgs^ e)
{
    // Натиснута клавіша "Enter"?
    if( e->KeyCode == Keys::Enter)
    {
        try
        {
            // Спробувати одержати введений номер документа для
            перегляду
            ddCurrentIndex = Convert::ToInt32(txtPosition->Text) -
            1;
        }
        catch(...)
        {
            // Якщо в процесі перетворення номера з рядка в число
            // виникла помилка, те ніяк на це не реагувати
        }
        // Обновити форму
        UpdateView();
    }
    else
    {
        // Текстове поле містить символ "/"?
        if(txtPosition->Text->IndexOf("/") != -1)
        {
            // Для початку введення нового номера позиції,
            необхідно
            // очистити текстове поле
            txtPosition->Text = "";
        }
    }
}
/*****
private: System::Void menuHelpAbout_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Відобразити форму з інформацією про програму
    UsrsManager::frmAbout^ frmNew = gcnew UsrsManager::frmAbout();
    frmNew->Show();
}
/*****
};
}

```

Файл Reader.cpp - вікно користувача інформаційної системи підприємства

```

/*****/
#include "StdAfx.h"
#include "Reader.h"
#include "Worker.h"
#include "Logger.h"
/*****/
using namespace System;
using namespace System::IO;
using namespace System::Collections;
using namespace System::Windows::Forms;
/*****/
#define READER_FILE_EXTENSION ".LMR"
/*****/
// NAME:          CReader
// DESCRIPTION:   Конструктор класу
// INPUT:         strUserName - ім'я облікового запису користувача інформаційної
системи підприємства
/*****/
CReader::CReader(String^ strUserName)
{
    // Сформувати ім'я файлу облікового запису
    strFileName = strUserName + READER_FILE_EXTENSION;
    // Створити список для документів користувача інформаційної системи
    підприємства
    listMyDoc = gcnew ArrayList();
    // Створити об'єкт для ведення історії
    logUsing = gcnew CLogger();
}
/*****/
// NAME:          Load
// DESCRIPTION:   Функція завантаження списку документів користувача
інформаційної системи підприємства з файлу
// INPUT:         Worker - об'єкт, через який здійснюється одержання інформації
про
//               книзі по вхідному номеру документа
/*****/
Boolean CReader::Load(CWorker^ Worker)
{
    // Перевірка існування файлу користувача інформаційної системи
    підприємства
    if(!File::Exists(strFileName))
    {
        // Видати запит на створення нового файлу
        if(MessageBox::Show("Файл користувача інформаційної системи
    підприємства не знайдений, створити новий?",
            "Система УЕД", MessageBoxButtons::YesNo,
            MessageBoxIcon::Warning) == DialogResult::Yes)
        {
            // Створити файловий потік
            FileStream^ fsReaderFile = gcnew FileStream(strFileName,
                FileMode::CreateNew);
            // Відкрити файл користувача інформаційної системи
    підприємства
            StreamWriter^ swReaderFile = gcnew StreamWriter(fsReaderFile);
            // Кількість документів у користувача інформаційної системи
    підприємства дорівнює 0
            swReaderFile->WriteLine("0");
            // Закрити файл
            swReaderFile->Close();
            // Закрити файловий потік
            fsReaderFile->Close();
            // Закінчити виконання функції, але виконати вхід
            return true;
        }
    }
    else

```

```

        {
            // Закінчити виконання функції й не виконувати вхід
            return false;
        }
    }
    // Відкрити файл користувача інформаційної системи підприємства
    StreamReader^ srReaderFile = gcnew StreamReader(strFileName);
    // Завантажити кількість документів
    Int32 ddDocsNumber = Convert::ToInt32(srReaderFile->ReadLine());
    // Цикл зчитування інформації про документи
    for(Int32 i = 0; i < ddDocsNumber; i++)
    {
        // Зчитування вхідний номер документа з файлу
        Int32 ddISBNHash = Convert::ToInt32(srReaderFile->ReadLine());

        // Одержання інформації й додавання документи в список документів
        користувача інформаційної системи підприємства
        listMyDoc->Add(Worker->ViewDoc(ddISBNHash));
    }

    // Кінець файлу не досягнуть?
    if(!srReaderFile->EndOfStream)
    {
        // Завантажити історію одержань і повернень документів
        logUsing->strLog = srReaderFile->ReadToEnd();
    }
    else
    {
        // Обнулити рядок історії
        logUsing->strLog = "";
    }

    // Закрити файл користувача інформаційної системи підприємства
    srReaderFile->Close();

    // Закінчити виконання функції й виконати вхід
    return true;
}
/*****
// NAME:         Save
// DESCRIPTION:   Функція збереження списку документів користувача інформаційної
системи підприємства у файл
// INPUT:        N/D
*****/
void CReader::Save()
{
    // Перевірка існування файлу користувача інформаційної системи
    підприємства
    if(File::Exists(strFileName))
    {
        // Видалення старого файлу
        File::Delete(strFileName);
    }
    // Відкрити новий файл користувача інформаційної системи підприємства
    StreamWriter^ swReaderFile = gcnew StreamWriter(strFileName);

    // Записати кількість документів
    Int32 ddDocsNumber = listMyDoc->Count;
    swReaderFile->WriteLine(Convert::ToString(ddDocsNumber));
    // Цикл запису інформації про документи
    for(Int32 i = 0; i < ddDocsNumber; i++)
    {
        // Запис вхідний номер документа у файл
        swReaderFile->WriteLine(((CDoc^)listMyDoc[i])->GetISBNHash());
    }
    // Записати лог
    swReaderFile->WriteLine(logUsing->strLog);
    // Закрити файл користувача інформаційної системи підприємства
    swReaderFile->Close();
}

```

```

}
/*****/
// NAME:          GetMyDocsList
// DESCRIPTION:   Функція одержання списку документів користувача інформаційної
системи підприємства
// INPUT:         N/D
/*****/
ArrayList^ CReader::GetMyDocsList()
{
    return listMyDoc;
}
/*****/
// NAME:          RequireDoc
// DESCRIPTION:   Функція запиту заданого документа
// INPUT:         Worker - об'єкт "Адміністратор", через який здійснюється
одержання
//              ddHash - хеш-значення вхідного номера документа, що
потрібно одержати
/*****/
Boolean CReader::RequireDoc(CWorker^ Worker, Int32 ddHash)
{
    // Запит на одержання документи до "адміністратора інформаційної системи
підприємства"
    Boolean fResult = Worker->GiveDoc(listMyDoc, ddHash);
    // Результат одержання документів позитивний?
    if(fResult)
    {
        // Індекс останньої документи
        Int32 ddIndex = listMyDoc->Count - 1;
        // Записати операцію в лог
        logUsing->WriteEvent("Отриманий документ '" +
((CDoc^)listMyDoc[ddIndex])->GetName() +
"', Автор '" + ((CDoc^)listMyDoc[ddIndex])->GetAuthor() +
"', що входить номер документа '" +
((CDoc^)listMyDoc[ddIndex])->GetISBN());

        // Операція пройшла успішно
        return true;
    }
    else
    {
        // Такого документа в наявності не є
        return false;
    }
}
/*****/
// NAME:          ReleaseDoc
// DESCRIPTION:   Функція повернення заданого документа
// INPUT:         Worker - об'єкт "Адміністратор", через який здійснюється
повернення
//              ddIndex - індекс документа в списку користувача
інформаційної системи підприємства
/*****/
void CReader::ReleaseDoc(CWorker^ Worker, Int32 ddIndex)
{
    // Повернути документ
    Worker->TakeDoc(listMyDoc, ((CDoc^)listMyDoc[ddIndex])->GetISBNHash());
    // Записати операцію в лог
    logUsing->WriteEvent(
        "Повернутий документ '" + ((CDoc^)listMyDoc[ddIndex])->GetName() +
        "', Автор '" + ((CDoc^)listMyDoc[ddIndex])->GetName() +
        "', що входить номер документа '" + ((CDoc^)listMyDoc[ddIndex])->
GetISBN());
    // Видалити документ зі списку користувача інформаційної системи
підприємства
    listMyDoc->RemoveAt(ddIndex);
}
/*****/
// NAME:          ViewLog

```

```
// DESCRIPTION:  Функція перегляду історії
// INPUT:       N/D
/*****/
String^ CReader::ViewLog()
{
    return logUsing->strLog;
}
/*****/
```

КБПЗ_2023

Файл Reader.resx - xml опис вікна користувача інформаційної системи підприємства

```

<?xml version="1.0" encoding=" utf-8"?>
<root>
  <xsd:schema id="root" xmlns="" xmlns:xsd=" " xmlns:msdata="urn: schemas-
microsoft-com: xml-msdata">
    <xsd:import namespace=" " />
    <xsd:element name="root" msdata:IsDataSet="true">
      <xsd:complexType>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element name="metadata">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0" />
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="assembly">
            <xsd:complexType>
              <xsd:attribute name="alias" type="xsd:string" />
              <xsd:attribute name="name" type="xsd:string" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="data">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
                <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" use="required"
msdata:Ordinal="1" />
              <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
              <xsd:attribute name="mimetype" type="xsd:string"
msdata:Ordinal="4" />
              <xsd:attribute ref="xml:space" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="resheader">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" use="required" />
            </xsd:complexType>
          </xsd:element>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
  <resheader name="resmimetype">
    <value>text/ microsoft-resx</value>
  </resheader>
  <resheader name="version">
    <value>2.0</value>
  </resheader>
  <resheader name="reader">
    <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
  <resheader name="writer">

```

```
<value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<metadata name="statusStrip.TrayLocation" type="System.Drawing.Point,
System.Drawing, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a">
  <value>436, 17</value>
</metadata>
<metadata name="menuStrip.TrayLocation" type="System.Drawing.Point,
System.Drawing, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a">
  <value>540, 17</value>
</metadata>
</root>
```

K6ПЗ_2023

Файл Reader.h - бібліотека для файлу frmAbout.cpp

```

#pragma once
/*****/
#include "Worker.h"
#include "Doc.h"
#include "Reader.h"

#include "frmAbout.h"
/*****/
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
/*****/
namespace UsrsManager
{
    public ref class frmReader : public System::Windows::Forms::Form
    {
/*****/
    private:
        CWorker^ Worker;        // Об'єкт класу "Адміністратор"
        CReader^ Reader;        // Об'єкт класу "Користувач"

        ArrayList^ listView;    // Поточний відображуваний список документів

        Int32 ddCurrentIndex;    // Індекс поточного відображуваного елемента

        String^ strUserName;    // Поточне ім'я користувача інформаційної системи
        підприємства
/*****/
    public:
        frmReader(String^ strReaderName)
        {
            // Ініціалізація елементів керування на формі
            InitializeComponent();

            // Створити список відображуваних документів
            listView = gcnew ArrayList();

            // Поточний індекс не визначений
            ddCurrentIndex = -1;

            // Зберегти ім'я користувача інформаційної системи підприємства
            strUserName = strReaderName;
        }
/*****/
    protected: ~frmReader()
    {
        if (components)
        {
            delete components;
        }
    }
/*****/
    private: System::Windows::Forms::Button^ btnRequest;
    private: System::Windows::Forms::ToolStripStatusLabel^ lblStatus;
    private: System::Windows::Forms::StatusStrip^ statusStrip;
    private: System::Windows::Forms::TabPage^ tabFinding;
    private: System::Windows::Forms::GroupBox^ groupBox1;
    private: System::Windows::Forms::Label^ lblPagesNumber;
    private: System::Windows::Forms::Label^ lblTheme;
    private: System::Windows::Forms::Label^ lblISBN;
    private: System::Windows::Forms::Label^ lblAuthor;
    private: System::Windows::Forms::Label^ lblDocName;
}

```

```

private: System::Windows::Forms::GroupBox^ groupFinding;
private: System::Windows::Forms::Button^ btnFind;
private: System::Windows::Forms::TextBox^ txtFindString;
private: System::Windows::Forms::TabControl^ tabControl;
private: System::Windows::Forms::TabPage^ tabMyDocs;
private: System::Windows::Forms::Button^ btnReturn;
private: System::Windows::Forms::GroupBox^ groupBox2;
private: System::Windows::Forms::ListBox^ lstMyDocs;
private: System::Windows::Forms::TabPage^ tabHistory;
private: System::Windows::Forms::GroupBox^ groupBox3;
private: System::Windows::Forms::TextBox^ txtHistory;
private: System::Windows::Forms::ToolStripMenuItem^ menuHelp;

private: System::Windows::Forms::ToolStripMenuItem^ menuHelpAbout;
private: System::Windows::Forms::ToolStripMenuItem^ menuFileExit;
private: System::Windows::Forms::MenuStrip^ menuStrip;
private: System::Windows::Forms::ToolStripMenuItem^ menuFile;
private: System::Windows::Forms::ToolStripMenuItem^ menuFileSave;
private: System::Windows::Forms::TextBox^ txtPosition;
private: System::Windows::Forms::Button^ btnFirst;
private: System::Windows::Forms::Button^ btnPrev;
private: System::Windows::Forms::Button^ btnNext;
private: System::Windows::Forms::Button^ btnLast;
private: System::Windows::Forms::TextBox^ txtPagesNumber;
private: System::Windows::Forms::TextBox^ txtTheme;
private: System::Windows::Forms::TextBox^ txtISBN;
private: System::Windows::Forms::TextBox^ txtAuthor;
private: System::Windows::Forms::TextBox^ txtDocName;

private:
/// <summary>
/// Required designer variable.
/// </summary>
System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
/// <summary>
/// Необхідний метод для підтримки Розроблювача - не модифікувати
/// зміст цього методу з кодовим редактором.
/// </summary>
void InitializeComponent(void)
{
    this->btnRequest = (gcnew System::Windows::Forms::Button());
    this->lblStatus = (gcnew
System::Windows::Forms::ToolStripStatusLabel());
    this->statusStrip = (gcnew System::Windows::Forms::StatusStrip());
    this->tabFinding = (gcnew System::Windows::Forms::TabPage());
    this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
    this->txtPosition = (gcnew System::Windows::Forms::TextBox());
    this->btnFirst = (gcnew System::Windows::Forms::Button());
    this->btnPrev = (gcnew System::Windows::Forms::Button());
    this->btnNext = (gcnew System::Windows::Forms::Button());
    this->btnLast = (gcnew System::Windows::Forms::Button());
    this->txtPagesNumber = (gcnew System::Windows::Forms::TextBox());
    this->txtTheme = (gcnew System::Windows::Forms::TextBox());
    this->txtISBN = (gcnew System::Windows::Forms::TextBox());
    this->txtAuthor = (gcnew System::Windows::Forms::TextBox());
    this->txtDocName = (gcnew System::Windows::Forms::TextBox());
    this->lblPagesNumber = (gcnew System::Windows::Forms::Label());
    this->lblTheme = (gcnew System::Windows::Forms::Label());
    this->lblISBN = (gcnew System::Windows::Forms::Label());
    this->lblAuthor = (gcnew System::Windows::Forms::Label());
    this->lblDocName = (gcnew System::Windows::Forms::Label());
    this->groupFinding = (gcnew System::Windows::Forms::GroupBox());
    this->btnFind = (gcnew System::Windows::Forms::Button());
    this->txtFindString = (gcnew System::Windows::Forms::TextBox());
    this->tabControl = (gcnew System::Windows::Forms::TabControl());
    this->tabMyDocs = (gcnew System::Windows::Forms::TabPage());
    this->btnReturn = (gcnew System::Windows::Forms::Button());

```

```

        this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());
        this->lstMyDocs = (gcnew System::Windows::Forms::ListBox());
        this->tabHistory = (gcnew System::Windows::Forms::TabPage());
        this->groupBox3 = (gcnew System::Windows::Forms::GroupBox());
        this->txtHistory = (gcnew System::Windows::Forms::TextBox());
        this->menuHelp = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuHelpAbout = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuFileExit = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuStrip = (gcnew System::Windows::Forms::MenuStrip());
        this->menuFile = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->menuFileSave = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->statusStrip->SuspendLayout();
        this->tabFinding->SuspendLayout();
        this->groupBox 1-1->SuspendLayout();
        this->groupFinding->SuspendLayout();
        this->tabControl->SuspendLayout();
        this->tabMyDocs->SuspendLayout();
        this->groupBox 2-2->SuspendLayout();
        this->tabHistory->SuspendLayout();
        this->groupBox 3-3->SuspendLayout();
        this->menuStrip->SuspendLayout();
        this->SuspendLayout();
        //
        // btnRequest
        //
        this->btnRequest->Location = System::Drawing::Point(567, 285);
        this->btnRequest->Name = L"btnRequest";
        this->btnRequest->Size = System::Drawing::Size(144, 23);
        this->btnRequest->TabIndex = 12;
        this->btnRequest->Text = L"Здійснити запит";
        this->btnRequest->UseVisualStyleBackColor = true;
        this->btnRequest->Click += gcnew System::EventHandler(this,
&frmReader::btnRequest_Click);
        //
        // lblStatus
        //
        this->lblStatus->Name = L"lblStatus";
        this->lblStatus->Size = System::Drawing::Size(109, 17);
        this->lblStatus->Text = L"toolStripStatusLabel1";
        this->lblStatus->Click += gcnew System::EventHandler(this,
&frmReader::lblStatus_Click);
        //
        // statusStrip
        //
        this->statusStrip->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) {this->lblStatus});
        this->statusStrip->Location = System::Drawing::Point(0, 365);
        this->statusStrip->Name = L"statusStrip";
        this->statusStrip->Size = System::Drawing::Size(724, 22);
        this->statusStrip->TabIndex = 13;
        this->statusStrip->Text = L"statusStrip1";
        //
        // tabFinding
        //
        this->tabFinding->Controls->Add(this->btnRequest);
        this->tabFinding->Controls->Add(this->groupBox1);
        this->tabFinding->Controls->Add(this->groupFinding);
        this->tabFinding->Location = System::Drawing::Point(4, 22);
        this->tabFinding->Name = L"tabFinding";
        this->tabFinding->Padding = System::Windows::Forms::Padding(3);
        this->tabFinding->Size = System::Drawing::Size(717, 314);
        this->tabFinding->TabIndex = 0;
        this->tabFinding->Text = L"Пошук";
        this->tabFinding->UseVisualStyleBackColor = true;

```

```

//
// groupBox1
//
this->groupBox 1-1->Controls->Add(this->txtPosition);
this->groupBox 1-1->Controls->Add(this->btnFirst);
this->groupBox 1-1->Controls->Add(this->btnPrev);
this->groupBox 1-1->Controls->Add(this->btnNext);
this->groupBox 1-1->Controls->Add(this->btnLast);
this->groupBox 1-1->Controls->Add(this->txtPagesNumber);
this->groupBox 1-1->Controls->Add(this->txtTheme);
this->groupBox 1-1->Controls->Add(this->txtISBN);
this->groupBox 1-1->Controls->Add(this->txtAuthor);
this->groupBox 1-1->Controls->Add(this->txtDocName);
this->groupBox 1-1->Controls->Add(this->lblPagesNumber);
this->groupBox 1-1->Controls->Add(this->lblTheme);
this->groupBox 1-1->Controls->Add(this->lblISBN);
this->groupBox 1-1->Controls->Add(this->lblAuthor);
this->groupBox 1-1->Controls->Add(this->lblDocName);
this->groupBox 1-1->Location = System::Drawing::Point(6, 65);
this->groupBox 1-1->Name = L"groupBox1";
this->groupBox 1-1->Size = System::Drawing::Size(702, 214);
this->groupBox 1-1->TabIndex = 9;
this->groupBox 1-1->TabStop = false;
this->groupBox 1-1->Text = L"Інформація про документ";
//
// txtPosition
//
this->txtPosition->Location = System::Drawing::Point(568, 187);
this->txtPosition->Name = L"txtPosition";
this->txtPosition->Size = System::Drawing::Size(58, 20);
this->txtPosition->TabIndex = 9;
this->txtPosition->TextAlign =
System::Windows::Forms::HorizontalAlignment::Center;
this->txtPosition->KeyDown += gcnew
System::Windows::Forms::EventHandler(this, &frmReader::txtPosition_KeyDown);
//
// btnFirst
//
this->btnFirst->Location = System::Drawing::Point(494, 187);
this->btnFirst->Name = L"btnFirst";
this->btnFirst->Size = System::Drawing::Size(31, 20);
this->btnFirst->TabIndex = 7;
this->btnFirst->Text = L"|<<";
this->btnFirst->UseVisualStyleBackColor = true;
this->btnFirst->Click += gcnew System::EventHandler(this,
&frmReader::btnFirst_Click);
//
// btnPrev
//
this->btnPrev->Location = System::Drawing::Point(531, 187);
this->btnPrev->Name = L"btnPrev";
this->btnPrev->Size = System::Drawing::Size(31, 20);
this->btnPrev->TabIndex = 8;
this->btnPrev->Text = L"<<";
this->btnPrev->UseVisualStyleBackColor = true;
this->btnPrev->Click += gcnew System::EventHandler(this,
&frmReader::btnPrev_Click);
//
// btnNext
//
this->btnNext->Location = System::Drawing::Point(632, 186);
this->btnNext->Name = L"btnNext";
this->btnNext->Size = System::Drawing::Size(31, 20);
this->btnNext->TabIndex = 10;
this->btnNext->Text = L">>";
this->btnNext->UseVisualStyleBackColor = true;
this->btnNext->Click += gcnew System::EventHandler(this,
&frmReader::btnNext_Click);
//

```

```

// btnLast
//
this->btnLast->Location = System::Drawing::Point(669, 186);
this->btnLast->Name = L"btnLast";
this->btnLast->Size = System::Drawing::Size(31, 20);
this->btnLast->TabIndex = 11;
this->btnLast->Text = L">>|";
this->btnLast->UseVisualStyleBackColor = true;
this->btnLast->Click += gcnew System::EventHandler(this,
&frmReader::btnLast_Click);
//
// txtPagesNumber
//
this->txtPagesNumber->Location = System::Drawing::Point(139, 161);
this->txtPagesNumber->Name = L"txtPagesNumber";
this->txtPagesNumber->ReadOnly = true;
this->txtPagesNumber->Size = System::Drawing::Size(48, 20);
this->txtPagesNumber->TabIndex = 6;
//
// txtTheme
//
this->txtTheme->Location = System::Drawing::Point(139, 128);
this->txtTheme->Name = L"txtTheme";
this->txtTheme->ReadOnly = true;
this->txtTheme->Size = System::Drawing::Size(302, 20);
this->txtTheme->TabIndex = 5;
//
// txtISBN
//
this->txtISBN->Location = System::Drawing::Point(139, 94);
this->txtISBN->Name = L"txtISBN";
this->txtISBN->ReadOnly = true;
this->txtISBN->Size = System::Drawing::Size(163, 20);
this->txtISBN->TabIndex = 4;
//
// txtAuthor
//
this->txtAuthor->Location = System::Drawing::Point(139, 60);
this->txtAuthor->Name = L"txtAuthor";
this->txtAuthor->ReadOnly = true;
this->txtAuthor->Size = System::Drawing::Size(505, 20);
this->txtAuthor->TabIndex = 3;
//
// txtDocName
//
this->txtDocName->Location = System::Drawing::Point(139, 26);
this->txtDocName->Name = L"txtDocName";
this->txtDocName->ReadOnly = true;
this->txtDocName->Size = System::Drawing::Size(557, 20);
this->txtDocName->TabIndex = 2;
//
// lblPagesNumber
//
this->lblPagesNumber->AutoSize = true;
this->lblPagesNumber->Location = System::Drawing::Point(40, 164);
this->lblPagesNumber->Name = L"lblPagesNumber";
this->lblPagesNumber->Size = System::Drawing::Size(99, 13);
this->lblPagesNumber->TabIndex = 4;
this->lblPagesNumber->Text = L"Кількість сторінок";
//
// lblTheme
//
this->lblTheme->AutoSize = true;
this->lblTheme->Location = System::Drawing::Point(108, 128);
this->lblTheme->Name = L"lblTheme";
this->lblTheme->Size = System::Drawing::Size(31, 13);
this->lblTheme->TabIndex = 3;
this->lblTheme->Text = L"Тема";
//

```

```

// lblISBN
//
this->lblISBN->AutoSize = true;
this->lblISBN->Location = System::Drawing::Point(2, 94);
this->lblISBN->Name = L"lblISBN";
this->lblISBN->Size = System::Drawing::Size(137, 13);
this->lblISBN->TabIndex = 2;
this->lblISBN->Text = L"Вхідний номер документу";
//
// lblAuthor
//
this->lblAuthor->AutoSize = true;
this->lblAuthor->Location = System::Drawing::Point(101, 60);
this->lblAuthor->Name = L"lblAuthor";
this->lblAuthor->Size = System::Drawing::Size(38, 13);
this->lblAuthor->TabIndex = 1;
this->lblAuthor->Text = L"Автор";
//
// lblDocName
//
this->lblDocName->AutoSize = true;
this->lblDocName->Location = System::Drawing::Point(44, 26);
this->lblDocName->Name = L"lblDocName";
this->lblDocName->Size = System::Drawing::Size(95, 13);
this->lblDocName->TabIndex = 0;
this->lblDocName->Text = L"Назва документу";
//
// groupFinding
//
this->groupFinding->Controls->Add(this->btnFind);
this->groupFinding->Controls->Add(this->txtFindString);
this->groupFinding->Location = System::Drawing::Point(6, 6);
this->groupFinding->Name = L"groupFinding";
this->groupFinding->Size = System::Drawing::Size(702, 53);
this->groupFinding->TabIndex = 8;
this->groupFinding->TabStop = false;
this->groupFinding->Text = L"Ключові рядки для пошуку";
//
// btnFind
//
this->btnFind->Location = System::Drawing::Point(621, 19);
this->btnFind->Name = L"btnFind";
this->btnFind->Size = System::Drawing::Size(75, 23);
this->btnFind->TabIndex = 1;
this->btnFind->Text = L"Знайти";
this->btnFind->UseVisualStyleBackColor = true;
this->btnFind->Click += gcnew System::EventHandler(this,
&frmReader::btnFind_Click);
//
// txtFindString
//
this->txtFindString->Location = System::Drawing::Point(6, 19);
this->txtFindString->Name = L"txtFindString";
this->txtFindString->Size = System::Drawing::Size(609, 20);
this->txtFindString->TabIndex = 0;
//
// tabControl
//
this->tabControl->Controls->Add(this->tabFinding);
this->tabControl->Controls->Add(this->tabMyDocs);
this->tabControl->Controls->Add(this->tabHistory);
this->tabControl->Location = System::Drawing::Point(0, 27);
this->tabControl->Name = L"tabControl";
this->tabControl->SelectedIndex = 0;
this->tabControl->Size = System::Drawing::Size(725, 340);
this->tabControl->TabIndex = 14;
//
// tabMyDocs
//

```

```

this->tabMyDocs->Controls->Add(this->btnReturn);
this->tabMyDocs->Controls->Add(this->groupBox2);
this->tabMyDocs->Location = System::Drawing::Point(4, 22);
this->tabMyDocs->Name = L"tabMyDocs";
this->tabMyDocs->Padding = System::Windows::Forms::Padding(3);
this->tabMyDocs->Size = System::Drawing::Size(717, 314);
this->tabMyDocs->TabIndex = 1;
this->tabMyDocs->Text = L"Документи";
this->tabMyDocs->UseVisualStyleBackColor = true;
//
// btnReturn
//
this->btnReturn->Location = System::Drawing::Point(622, 285);
this->btnReturn->Name = L"btnReturn";
this->btnReturn->Size = System::Drawing::Size(85, 23);
this->btnReturn->TabIndex = 1;
this->btnReturn->Text = L"Повернути";
this->btnReturn->UseVisualStyleBackColor = true;
this->btnReturn->Click += gcnew System::EventHandler(this,
&frmReader::btnReturn_Click);
//
// groupBox2
//
this->groupBox 2-2->Controls->Add(this->lstMyDocs);
this->groupBox 2-2->Location = System::Drawing::Point(8, 6);
this->groupBox 2-2->Name = L"groupBox2";
this->groupBox 2-2->Size = System::Drawing::Size(699, 273);
this->groupBox 2-2->TabIndex = 0;
this->groupBox 2-2->TabStop = false;
this->groupBox 2-2->Text = L"Список документів користувача
інформаційної системи підприємства";
this->groupBox 2-2->Enter += gcnew System::EventHandler(this,
&frmReader::groupBox2_Enter);
//
// lstMyDocs
//
this->lstMyDocs->FormattingEnabled = true;
this->lstMyDocs->HorizontalScrollbar = true;
this->lstMyDocs->Location = System::Drawing::Point(6, 19);
this->lstMyDocs->Name = L"lstMyDocs";
this->lstMyDocs->Size = System::Drawing::Size(687, 251);
this->lstMyDocs->TabIndex = 0;
//
// tabHistory
//
this->tabHistory->Controls->Add(this->groupBox3);
this->tabHistory->Location = System::Drawing::Point(4, 22);
this->tabHistory->Name = L"tabHistory";
this->tabHistory->Size = System::Drawing::Size(717, 314);
this->tabHistory->TabIndex = 2;
this->tabHistory->Text = L"Історія";
this->tabHistory->UseVisualStyleBackColor = true;
//
// groupBox3
//
this->groupBox 3-3->Controls->Add(this->txtHistory);
this->groupBox 3-3->Location = System::Drawing::Point(8, 8);
this->groupBox 3-3->Name = L"groupBox3";
this->groupBox 3-3->Size = System::Drawing::Size(700, 303);
this->groupBox 3-3->TabIndex = 2;
this->groupBox 3-3->TabStop = false;
this->groupBox 3-3->Text = L"Історія перегляду документів";
//
// txtHistory
//
this->txtHistory->Location = System::Drawing::Point(6, 19);
this->txtHistory->Multiline = true;
this->txtHistory->Name = L"txtHistory";
this->txtHistory->ReadOnly = true;

```

```

        this->txtHistory->ScrollBars =
System::Windows::Forms::ScrollBars::Both;
        this->txtHistory->Size = System::Drawing::Size(688, 278);
        this->txtHistory->TabIndex = 0;
        //
        // menuHelp
        //
        this->menuHelp->DropDownItems->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) {this->menuHelpAbout});
        this->menuHelp->Name = L"menuHelp";
        this->menuHelp->Size = System::Drawing::Size(60, 20);
        this->menuHelp->Text = L"Довідка";
        //
        // menuHelpAbout
        //
        this->menuHelpAbout->Name = L"menuHelpAbout";
        this->menuHelpAbout->Size = System::Drawing::Size(166, 22);
        this->menuHelpAbout->Text = L"Про програму...";
        this->menuHelpAbout->Click += gcnew System::EventHandler(this,
&frmReader::menuHelpAbout_Click);
        //
        // menuFileExit
        //
        this->menuFileExit->Name = L"menuFileExit";
        this->menuFileExit->Size = System::Drawing::Size(152, 22);
        this->menuFileExit->Text = L"Вихід";
        this->menuFileExit->Click += gcnew System::EventHandler(this,
&frmReader::menuFileExit_Click);
        //
        // menuStrip
        //
        this->menuStrip->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(2) {this->menuFile, this->menuHelp});
        this->menuStrip->Location = System::Drawing::Point(0, 0);
        this->menuStrip->Name = L"menuStrip";
        this->menuStrip->Size = System::Drawing::Size(724, 24);
        this->menuStrip->TabIndex = 12;
        this->menuStrip->Text = L"menuStrip1";
        //
        // menuFile
        //
        this->menuFile->DropDownItems->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(2) {this->menuFileSave,
        this->menuFileExit});
        this->menuFile->Name = L"menuFile";
        this->menuFile->Size = System::Drawing::Size(45, 20);
        this->menuFile->Text = L"Файл";
        //
        // menuFileSave
        //
        this->menuFileSave->Name = L"menuFileSave";
        this->menuFileSave->Size = System::Drawing::Size(152, 22);
        this->menuFileSave->Text = L"Зберегти";
        this->menuFileSave->Click += gcnew System::EventHandler(this,
&frmReader::menuFileSave_Click);
        //
        // frmReader
        //
        this->AutoScaleDimensions = System::Drawing::Size(6, 13);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(724, 387);
        this->Controls->Add(this->statusStrip);
        this->Controls->Add(this->tabControl);
        this->Controls->Add(this->menuStrip);
        this->MaximizeBox = false;
        this->MinimizeBox = false;
        this->Name = L"frmReader";
        this->Text = L"Режим користувача інформаційної системи підприємства
- Система Управління Електронним Документообігом ";

```

```

        this->Load += gcnew System::EventHandler(this,
&frmReader::frmReader_Load);
        this->FormClosed += gcnew
System::Windows::Forms::FormClosedEventHandler(this,
&frmReader::frmReader_FormClosed);
        this->statusStrip->ResumeLayout(false);
        this->statusStrip->PerformLayout();
        this->tabFinding->ResumeLayout(false);
        this->groupBox 1-1->ResumeLayout(false);
        this->groupBox 1-1->PerformLayout();
        this->groupBoxFinding->ResumeLayout(false);
        this->groupBoxFinding->PerformLayout();
        this->tabControl->ResumeLayout(false);
        this->tabMyDocs->ResumeLayout(false);
        this->groupBox 2-2->ResumeLayout(false);
        this->tabHistory->ResumeLayout(false);
        this->groupBox 3-3->ResumeLayout(false);
        this->groupBox 3-3->PerformLayout();
        this->menuStrip->ResumeLayout(false);
        this->menuStrip->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
/*****
private: System::Void UpdateView()
{
    // Список не порожній?
    if(listView->Count != 0)
    {
        // Перевірити індекс поточного елемента
        if(ddCurrentIndex >= listView->Count)
        {
            // Установити покажчик на останній елемент
            ddCurrentIndex = listView->Count - 1;
        }

        // Поточний індекс менше нуля?
        if(ddCurrentIndex < 0)
        {
            // Установити індекс на перший елемент
            ddCurrentIndex = 0;
        }

        // Одержати характеристики документа й заповнити поля форми
        txtDocName->Text = ((CDoc^)listView[ddCurrentIndex])-
>GetName();
        txtAuthor->Text = ((CDoc^)listView[ddCurrentIndex])-
>GetAuthor();
        txtISBN->Text = ((CDoc^)listView[ddCurrentIndex])->GetISBN();
        txtTheme->Text = ((CDoc^)listView[ddCurrentIndex])-
>GetTheme();
        txtPagesNumber->Text =
Convert::ToString(((CDoc^)listView[ddCurrentIndex])->GetPages());

        // Документ є в наявності?
        if(((CDoc^)listView[ddCurrentIndex])->GetFreeNumber() > 0)
        {
            // Розблокувати кнопку запису
            btnRequest->Enabled = true;
        }
        else
        {
            // Заблокувати кнопку запису
            btnRequest->Enabled = false;
        }

        // Установити й вивести поточну позицію

```

```

txtPosition->Text = Convert::ToString(ddCurrentIndex + 1) + "
/ " +
        Convert::ToString(listView->Count);
}
else
{
    // Очистити всі поля
    txtDocName->Text = "";
    txtAuthor->Text = "";
    txtISBN->Text = "";
    txtTheme->Text = "";
    txtPagesNumber->Text = "";
    // Поточна позиція - 0
    txtPosition->Text = "0/0";
    // Установити індекс поточного елемента
    ddCurrentIndex = 0;
    // Вивести стан
    lblStatus->Text = "Немає записів для відображення";
}

// Вивести історію
txtHistory->Text = Reader->ViewLog();

// Одержати список документів користувача інформаційної системи
підприємства
ArrayList^ listTemp = gnew ArrayList();
listTemp = Reader->GetMyDocsList();

// Очистити список на формі
lstMyDocs->Items->Clear();
// Кількість документів не дорівнює нулю?
if(listTemp->Count != 0)
{
    // Заповнити список документів на формі
    for(Int32 i = 0; i < listTemp->Count; i++)
    {
        // Одержати характеристики документа й вивести їх у
        список
        lstMyDocs->Items->Add(Convert::ToString(i + 1) + ". " +
        +
            ((CDoc^)listTemp[i])->GetName() + ", " +
            ((CDoc^)listTemp[i])->GetAuthor() + ", вхідний
номер документа " +
            ((CDoc^)listTemp[i])->GetISBN());
    }
}

/*****/
private: System::Void frmReader_Load(System::Object^ sender, System::EventArgs^
e)
{
    // Створити об'єкт "Адміністратор"
    Worker = gnew CWorker();
    // Завантажити список документів
    Worker->LoadDocList();

    // Створити об'єкт класу "Користувач"
    Reader = gnew CReader(strUserName);
    // Завантажити список документів, що перебувають у користувача
інформаційної системи підприємства
    Reader->Load(Worker);

    // Вивести стан
    lblStatus->Text = "Для відображення списку потрібних документів
скористайтеся пошуком";
    // Обновити форму
    UpdateView();
}

/*****/

```

```

private: System::Void btnFind_Click(System::Object^ sender, System::EventArgs^
e)
{
    // Зabrати з рядку пошуку пробіли, перетворити до нижнього регістра
    й зберегти
    String^ strValue = txtFindString->Text->Trim()->ToLower();

    // Перевірка рядка
    if(String::IsNullOrEmpty(strValue))
    {
        // Рядок порожня, запропонувати вивести весь список
        if(MessageBox::Show("Рядок пошуку не завдань, вивести весь
список?",
                                "Система УЕД",
                                MessageBoxButtons::YesNo,
                                MessageBoxIcon::Question) ==
::DialogResult::Yes)
        {
            // Перейти на вивід списку
            goto OUT_LIST;
        }
        // Завершити роботу функції
        return;
    }

OUT_LIST: // Виконати функцію пошуку
listView = Worker->FindDoc(strValue);

    // Вивести стан
    lblStatus->Text = "Пошук завершено";
    // Оновити форму
    UpdateView();
}
/*****/
private: System::Void btnFirst_Click(System::Object^ sender, System::EventArgs^
e)
{
    // Установити індекс на перший елемент
    ddCurrentIndex = 0;
    // Оновити форму
    UpdateView();
}
/*****/
private: System::Void btnPrev_Click(System::Object^ sender, System::EventArgs^
e)
{
    // Установити індекс на попередній елемент
    ddCurrentIndex ---i;
    // Оновити форму
    UpdateView();
}
/*****/
private: System::Void btnNext_Click(System::Object^ sender, System::EventArgs^
e)
{
    // Установити індекс на наступний елемент
    ddCurrentIndex ++;
    // Оновити форму
    UpdateView();
}
/*****/
private: System::Void btnLast_Click(System::Object^ sender, System::EventArgs^
e)
{
    // Установити індекс на останній елемент
    ddCurrentIndex = listView->Count - 1;
    // Оновити форму
    UpdateView();
}
}

```

```

/*****/
private: System::Void txtPosition_KeyDown(System::Object^ sender,
System::Windows::Forms::KeyEventArgs^ e)
{
    // Натиснута клавіша "Enter"?
    if( e->KeyCode == Keys::Enter)
    {
        try
        {
            // Спробувати одержати введений номер документа для
перегляду
            ddCurrentIndex = Convert::ToInt32(txtPosition->Text) -
1;
        }
        catch(...)
        {
            // Якщо в процесі перетворення номера з рядка в число
            // виникла помилка, те ніяк на це не реагувати
        }

        // Обновити форму
        UpdateView();
    }
    else
    {
        // Текстове поле містить символ "/"?
        if(txtPosition->Text->IndexOf("/") != -1)
        {
            // Для початку введення нового номера позиції,
необхідно
            // очистити текстове поле
            txtPosition->Text = "";
        }
    }
}
/*****/
private: System::Void btnRequest_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Запросити документ, попередньо перетворивши значення вхідного
номера документа в текстовому
// поле до нижнього регістра й забравши бічні пробіли
if(Reader->RequireDoc(Worker, txtISBN->Text->ToLower()->Trim()-
>GetHashCode()))
{
    // Запит пройшов вдало, видалити документ із поточного списку
перегляду
    listView->RemoveAt(ddCurrentIndex);

    // Обновити форму
    UpdateView();
    // Вивести стан
    lblStatus->Text = "Документ успішно отримано";
}
else
{
    // Неможливо одержати документ, вивести повідомлення про
цьому
    lblStatus->Text = "Вибацте, але на даний момент немає жодного
документу по даному запиту";
}
}
/*****/
private: System::Void btnReturn_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // У списку документів є обраний елемент?
    if(lstMyDocs->SelectedIndex != -1)
    {

```

```

// Повернути обрану документ
Reader->ReleaseDoc(Worker, lstMyDocs->SelectedIndex);

// Обновити форму
UpdateView();
// Вивести стан
lblStatus->Text = "Документ виконано";
}
else
{
// Вивести повідомлення про помилку
lblStatus->Text = "Спочатку виберіть документ для виконання";
}
}
/*****/
private: System::Void menuFileSave_Click(System::Object^ sender,
System::EventArgs^ e)
{
// Збереження модифікованого списку документів у файл
Worker->SaveDocList();

// Збереження списку документів користувача інформаційної системи
підприємства
Reader->Save();
}
/*****/
private: System::Void menuFileExit_Click(System::Object^ sender,
System::EventArgs^ e)
{
// Закрити додаток
Application::Exit();
}
/*****/
private: System::Void frmReader_FormClosed(System::Object^ sender,
System::Windows::Forms::FormClosedEventArgs^ e)
{
// Закрити додаток
Application::Exit();
}
/*****/
private: System::Void menuHelpAbout_Click(System::Object^ sender,
System::EventArgs^ e)
{
// Відобразити форму з інформацією про програму
UsrsManager::frmAbout^ frmNew = gnew UsrsManager::frmAbout();
frmNew->Show();
}
/*****/
private: System::Void groupBox2_Enter(System::Object^ sender,
System::EventArgs^ e) {
}
private: System::Void lblStatus_Click(System::Object^ sender,
System::EventArgs^ e) {
}
};
}

```

```

<?xml version="1.0" encoding=" utf-8"?>
<root>
  <xsd:schema id="root" xmlns="" xmlns:xsd=" " xmlns:msdata="urn: schemas-
microsoft-com: xml-msdata">
    <xsd:import namespace=" " />
    <xsd:element name="root" msdata:IsDataSet="true">
      <xsd:complexType>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element name="metadata">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0" />
              </xsd:sequence>
              <xsd:attribute name="name" use="required" type="xsd:string" />
              <xsd:attribute name="type" type="xsd:string" />
              <xsd:attribute name="mimetype" type="xsd:string" />
              <xsd:attribute ref="xml:space" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="assembly">
            <xsd:complexType>
              <xsd:attribute name="alias" type="xsd:string" />
              <xsd:attribute name="name" type="xsd:string" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="data">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
                <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" use="required"
msdata:Ordinal="1" />
              <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
              <xsd:attribute name="mimetype" type="xsd:string"
msdata:Ordinal="4" />
              <xsd:attribute ref="xml:space" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="resheader">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" use="required" />
            </xsd:complexType>
          </xsd:element>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
  <resheader name="resmimetype">
    <value>text/ microsoft-resx</value>
  </resheader>
  <resheader name="version">
    <value>2.0</value>
  </resheader>
  <resheader name="reader">
    <value>System.Resources.ResXResourceReader, System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
  <resheader name="writer">
    <value>System.Resources.ResXResourceWriter, System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
</root>

```

Файл frmLogin.h - бібліотека для файлу frmLogin.resx

```

/*****
#pragma once
/*****
#include "frmWorker.h"
#include "frmReader.h"
#include "Usrs.h"
/*****
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
/*****
#define WORKER_LOGIN "worker"
/*****
namespace UsrsManager
{
    public ref class frmLogin : public System::Windows::Forms::Form
    {
/*****
    private:
        CUsrs^ Usrs; // Керуючий об'єкт класу "Система"
/*****
public:
    frmLogin(void)
    {
        InitializeComponent();
        //
        //Додавання цього коду в конструктор
        //
    }
/*****
protected: ~frmLogin()
    {
        if (components)
        {
            delete components;
        }
    }
/*****
    private: System::Windows::Forms::GroupBox^ groupBox;
    private: System::Windows::Forms::TextBox^ txtPassword;
    private: System::Windows::Forms::TextBox^ txtLogin;
    private: System::Windows::Forms::Label^ lblPassword;
    private: System::Windows::Forms::Label^ lblLogin;
    private: System::Windows::Forms::Button^ btnEnter;
    private: System::Windows::Forms::Button^ btnExit;
    private: System::Windows::Forms::Button^ btnDelete;
    private: System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Необхідний метод для підтримки Розроблювача - не модифікувати
    /// зміст цього методу з кодовим редактором.
    /// </summary>
    void InitializeComponent(void)
    {
        this->groupBox = (gcnew System::Windows::Forms::GroupBox());
        this->txtPassword = (gcnew System::Windows::Forms::TextBox());
        this->txtLogin = (gcnew System::Windows::Forms::TextBox());
        this->lblPassword = (gcnew System::Windows::Forms::Label());
        this->lblLogin = (gcnew System::Windows::Forms::Label());
        this->btnEnter = (gcnew System::Windows::Forms::Button());
        this->btnExit = (gcnew System::Windows::Forms::Button());
        this->btnDelete = (gcnew System::Windows::Forms::Button());
        this->groupBox->SuspendLayout();
        this->SuspendLayout();
    }
}
/*****

```

```

//
// groupBox
//
this->groupBox->Controls->Add(this->txtPassword);
this->groupBox->Controls->Add(this->txtLogin);
this->groupBox->Controls->Add(this->lblPassword);
this->groupBox->Controls->Add(this->lblLogin);
this->groupBox->Location = System::Drawing::Point(12, 11);
this->groupBox->Name = L"groupBox";
this->groupBox->Size = System::Drawing::Size(237, 80);
this->groupBox->TabIndex = 8;
this->groupBox->TabStop = false;
this->groupBox->Text = L"Параметри облікового запису";
//
// txtPassword
//
this->txtPassword->Location = System::Drawing::Point(57, 45);
this->txtPassword->Name = L"txtPassword";
this->txtPassword->PasswordChar = '*';
this->txtPassword->Size = System::Drawing::Size(174, 20);
this->txtPassword->TabIndex = 1;
//
// txtLogin
//
this->txtLogin->Location = System::Drawing::Point(57, 19);
this->txtLogin->Name = L"txtLogin";
this->txtLogin->Size = System::Drawing::Size(174, 20);
this->txtLogin->TabIndex = 0;
//
// lblPassword
//
this->lblPassword->AutoSize = true;
this->lblPassword->Location = System::Drawing::Point(6, 48);
this->lblPassword->Name = L"lblPassword";
this->lblPassword->Size = System::Drawing::Size(44, 13);
this->lblPassword->TabIndex = 1;
this->lblPassword->Text = L"Пароль";
//
// lblLogin
//
this->lblLogin->AutoSize = true;
this->lblLogin->Location = System::Drawing::Point(13, 19);
this->lblLogin->Name = L"lblLogin";
this->lblLogin->Size = System::Drawing::Size(33, 13);
this->lblLogin->TabIndex = 0;
this->lblLogin->Text = L"Логін";
//
// btnEnter
//
this->btnEnter->Location = System::Drawing::Point(12, 98);
this->btnEnter->Name = L"btnEnter";
this->btnEnter->Size = System::Drawing::Size(75, 23);
this->btnEnter->TabIndex = 2;
this->btnEnter->Text = L"Вхід";
this->btnEnter->UseVisualStyleBackColor = true;
this->btnEnter->Click += gcnew System::EventHandler(this,
&frmLogin::btnEnter_Click);
//
// btnExit
//
this->btnExit->Location = System::Drawing::Point(174, 98);
this->btnExit->Name = L"btnExit";
this->btnExit->Size = System::Drawing::Size(75, 23);
this->btnExit->TabIndex = 3;
this->btnExit->Text = L"Вихід";
this->btnExit->UseVisualStyleBackColor = true;
this->btnExit->Click += gcnew System::EventHandler(this,
&frmLogin::btnExit_Click);
//

```

```

        // btnDelete
        //
        this->btnDelete->Location = System::Drawing::Point(93, 98);
        this->btnDelete->Name = L"btnDelete";
        this->btnDelete->Size = System::Drawing::Size(75, 23);
        this->btnDelete->TabIndex = 4;
        this->btnDelete->Text = L"Видалення";
        this->btnDelete->UseVisualStyleBackColor = true;
        this->btnDelete->Click += gnew System::EventHandler(this,
&frmLogin::btnDelete_Click);
        //
        // frmLogin
        //
        this->AutoScaleDimensions = System::Drawing::Size(6, 13);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(260, 129);
        this->Controls->Add(this->groupBox);
        this->Controls->Add(this->btnEnter);
        this->Controls->Add(this->btnExit);
        this->Controls->Add(this->btnDelete);
        this->MaximizeBox = false;
        this->MinimizeBox = false;
        this->Name = L"frmLogin";
        this->Text = L"Вхід у систему УЕД";
        this->groupBox->ResumeLayout(false);
        this->groupBox->PerformLayout();
        this->ResumeLayout(false);
    }
#pragma endregion
/*****
private: System::Boolean PrepareLoginPassword()
    {
        // Перетворити ім'я до нижнього регістра
        txtLogin->Text = txtLogin->Text->ToLower();
        // Забрати в поле ім'я бічні пробіли
        txtLogin->Text = txtLogin->Text->Trim();

        // Поле для введення ім'я не містить значення?
        if(String::IsNullOrEmpty(txtLogin->Text))
        {
            // Видати повідомлення про те, поле не містить значення
            MessageBox::Show("Введіть ім'я користувача
інформаційної системи підприємства.", "Usrs Manager",
                MessageBoxButtons::OK, MessageBoxIcon::Warning);
            // Вийти
            return false;
        }

        // Перевірити введені символи
        for(Int32 i = 0; i < txtLogin->Text->Length; i++)
        {
            // Перевіряється символ
            Char chTest = txtLogin->Text[i];

            // Перевірка на коректність
            if(!(chTest >= 'A' && chTest <= 'Z') &&
                !(chTest >= 'a' && chTest <= 'z') &&
                !(chTest >= 'А' && chTest <= 'Я') &&
                !(chTest >= 'а' && chTest <= 'я') &&
                !(chTest == '_'))
            {
                // Видати повідомлення про те, поле містить
                неприпустимий символ
                MessageBox::Show("Логін містить неприпустимі
                символи!",
                    "Usrs Manager", MessageBoxButtons::OK,
                    MessageBoxIcon::Warning);
            }
        }
    }

```

```

        // Вийти
        return false;
    }
}

return true;
}
/*****
private: System::Void btnEnter_Click(System::Object^ sender, System::EventArgs^ e)
{
    // Підготовка значень
    if(!PrepareLoginPassword())
    {
        return;
    }

    // Уміст поля ім'я - логін адміністратора інформаційної
    системи підприємства?
    if(txtLogin->Text == WORKER_LOGIN)
    {
        // Увійти в обліковий запис адміністратора
        інформаційної системи підприємства
        if(Usrs->WorkerLogin(txtPassword->Text))
        {
            // Відобразити форму адміністратора інформаційної
            системи підприємства
            UsrsManager::frmWorker^ frmNew = gcnew
            UsrsManager::frmWorker;
            frmNew->Show();
        }
        else
        {
            // Вхід неможливий, закінчити роботу функції
            return;
        }
    }
    else
    {
        // Увійти в обліковий запис користувача інформаційної
        системи підприємства
        if(Usrs->ReaderLogin(txtLogin->Text, txtPassword->Text))
        {
            // Відобразити форму для користувача
            інформаційної системи підприємства
            UsrsManager::frmReader^ frmNew =
            gcnew UsrsManager::frmReader(txtLogin->Text);
            frmNew->Show();
        }
        else
        {
            // Вхід неможливий, закінчити роботу функції
            return;
        }
    }

    // Сховати форму входу
    this->Hide();
}
/*****
private: System::Void btnExit_Click(System::Object^ sender, System::EventArgs^ e)
{
    // Закрити додаток
    Application::Exit();
}
/*****

```

```

private: System::Void btnDelete_Click(System::Object^ sender,
System::EventArgs^ e)
{
    // Одержати підтверження на видалення
    if(MessageBox::Show("Ви дійсно хочете зробити видалення?",
        "Usrs Manager", MessageBoxButtons::YesNo,
        MessageBoxIcon::Information) == ::DialogResult::Yes)
    {
        // Підготовка значень логіна й пароля
        if(!PrepareLoginPassword())
        {
            // Якщо виникли помилки, то завершити роботу
            return;
        }

        // Уміст поля ім'я - логін адміністратора інформаційної
        системи підприємства?
        if(txtLogin->Text == WORKER_LOGIN)
        {
            // Обліковий запис адміністратора
            інформаційної системи підприємства видалити не можна
            MessageBox::Show("Неможливо видалити
            обліковий запис адміністратора інформаційної системи підприємства!",
                "Usrs Manager", MessageBoxButtons::OK,
                MessageBoxIcon::Error);
        }
        else
        {
            // Видалити обліковий запис користувача
            інформаційної системи підприємства
            if(Usrs->RemoveReader(txtLogin->Text,
            txtPassword->Text))
            {
                // Видалення пройшло успішно
                MessageBox::Show("Обліковий запис успішно
                вилучений",
                    "Usrs Manager",
                    MessageBoxButtons::OK,
                    MessageBoxIcon::Information);
            }
            else
            {
                // При видаленні відбулася помилка
                MessageBox::Show("Неможливо видалити
                обліковий запис",
                    "Usrs Manager",
                    MessageBoxButtons::OK,
                    MessageBoxIcon::Error);
            }
        }
    }
}

/*****
*/;
}

```

Файл Doc.cpp - робота з документами

```

/*****
#include "StdAfx.h"
#include "Doc.h"
/*****
using namespace System;
/*****
// NAME:          CDoc
// DESCRIPTION:   Конструктор класу
// INPUT:         N/D
// OUTPUT:        N/D
/*****
CDoc::CDoc(void)
{
    ddTotalNumber = 0;        // Загальна кількість екземплярів дорівнює 0
    ddFreeNumber = 0;        // Кількість вільних екземплярів дорівнює 0
}
/*****
// NAME:          GetName
// DESCRIPTION:   Функція одержання назви документів
// INPUT:         N/D
// OUTPUT:        Назва документа
/*****
String^ CDoc::GetName()
{
    return strName;
}
/*****
// NAME:          SetName
// DESCRIPTION:   Функція установки назви документів
// INPUT:         strValue - назва документа
// OUTPUT:        TRUE - Назва документа встановлена
//               FALSE - Назва не встановлена, рядок-параметр порожня
/*****
Boolean CDoc::SetName(String^ strValue)
{
    // Перевірити, чи не порожній рядок
    if(!String::IsNullOrEmpty(strValue))
    {
        // Установити значення
        strName = strValue;
        return true;
    }
    else
    {
        return false;
    }
}
/*****
// NAME:          GetAuthor
// DESCRIPTION:   Функція одержання автора документа
// INPUT:         N/D
// OUTPUT:        Автор документи
/*****
String^ CDoc::GetAuthor()
{
    // Повернути значення
    return strAuthor;
}
/*****
// NAME:          SetAuthor
// DESCRIPTION:   Функція установки автора документа
// INPUT:         strValue - ім'я автора документа
// OUTPUT:        N/D
/*****
void CDoc::SetAuthor(String^ strValue)

```

```

{
    // Установити значення
    strAuthor = strValue;
}
/*****
// NAME:          GetISBN
// DESCRIPTION:   Функція одержання вхідний номер документа
// INPUT:         N/D
// OUTPUT:        вхідний номер документа
*****/
String^ CDoc::GetISBN()
{
    // Повернути значення
    return strISBN;
}
/*****
// NAME:          SetISBN
// DESCRIPTION:   Функція установки вхідний номер документа
// INPUT:         strValue - вхідний номер документа
// OUTPUT:        TRUE - вхідний номер документа встановлений
//               FALSE - вхідний номер документа не встановлений, рядок-
параметр порожня
*****/
Boolean CDoc::SetISBN(String^ strValue)
{
    // Перевірити, чи не порожній рядок
    if(!String::IsNullOrEmpty(strValue))
    {
        // Установити значення
        strISBN = strValue;
        // Обчислити й установити хеш-код вхідний номер документа
        ddISBNHash = strISBN->GetHashCode();
        return true;
    }
    else
    {
        return false;
    }
}
/*****
// NAME:          GetTheme
// DESCRIPTION:   Функція одержання теми
// INPUT:         N/D
// OUTPUT:        Тема документи
*****/
String^ CDoc::GetTheme()
{
    // Повернути значення
    return strTheme;
}
/*****
// NAME:          SetTheme
// DESCRIPTION:   Функція установки теми
// INPUT:         strValue - тема документи
// OUTPUT:        N/D
*****/
void CDoc::SetTheme(String^ strValue)
{
    // Установити значення
    strTheme = strValue;
}
/*****
// NAME:          GetPages
// DESCRIPTION:   Функція одержання кількості сторінок
// INPUT:         N/D
// OUTPUT:        Кількість сторінок у книзі
*****/
Int32 CDoc::GetPages()
{

```

```

        // Повернути значення
        return ddPages;
    }
/*****/
// NAME:          SetPages
// DESCRIPTION:   Функція установки кількості сторінок
// INPUT:         ddValue - кількість сторінок
// OUTPUT:        TRUE - кількість сторінок установлена
//               FALSE - кількість сторінок установлена, неприпустиме
значення аргументу
/*****/
Boolean    CDoc::SetPages(Int32 ddValue)
{
    // Параметр має припустиме значення?
    if(ddValue >= 0)
    {
        // Установити значення
        ddPages = ddValue;
        return true;
    }
    else
    {
        return false;
    }
}
/*****/
// NAME:          GetTotalNumber
// DESCRIPTION:   Функція одержання загальної кількості екземплярів
// INPUT:         N/D
// OUTPUT:        Загальна кількість екземплярів документи
/*****/
Int32 CDoc::GetTotalNumber()
{
    // Повернути значення
    return ddTotalNumber;
}
/*****/
// NAME:          SetTotalNumber
// DESCRIPTION:   Функція установки загальне кількості екземплярів
// INPUT:         ddValue - загальна кількість екземплярів
// OUTPUT:        TRUE - кількість екземплярів установлена
//               FALSE - кількість екземплярів установлена, негативне
значення
//               аргумента або аргумент перевищує кількість екземплярів,
що перебуває в
//               користуваців
/*****/
Boolean    CDoc::SetTotalNumber(Int32 ddValue)
{
    // Перевірка параметра
    if(ddValue >= 0 && ddValue >= ddTotalNumber - ddFreeNumber)
    {
        // Установити значення
        ddTotalNumber = ddValue;
        return true;
    }
    else
    {
        return false;
    }
}
/*****/
// NAME:          GetFreeNumber
// DESCRIPTION:   Функція одержання кількості вільних екземплярів
// INPUT:         N/D
// OUTPUT:        Кількість вільних екземплярів
/*****/
Int32 CDoc::GetFreeNumber()
{

```

```

// Повернути значення
return ddFreeNumber;
}
/*****/
// NAME:          SetFreeNumber
// DESCRIPTION:   Функція установки кількості вільних екземплярів
// INPUT:         ddValue - кількість вільних екземплярів
// OUTPUT:        TRUE - кількість вільних екземплярів установлене
//                FALSE - значення параметра перевищує загальна кількість
екземплярів
/*****/
Boolean    CDoc::SetFreeNumber(Int32 ddValue)
{
    // Порівняння параметра із загальною кількістю екземплярів
    if(ddTotalNumber >= ddValue)
    {
        // Установити значення
        ddFreeNumber = ddValue;
        return true;
    }
    else
    {
        // ddFreeNumber = ddTotalNumber;
        return false;
    }
}
/*****/
// NAME:          GetISBNHash
// DESCRIPTION:   Функція одержання хеш-коду від вхідного номера документа
// INPUT:         N/D
// OUTPUT:        Хеш-      Код вхідний номер документа
/*****/
Int32 CDoc::GetISBNHash()
{
    // Повернути хеш-значення вхідного номера документа
    return ddISBNHash;
}
/*****/
// NAME:          DecFreeNumber
// DESCRIPTION:   Функція зменшення кількості вільних екземплярів
// INPUT:         N/D
// OUTPUT:        TRUE - кількість вільних екземплярів декрементовано
//                FALSE - зменшення неприпустимо, кількість вільних екземплярів
дорівнює 0
/*****/
Boolean    CDoc::DecFreeNumber()
{
    // Кількість вільних екземплярів не дорівнює 0?
    if(ddFreeNumber != 0)
    {
        // Зменшити значення
        ddFreeNumber ---i;
        return true;
    }
    else
    {
        return false;
    }
}
/*****/
// NAME:          IncFreeNumber
// DESCRIPTION:   Функція збільшення кількості вільних екземплярів
// INPUT:         N/D
// OUTPUT:        TRUE - кількість вільних екземплярів інкрементовано
//                FALSE - значення неприпустимо, кількість вільних екземплярів
дорівнює
//                загальній кількості екземплярів
/*****/
Boolean    CDoc::IncFreeNumber()

```

```
{
    // Кількість вільних екземплярів не дорівнює загальній кількості
    екземплярів?
    if(ddFreeNumber != ddTotalNumber)
    {
        // Збільшити значення
        ddFreeNumber ++;
        return true;
    }
    else
    {
        return false;
    }
}
/*****/
```

КБПЗ_2023

Файл Doc.h - бібліотека для файлу Doc. cpp

```

#pragma once
/*****
using namespace System;
/*****
ref class CDoc // Клас "Документ"
{
private:
    String^ strName; // Назва
    String^ strAuthor; // Автор
    String^ strISBN; // вхідний номер документа
    String^ strTheme; // Тема
    Int32 ddPages; // Кількість сторінок
    Int32 ddTotalNumber; // Загальна кількість
екземплярів
    Int32 ddFreeNumber; // Кількість вільних
(перебувають в // системі) екземплярів
    Int32 ddISBNHash; // Значення хеш-функції від
вхідний номер документа

public:
    CDoc(void); // Конструктор класу

    String^ GetName(); // Функція одержання назви
документів
    String^ GetAuthor(); // Функція одержання автора
документа
    String^ GetISBN(); // Функція одержання
вхідний номер документа
    String^ GetTheme(); // Функція одержання теми
    Int32 GetPages(); // Функція одержання
кількості сторінок
    Int32 GetTotalNumber(); // Функція одержання
загальне кількості екземплярів
    Int32 GetFreeNumber(); // Функція одержання
кількості вільних екземплярів
    Int32 GetISBNHash(); // Функція одержання хеша-
коду від вхідний номер документа

    Boolean SetName(String^ strValue); // Функція установки назви
документів
    void SetAuthor(String^ strValue); // Функція установки автора
документа
    Boolean SetISBN(String^ strValue); // Функція установки вхідний
номер документа
    void SetTheme(String^ strValue); // Функція установки теми
    Boolean SetPages(Int32 ddValue); // Функція установки
кількості сторінок
    Boolean SetTotalNumber(Int32 ddValue); // Функція установки
загальне кількості екземплярів
    Boolean SetFreeNumber(Int32 ddValue); // Функція установки кількості
вільних екземплярів

    Boolean DecFreeNumber(); // Функція зменшення
кількості вільних екземплярів
    Boolean IncFreeNumber(); // Функція збільшення
кількості вільних екземплярів
};
/*****

```

Файл Usrs.cpp - робота з обліковими записами

```

/*****/
#include "StdAfx.h"
#include "Usrs.h"
/*****/
using namespace System;
using namespace System::IO;
using namespace System::Windows::Forms;
using namespace System::Collections;
/*****/
#define USER_LIST_FILE          "USERLIST.TXT"
#define WORKER_NAME              "ADMINISTRATOR"
#define WORKER_FILE_EXTENSION ".LMW"
#define READER_FILE_EXTENSION ".LMR"
/*****/
// NAME:          CUsrcs
// DESCRIPTION:   Конструктор класу
// INPUT:         N/D
/*****/
CUsrcs::CUsrcs(void)
{
}
/*****/
// NAME:          WorkerLogin
// DESCRIPTION:   Функція входу й завантаження даних про адміністратора
інформаційної системи підприємства
// INPUT:         strPassword - пароль адміністратора інформаційної системи
підприємства
/*****/
Boolean CUsrcs::WorkerLogin(System::String ^strPassword)
{
    // Перевірити існування файлу зі списком імен і паролів
    if(File::Exists(USER_LIST_FILE))
    {
        // Відкрити файл
        StreamReader^ srUserList = gcnew StreamReader(USER_LIST_FILE);

        // Завантажити перший рядок з ім'ям адміністратора інформаційної системи
підприємства
        String^ strLogin = srUserList->ReadLine();

        // Зрівняти введене ім'я з ім'ям у файлі
        if(String::Compare(strLogin, WORKER_NAME) != 0)
        {
            // Логіни не збігаються
            MessageBox::Show("Обліковий запис не знайдено.");
            // Повернути помилку
            return false;
        }

        // Завантажити другий рядок з паролем адміністратора інформаційної системи
підприємства
        String^ strPasswordInFile = srUserList->ReadLine();

        // Закрити файл
        srUserList->Close();

        // Зрівняти введений пароль із паролем у файлі
        if(String::Compare(strPassword, strPasswordInFile) != 0)
        {
            // Паролі не збігаються
            MessageBox::Show("Невірний пароль!", "Система УЕД",
                MessageBoxButtons::OK, MessageBoxIcon::Error);
            // Повернути помилку
            return false;
        }
    }
}

```

```

// Вхід виконаний
return true;
}
else
{
// Створити файл зі списком користувачів
StreamWriter^ swUserList = gcnew StreamWriter(USER_LIST_FILE);

// Зберегти ім'я користувача інформаційної системи підприємства й пароль
swUserList->WriteLine(WORKER_NAME);
swUserList->WriteLine(strPassword);

// Закрити файл
swUserList->Close();
}

return true;
}
/*****/
// NAME:      ReaderLogin
// DESCRIPTION: Функція входу й завантаження даних про користувача
інформаційної системи підприємства
// INPUT:     strName - ім'я облікового запису користувача інформаційної
системи підприємства
//           strPassword - пароль для входу
/*****/
Boolean CUsers::ReaderLogin(System::String^ strName, System::String^ strPassword)
{
// Прапор удалого входу
Boolean IsLoginValid = false,
IsPasswordValid = false;

// Перевірка існування файлу зі списком користувачів
if(!File::Exists(USER_LIST_FILE))
{
// Видати повідомлення про помилку
MessageBox::Show("Помилка! Для початку роботи необхідно створити обліковий
запис.",
"Система УЕД", MessageBoxButtons::OK, MessageBoxIcon::Error);
// Повернути помилку
return false;
}

// Відкрити файл
StreamReader^ srUserList = gcnew StreamReader(USER_LIST_FILE);

do
{
// Завантажити ім'я й пароль
String^ strNameInFile = srUserList->ReadLine();
String^ strPasswordInFile = srUserList->ReadLine();

if(String::Compare(strName, strNameInFile) == 0)
{
// Ім'я користувача інформаційної системи підприємства знайдене
IsLoginValid = true;

// Зрівняти введений пароль із паролем у файлі
if(String::Compare(strPassword, strPasswordInFile) == 0)
{
// Пароль збігається
IsPasswordValid = true;
// Закінчити цикл пошуку
break;
}
}
else
{
// Ім'я знайдене, але пароль не збігається
IsPasswordValid = false;
}
}
}
}

```

```

        // Закінчити цикл пошуку
        break;
    }
}
while(srUserList->EndOfStream == false);

// Закрити файл
srUserList->Close();

// Якщо логін збігається, а пароль не збігається, то перервати виконання
if(IsLoginValid && !IsPasswordValid)
{
    // Повідомити користувача інформаційної системи підприємства
    MessageBox::Show("Невірний пароль!", "Система УЕД",
        MessageBoxButtons::OK, MessageBoxIcon::Error);
    // Повернути помилку
    return false;
}

// Якщо ні логін, ні пароль не збігаються, то створити новий обліковий запис
if(!IsLoginValid && !IsPasswordValid)
{
    // Запропонувати створити користувачем новий обліковий запис
    if (MessageBox::Show("Облікового запису з таким ім'ям користувача
інформаційної системи підприємства не знайдено, створити новий обліковий
запис?",
        "Система УЕД", MessageBoxButtons::YesNo,
        MessageBoxIcon::Question) == DialogResult::Yes)
    {
        // Створити новий обліковий запис
        if(!AddReader(strName, strPassword))
        {
            MessageBox::Show("При створенні нового облікового запису виникла
помилка", "Система УЕД", MessageBoxButtons::OK,
                MessageBoxIcon::Error);

            // Не виконувати вхід
            return false;
        }
    }
    else
    {
        // Не виконувати вхід
        return false;
    }
}

return true;
}
/*****/
// NAME:          AddReader
// DESCRIPTION:   Функція додавання нового облікового запису користувача
інформаційної системи підприємства
// INPUT:        strName - ім'я користувача інформаційної системи підприємства
//              strPassword - пароль
/*****/
Boolean CUsers::AddReader(System::String ^strName, System::String ^strPassword)
{
    // Створити файловий потік
    FileStream^ fsUserList = gcnew FileStream(USER_LIST_FILE, FileMode::Append);
    // Відкрити файл зі списком користувачів
    StreamWriter^ swUserList = gcnew StreamWriter(fsUserList);

    // Записати ім'я й пароль
    swUserList->WriteLine(strName + Convert::ToChar(13) + Convert::ToChar(10) +
strPassword);

    // Закрити файл

```

```

swUserList->Close();
// Закрити файловий потік
fsUserList->Close();

// Створити файловий потік
FileStream^ fsReaderFile = gcnew FileStream(strName->ToUpper() +
READER_FILE_EXTENSION,
                                           FileMode::CreateNew);

// Відкрити файл користувача інформаційної системи підприємства
StreamWriter^ swReaderFile = gcnew StreamWriter(fsReaderFile);
// Кількість документів у користувача інформаційної системи підприємства
дорівнює 0
swReaderFile->WriteLine("0");
// Закрити файл
swReaderFile->Close();
// Закрити файловий потік
fsReaderFile->Close();

return true;
}
/*****
// NAME:          RemoveReader
// DESCRIPTION:   Функція видалення облікового запису користувача інформаційної
системи підприємства
// INPUT:        strName - ім'я облікового запису
//               strPassword - пароль
*****/
Boolean CUsers::RemoveReader(System::String ^strName, System::String
^strPassword)
{
    // Створити файловий потік
    FileStream^ fsUserList = gcnew FileStream(USER_LIST_FILE, FileMode::Open);

    // Пошук позиції рядка із заданим ім'ям користувача інформаційної системи
підприємства

    // Тимчасовий рядок
    String^ strTmp = "";
    // Лічильник символів
    Int64 dqCounter = 0;
    // Непарні рядки - рядка з іменами
    Boolean IsLoginString = true;
    // Цикл пошуку
    do
    {
        // Завантажити байт
        Char dbChar = fsUserList->ReadByte();
        // Якщо байт службовий або досягнуть кінець файлу
        if(dbChar == 0x0D || fsUserList->Position == fsUserList->Length)
        {
            // Якщо рядок ім'я
            if(IsLoginString)
            {
                // Зрівняти ім'я користувача інформаційної системи підприємства з
виділеним рядком
                if(String::Compare(strTmp, strName) == 0)
                {
                    // Скорегувати покажчик
                    dqCounter -= strTmp->Length;
                    // Перервати виконання циклу
                    break;
                }
            }

            // Обнулити рядок
            strTmp = "";
            // Інвертувати прапор рядка ім'я
            IsLoginString = !IsLoginString;
            // Якщо не кінець файлу

```

```
if(fsUserList->Position + 1 != fsUserList->Length)
{
    // Пропустити службовий символ
    fsUserList->ReadByte();
    dqCounter++;
}
else
{
    // Додати лічений символ до тимчасового рядка
    strTmp += dbChar;
}
// Збільшити покажчик
dqCounter++;
}
// Продовжувати, поки не кінець файлу
while(fsUserList->Position != fsUserList->Length);

// Переміститися у файлі до знайденого рядка
fsUserList->Position = dqCounter;
// Заповнити ім'я пробілами
for(Int32 i = 0; i < strTmp->Length; i++)
{
    fsUserList->WriteByte(' ');
}

// Закрити файловий потік
fsUserList->Close();
// Видалити файл користувача інформаційної системи підприємства
File::Delete(strName + READER_FILE_EXTENSION);

return true;
}
/*****/
```

Файл Usrs.h - бібліотека для файлу Usrs.cpp

```

    /*****
    /
    #pragma once
    /*****/
    using namespace System;
    /*****/
    ref class CUsrs // Клас реєстрації, видалення й авторизації користувачів
    {
    public:
        // Конструктор класу
        CUsrs(void);

        // Функція додавання нового облікового запису користувача інформаційної
        системи підприємства
        Boolean AddReader(System::String^ strName, System::String^ strPassword);
        // Функція видалення облікового запису користувача інформаційної системи
        підприємства
        Boolean RemoveReader(System::String^ strName, System::String^ strPassword);

        // Функція входу й завантаження даних про користувача інформаційної системи
        підприємства
        Boolean ReaderLogin(System::String^ strName, System::String^ strPassword);
        // Функція входу й завантаження даних про адміністратора інформаційної
        системи підприємства
        Boolean WorkerLogin(System::String^ strName);
    };
    /*****/

```