

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
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 першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“ Програмне забезпечення інформаційної каталожної системи ”**

Виконав здобувач вищої освіти  
IV курсу, групи КМ-19  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Литвиненко Д.О.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Пархоменко Ю.М.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.  
Рецензент \_\_\_\_\_

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний

Кафедра Кібербезпеки та програмного забезпечення

Рівень вищої освіти бакалавр

Галузь знань 12 “Інформаційні технології”

Спеціальність 123 “Комп’ютерна інженерія”

Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

“ ” 2023 року

**ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

*Литвиненку Дмитру Олеговичу*

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення інформаційної каталожної системи

2. Керівник роботи Пархоменко Юрій Михайлович, канд. техн. наук, доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 5 ” січня 2023 року № 10-02

3. Строк подання роботи до захисту 22.05.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи. *Метою роботи є розробка програмного забезпечення інформаційної каталожної системи*

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

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію*

*6. Висновки.*

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

*Структурна схема системи* 1 аркуш

*Функціональна схема системи* 1 аркуш

*Діаграма процесів* 1 аркуш

*Блок-схема алгоритму роботи додатку* 2 аркуша

7. Дата видачі завдання «  »    20   р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих каталожних систем	10.03.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2023 р.	
3.	Розробка моделі компонента	20.03.2023 р.	
4.	Розробка структур даних	25.03.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2023 р.	
6.	Програмування алгоритмів	10.04.2023 р.	
7.	Оформлення ПЗ	17.05.2023 р.	
8.	Попередній захист роботи	22.05.2023 р.	

Дата видачі завдання  
«  »    2023р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
«  »    2023р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Литвиненко Д.О. Програмне забезпечення інформаційної каталожної системи. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.**

В даній бакалаврській роботі розроблена інформаційна каталожна система.

Метою роботи є дослідження та програмна реалізація інформаційної каталожної системи.

Об'єктом дослідження є процес ведення електронного каталогу.

Предметом дослідження є методи каталогізації та систематизації.

Методи дослідження базуються на методах обробки даних, методах каталогізації та систематизації, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація інформаційної каталожної системи.

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 призначена для виконання під управлінням операційної системи сімейства Windows.

Програму розроблено в середовищі Delphi 7.

**Ключові слова:** комп'ютерна інженерія, інформація, інформаційна каталожна система.

## ABSTRACT

**Litvinenko D.O. Information catalog system software.  
123 Computer Engineering. Central Ukrainian National Technical University.  
Kropyvnytskyi. 2023**

An information catalog system has been developed in this bachelor's thesis.

The purpose of the work is the study and software implementation of information catalog system.

The object of the study is the process of maintaining an electronic catalog.

The subject of the study is methods of cataloging and systematization.

The research methods are based on data processing methods, methods of cataloging and systematization, and the methods of software development.

The result of the work is software implementation of information catalog system.

The analysis of the existing hardware and software methods was carried out in the process of working on the software model. All components of the developed software are fully described.

A user-friendly interface has been developed. The instructions for working with the software are given.

The software is designed to run on Windows operating system.

The software was developed in Delphi 7.

**Key words:** computer engineering, information, of information catalog system.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП .....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ .....	5
1.1 Призначення системи.....	5
1.2 Область застосування .....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ.....	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти) .....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування .....	18
2.3 Розгорнута постановка завдання .....	19
3 ОПИС І ОБґРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	21
3.1 Опис функціонування системи .....	21
3.2 Розробка структурної схеми.....	24
3.3 Розробка функціональної схеми .....	27
3.4 Розробка діаграми процесів .....	29
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	31

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>			
<b>Ви</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	<b>Програмне забезпечення інформаційної каталожної системи</b>	<b>Літ.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<b>Розроб.</b>		<i>Литяченко Д.О.</i>				<b>М</b>	1	55
<b>Перев.</b>		<i>Пархоменко Ю.М.</i>						
<b>Н.контр.</b>		<i>Гермак В.С.</i>						
<b>Затв.</b>		<i>Смірнов О.А.</i>						

4.1 Розробка блок-схем та опис алгоритмів функціонування системи .....	31
4.2 Захист розробленого програмного забезпечення.....	43
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	45
6 ОСНОВНІ ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52

Кафедра \_ КБПЗ \_ 2023 рік

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		2





# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

## 1.1. Призначення системи

Інформаційний довідково-пошуковий апарат ресурсів може бути представлений традиційним картковим (паперовим) і електронним каталогами, а також різними картотеками. З його допомогою здійснюється запит на спрощення доступу до необхідних об'єктів (документам, інструментам, запчастинам і так далі) щоб отримати їх для роботи. Він значно полегшує і прискорює отримання необхідного ресурсу, якщо в заявці точно вказані усі дані на його знаходження (особливо важливий шифр, що показує місце зберігання необхідного).

Саме слово каталог - слово грецького походження і означає "збори", "зосередження" чого-небудь (наприклад: перелік творів і інших документів, наявних у відділі інформаційних ресурсів, складений за певним принципом що розкриває склад і зміст фонду).

Каталоги бувають найрізноманітніші: за призначенням - службові або загального призначення (наприклад читацькі); за способом угруповання матеріалу - алфавітні, систематичні; по видах об'єктів, що каталогізуються (наприклад відносно носіїв документів: CD-диски, відеокасети, тощо).

У багатьох сучасних бібліотеках традиційний каталог - картотека витісняється комп'ютерним каталогом (електронним), що виконує аналогічні довідково-бібліографічні і пошукові функції. Алфавітний каталог, систематичний каталог, картотека персоналій, картотека заголовків художніх творів і картотека прози - знаходяться в читальному залі відділу інформаційних ресурсів. Систематична картотека статей і тематичні картотеки розташовані в читальній залі періодичних видань (зазвичай скористатися електронним каталогами можна в приміщенні бібліотека або за допомогою мережевих ресурсів - спеціалізованих сайтів) [1, 2].

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Основним призначенням каталожних систем є одержання інформаційних відомостей про об'єкт пошуку, на основі обробки метаданих. Реалізація основної задачі передбачає вирішення задачі підготовки метаданих для організації пошуку, яка включає наступний функціонал: початкове введення даних, їх збереження та супроводження, що враховує можливість їх змін з плином часу.

До того ж слід зазначити, що для забезпечення основної цілі, метадані мають бути жорстко структуровані та стандартизовані, хоч це і не є на пряму призначенням або функціональними можливостями системи, та все ж це роботи по забезпеченню працездатності системи.

## 1.2 Область застосування

Каталожні системи можуть використовуватися практично в усіх сферах людської діяльності, де вимагається пошук, який будуються на основі відбору з множини подібних об'єктів, що розміщено в систематизованому порядку, який відображає термінологічну систему понять в області каталогізації: у органах державного управління і місцевого самоврядування; у військовій справі; у медицині; у всіх наявних галузях промисловості, будівництві та транспорті; у торгівлі; у туризмі; у юриспруденції; у банківській справі та страхуванні; науково-дослідницькими та освітніми установами; бібліотеками; архівами та фізичними особами.

Зазначимо, що класичним застосуванням для каталожних систем вважається бібліотечна та архівна справа.

Таким чином, виходячи з вищенаведеного, дослідження та програмна реалізація каталожної системи є актуальною задачею, яка потребує вирішення у даній бакалаврській роботі.

					ВКРБ-123.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти)

У сучасному інформаційному світі так звані електронні каталоги це технологія обліку об'єктів, яка заснована на каталогізації і створенні масивів вторинних інформаційних ресурсів (метаданих). Необхідність у вторинних інформаційних ресурсах пояснюється різноманітністю первинної інформації, що підлягає систематизації для організації пошуку, яка нині дуже широко застосовна і відповідно представлена безліччю різноманітних рішень в тій або іншій області людської діяльності [3, 4, 5].

Огляд існуючих рішень обмежимо розглядом класичним застосуванням електронних каталогів, які призначені для забезпечення функціонування інформаційних фондів, інформаційних сховищ, архівів, бібліотек.

Один з класів таких програм є бібліографічні менеджери. Бібліографічні менеджери - це програми, розроблені для зберігання бібліографічних даних, повних текстів і оформлення посилань і списків літератури. При роботі з бібліографічним менеджером програма стаціонарно встановлена на комп'ютер. До універсальних бібліографічних менеджерів відносяться такі як Mendeley, EndNote, Zotero та інші, які можуть використовуватися як індивідуальними користувачами, так і науковими колективами [6].

Mendeley - безкоштовна програма для управління бібліографічною інформацією, що дозволяє зберігати і переглядати дослідницькі праці у

					ВКРБ-123.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

форматі PDF, а також має підключення до академічної соціальної мережі. Окрім стандартного функціонала бібліографічного менеджера Mendeley дозволяє знаходити за списками літератури колег, які цитують ті ж джерела. Mendeley дозволяє завантажувати статті безпосередньо з таких платформ, як Science Direct; зберігати документи в особистій бібліотеці, а також управляти і користуватися ними у будь-який час як онлайн, так і оффлайн; виділяти фрагменти статей, коментувати їх і ділитися інформацією з членами своєї дослідницької групи; створювати бібліографічні списки і посилання для роботи одним натисненням кнопки, вибираючи відповідний стиль. [6, 7, 8].

EndNote - комерційна система управління бібліографічною інформацією, вживана для управління посиланнями і бібліографією, яка дозволяє відформатувати їх згідно з стандартами цитування. EndNote дозволяє: додавати посилання у свою бібліотеку безпосередньо з Web of Science; розміщувати посилання в теках для доступу і використання їх у будь-який час; створювати форматований список посилань з використанням різних стилів публікацій; діставати доступ до бібліотеки EndNote з будь-якого комп'ютера [6, 9, 10].

Zotero - програма з відкритим вихідним кодом, є додатком (плагіном) до браузеру Firefox. Дозволяє користувачеві збирати, упорядковувати, та організовувати інформацію для власної дослідницької роботи з усіх типів джерел, що можливі для відображення у браузері. Zotero дозволяє: зберігати бібліографічну інформацію не лише в PDF форматі; створювати бібліографію і вставки посилань на роботи зі збережених даних; синхронізуватися з сервером для зберігання вашої бази даних; індексувати бази даних; створювати ваш профіль online. [6, 11, 12].

На теперішній час системи каталогів і картотек вбудовані в системи комплексної автоматизації бібліотечних процесів і представлені на ринку рядом рішень [13].

1С:Бібліотека - програма для автоматизації діяльності бібліотек будь-якого типу і призначення, рисунок 2.1.

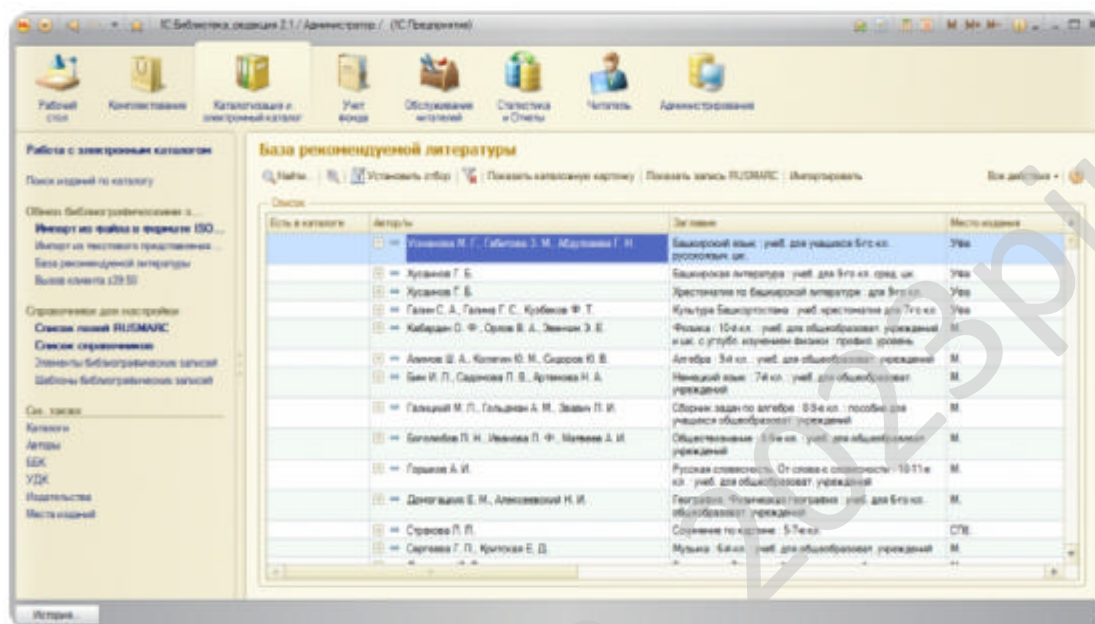


Рисунок 2.1 1С:Бібліотека

1С:Бібліотека дозволяє автоматизувати робочі процеси бібліотеки, залежно від її призначення, типу, складу фондів, може бути інтегрований з іншими типовими рішеннями "1С". За допомогою системи "1С:Бібліотека" може бути автоматизована діяльність бібліотек різних призначень і типів [14, 15, 16, 17].

Рішення, що цікавить, - каталогізація забезпечує:

- опис будь-яких видів видань, включаючи документи в електронному форматі, аудіо-, відео-, нотні, картографічні, ізовидання і ін.;
- контроль даних, що вводяться, на їх відповідність вимогам стандартів;
- автоматичну індексацію, на основі якої реалізується швидкий пошук по будь-яких елементах бібліографічного опису і їх поєднаннях;
- можливість використати будь-які графічні матеріали (зображення титульних листів, артефактів, сторінок тексту і так далі);

- експорт/імпорт даних в MARC -форматах;
- зберігання необмеженої кількості бібліографічних записів.

АБІС ІРБІС являє собою типове інтегроване рішення в області автоматизації бібліотечних технологій і призначена для використання в бібліотеках будь-якого типу й профілю. , рисунок 2.2.

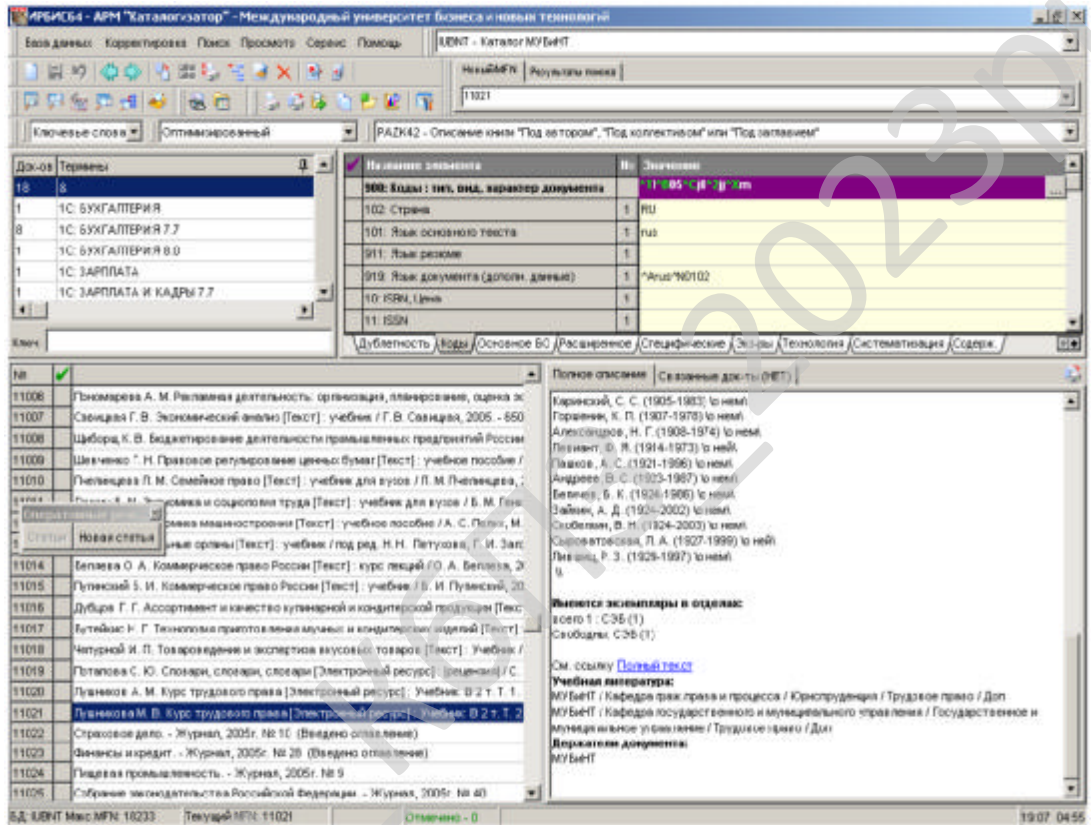


Рисунок 2.2 АБІС ІРБІС

Система цілком відповідає міжнародним вимогам, пропонованим до подібного роду систем, і підтримує усі вітчизняні бібліографічні стандарти й формати. Система дозволяє описувати усі види видань. Інтерфейси системи максимально наближені до потреб користувача і легко освоюються [14, 18, 19, 20 ].

Рішення АРМ “Каталогізатор” АБІС ІРБІС забезпечує

- обробку будь-яких видів видань, включаючи аудіо- і відеоматеріали, електронні ресурси, картографічні матеріали, ноти і так далі, будь-яка повнота опису, включаючи зміст журналів і зміст збірок;

- опис періодичних видань на звідному рівні і на рівні окремих номерів і "підшивок" з урахуванням відомостей про статті, що входять в них;

- технологію індексування видань (систематизація, предметизація), що включає автоматичне формування авторського знаку і апарат навігації по рубризатору, алфавітно-предметному покажчику УДК/ББК, авторитетному файлу предметних рубрик і тезаурусу;

- технологію копіювання даних, що виключає повторне введення при створенні аналогічних і пов'язаних бібліографічних описів, зокрема, при створенні аналітичних описів;

- систему формально-логічного контролю даних - як на рівні окремих бібліографічних елементів, так і на рівні опису в цілому;

- технологію автоматичного звірення на ідентичність, що виключає повторне введення в електронний каталог;

- рішення задачі книгозабезпеченості і безінвентарний облік багатопримірникової літератури для бібліотек навчальних закладів;

- вихідні форми - покажчики, інвентарні списки, списки номерів журналів та ін., що не поступили.

АБІС Коha - перша вільна АБІС з відкритими кодами автоматизована бібліотечна інформаційна система, рисунок 2.3..

АБІС Коha призначена для підтримки традиційних бібліотечних технологічних процесів. Коha створена на основі бібліотечних стандартів та протоколів, що забезпечує можливість взаємодії між АБІС Коha та іншими бібліотечними системами і технологіями [14, 21, 22, 23]..

Рішення вибудовано на модулі каталогізації з вбудованим клієнтом Z39.50, що базується на наступних бібліографічних запитах доступних шаблонів або Marc-структур:

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

- структура за умовчанням;
- аудіоматеріали на касетах, CD;
- відеоматеріали (DVD, VHS);
- книги (монографічні видання);
- комплекти, набори змішаних матеріалів;
- моделі;
- палітурка (видавничий, поширення, власницький);
- тривалі (серійні) видання;
- структура швидкого додавання;
- електронні ресурси (диски, веб, файли, .).



Рисунок 2.3 АБІС Koha

АБІС МегаПро - програмне рішення для комплексної автоматизації інформаційно-бібліотечної діяльності, побудови бібліотечних мереж, ефективного управління інформаційними ресурсами і організації доступу до них на основі web-технологій, рисунок 2.4.

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

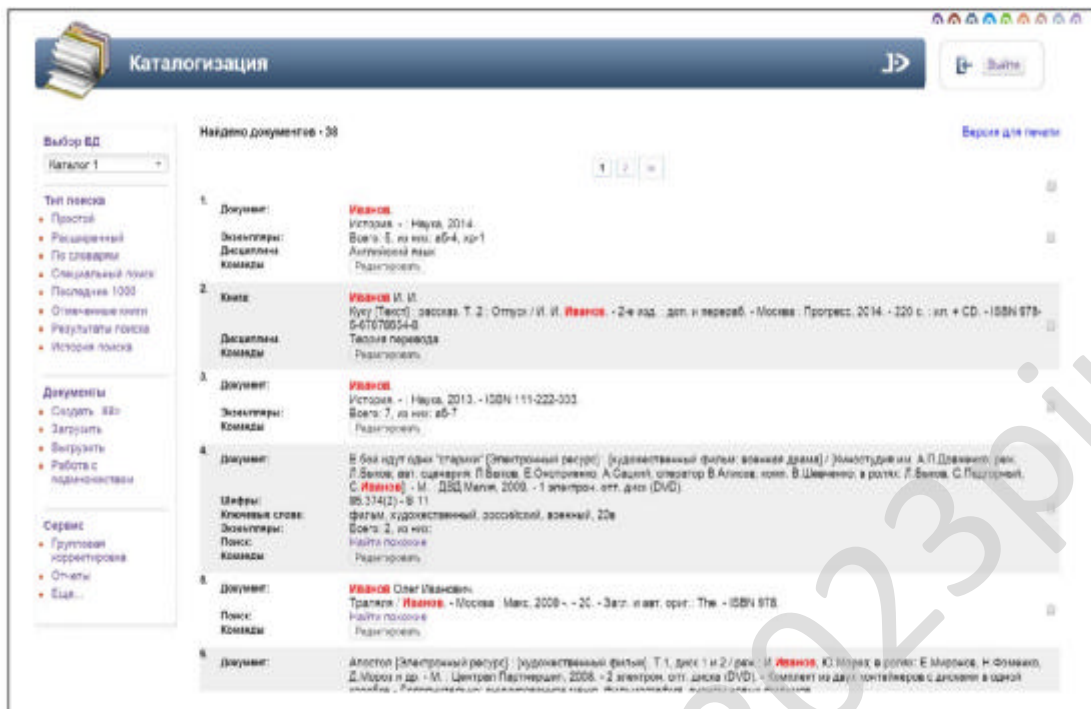


Рисунок 2.4 АБІС МегаПро

Являється web-системою нового покоління, може використовуватися як в традиційному, так і в «хмарному» варіантах [14, 24, 25, 26].

Модуль «Каталогізація» АБІС МегаПро призначений для ефективної автоматизації процесів, пов'язаних з каталогізацією у бібліотеці. Робота з модулем виконується авторизованим персоналом (співробітники бібліотеки) за допомогою web-браузера і може виконуватися як в локальній обчислювальній мережі, так і видалено, з будь-якої точки Інтернет.

Модуль «Каталогізація» забезпечує ефективне виконання функцій каталогізації в автоматизованому режимі. Для підтримки роботи модуля з RFID-обладнанням потрібний додаток «МегаRFID». Модуль «Каталогізація» розроблений на основі відкритих стандартів і форматів даних. У ній реалізована триланкова архітектура «тонкий клієнт - сервер додатків - СУБД» на основі Web-технологій. Доступ до функцій і ресурсів здійснюється через стандартний web-браузер. Реалізація модуля виконана на програмних засобах платформи .NET. Встановлюється на IIS 7.0 і вище. Для

роботи з модулем використовуються браузері: Internet Explorer; Google Chrome. Основні функціональні можливості:

- ведення електронного каталогу бібліотеки;
- введення і редагування бібліографічних записів усіх видів документів в MARC-форматі в електронному каталозі;
- підтримка багаторівневої структури бібліографічних записів - створення пов'язаних записів на рівні набору;
- підтримка UNICODE (підтримка даних на різних мовах);
- наявність вбудованої віртуальної клавіатури з підтримкою більше 40 мов світу для введення бібліографічних описів;
- підготовка аналітичних описів;
- видача звітно-статистичних даних;
- облік фондів бібліотеки в автоматизованому режимі;
- введення інформації про пункти книговидачі і визначення їх статусів;
- формування і друк стандартних вихідних форм;
- пошук інформації по будь-яких елементах бібліографічного опису і їх поєднаннях;
- завантаження/вивантаження бібліографічних записів в MARC - формати (MARC21, RUSMARC);
- отримання статистичних відомостей по стану, руху і використанню фонду бібліотеки.

АІБС МАРК - є інтегрованою системою, призначеною для автоматизації інформаційних технологій в області бібліотечної справи, і забезпечує реалізацію повних технологічних циклів від замовлення літератури до видачі її читачеві, рисунок 2.5.

Вона забезпечує комплексну автоматизацію усіх бібліотечних процесів : комплектування літератури; створення і ведення електронного каталогу; систематизацію; обробку видань, що поступають; довідково-інформаційне обслуговування; обслуговування читачів; облік бібліотечного фонду;

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14





- зберігання конфігурації для усіх робочих місць (співробітників) централізований на сервері;
- підтримка формату RUSMARC, включаючи механізми зв'язування бібліографічних записів, авторитетних файлів;
- управління схемою RUSMARC для авторитетних і бібліографічних записів; створення довідників, генераторів і зв'язування їх з елементами запису; - замовлення книг;
- підписка і реєстрація періодичних видань;
- традиційні функції комплектування одиниць зберігання, дроту рахунків;
- формування типових вихідних форм у форматі MS Excel (інвентарні книги, КСУ, рахунки, акт);
- контроль стадій обробки записів по відділах;
- можливість налаштування формату запису у рамках специфікації RUSMARC і полів локального використання;
- створення RUSMARC записів з використанням шаблонів (робочих листів);
- створення шаблонів для каталогізації різних видів документів, включаючи нетрадиційні (аудіо, відео, електронні ресурси, ноти, т.д.);
- виявлення записів-дублів;
- підтримка введення даних на різних мовах в стандартному кодуванні UNICODE з використанням вбудованої віртуальної клавіатури;
- забезпечення надійності операцій видаленої зміни даних;
- можливість ведення словників і довідників;
- видалена взаємодія з джерелами бібліографічних і авторитетних даних у форматах RUSMARC, доступних по протоколу Z39.50 або SRU;
- підтримка «каталогізації запозиченням» - копіювання записів із зовнішніх джерел з можливістю подальшої перевірки, редагування і запису в локальний каталог;
- формування бюлетеня нових надходжень і інших списків документів на основі RUSMARC записів;

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- завантаження/вивантаження MARC записів з/у в різних кодуваннях у форматах ISO2709, JSON, XML;
- підтримка функцій обробки RFID-міток документів (за умови підтримки виробником устаткування протоколу SIP 2.0).

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Базуючись на наявності в якості засобу розробки та наявних можливостях для реалізації проекту й побудови електронної каталожної системи обрано Delphi версії 7.

Система програмування Delphi версії 7 фірми Enterprise (Borland) надає широкі можливості для програмування додатків ОС Windows. Delphi - це продукт Borland International для швидкого створення додатків. Цей високопродуктивний інструмент візуальної побудови додатків включає справжній компілятор коду і надає засоби візуального програмування, дещо схожі на ті, які можна виявити в Microsoft Visual Basic або в інших інструментах візуального проектування.

В основі Delphi лежить мова Object Pascal, яка є розширенням об'єктно-орієнтованої мови Pascal.

До Delphi також входять локальний SQL-сервер, генератори звітів, бібліотеки візуальних компонентів, і інше, необхідне для того, щоб почувати себе абсолютно упевненим при професійній розробці інформаційних систем або просто програм для Windows середовища.

Сам Delphi призначений для професійних розробників, бажаючих дуже швидко розробляти додатки в архітектурі клієнт-сервер. Delphi робить невеликі за розмірами високоефективні виконувані модулі (.exe і .dll), тому в Delphi мають бути, передусім, зацікавлені ті, хто розробляє продукти на продаж. З іншого боку, невеликі за розмірами і швидко виконувані модулі означають, що вимоги до клієнтських робочих місць істотно знижуються - це має важливе значення і

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>18</b>

для кінцевих користувачів.

Переваги Delphi у порівнянні з аналогічними програмними продуктами:

- швидкість розробки додатка (RAD);
- висока продуктивність розробленого застосування;
- низькі вимоги розробленого додатка до ресурсів комп'ютера;
- можливість розробки нових компонентів і інструментів власними засобами Delphi (існуючі компоненти й інструменти доступні в початкових кодах);
- вдале опрацювання ієрархії об'єктів.

Система програмування Delphi розрахована на програмування різних застосувань і надає велику кількість компонентів для цього. До того ж працедавців цікавить, передусім, швидкість і якість створення програм, а ці характеристики може забезпечити тільки середовище візуального проектування, здатне взяти на себе значні об'єми рутинної роботи з підготовки додатків, а також погоджувати діяльність групи постановників, кодувальників, тестерів і технічних письменників. Можливості Delphi повністю відповідають подібним вимогам і підходять для створення систем будь-якої складності [32, 33, 34, 35, 36, 37, 38, 39].

В якості бази даних планується скористатися Microsoft Office Access. Вибір базується на наявності засобів підключення до вибраної БД в пакеті поставки Delphi 7 та наявності самої Microsoft Office Access (як засіб розробки вона не використовується, але для поточного контролю БД під час розробки зовсім і не лишня).

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для каталожних систем.

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>19</b>

Підставою для розробки служить завдання на бакалаврську роботу, видане на кафедрі програмування комп'ютерних систем і мереж

У процесі розробки бакалаврської роботи 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 методику побудови системи. Розробити функціональну та структурну схеми системи;

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

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

д) розробити рекомендації щодо організаційних та методичних заходів, які забезпечать впровадження розробки в промислову експлуатацію та її подальшу успішну експлуатацію;

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

					ВКРБ-123.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Інформаційний пошук - напрям досліджень, що вивчає питання пошуку документів, обробки результатів пошуку, а також цілу низку суміжних запитань : моделювання, класифікації, кластеризації і фільтрації документів, проектування архітектури пошукових систем і призначених для користувача інтерфейсів, мови запитів, і так далі. Документ - це змістовно закінчена одиниця інформації, представлена на якій-небудь природній мові, яка ідентифікується унікальним чином. Документ - це порція інформації, якою оперують інформаційно-пошукові системи. Інформаційно-пошукова система - це комплекс програмних засобів, що забезпечують виборчий відбір за заданими ознаками документів, що зберігаються в електронному (оцифрованому) уявленні. Одним з ключових понять, що характеризує вибір того або іншого методу аналізу текстової інформації, а також реалізацію конкретного варіанту пошуку, є модель пошуку [40, 41, 42, 43].

Класично модель інформаційного пошуку побудована на системі каталогів і картотек. Система каталогів і картотек - сукупність планомірно організованих, доповнюючих один одного взаємозв'язаних каталогів і картотек, які розкривають зміст і структуру фондів в різних аспектах. Вони є важливим засобом розкриття фондів та надання інформації користувачам про наявні в них матеріали, а також дозволяють користувачам з найменшими витратами часу підібрати необхідне враховуючі наявні потреби [44].

Каталоги і картотеки містять про наявний у фонді документ дані, які складені згідно з існуючими правилами бібліографічного опису. Каталоги і картотеки розрізняються по видах відбитих документів, призначенні, охопленні фондів, способі угруповання і формі представлення інформації [45].

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Електронний бібліотечний каталог - сукупність програмних і апаратних засобів по забезпеченню діяльності бібліотеки за замовленням, каталогізації, пошуку, видачі книг, рішення різних завдань по звітності і книгозабезпеченні читачів та ін. як в локальній обчислювальній мережі, так і через web-вміст [46].

І так, в каталозі інформація про зміст документів архіву згрупована по предметах (темам, галузям), розташованих відповідно до прийнятої для цього каталогу схеми класифікації документної інформації. Сукупність різних каталогів складає систему каталогів. Сукупність робіт з підготовки, створення і ведення каталогів називається каталогізацією. Каталогізація включає наступні види робіт : складання схеми класифікації, виявлення і відбір документів, їх опис і систематизація, ведення каталогів. Вибір схеми класифікації визначається складом об'єктів (документів, ...) і завданнями їх використання. Залежно від схеми побудови каталоги підрозділяються на систематичні, тематичні, предметні і їх різновиди (іменний, географічний, об'єктний). Об'єктом опису може виступати документ (група документів, частина документу), одиниця зберігання (одиниця обліку), опис, комплект, фонд (група фондів), що містять інформацію по певній темі (питанню). До складу описової статті входять: назва архіву, індекс, рубрика, підрубрика, дата події, місце події, зміст, номер фонду, назва фонду, номери опису, справи, листів, мова документу, спосіб відтворення, для карткового каталогу також прізвище укладача і дата складання описової статті [47].

Як правило, найбільш поширені являються систематичний і іменний каталоги. В систематичному каталозі документи класифікуються по галузях знань і практичної діяльності, та розташовуються в логічній послідовності. У іменному каталозі документи класифікуються за абеткою прізвищ осіб, що згадуються в документах або є їх авторами. Подальша систематизація проводиться в хронологічній або логічній послідовності [47].

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Отже для розробки проекту оберемо в якості моделей організації пошуку - класичні на базі яких будуються систематичні і іменні каталоги,

По суті, робота системи зводиться до вибору інформації з бази даних, до якої вона попередньо заносилася та корекція самої бази даних. Наявні також сервісні операції пов'язані з адмініструванням програми

Виходячи із вибраної моделі пошуку, в системі має бути реалізований наступний функціонал:

- запуск програми та забезпечення її працездатності при встановленому факту на правове використання програмного продукту;
- блокування працездатності при виявленні факту неправомірного використання програмного продукту;
- завершення роботи програми;
- авторизація користувача;
- вибір режиму роботи програми для авторизованого користувача: завершення роботи користувача, адміністрування програми в цілому та роботою користувачів зокрема, керування інформаційними ресурсами, отримання довідкової інформації:
- для режиму адміністрування програми в цілому та роботою користувачів зокрема доступний наступний набір операцій та функції: введення та видалення користувача, визначення та перевизначення його пароля на ідентифікацію програмою, визначення його прав доступу до ресурсів програми, групові операції над інформаційним вмістом програми, видалення інформаційного вмісту програми, зміна або корекція інформаційного вмісту програми; доповнення інформаційного вмісту програми, керування пошуковим запитом та перегляд віднайденого;

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

- для режиму керування інформаційними ресурсами доступний наступний набір операцій та функцій: видалення інформаційного вмісту програми, зміна або корекція інформаційного вмісту програми; доповнення інформаційного вмісту програми, керування пошуковим запитом та перегляд віднайденого;

- для режиму отримання довідкової інформації доступно: керування пошуковим запитом та перегляд віднайденого.

### 3.2 Розробка структурної схеми

Розгляд структури програми можливий з двох позицій: розгляд, як структурних одиниць програми самої Delphi або як описання зв'язків між різними структурними одиницями програми (структурні одиниці програми розглядаються в якості окремих частини - блоків, що виконують різні функції) та проходженням інформаційних потоків через них. Будь-яка програма Delphi складається з файлу проекту (файл з розширенням DPR) і одного чи кількох модулів (файли з розширенням PAS). Структурно модулі - це окремі програмні одиниці, що реалізують окремі частини програми. [48]. Тож для побудови програми скористаємося можливістю мінімального використання модулів - один модуль.

Розглянемо структуру програми через призму її виконання з боку функції. Структурна схема системи зображена на рисунку 3.1.

Зі схеми можна побачити, що до програми поступають наступні інформаційні потоки:

- Login - текстова послідовність, якою задається ім'я користувача для внесення в базу користувачів або дозволу на опрацювання режимів роботи програми з урахуванням обмежень на функціонал програми та прав доступу до тих чи інших ресурсів ;

					ВКРБ-123.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24



- sn - серійний номер обладнання (серійний номер жорсткого диску) для організації захисту програмного забезпечення від можливого небажаного розповсюдження;

- I - масив інформації, для створення метаданих, що попередньо вноситься до каталогу;

- Z - інформаційний запит, на відшукування інформації з масиву метаданих, який зберігається в відповідній база даних.

Також, із схеми можна побачити, що програма генерує:

- i - відібрані метадані дані, які задовольняються інформаційний запит на відбір.

На схемі також показані внутрішні інформаційні зв'язки:

- зв'язок між блоком підпрограм керування та підпрограмою адміністрування, по команді керування запускається підпрограма адміністрування, а після її виконання проходить повернення до місці виклику підпрограми;

- зв'язок між блоком підпрограм керування та підпрограмою підготовки метаданих, по команді керування запускається підпрограма підготовки метаданих, а після її виконання проходить повернення до місці виклику підпрограми;

- зв'язок між блоком підпрограм керування та підпрограмою відбору метаданих, по команді керування запускається підпрограма відбору метаданих, а після її виконання проходить повернення до місці виклику підпрограми;

- зв'язок між підпрограмою адміністрування та бази даних користувачі, підпрограма звертається до бази даних на одержання або запис прав, паролів та логіків користувачів;

- зв'язок між підпрограмою підготовки метаданих та бази даних довідники, підпрограма звертається до бази даних на одержання або модифікації довідкової інформації;

- зв'язок між підпрограмою підготовки метаданих та бази даних

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

метадані, підпрограма звертається до бази даних на запис або модифікації метаданих;

- зв'язок між підпрограмою відбору метаданих та базо даних довідники, підпрограма звертається до бази даних на одержання довідкової інформації;

- зв'язок між підпрограмою відбору метаданих та базо даних метадані, підпрограма звертається до бази даних на одержання метаданих.

### 3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2.

З рисунка видно, що роботу розробленої системи можна представити функціонуванням сукупності частин: підпрограми автентифікації, підпрограми керування режимами, підпрограми адміністрування, підпрограми підготовки метаданих, підпрограми відбору метаданих, бази даних користувачі, бази даних довідники, бази даних метадані.

Розглянемо їх роботу. Підпрограми автентифікації в заданості від серійного номера диску sn визначить право на працездатність програми та встановить користувача і його права через підпрограму адміністрування керуванням UAB проаналізувавши відповідь АВ. Далі, випадку всіх наявних дозволів, підпрограми автентифікації передаватиме підпрограмі керування режимами керування U для подальшого опрацювання. Підпрограма керування режимами відповідно до режиму, який визначається керуванням U, формує керування UA (для режиму адміністрування), керування UP (для режиму підготовки метаданих), керування UP (для режиму відбору метаданих) ц передає відповідній підпрограмі керування.

Підпрограма адміністрування аналізує сигнал керування та може виконувати порівняння зовнішніх сигналів Login (імені користувача) і Password (пароля) з одержаними за запитом ZA з бази даних користувачі

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

сигналів LA та PA й видавати відповідь AB, або може заносити до бази даних користувачів сигналами LA та PA ім'я та пароль що надійшли зовнішніми сигналами Logon і Password керуванням на запис ZA.

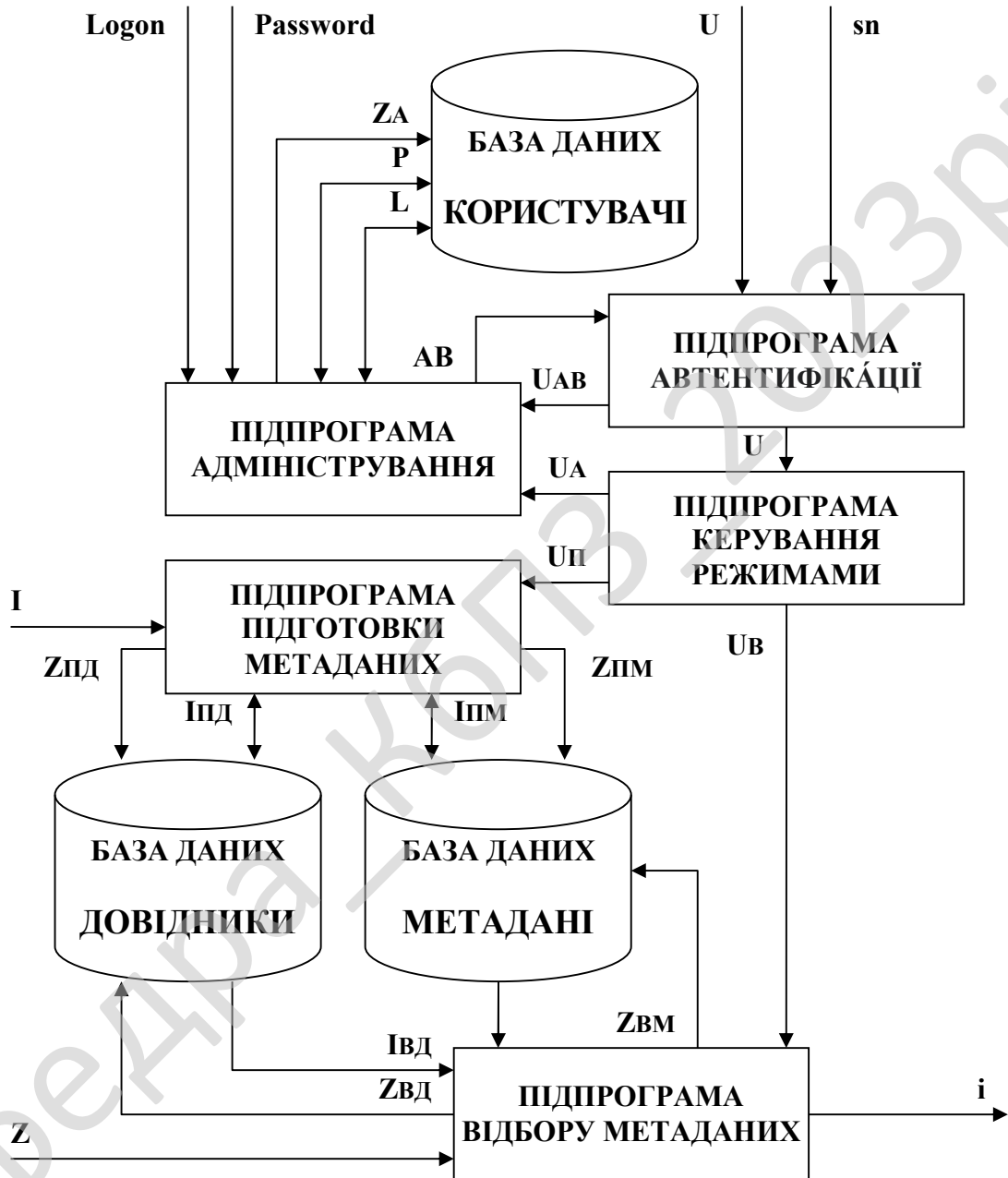


Рисунок 3.2 - Функціональна схема

Підпрограма підготовки метаданих по керуванню UP може робити запис до бази даних метадані або модифікувати дані в базі даних метадані

відповідно до одержаного масиву І з урахуванням вибраної за запитом ЗПД інформації ІПД від бази даних довідники, відпрацювання з базою даних метадані проходить за запитом ЗПМ, передача даних ІПМ. Також підпрограма підготовки метаданих по керуванню УП може виконувати модифікувати дані в базі даних довідники відпрацювання з базою проходить за запитом ЗПД, передача даних ІПД..

Підпрограма відбору метаданих по керуванню UB може виконувати тільки зчитування бази даних метадані IBM за запитом ZBM створюючи його за запитом Z та інформацією ІВД від бази даних словники одержаною за запитом ZВД. Вихідні метадані і підпрограма відбору метаданих формує на основі одержаних даних IBM з бази даних метадані.

### 3.4 Розробка діаграми процесів

Оскільки для опису архітектури інформаційної системи можна скористатися одним із п'яти видів представлень, кожна з яких є одна з можливих проекцій організації і структури системи і відповідає окремому аспекту її функціонування (вид з погляду прецедентів використання, вид з погляду проектування, вид з погляду процесів, вид з погляду реалізації, вид з погляду розгортання системи,) то для опису механізмів синхронізації взаємодій, станів і дій скористаємося діаграмою процесів [49].

Діаграма процесів розробленої системи зображена на рисунку 3.3.

Діаграма процесів представлена у вигляді напрямленого графа, що описує переходи від одного процесу до іншого. Проаналізувавши діаграму процесів, можна визначити безпосередньо послідовність дій системи, що розробляється.

Так, процес роботи програми починається з визначення автентифікації, після чого у випадку коли вона не пройшла робота програми припиняється а у випадку проходження автентифікації програма переходить до вибору трьох режимів роботи або закінчення роботи програми. Закінчення режиму роботи адміністрування

					ВКРБ-123.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

призводить до повернення на процес режиму роботи. Режим підготовка метаданих поділяється на три підрежими введення інформації, корекція довідників, корекція метаданих, підготовка метаданих. Закінчення режиму підготовка метаданих теж приводить до повернення на вибір режиму. Режим відбір метаданих складається з двох процесів які слідують один за одним - спочатку вводиться запит, в наслідок якого формується та видається інформація. Завершується режим переходом до вибору режиму.

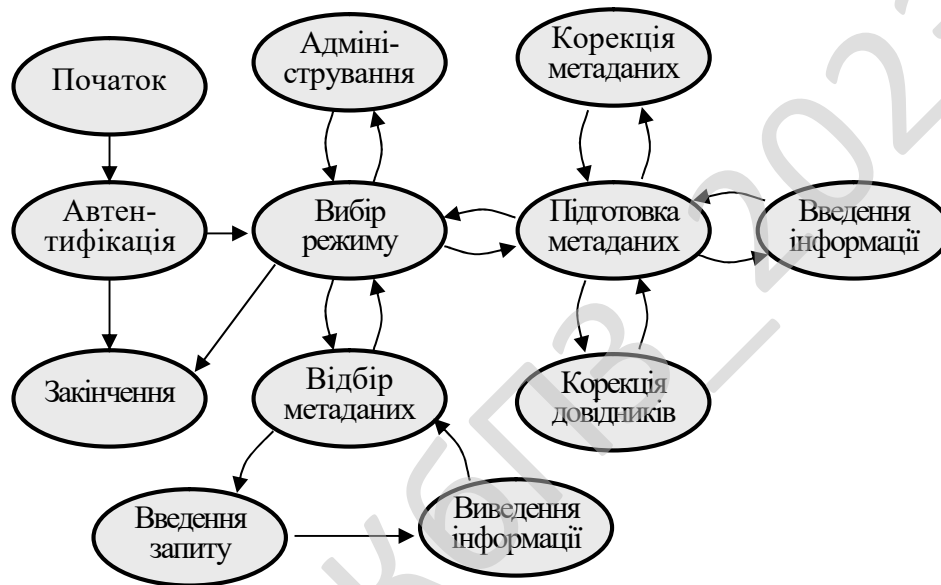


Рисунок 3.3 - Діаграма процесів

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.

## 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

### 4.1 Розробка блок-схем та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи програми. Блок-схема алгоритму роботи основної програми зображена на рисунку 4.1.

З представленої блок-схеми роботи програми видно, що спочатку відбувається введення User name а потім Password - це ім'я користувача та пароль на підключення до програми баз даних, де зберігаються метадані та довідник УДК. Після введення зазначених параметрів і відбувається саме підключення до баз. Наступним процесом роботи програми є запуск підпрограми автентифікації для отримання параметра Av, за яким визначається правомірність використання програмного продукту, законність його придбання. Далі проводиться аналіз отриманого значення параметра Av, що являє собою змінну логічного (булево) типу. У випадку прийняття Av значення хиба (false) приймається рішення, що відбувається спроба незаконного використання програмного продукту. За розробленим алгоритмом програма далі сповіщає зловмисника видачею повідомлення про його намагання використати неліцензійне програмне забезпечення. Наступним виконанням програми є повідомлення про непрацездатність програми. На далі програма надає час на прочитання повідомлень та осмисленого виходу з програми і у випадку відмови виходу час прочитання та осмислений вихід подовжується. У випадку прийняття Av значення істина (true), програма передає курування підпрограмі авторизації, визначення надання функціональних можливостей користувачу програми відповідно до його заявлених прав на використання програмного продукту. Наступною дією програми є запит на введення режиму (команди на виконання опрацювань).

					ВКРБ-123.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

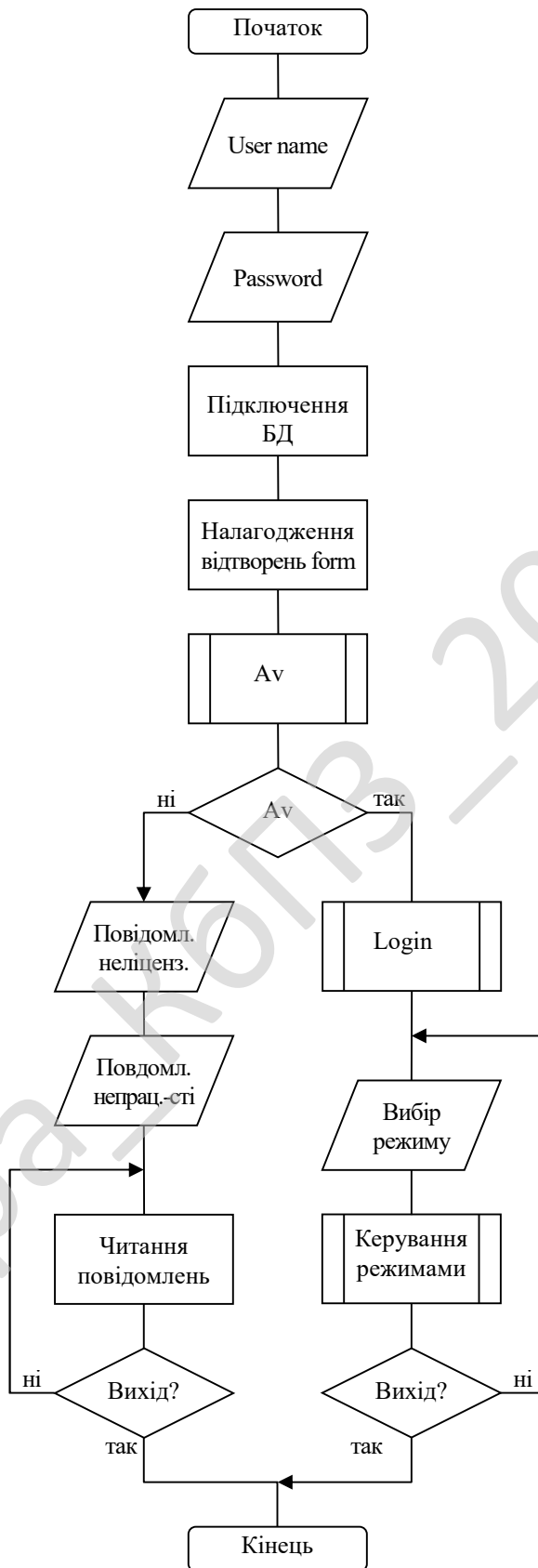


Рисунок 4.1 - Блок-схема роботи основної програми.



З представленої блок-схеми роботи підпрограми видно, що спочатку відбувається зчитування SN HDD серійного номера жорсткого диску. Потім відкривається системний реєстр. Наступним для відкриття є розділ системного реєстру HKEY\_LOCAL\_MACHINE. Далі відкривається блок системного реєстру HardWare. Наступним є зчитування з системного реєстру (попередньо записане значення серійного номера жорсткого диску в якості секретного ключа на ідентифікацію комп'ютера до якого ліцензійно встановлена програма) до змінної snr. Після чого порівнюються SN HDD та snr у випадку їх рівності значення функції Av (яке повертається) визначається як true (істина) та позначає наявність ліцензійного погодження на програмне забезпечення, а в разі нерівності Av приймає значення false (хиба) - відсутність.

Розглянемо алгоритм роботи підпрограми авторизації (Login). Його блок-схема зображена на рисунку 4.3.

З представленої блок-схеми роботи підпрограми видно, що спочатку відбувається пошук файлу з паролями. Якщо він відсутній, то він створюється й до нього зноситься користувач ADMIN з паролем 1 й правами адміністратора (prava=1) А якщо таких файл існує то ці процеси пропускаються. Далі водиться Login найменування користувача що бажає працювати з програмою. Наступним вводиться пароль для підтвердження особистості. Якщо бажає працювати з програмою. Відповідно до наданих даних визначаються права користувача та для визначених прав встановлюється доступ до функціоналу програми, шляхом обмеження певних функцій для запобігання дій, які обмежені для виконання тими чи іншими категоріями персоналу.

Розглянемо алгоритм роботи підпрограми керування режимами. Його блок-схема зображена на рисунку 4.4.

З представленої блок-схеми роботи підпрограми видно, що перевіряються по чергово команди на запуск режимів і потім запускається відповідна підпрограма.

					ВКРБ-123.23.0006.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34



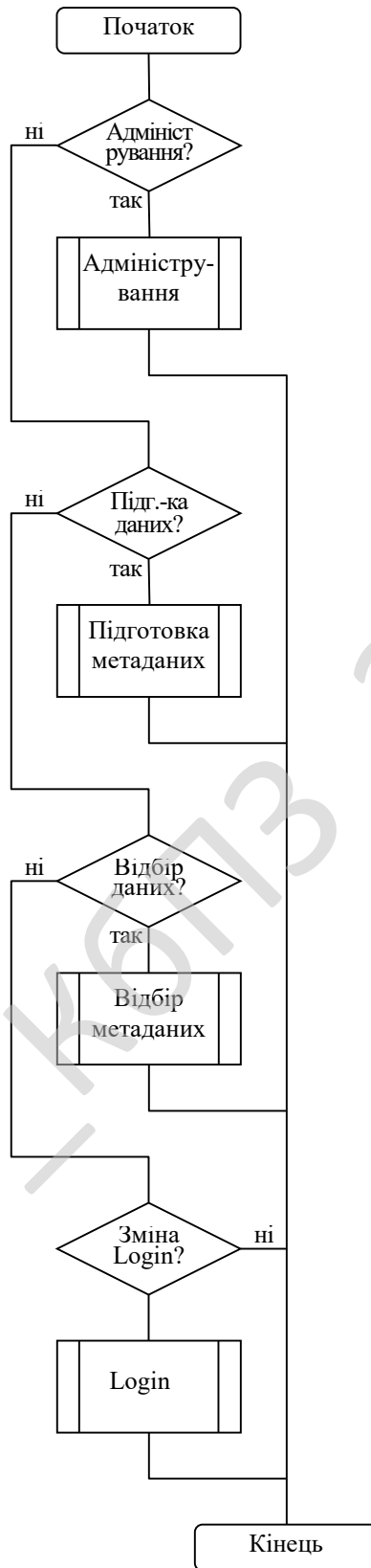


Рисунок 4.4 - Блок-схема роботи підпрограми керування режимами

Зауважимо що наявний вихід режиму роботи, шляхом вибору іншого (в подібність перериванню виконання)

Розглянемо алгоритм роботи підпрограми адміністрування. Його блок-схема зображена на рисунку 4.5.

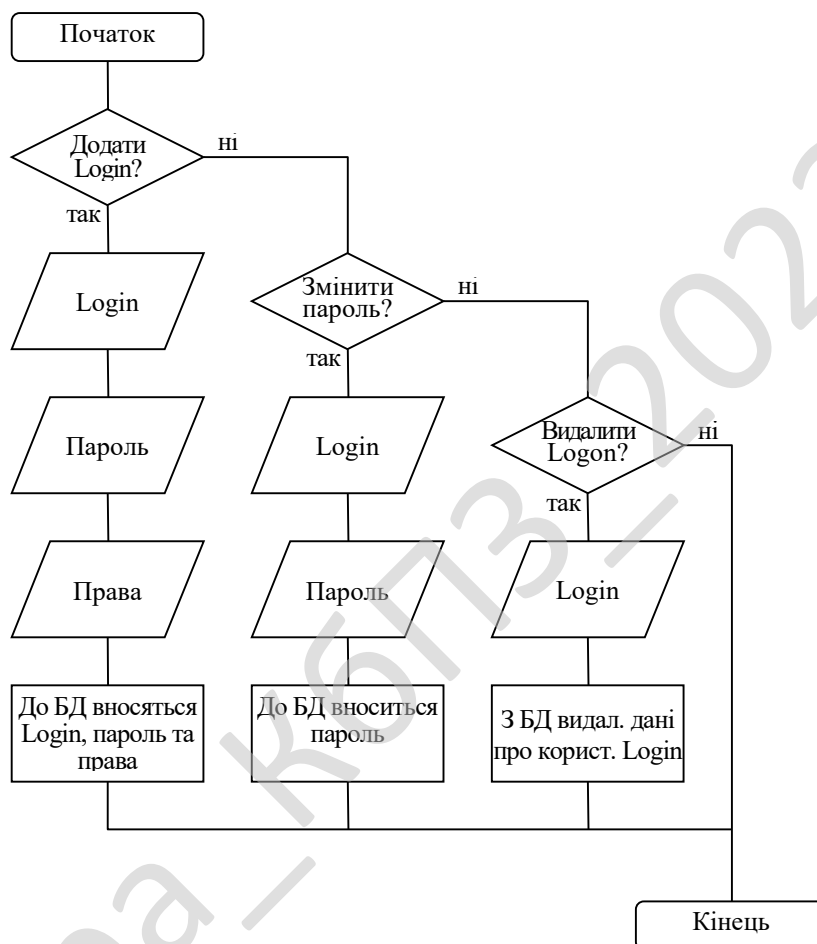


Рисунок 4.5 - Блок-схема роботи підпрограми адміністрування

З представленої блок-схеми роботи підпрограми видно, що перевіряються по чергово команди режиму адміністрування. Після входження в процес виконання команд відсутня програмно передбачена можливість виходу до завершення процесу виконання (крім аварійного переривання роботи самої програми - виходу з неї). Додавання користувача проходить за наступною послідовністю введення Login, введення пароля, вибір права та занесення даних до БД. Зміна пароля користувача

проходить за наступною послідовністю введення Login, введення пароля та занесення даних до БД. Видалення користувача (Login) проходить за наступною послідовністю введення Login та вилучення даних з БД.

Розглянемо алгоритм роботи підпрограми підготовки метаданих. Його блок-схема зображена на рисунку 4.6.

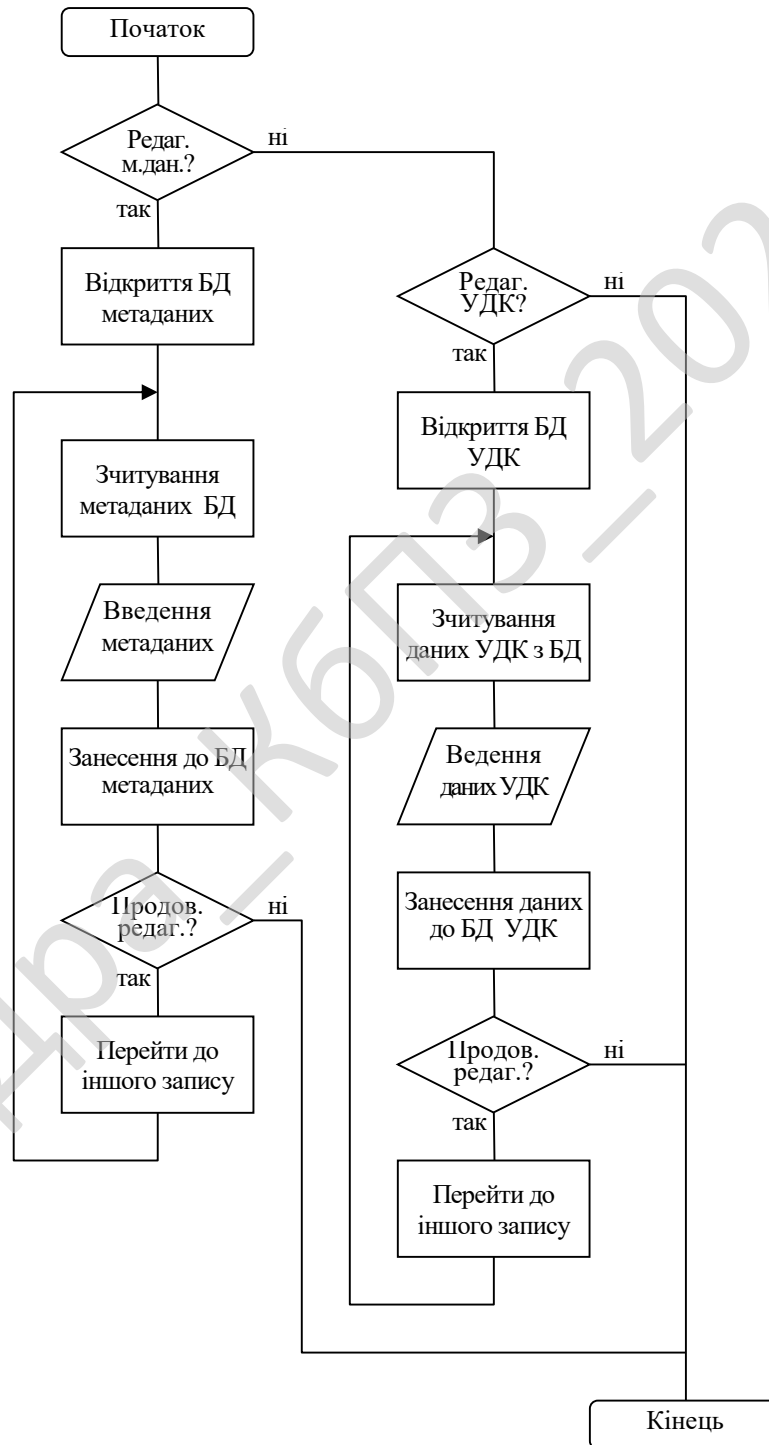


Рисунок 4.6 - Блок-схема роботи підпрограми підготовки метаданих

З представленої блок-схеми роботи підпрограми видно, що перевіряються по чергово команди режиму підготовки метаданих. Редагування метаданих проходить за наступною послідовністю відкриття БД метаданих, зчитування даних з БД, введення даних, занесення даних до БД, перевірка на продовження процесу редагування, якщо так - то переведення кажи до іншого запису і процес повторюється з зчитування даних з БД, якщо ні - вихід з режиму. Редагування (корекція) УДК проходить за наступною послідовністю відкриття БД УДК, зчитування даних з БД, введення даних, занесення даних до БД, перевірка на продовження процесу редагування, якщо так - то переведення кажи до іншого запису і процес повторюється з зчитування даних з БД, якщо ні - вихід з режиму

Розглянемо алгоритм роботи підпрограми відбору метаданих. Його блок-схема зображена на рисунках 4.7. та 4.8

З представленої блок-схеми роботи підпрограми видно, що перевіряються по чергово команди режиму відбору метаданих. Після входження в процес виконання команд наявна програмно передбачена можливість виходу до завершення процесу виконання. Пошук метаданих з врахуванням УДК проходить за наступною послідовністю: відкриття БД УДК, зчитування даних УДК з БД, надання УДК для спрощення та осмисленого вибору напрямку пошуку метаданих, відкриття БД метаданих, відбір метаданих по УДК, множинне представлення сукупності метаданих відібраних за запитом, одиничний сфокусований перегляд окремого елемента з множини відібраної за запитом. В подальшому виконання операції можна вважати кінцевою й вийти з режиму відбору метаданих або провести уточнення результату пошуку то назві за наступною послідовністю: введення назви, відбір метаданих з урахуванням нової додаткової умови, множинне представлення сукупності метаданих відібраних за запитом, одиничний сфокусований перегляд окремого елемента з множини відібраної за запитом; або провести уточнення результату пошуку то автору за наступною послідовністю: введення даних автора, відбір метаданих з урахуванням нової додаткової умови, множинне представлення сукупності метаданих відібраних за запитом, одиничний сфокусований перегляд окремого елемента з множини відібраної за запитом.

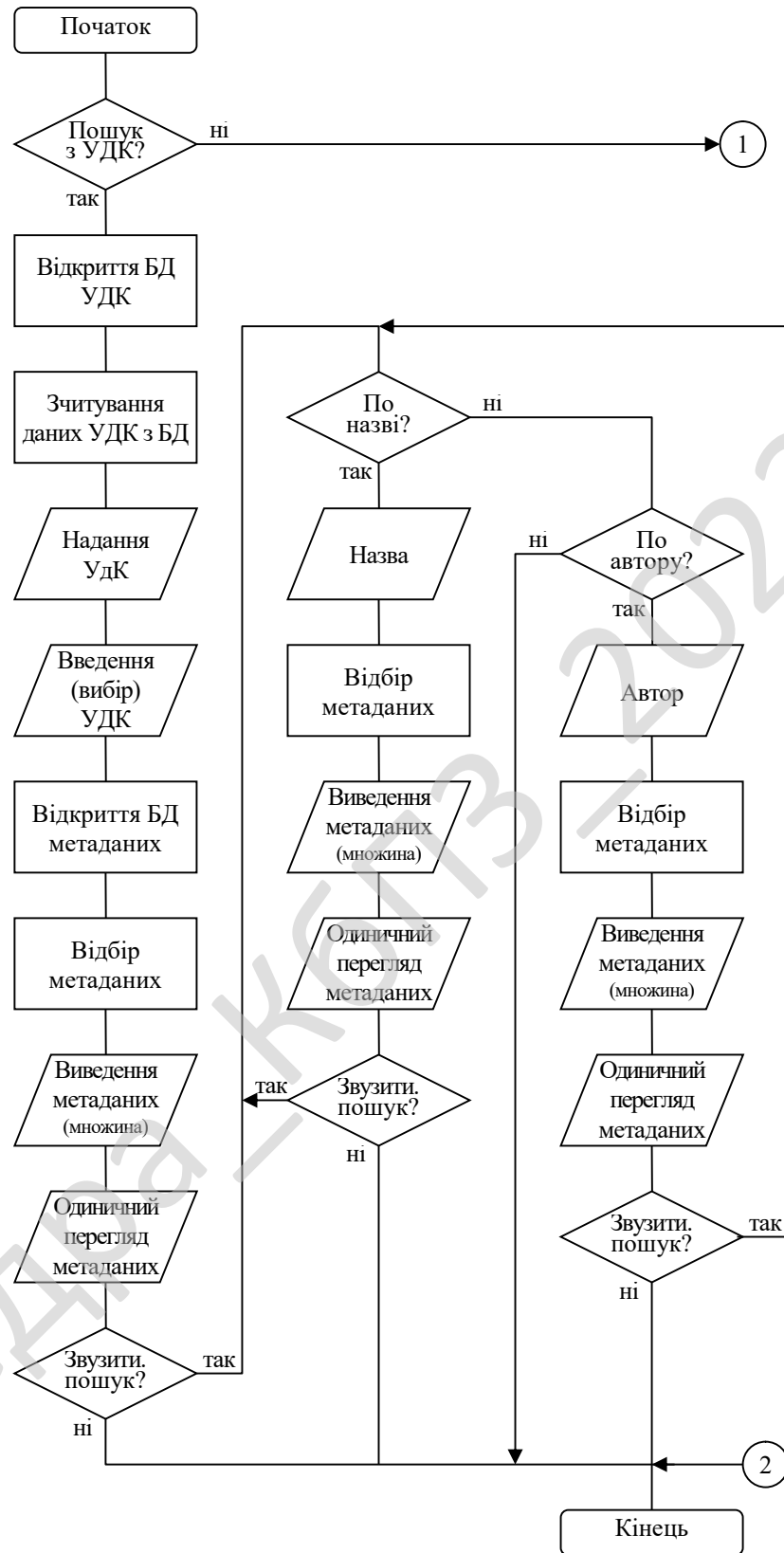


Рисунок 4.7 - Блок-схема роботи підпрограми відбору метаданих, частина з розширеним пошуком

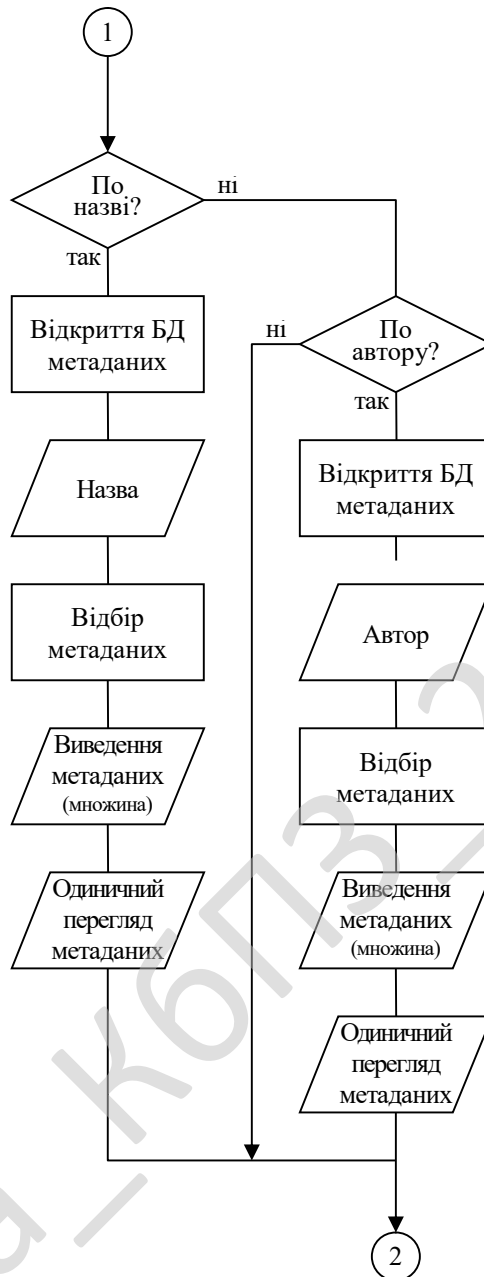


Рисунок 4.8 - Блок-схема роботи підпрограми відбору метаданих, частина з пошуком за назвою та авторами

Після завершення уточнення в обох випадках доступний перехід на повторне уточнення, альтернативний розвиток події вихід з режиму відбору метаданих.

Альтернативою до пошуку по УДК є пошук по назві, вибір якого може слідувати випадку не активації команди вибору па УДК. Проводиться

пошуку по назві (у випадку активації) за наступною послідовністю: відкриття БД метадані, введення назви, відбір метаданих з урахуванням назви, множинне представлення сукупності метаданих відібраних за запитом, одиничний сфокусований перегляд окремого елемента з множини відібраної за запитом, після чого проходить вихід з режиму відбору метаданих. У випадку коли не активації операції пошуку по назві, проводиться операція (у випадку активації) пошуку по автору за наступною послідовністю: відкриття БД метадані, введення даних автора, відбір метаданих з урахуванням умови пошуку, множинне представлення сукупності метаданих відібраних за запитом, одиничний сфокусований перегляд окремого елемента з множини відібраної за запитом, після чого теж проходить вихід з режиму відбору метаданих.

Після входження в процес відбору метаданих програмно передбачена можливість виходу з процесу відбору до завершення його виконання.

Для зберігання даних, які використовуються програмним продуктом передбачена наступна їх наступна організація:

- дані про користувачів, їх паролі та права, враховуючи відносно невеликий їх об'єм, пропонується зберігати в типізованому файлі відповідно наступній структурі: ФІО рядкове поле довжиною 20 символів, wr рядкове поле довжиною 20 символів, prav цілочислене поле;

- дані УДК та його опис, враховуючи їх об'єм, пропонується зберігати в БД відповідно наступній структурі: UDR текстове поле довжиною 20 символів, ОР текстове поле довжиною 255 символів;

- метадані: УДК, назву джерела, найменування авторів, місце видавництва, рік видавництва, видавництво, кількість сторінок, рефератний опис та ББК, враховуючи їх об'єм, пропонується зберігати в БД відповідно наступній структурі: UDR текстове поле довжиною 20 символів, NAME

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42





## 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Загалом використання програмного забезпечення вимагає розгляду наступних завдань: підготовка користувачів програмного забезпечення, встановлення програмного забезпечення, його запуск та налагодження, контроль працездатності та супровід програмного забезпечення.

Підготовка користувачів програмного забезпечення, що розроблялося, загалом не вимагає ніяких специфічних навиків. Програма оснащена простим і інтуїтивно понятим інтерфейсом, і користувача, що має загальні навички роботи з комп'ютерними програмами, без будь-яких труднощів він може відразу в ній працювати. То ж перед введенням в експлуатацію системи немає необхідності у проведенні спеціальних навчань персоналу, а тому вони і не проводяться.

Запуск та налагодження програми є стандартним для програм, написаних для сімейства ОС Windows. Враховуючи те, що файлом програми є E-Katalog.exe, запуск програми може проводитися так: з меню «програми» операційної ОС (якщо прописано); з меню «виконати» операційної ОС; з робочого столу ярликом програми (якщо встановлено сам ярлик на робочому столі для програми); з панелі задач (якщо встановлено ярлик на панелі швидкого запуску); шляхом натиснення комбінацій гарячих кнопок для запуску програми (якщо така комбінація прописана для даної програми); автоматично при завантаженні ОС (якщо вказано автоматичне завантаження при запуску ОС); автоматично при підключенні зовнішнього з'ємного носія (якщо автоматичний запуск програми прописано); з файлових менеджерів при запуску файла програми тощо.

Проте для запуску програмного забезпечення необхідно виконати ряд первинних налаштувань - встановлення ПЗ (інсталяція). Порядок встановлення розробленого програмного забезпечення не відрізняється від

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

загально прийнятих - для запуску програми необхідно й достатньо організувати доступ до файлів програми, які можуть бути записані на жорсткий диск комп'ютера (перенесені, як звичайні файли, які були скопійовані з іншого комп'ютера із встановленою програмою; інсталювані з дистрибутива - спеціально скомпонованого файлу для завантаження; розпаковано з архіву, скопійовано з мережі, тощо) або принесені (монтовані) на зовнішньому з'ємному носію.

Обов'язковою умовою для працездатності системи є наявний доступ до системного реєстру, в якому зберігаються або заносяться дані. В даному конкретному випадку до системного реєстру в розділ HKEY\_LOCAL\_MACHINE блок параметрів HardWare заноситься параметр sp значення якого має співпадати з серійним номером жорсткого диска, порівнюючи згадані величини програмна визначає правомірність застосування програмного забезпечення. Для занесення sp до системного реєстру рекомендовано використати редактор системного реєстру regedit запуск якого можна провести набрав його назву в командному рядку операційної системи. Сам редактор входить до поставки ОС сімейства WINDOWS.

Іною необхідною умовою для забезпечення працездатності є підключення БД. Для реалізації демонстраційного прикладу в якості розміщення обрано файл db.mdb, метадані - таблиця A та сортований в порядку зростання NAME запит B, а УДК - таблиця X та сортований в порядку зростання UDK запит Y. Структуру БД описано в 4.2 Доступ до даних організовано за допомогою BDE Administrator через DataBaseName = Z Загалом модно використовувати різноманітні контейнери для зберігання БД, основною умовою їх використання є їх внутрішня структура, обмеженням виступають тільки можливості BDE Administrator на їх підключення (обмежений набір типів БД, що підключаються). До функціоналу BDE Administrator входить керування доступом до БД встановленням User Name та Password3 значення яких

там і прописується (задається). Під час запуску програми для опосередкованого підключення через BDE Administrator до БД вимагатиметься їх введення на запит, рисунок 5.1.

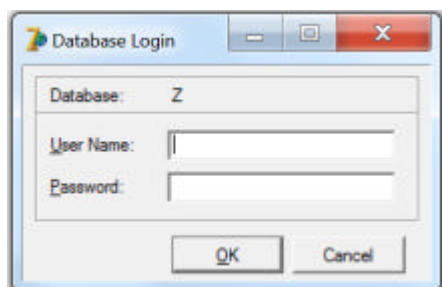


Рисунок 5.1 – Запит на підключення БД

Типізований файл збереження Login, пароля та прав носить назву USERS. Загалом його наявність при первинному запуску програмного забезпечення необов'язкова. При відсутності файлу він створюється автоматично куди автоматично додається користувач ADMIN з встановленим паролем 1 й необмеженими правами. Запуск ПЗ проводиться після авторизації користувача, рисунок 5.2.

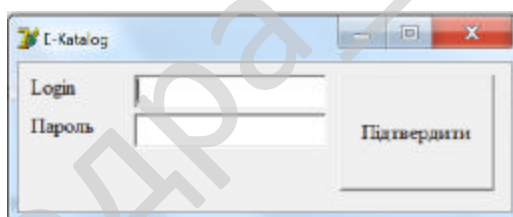


Рисунок 5.2 – Авторизація користувача

Функціонал ADMIN дозволяє вносити та видаляти користувачів, міняти їм паролі - виконувати функції адміністрування програмою, рисунок 5.3. Загалом функціонал ADMIN нічим не обмежений, до його можливостей входять всі заявлені функції, рисунок 5.3.

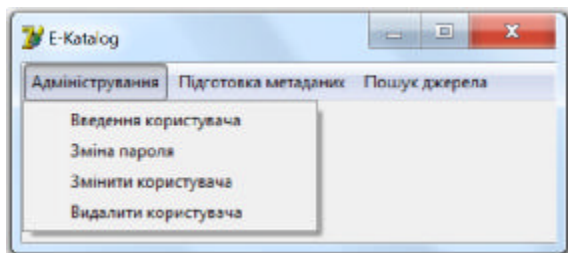


Рисунок 5.3– Функції адміністрування

Функціонал користувача з правами спеціаліста включає тільки функцію зміна користувача з функцій адміністрування програмою та повними наборами функцій підготовки метаданих (рисунок 5.4.) а пошук джерела (відбору метаданих, рисунок 5.5)

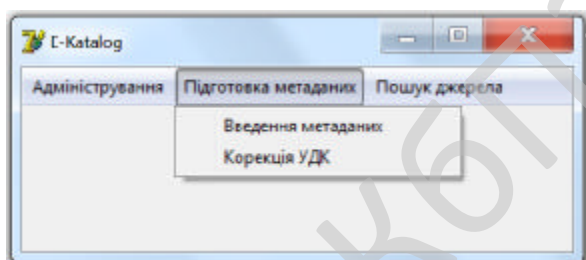


Рисунок 5.4 - Функції підготовки метаданих

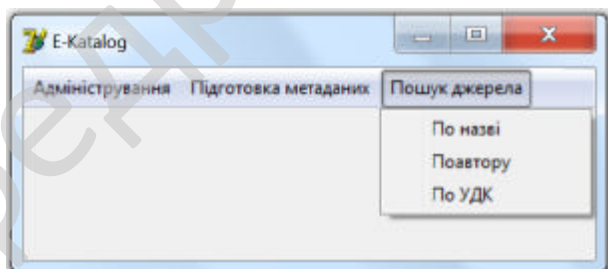


Рисунок 5.5 - Функції пошук джерела (відбору метаданих)



## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання бакалаврської роботи, призначене для ведення електронних каталогів.

У бакалаврській роботі наведено теоретичне узагальнення й вирішення наукового завдання дослідження методів побудови електронних каталогів.

Вирішення даного завдання полягало у здійсненні наступних задач:

- було проведено огляд існуючих програмних рішень для ведення електронних каталогів;
- досліджено та обрано засоби розробки електронного каталогу;
- розроблено структуру та принципи функціонування електронного каталогу;
- проведено аналіз процесів що протікають при відпрацюванні електронного каталогу;
- розроблено алгоритм роботи електронного каталогу;
- на основі отриманих результатів досліджень створена програмна реалізація електронного каталогу.

Розроблені під час виконання бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання ведення електронного каталогу.

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; побудовано алгоритм і вибрано середовище розробки.

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

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>50</b>

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

Програма реалізована на мові високого рівня Delphi. Саме ця 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 багатозначної операційної системи сімейства Windows.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати один із загально розповсюджених методів - прив'язка до параметрів комп'ютера, в нашому випадку організовано прив'язку до жорсткого диску.

У 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.

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51





27. Система MAPK-SQL. [Электронный ресурс] - Режим доступа: <http://www.gpntb.ru/win/inter-events/crimea2001/tom/sec2a/Doc5.html>
28. АБИС [Электронный ресурс] - Режим доступа: <http://www.artinfo.ru/eva/EVA2000M/eva-papers/200015/Popov-R.html>
29. Автоматизированная библиотечноинформационная система Руслан-Нео / Общее описание функциональные характеристики программного обеспечения и информация, необходимая для установки и эксплуатации программного обеспечения. ООО «Открытые Библиотечные Системы» 2019. -18сс.
30. Руслан. Назначение и принципы организации системы [Электронный ресурс] - Режим доступа: <http://abis.ruslan.ru/ruslan/products/ruslanls/common>
31. АБИС «Руслан-Нео» [Электронный ресурс] - Режим доступа: <https://obs.ruslan.ru/products/abis-ruslan-neo/>
32. Delphi 7 Обоснование выбора [Электронный ресурс] - Режим доступа: [http://barsicne.narod.ru/publ/delphi\\_7\\_obosnovanie\\_vybora/1-1-0-1](http://barsicne.narod.ru/publ/delphi_7_obosnovanie_vybora/1-1-0-1)
33. Культин Н. Основы программирования в Delphi 7 / Никита Культин. - М.: БХВ-Петербург, 2013. - 640 с.
34. Осипов Д. Delphi. Профессиональное программирование / Дмитрий Осипов. - М.: Символ-плюс, 2013. -820с.
35. Кэнту М. Delphi 7: Для профессионалов/ Кэнту М.- СПб.: Питер, 2004. - 1101 с.
36. Архангельский А.Я. Программирование в Delphi 7./ Архангельский А.Я. - М.: ООО «Бином-Пресс», 2003 г. -1152с.
37. Дарахвелидзе П. Г., Марков Е. П. Программирование в Delphi 7. / Дарахвелидзе П. Г., Марков Е. П.- СПб.: БХВ-Петербург, 2003. - 784 с.
38. Delphi. Работа с HDD, CD-ROM, ZIP - [Электронный ресурс] - Режим доступа: <http://z-oleg.com/delphi/hardw1.htm>
39. Delphi FAQ [Электронный ресурс] - Режим доступа: [http://www.delphisources.ru/pages/faq/base/registry\\_influence.html](http://www.delphisources.ru/pages/faq/base/registry_influence.html)
40. Когаловский М. Р. Перспективные технологии информационных систем. - М.: ДМК Пресс; М.: Компания АйТи, 2003. - 288 с.

					<b>ВКРБ-123.23.0006.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54



Додаток А  
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування	Ошибка! Закладка не определена.
2	Підстава для розробки	2
3	Мета та призначення розробки	2
4	Джерела розробки	2
5	Технічні вимоги	2
5.1	Склад продукції	2
5.2	Показники призначення	3
5.3	Вимоги до функціональних характеристик	3
5.4	Вимоги до архітектури	3
5.5	Вимоги до надійності	4
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	5
5.8.3	Вхідні дані	5
5.8.4	Вихідні дані	5
6	Вимоги до програмної документації	5
7	Перелік документів, які необхідно розробити:	5
8	Етапи розробки	6
9	Порядок контролю і приймання	6

					<b>ВКРБ-123.23.0006.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив		Литвиненко Д.О.			<i>Програмне забезпечення інформаційної каталожної системи</i>	Літ.	Аркуш	Аркушів
Перевірів		Пархоменко Ю.М.				Б	1	6
Н. Контр.		Гермак В.С.			ЦНТУ КМ-19			
Затв.		Смірнов О.А.						

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку інформаційної каталожної системи.

## 2 Підстава для розробки

Підставою для розробки служить завдання на бакалаврську дипломну роботу, видане на кафедрі програмування комп'ютерних систем та мереж (нак. №10-02 від 5.01.2023 року).

## 3 Мета та призначення розробки

Метою бакалаврської дипломної роботи є дослідження та програмна реалізація інформаційної каталожної системи.

## 4 Джерела розробки

Джерелом цієї бакалаврської роботи є стосовна до теми література й існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- аналіз існуючих комп'ютеризованих АБІС;
- вибір і обґрунтування методів та засобів реалізації проекту;
- розробка структур даних і механізму їхньої взаємодії,
- також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи.

					<b>ВКРБ-123.23.0006.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

## 5.2 Показники призначення

Система повинна забезпечувати:

- виконання сукупності процесів, що направлені на створення та підтримку функціонування каталогів (процесів створення, зберігання та перетворення метаданих які представляють інформаційні ресурси: книги, документи, аудіо- та відеозаписи, надаючи таку інформацію як імена авторів та назви ресурсів, рік та місце їх створення, іншу описову інформації ресурсу);
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно містити обмежень на версію драйверів операційної системи, розширення структур БД, форм, засобів, звітів і запитів.

## 5.4 Вимоги до архітектури

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

					<b>ВКРБ-123.23.0006.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.5 Вимоги до надійності

Програмні модулі написані за всіма правилами, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні відповідати 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 сімейства Windows 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 сімейства Windows.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/1024 Мб/40 Gb/VGA 17" 128Мб або сумісні з ним.

					<b>ВКРБ-123.23.0006.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

## 5.8.2 Мова програмування

Середовище Delphi версії 7.

## 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 Перелік документів, які необхідно розробити:

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму – 1 аркуш.
- Блок-схема алгоритму роботи – 1 аркуш.
- Пояснювальна записка – 55 аркушів.

					<b>ВКРБ-123.23.0006.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 8 Етапи розробки

На рівні проекту розробляються (терміни виконання етапів див. у "Завданні на бакалаврську дипломну роботу"):

8.1 Збір і обробка інформації по темі бакалаврської дипломної роботи.  
Постановка задачі на виконання бакалаврської дипломної роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень бакалаврської дипломної роботи.

8.3 Розробка функціональних схем, блок-схем алгоритмів роботи програмного забезпечення компоненту.

8.4 Побудова схем взаємодії структур даних.

8.5 Створення програмного продукту.

8.6 Відлагодження компоненту, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 9 Порядок контролю і приймання

9.1 Подання бакалаврської дипломної роботи на попередній захист

16.05. 2023 р.

9.2 Подання бакалаврської дипломної роботи на захист 21.05.2023 р.

					<b>ВКРБ-123.23.0006.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

\_\_\_\_\_ Пархоменко Ю.М.

*Програмне забезпечення інформаційної каталожної системи*

Лістинг програми

Код документу 12

Носій: DVD-RW диск

Загальна кількість аркушів: 26

Літера: РП

Кропивницький 2023

```

// ----- u.pas ----- Програмный модуль
unit Unit1;
interface
uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms, Dialogs, StdCtrls, Registry, Menus,
    Grids, DB, DBTables, DBGrids, ExtCtrls, DBCtrls;
type
TForm1 = class(TForm)
Memo1: TMemo; Label1: TLabel; Label2: TLabel; Edit1: TEdit;
Edit2: TEdit; Button1: TButton; MainMenu1: TMainMenu;
N11: TMenuItem; N21: TMenuItem; N31: TMenuItem; N41: TMenuItem;
N1: TMenuItem; N2: TMenuItem; N3: TMenuItem; N4: TMenuItem;
N5: TMenuItem; N6: TMenuItem; N7: TMenuItem; Button2: TButton;
Button3: TButton; N9: TMenuItem; StringGrid1: TStringGrid;
StringGrid2: TStringGrid; Button4: TButton; Button5: TButton;
Table1: TTable; DataSource1: TDataSource; DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator; StringGrid3: TStringGrid;
Button6: TButton; Label3: TLabel; Label4: TLabel;
Label5: TLabel; Label6: TLabel; Edit3: TEdit; Button7: TButton;
Table2: TTable; DataSource2: TDataSource; DBGrid2: TDBGrid;
DBNavigator2: TDBNavigator; CheckBox1: TCheckBox;
CheckBox2: TCheckBox; Button8: TButton; Edit4: TEdit;
Edit5: TEdit; Label7: TLabel; Label8: TLabel;
procedure AddK(Sender: TObject); procedure MM1(Sender: TObject);
procedure NotVisible(Sender: TObject); procedure login(Sender: TObject);
function av(Sender: TObject):boolean; procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject); procedure N2Click(Sender: TObject);
procedure N21Click(Sender: TObject); procedure Button2Click(Sender:
TObject);
    procedure Button3Click(Sender: TObject); procedure N9Click(Sender: TObject);
    procedure StringGrid1SelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean); procedure N1Click(Sender: TObject);
    procedure StringGrid1SelectCel2(Sender: TObject; ACol, ARow: Integer; var
CanSelect: Boolean); procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject); procedure dd(Sender: TObject);
    procedure Button6Click(Sender: TObject); procedure SEEK_NAME(Sender:
TObject);
    Procedure StringGrid3SelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean); procedure N5Click(Sender: TObject);
    procedure N41Click(Sender: TObject); procedure Button7Click(Sender:
TObject);
    procedure N6Click(Sender: TObject); procedure SEEK_avtor(Sender: TObject);
    procedure N3Click(Sender: TObject); procedure N7Click(Sender: TObject);

```

```

procedure BK(Sender: TObject); procedure DBGrid2CellClick(Column: TColumn);
procedure Button8Click(Sender: TObject); procedure CheckBox2Click(Sender:
TObject);
procedure CheckBox1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
tuser = record
  FIO:string[20];    PW:string[20];    prav:integer;
end;
var
  Form1: TForm1; FT,FL,FH,FW,prava,K,KN,JJJ:integer;
  users:array [0..100] of tuser;  Fusers:File of tuser ;
implementation
{$R *.dfm}
procedure TForm1.AddK(Sender: TObject);
var i:integer;
begin
IF (NOT(EDIT1.TEXT='')) AND (NOT(EDIT2.TEXT='')) THEN
  BEGIN users[k].FIO:=EDIT1.TEXT;  users[k].PW:=EDIT2.TEXT;
  AssignFile(Fusers,'users');  Rewrite(Fusers);
  FOR i:=0 to k do  Write(Fusers,users[I]);
  CloseFile(Fusers);  k:=k+1;  MM1(Sender);  END;
END;
procedure TForm1.MM1(Sender: TObject);
var NewItem: TMenuItem;
begin
NotVisible(Sender);
if (prava=3) or (prava=0)  then begin FORM1.MainMenu1.Items[0].Visible :=
TRUE;
FORM1.MainMenu1.Items[0][0].Visible:= false;
FORM1.MainMenu1.Items[0][1].Visible:= false;
FORM1.MainMenu1.Items[0][3].Visible:= false;
FORM1.MainMenu1.Items[2].Visible := TRUE; end;
if prava=2 then begin FORM1.MainMenu1.Items[0].Visible := TRUE;
FORM1.MainMenu1.Items[0][0].Visible:= false;
FORM1.MainMenu1.Items[0][1].Visible:= false;
FORM1.MainMenu1.Items[0][3].Visible:= false;
FORM1.MainMenu1.Items[1].Visible := TRUE;
FORM1.MainMenu1.Items[2].Visible := TRUE; end;
if prava=1 then begin  FORM1.MainMenu1.Items[0][0].Visible:= TRUE;

```

```

FORM1.MainMenu1.Items[0][1].Visible:= TRUE;
FORM1.MainMenu1.Items[0][2].Visible:= TRUE;
FORM1.MainMenu1.Items[0][3].Visible:= TRUE;
FORM1.MainMenu1.Items[0].Visible := TRUE;
FORM1.MainMenu1.Items[1].Visible := TRUE;
FORM1.MainMenu1.Items[2].Visible := TRUE;  end;

END;

procedure TForm1.NotVisible(Sender: TObject);
begin
  dbgrid1.visible:=false; DBNavigator1.visible:=false;
  dbgrid2.visible:=false; DBNavigator2.visible:=false;
  CheckBox1.Visible:=false; CheckBox2.Visible:=false;
  form1.Top:=FT; form1.Left:=FL; form1.Height:=FH; form1.Width:=FW;
  StringGrid1.visible:=false; StringGrid2.visible:=false;
  StringGrid3.visible:=false; memo1.visible:=false;
StringGrid1.visible:=false;
  label11.visible:=false; label2.visible:=false; label3.visible:=false;
  label4.visible:=false;label5.visible:=false; label6.visible:=false;
  label7.visible:=false;label8.visible:=false;
  label11.caption:=''; label2.caption:=''; label3.caption:='';
  label4.caption:=''; label5.caption:=''; label6.caption:='';
  label7.caption:='';label8.caption:=''; edit1.visible:=false;
  edit2.visible:=false; edit3.visible:=false; edit4.visible:=false;
  edit5.visible:=false; button1.visible:=false; button2.visible:=false;
  button3.visible:=false; button4.visible:=false;
button5.visible:=false;
  button6.visible:=false; button7.visible:=false;
button8.visible:=false;
  end;
  procedure TForm1.login(Sender: TObject);
  begin
    FT:= form1.Top; FL:=form1.left; FH:= form1.Height; FW:= form1.Width;
    NotVisible(Sender); prava:=0; FORM1.MainMenu1.Items[0].Visible :=
false;
    FORM1.MainMenu1.Items[1].Visible := false;FORM1.MainMenu1.Items[2].Visible :=
false;
    label11.visible:=true; label2.visible:=true; label11.caption:='Login';
    label2.caption:='Пароль'; form1.Top:=round(Screen.Height/2-80);
    form1.Left:=round(Screen.Width/2-200); form1.Height:=160;
    form1.Width:=400; label11.top:=10; label11.Left:=10; label2.top:=40;
    label2.Left:=10; label11.font.name:='Times New Roman';
    label11.Font.Size:=12; label2.font.name:='Times New Roman';
    label2.Font.Size:=12; edit1.top:=10; edit1.Left:=90; edit2.top:=40;

```

```

edit2.Left:=90; edit1.font.name:='Times New Roman';
edit1.Font.Size:=12;
edit2.font.name:='Times New Roman'; edit2.Font.Size:=12;
edit1.visible:=true;
edit2.visible:=true; edit1.Width:=150; edit2.Width:=150;
edit1.text:='';
edit2.text:=''; button1.visible:=true; button1.caption:='Підтвердити';
button1.font.name:='Times New Roman'; button1.Font.Size:=12;
button1.top:=10; button1.Left:=250; button1.Width:=120;
button1.Height:=90;
if not (FileExists('users')) then
begin Memol.Lines.Add('паролей нет'); users[0].FIO:='ADMIN';
users[0].PW:='1'; users[0].prav:=1; AssignFile(Fusers,'users');
Rewrite(Fusers); Write(Fusers,users[0]); CloseFile(Fusers); end;
end;
function TForm1.av(Sender: TObject):boolean;
var VolumeName,FileSystemName : array [0..MAX_PATH-1] of Char;
VolumeSerialNo,snr : DWord;MaxComponentLength,FileSystemFlags:
Cardinal;
reg:TRegistry; RDate:tDate;
begin
GetVolumeInformation('C:\',VolumeName,MAX_PATH,@VolumeSerialNo,
MaxComponentLength,FileSystemFlags, FileSystemName,MAX_PATH);
reg:=TRegistry.Create; reg.RootKey:=HKEY_LOCAL_MACHINE;
reg.OpenKey('HardWare', true); snr:=REG.readInteger('snr');
if snr=VolumeSerialNo then Result:=true else Result:=false;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
form1.Top:=0;form1.Left:=0;form1.Height:=Screen.Height;
form1.Width:=Screen.Width;
FT:= form1.Top;FL:=form1.Top; FH:= form1.Height; FW:= form1.Width;
NotVisible(Sender); form1.caption:='E-Katalog';
if av(Sender)
then begin login(Sender); end
else begin FORM1.COLOR:=CLWHITE;
form1.caption:='Програма E-Katalog неліцензована';
Memol.Lines.Add('ПРОГРАМА НЕ ЛІЦЕНЗОВАНА');
Memol.Lines.Add('РОБОТУ ПРОГРАМИ ЗАБЛОКОВАНО!!!');
memol.visible:=true; Memol.ReadOnly:=true;
Memol.font.name:='Times New Roman'; Memol.Font.Size:=20;
end;
end;
end;

```

```

procedure TForm1.Button1Click(Sender: TObject);
var i: integer;
begin
  prava:=3; Mem1.Lines.clear; AssignFile(Fusers,'users');
ReSet(Fusers);I:=0;
  WHILE NOT( EOF(Fusers)) DO
    begin
      read(Fusers,users[i]);
      if (users[i].FIO=edit1.text) and (users[i].PW=edit2.text)
        then prava:=users[i].prav;
      Mem1.top:=300; Mem1.Left:=0; Mem1.Height:=form1.Height;
      Mem1.Width:= form1.Width; I:=I+1; end;
    k:=i; CloseFile(Fusers); MM1(Sender);
  end;
procedure TForm1.N2Click(Sender: TObject);
begin NotVisible(Sender); LOGIN(Sender); end;
procedure TForm1.N21Click(Sender: TObject);
begin
  NotVisible(Sender); FT:= form1.Top; FL:=form1.left; FH:= form1.Height;
  FW:= form1.Width; FORM1.MainMenu1.Items[0].Visible := false;
  FORM1.MainMenu1.Items[1].Visible := false; FORM1.MainMenu1.Items[2].Visible
:= false;
  label1.visible:=true; label2.visible:=true; label1.caption:='Login';
  label2.caption:='Пароль'; form1.Top:=round(Screen.Height/2-80);
  form1.Left:=round(Screen.Width/2-200); form1.Height:=160;
  form1.Width:=400; label1.top:=10; label1.Left:=10; label2.top:=40;
  label2.Left:=10; label1.font.name:='Times New Roman';
label1.Font.Size:=12;
  label2.font.name:='Times New Roman'; label2.Font.Size:=12;
edit1.top:=10;
  edit1.Left:=90; edit2.top:=40; edit2.Left:=90; edit1.font.name:='Times New
Roman';
  edit1.Font.Size:=12; edit2.font.name:='Times New Roman';
  edit2.Font.Size:=12; edit1.visible:=true; edit2.visible:=true;
  edit1.Width:=150; edit2.Width:=150; edit1.ReadOnly:=FALSE;
  edit1.text:='';
  edit2.text:=''; button2.visible:=true; button2.caption:='АДМІН';
  button2.font.name:='Times New Roman'; button2.Font.Size:=12;
  button2.top:=10; button2.Left:=250; button2.Width:=120;
button2.Height:=40;
  button3.visible:=true; button3.caption:='СПЕЦІАЛІСТ';
  button3.font.name:='Times New Roman'; button3.Font.Size:=12;
button3.top:=55;

```

```

button3.Left:=250;button3.Width:=120;button3.Height:=40;
end;
procedure TForm1.Button2Click(Sender: TObject);
VAR FFF:BOOLEAN;i:integer;
begin
    FFF:=true; FOR i:=0 to k do if users[i].FIO=edit1.text then
FFF:=false;
        if fff then begin sers[k].prav:=1;AddK(Sender); end else
edit1.text:='';
        end;
    procedure TForm1.Button3Click(Sender: TObject);
    VAR FFF:BOOLEAN; i:integer;
    begin
        FFF:=true; FOR i:=0 to k do if users[i].FIO=edit1.text then
FFF:=false;
            if fff then begin users[k].prav:=2; AddK(Sender); end else
edit1.text:='';
            end;
        procedure TForm1.N9Click(Sender: TObject);
        VAR i:integer;
        begin
            NotVisible(Sender); FORM1.MainMenu1.Items[0].Visible:= false;
            FORM1.MainMenu1.Items[1].Visible:= false; FORM1.MainMenu1.Items[2].Visible:=
false;
            StringGrid1.Cells[0,0]:='Видалити користувача ';
            StringGrid1.top:=0; StringGrid1.Left:=0;
StringGrid1.Height:=form1.Height-75;
            StringGrid1.Width:= form1.Width; StringGrid1.visible:=true;
            StringGrid1.RowCount:=k; AssignFile(Fusers,'users'); ReSet(Fusers);
            read(Fusers,users[0]);CloseFile(Fusers);StringGrid1.RowCount:=k;
            StringGrid1.ColCount:=1; StringGrid1.ColWidths[0]:= form1.Width-20;
            for i:=1 to k do
                if users[i].prav=1 then StringGrid1.Cells[0,i]:=' ADM -
'+users[i].FIO
                    else StringGrid1.Cells[0,i]:=' СПЕЦ -
'+users[i].FIO;
                end;
        procedure TForm1.StringGrid1SelectCell(Sender: TObject; ACol,
            ARow: Integer; var CanSelect: Boolean);
        var i:integer;
        begin
            if ARow>0

```

```

then begin
Memor1.Lines.Add(IntToStr(ARow));AssignFile(Fusers,'users');
        Rewrite(Fusers);write(Fusers);; Seek(Fusers, 0);
Truncate(Fusers);
        for i:=ARow to k+2 do users[i]:=users[i+1];
        for i:=0 to k-1 do Write(Fusers,users[i]);
        Truncate(Fusers); CloseFile(Fusers);k:=k-1; end;
MM1(Sender);
end;
procedure TForm1.N1Click(Sender: TObject);
VAR i:integer;
begin
NotVisible(Sender);FORM1.MainMenu1.Items[0].Visible:= false;
FORM1.MainMenu1.Items[1].Visible:= false;
FORM1.MainMenu1.Items[2].Visible:= false;
StringGrid1.Cells[0,0]:='Видалити користувача ';
StringGrid2.visible:=TRUE; StringGrid2.top:=0; StringGrid2.Left:=0;
StringGrid2.Height:=form1.Height-75; StringGrid2.Width:= form1.Width;
StringGrid2.visible:=true; StringGrid2.RowCount:=k;
AssignFile(Fusers,'users'); ReSet(Fusers);read(Fusers,users[0]);
CloseFile(Fusers); StringGrid2.RowHeights[0]:=0;
StringGrid2.RowCount:=k+1;
StringGrid2.ColCount:=1; StringGrid2.ColWidths[0]:= form1.Width-20;
for i:=0 to k do
    if users[i].prav=1 then StringGrid2.Cells[0,i+1]:=' ADM -
'+users[i].FIO
    else StringGrid2.Cells[0,i+1]:=' СПЕЦ -
'+users[i].FIO;
end;
procedure TForm1.StringGrid1SelectCel2(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
var i:integer;
begin
kN:=ARow-1; FT:= form1.Top; FL:=form1.left; FH:= form1.Height;
FW:= form1.Width; NotVisible(Sender); FORM1.MainMenu1.Items[0].Visible :=
false;
FORM1.MainMenu1.Items[1].Visible := false; FORM1.MainMenu1.Items[2].Visible :=
false;
label1.visible:=true; label2.visible:=true;
label1.caption:='Login';
label2.caption:='Пароль'; form1.Top:=round(Screen.Height/2-80);
form1.Left:=round(Screen.Width/2-200); form1.Height:=160;
form1.Width:=400;

```

```

    label1.top:=10; label1.Left:=10; label2.top:=40; label2.Left:=10;
    label1.font.name:='Times New Roman'; label1.Font.Size:=12;
    label2.font.name:='Times New Roman'; label2.Font.Size:=12;
edit1.top:=10;
    edit1.Left:=90; edit2.top:=40; edit2.Left:=90; edit1.font.name:='Times New
Roman';
    edit1.Font.Size:=12; edit2.font.name:='Times New Roman';
edit2.Font.Size:=12;
    edit1.visible:=true; edit2.visible:=true; edit1.Width:=150;
edit2.Width:=150;
    edit1.text:=users[KN].FIO; edit2.text:=users[KN].PW;
button4.visible:=true;
    button4.caption:='Підтвердити'; button4.font.name:='Times New Roman';
    button4.Font.Size:=12; button4.top:=10; button4.Left:=250;
button4.Width:=120;
    button4.Height:=90; button5.visible:=true;
button5.caption:='Відміна';
    button5.font.name:='Times New Roman'; button5.Font.Size:=12;
    button5.top:=75; button5.Left:=20; button5.Width:=200;
button5.Height:=25;
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
users[KN].PW:=EDIT2.TEXT; AssignFile(Fusers,'users'); ReSET(Fusers);
SEEK(Fusers,KN); Write(Fusers,users[KN]); CloseFile(Fusers);
edit1.ReadOnly:=FALSE; MM1(Sender)
end;
procedure TForm1.Button5Click(Sender: TObject);
begin edit1.ReadOnly:=FALSE; MM1(Sender) end;
procedure TForm1.dd(Sender: TObject);
begin
label3.Left:=10; label3.top:=40; label3.Height:=400;
label3.Width:=400;
label3.font.name:='Times New Roman'; label3.Font.Size:=16;
label3.CAPTION:=StringGrid3.Cells[1,JJJ]; label3.visible:=TRUE;
label4.Left:=10; label4.top:=80; label4.Height:=400;
label4.Width:=400; label4.font.name:='Times New Roman';
label4.Font.Size:=16; label4.CAPTION:=StringGrid3.Cells[6,JJJ];
label4.visible:=TRUE; label5.WordWrap:=TRUE; label5.Left:=200;
label5.top:=10; label5.Height:=form1.Height-350 ; label5.Height:=50
;
label5.Width:= form1.Width-250; label5.font.name:='Times New Roman';
label5.Font.Size:=18; label5.visible:=true;

```

```

label5.CAPTION:=StringGrid3.Cells[3,JJJ];
label6.WordWrap:=TRUE; label6.Left:=200; label6.top:=80;
label6.Width:=150 ; label6.Width:= form1.Width-300;
label6.font.name:='Times New Roman'; label6.Font.Size:=18;
label6.CAPTION:=StringGrid3.Cells[4,JJJ]; label6.visible:=true;
label7.Left:=200; label7.top:=150; label7.Width:=150 ;
label7.Width:= form1.Width-300; label7.font.name:='Times New Roman';
label7.Font.Size:=18; label7.CAPTION:=StringGrid3.Cells[5,JJJ];
label7.visible:=true; label8.WordWrap:=TRUE; label8.Left:=10;
label8.top:=180; label8.Width:=150 ; label8.Width:= form1.Width-40;
label8.font.name:='Times New Roman'; label8.Font.Size:=14;
label8.CAPTION:=StringGrid3.Cells[7,JJJ]; label8.visible:=true;
end;
procedure TForm1.Button6Click(Sender: TObject);
begin NotVisible(Sender); SEEK_NAME(Sender); end;
procedure TForm1.SEEK_NAME(Sender: TObject);
VAR I,ii:INTEGER; s1,s2:string;
begin
Table1.Active:=true; StringGrid3.visible:=true;
FOR I:=0 TO StringGrid3.RowCount DO
    FOR ii:=0 TO 8 DO StringGrid3.Cells[ii,i]:='';
StringGrid3.ColCount:=9; StringGrid3.ColWidths[0]:= 50;
StringGrid3.ColWidths[1]:= 130; StringGrid3.ColWidths[2]:=
form1.Width-210;
FOR I:=3 TO 8 DO StringGrid3.ColWidths[I]:= 0;
StringGrid3.FONT.Size:=14; StringGrid3.RowCount:=1;
StringGrid3.DefaultRowHeight:= 30; StringGrid3.ROWHEIGHTS[0]:= 0;
DBGrid1.DataSource.Dataset.First;s2:=edit3.text; ii:=0;
FOR I:=1 TO DBGrid1.DataSource.Dataset.RecordCount DO
BEGIN
s1:=DBGrid1.DataSource.Dataset.FieldName('NAME').AsString;
if (pos(TrimRight(s2),TrimRight(s1)))>0 then
begin ii:=ii+1; StringGrid3.RowCount:=StringGrid3.RowCount+1;
StringGrid3.Cells[0,II]:=' '+INTTOSTR(ii)+'.';
StringGrid3.Cells[1,ii]:=
    DBGrid1.DataSource.Dataset.FieldName('UDK').AsString;
StringGrid3.Cells[2,II]:=
    DBGrid1.DataSource.Dataset.FieldName('NAME').AsString
+' '+DBGrid1.DataSource.Dataset.FieldName('FIO').AsString
+'- '+DBGrid1.DataSource.Dataset.FieldName('PLACE').AsString
+': '+DBGrid1.DataSource.Dataset.FieldName('IZD').AsString
+', -
'+DBGrid1.DataSource.Dataset.FieldName('PAGE').AsString+'c.';

```

```

StringGrid3.Cells[3,II]:=
    DBGrid1.DataSource.DataSet.FieldByName('NAME').AsString;
StringGrid3.Cells[4,II]:=
    DBGrid1.DataSource.DataSet.FieldByName('FIO').AsString;
StringGrid3.Cells[5,II]:=
    DBGrid1.DataSource.DataSet.FieldByName('PLACE').AsString
    +': '+DBGrid1.DataSource.DataSet.FieldByName('IZD').AsString
    +' . -
'+DBGrid1.DataSource.DataSet.FieldByName('PAGE').AsString+'c.';
StringGrid3.Cells[6,II]:=
    DBGrid1.DataSource.DataSet.FieldByName('BBS').AsString;
StringGrid3.Cells[7,II]:=
    DBGrid1.DataSource.DataSet.FieldByName('OPIS').AsString;
end;
DBGrid1.DataSource.DataSet.Next;
END;
DBGrid1.Datasource.Dataset.First; StringGrid3.top:=270;
StringGrid3.Left:=0; StringGrid3.Height:=form1.Height-75-90-200;
StringGrid3.Width:= form1.Width;
IF ii<1
    THEN begin LABEL8.Caption:=' !!! ПОШУК РЕЗУЛЬТАТУ НЕ ДАВ';
        label8.WordWrap:=FALSE; label8.Left:=10; label8.top:=180;
        label8.Width:=150 ; label8.Width:= form1.Width-40;
        label8.font.name:='Times New Roman'; label8.Font.Size:=20;
        label8.visible:=true; end
    ELSE BEGIN JJJ:=1; DD(Sender); JJJ:=1; DD(Sender); END;
edit3.visible:=true; button6.visible:=true;
end;
Procedure TForm1.StringGrid3SelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
Begin if ARow>0 then begin JJJ:=ARow; DD(Sender); end; end;
procedure TForm1.N5Click(Sender: TObject);
begin
    NotVisible(Sender); Edit3.Left:=183;Edit3.top:=240; Edit3.Height:=28
;
    Edit3.Width:= form1.Width-210; Edit3.font.name:='Times New Roman';
    Edit3.Font.Size:=14; edit3.text:=''; edit3.visible:=true;
    button6.Left:=10; button6.top:=240; button6.Height:=28 ;
button6.Width:=160;
    button6.font.name:='Times New Roman'; button6.Font.Size:=14;
    button6.caption:='Знайти по назві: '; button6.visible:=true;
end;
procedure TForm1.N41Click(Sender: TObject);

```

```

begin
  NotVisible(Sender); DBGrid1.visible:=true;
DBGrid1.Datasource.Dataset.First;
  DBGrid1.top:=0; DBGrid1.Left:=0; DBGrid1.Height:=form1.Height-127;
  DBGrid1.Width:= form1.Width; DBGrid1.font.name:='Times New Roman';
  DBGrid1.Font.Size:=14; DBNavigator1.top:=form1.Height-120;
  DBNavigator1.Left:=0; DBNavigator1.visible:=true;
end;
procedure TForm1.Button7Click(Sender: TObject);
begin NotVisible(Sender); SEEK_автор(Sender); end;
procedure TForm1.SEEK_автор(Sender: TObject);
VAR I,ii:INTEGER; s1,s2:string;
begin
  Table1.Active:=true; StringGrid3.visible:=true;
  FOR I:=0 TO StringGrid3.RowCount DO
    FOR ii:=0 TO 8 DO StringGrid3.Cells[ii,i]:='';
  StringGrid3.ColCount:=9; StringGrid3.ColWidths[0]:= 50;
  StringGrid3.ColWidths[1]:= 130; StringGrid3.ColWidths[2]:=
form1.Width-210;
  FOR I:=3 TO 8 DO StringGrid3.ColWidths[I]:= 0;
  StringGrid3.FONT.Size:=14; StringGrid3.RowCount:=1;
  StringGrid3.DefaultRowHeight:= 30; StringGrid3.ROWHEIGHTS[0]:= 0;
  DBGrid1.Datasource.Dataset.First; s2:=edit3.text; ii:=0;
  FOR I:=1 TO DBGrid1.DataSource.DataSet.RecordCount DO
    BEGIN s1:=DBGrid1.DataSource.DataSet.FieldByName('FIO').AsString;
    if (pos(TrimRight(s2),TrimRight(s1)))>0 then
      begin
        ii:=ii+1; StringGrid3.RowCount:=StringGrid3.RowCount+1;
        StringGrid3.Cells[0,II]:=' '+INTTOSTR(ii)+'.';
        StringGrid3.Cells[1,ii]:=
          DBGrid1.DataSource.DataSet.FieldByName('UDK').AsString;
        StringGrid3.Cells[2,II]:=
          DBGrid1.DataSource.DataSet.FieldByName('NAME').AsString
          +'.' +DBGrid1.DataSource.DataSet.FieldByName('FIO').AsString
          +'- ' +DBGrid1.DataSource.DataSet.FieldByName('PLACE').AsString
          +': ' +DBGrid1.DataSource.DataSet.FieldByName('IZD').AsString
          +', -
'+DBGrid1.DataSource.DataSet.FieldByName('PAGE').AsString+'c.';
        StringGrid3.Cells[3,II]:=
          DBGrid1.DataSource.DataSet.FieldByName('NAME').AsString;
        StringGrid3.Cells[4,II]:=
          DBGrid1.DataSource.DataSet.FieldByName('FIO').AsString;
        StringGrid3.Cells[5,II]:=

```

```

        DBGrid1.DataSource.DataSet.FieldByName('PLACE').AsString
        +': '+DBGrid1.DataSource.DataSet.FieldByName('IZD').AsString
        +' . -
'+DBGrid1.DataSource.DataSet.FieldByName('PAGE').AsString+'c.';
    StringGrid3.Cells[6,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('BBS').AsString;
    StringGrid3.Cells[7,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('OPIS').AsString;
    end;
    DBGrid1.DataSource.DataSet.Next;
    END;

    DBGrid1.Datasource.Dataset.First; StringGrid3.top:=270;
StringGrid3.Left:=0;
    StringGrid3.Height:=form1.Height-75-90-200; StringGrid3.Width:=
form1.Width;
    IF ii<1
    THEN begin LABEL8.Caption:=' !!! ПОШУК РЕЗУЛЬТАТУ НЕ ДАВ';
        label8.WordWrap:=FALSE; label8.Left:=10; label8.top:=180;
        label8.Width:=150 ; label8.Width:= form1.Width-40;
        label8.font.name:='Times New Roman'; label8.Font.Size:=20;
        label8.visible:=true; end
    ELSE BEGIN JJJ:=1; DD(Sender); JJJ:=1; DD(Sender); END;
    edit3.visible:=true; button7.visible:=true;
end;
    procedure TForm1.N6Click(Sender: TObject);
    begin
        NotVisible(Sender); Edit3.Left:=183; Edit3.top:=240;
Edit3.Height:=28 ;
        Edit3.Width:= form1.Width-210; Edit3.font.name:='Times New Roman';
        Edit3.Font.Size:=14; edit3.text:=''; edit3.visible:=true;
        button7.Left:=10; button7.top:=240; button7.Height:=28 ;
        button7.Width:=160; button7.font.name:='Times New Roman';
        button7.Font.Size:=14; button7.caption:='Знайти по автору: ';
        button7.visible:=true;
    end;
    procedure TForm1.N3Click(Sender: TObject);
    begin
        NotVisible(Sender); DBGrid2.Columns.Items[0].Width := 120;
        DBGrid2.Columns.Items[1].Width := form1.Width-170;
        DBGrid2.visible:=true; DBGrid2.Datasource.Dataset.First;
        DBGrid2.top:=0; DBGrid2.Left:=0; DBGrid2.Height:=form1.Height-127;
        DBGrid2.Width:= form1.Width; DBGrid2.font.name:='Times New Roman';
        DBGrid2.Font.Size:=14; DBNavigator2.top:=form1.Height-120;

```

```

    DBNavigator2.Left:=0; DBNavigator2.Visible:=true;
end;
procedure TForm1.N7Click(Sender: TObject);
begin
    NotVisible(Sender); Edit3.Left:=183; Edit3.Top:=240; Edit3.Height:=28
;
    Edit3.Width:= form1.Width-210; Edit3.Font.Name:='Times New Roman';
    Edit3.Font.Size:=14; edit3.Text:=''; edit3.Visible:=true;
    edit3.Visible:=true; button8.Left:=10; button8.Top:=240;
    button8.Height:=28; button8.Width:=160; button8.Font.Name:='Times New
Roman';
    button8.Font.Size:=14; button8.Caption:='Знайти по УДК';
    button8.Visible:=true; DBGrid2.DataSource.Dataset.First;
DBGrid2.Top:=270;
    DBGrid2.Left:=0; DBGrid2.Height:=form1.Height-75-90-200;
    DBGrid2.Width:= form1.Width; DBGrid2.Visible:=true;
    DBGrid2.Columns.Items[0].Width := 120;
    DBGrid2.Columns.Items[1].Width := form1.Width-155;
    DBGrid2.ReadOnly:=true; DBGrid2.Visible:=true;
    DBGrid2.Font.Name:='Times New Roman'; DBGrid2.Font.Size:=14;
end;
procedure TForm1.BK(Sender: TObject);
VAR I,ii:INTEGER; s1,s2,s3,A1,A2,A3:string;
begin
    Table1.Active:=true; StringGrid3.Visible:=true;
    FOR I:=0 TO StringGrid3.RowCount DO
        FOR ii:=0 TO 8 DO StringGrid3.Cells[ii,i]:='';
    StringGrid3.ColCount:=9; StringGrid3.ColWidths[0]:= 50;
    StringGrid3.ColWidths[1]:= 130; StringGrid3.ColWidths[2]:=
form1.Width-210;
    FOR I:=3 TO 8 DO StringGrid3.ColWidths[I]:= 0;
    StringGrid3.Font.Size:=14; StringGrid3.RowCount:=1;
    StringGrid3.DefaultRowHeight:= 30; StringGrid3.RowHeights[0]:= 0;
    DBGrid1.DataSource.Dataset.First; s1:=edit3.Text; s3:=edit4.Text;
    s2:=edit5.Text; ii:=0;
    FOR I:=1 TO DBGrid1.DataSource.DataSet.RecordCount DO
        BEGIN
            A1:=DBGrid1.DataSource.DataSet.FieldByName('UDK').AsString;
            A2:=DBGrid1.DataSource.DataSet.FieldByName('NAME').AsString;
            A3:=DBGrid1.DataSource.DataSet.FieldByName('FIO').AsString;
            if ((pos(TrimRight(s1),TrimRight(A1)))>0)
                and (not(CheckBox1.Checked) or
((pos(TrimRight(s2),TrimRight(A2)))>0))

```

```

        and (not (CheckBox2.Checked) or
((pos (TrimRight (s3), TrimRight (A3)))>0))
        then begin ii:=ii+1; StringGrid3.RowCount:=StringGrid3.RowCount+1;
StringGrid3.Cells[0,II]:=' '+INTTOSTR(ii)+'.';
StringGrid3.Cells[1,ii]:=
        DBGrid1.DataSource.DataSet.FieldByName('UDK').AsString;
StringGrid3.Cells[2,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('NAME').AsString
        +'.'+DBGrid1.DataSource.DataSet.FieldByName('FIO').AsString
        +'-'+DBGrid1.DataSource.DataSet.FieldByName('PLACE').AsString
        +':'+DBGrid1.DataSource.DataSet.FieldByName('IZD').AsString
        +', -
'+DBGrid1.DataSource.DataSet.FieldByName('PAGE').AsString+'c.';
StringGrid3.Cells[3,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('NAME').AsString;
StringGrid3.Cells[4,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('FIO').AsString;
StringGrid3.Cells[5,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('PLACE').AsString
        +':'+DBGrid1.DataSource.DataSet.FieldByName('IZD').AsString
        +'.' -
'+DBGrid1.DataSource.DataSet.FieldByName('PAGE').AsString+'c.';
StringGrid3.Cells[6,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('BBS').AsString;
StringGrid3.Cells[7,II]:=
        DBGrid1.DataSource.DataSet.FieldByName('OPIS').AsString;
end;
DBGrid1.DataSource.DataSet.Next;
END;
DBGrid1.Datasource.Dataset.First; StringGrid3.top:=302;
StringGrid3.Left:=0; StringGrid3.Height:=form1.Height-403;
StringGrid3.Width:= form1.Width;
IF ii<1
THEN begin LABEL8.Caption:=' !!! ПОШУК РЕЗУЛЬТАТУ НЕ ДАВ';
label8.WordWrap:=FALSE; label8.Left:=10; label8.top:=180;
label8.Width:=150 ; label8.Width:= form1.Width-40;
label8.font.name:='Times New Roman';label8.Font.Size:=20;
label8.visible:=true;
if (not (CheckBox1.Checked) and (not (CheckBox2.Checked) )
then begin CheckBox1.visible:=false;
CheckBox2.visible:=false;
edit4.visible:=false; edit5.visible:=false; edit3.visible:=false;
end;

```

```

        LABEL4.Caption:=''; LABEL5.Caption:=''; LABEL6.Caption:='';
        LABEL7.Caption:=''; end
    ELSE BEGIN JJJ:=1; DD(Sender); JJJ:=1; DD(Sender); END;
end;
procedure TForm1.DBGrid2CellClick(Column: TColumn);
begin edit3.text:=DBGrid2.DataSource.DataSet.FieldByName('UDK').AsString;
end;
procedure TForm1.Button8Click(Sender: TObject);
begin
    if not(edit3.text='')
        then begin NotVisible(Sender); label3.Left:=10; label3.top:=40;
            label3.Height:=400; label3.Width:=400; label3.font.name:='Times New
Roman';
            label3.Font.Size:=16; label3.CAPTION:=edit3.text;
label3.visible:=true;
            Edit5.Left:=183; Edit5.top:=240; Edit5.Height:=28 ;
            Edit5.Width:= form1.Width-210; Edit5.font.name:='Times New
Roman';
            Edit5.Font.Size:=14; edit5.text:=''; edit5.visible:=true;
            Edit4.Left:=183; Edit4.top:=270; Edit4.Height:=28 ;
            Edit4.Width:= form1.Width-210;Edit4.font.name:='Times New
Roman';
            Edit4.Font.Size:=14; edit4.text:=''; edit4.visible:=true;
            label4.Left:=10; label4.top:=40; label4.Height:=400;
            label4.Width:=400; label4.font.name:='Times New Roman';
            label4.Font.Size:=16; label4.CAPTION:=edit3.text;
edit4.text:='';
            label3.visible:=TRUE; CheckBox2.Visible:=TRUE;
edit3.Visible:=TRUE;
            CheckBox1.Left:=10; CheckBox1.top:=240; CheckBox1.Height:=28 ;
            CheckBox1.Width:=160; CheckBox1.font.name:='Times New Roman';
            CheckBox1.Font.Size:=12; CheckBox1.caption:='Знайти по назві:';
            CheckBox1.visible:=true; CheckBox2.Left:=10; CheckBox2.top:=270;
            CheckBox2.Height:=28 ; CheckBox2.Width:=160;
            CheckBox2.font.name:='Times New Roman'; CheckBox2.Font.Size:=12;
            CheckBox2.caption:='Знайти по автору:'; CheckBox2.visible:=true;
            BK(Sender);end;
        end;
procedure TForm1.CheckBox2Click(Sender: TObject);
begin
    if CheckBox2.Checked=false
        then edit4.ReadOnly:=false
        else if edit4.text='' then CheckBox2.Checked:=false

```

```
        else edit4.ReadOnly:=true;
    if not(edit4.text='') then BK(Sender);
end;
procedure TForm1.CheckBox1Click(Sender: TObject);
begin
    if CheckBox1.Checked=false
        then edit5.ReadOnly:=false
        else if edit5.text='' then CheckBox1.Checked:=false
            else edit5.ReadOnly:=true;
    if not(edit5.text='') then BK(Sender);
end;
end.
```

Кафедра \_ КБПЗ \_ 2023рік

```
// ----- Form1.dfm ----- Опис Форми
```

```
Form1
```

```
object Form1: TForm1
  Left = 179
  Top = 246
  Width = 985
  Height = 368
  Caption = 'Form1'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  Menu = MainMenu1
  OldCreateOrder = False
  OnCreate = FormCreate
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 104
    Top = 0
    Width = 32
    Height = 13
    Caption = 'Label1'
  end
  object Label2: TLabel
    Left = 104
    Top = 16
    Width = 32
    Height = 13
    Caption = 'Label2'
  end
  object Label3: TLabel
    Left = 104
    Top = 32
    Width = 32
    Height = 13
    Caption = 'Label3'
  end
  object Label4: TLabel
    Left = 104
```

```
    Top = 48
    Width = 32
    Height = 13
    Caption = 'Label4'
    Color = clBtnFace
    ParentColor = False
end
object Label5: TLabel
    Left = 104
    Top = 64
    Width = 32
    Height = 13
    Caption = 'Label5'
end
object Label6: TLabel
    Left = 104
    Top = 80
    Width = 32
    Height = 13
    Caption = 'Label6'
end
object Label7: TLabel
    Left = 104
    Top = 96
    Width = 32
    Height = 13
    Caption = 'Label7'
end
object Label8: TLabel
    Left = 104
    Top = 112
    Width = 32
    Height = 13
    Caption = 'Label8'
end
object StringGrid1: TStringGrid
    Left = 280
    Top = 0
    Width = 73
    Height = 25
    FixedCols = 0
    RowCount = 3
    FixedRows = 0
```

```
    TabOrder = 0
    OnSelectCell = StringGrid1SelectCell
end
object Memo1: TMemo
    Left = 40
    Top = 0
    Width = 57
    Height = 25
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -13
    Font.Name = 'Courier'
    Font.Style = []
    Lines.Strings = (
        'Memo1')
    ParentFont = False
    TabOrder = 1
end
object Edit1: TEdit
    Left = 144
    Top = 0
    Width = 41
    Height = 21
    TabOrder = 2
    Text = 'Edit1'
end
object Edit2: TEdit
    Left = 144
    Top = 16
    Width = 41
    Height = 21
    TabOrder = 3
    Text = 'Edit2'
end
object Button1: TButton
    Left = 192
    Top = 0
    Width = 81
    Height = 25
    Caption = 'Button1'
    TabOrder = 4
    OnClick = Button1Click
end
```

```
object Button2: TButton
  Left = 192
  Top = 24
  Width = 81
  Height = 25
  Caption = 'Button2'
  TabOrder = 5
  OnClick = Button2Click
end
object Button3: TButton
  Left = 192
  Top = 48
  Width = 81
  Height = 25
  Caption = 'Button3'
  TabOrder = 6
  OnClick = Button3Click
end
object StringGrid2: TStringGrid
  Left = 280
  Top = 24
  Width = 81
  Height = 25
  TabOrder = 7
  OnSelectCell = StringGrid1SelectCel2
end
object Button4: TButton
  Left = 192
  Top = 72
  Width = 81
  Height = 25
  Caption = 'Button4'
  TabOrder = 8
  OnClick = Button4Click
end
object Button5: TButton
  Left = 192
  Top = 96
  Width = 81
  Height = 25
  Caption = 'Button5'
  TabOrder = 9
  OnClick = Button5Click
```

```
end
object DBGrid1: TDBGrid
  Left = 368
  Top = 32
  Width = 233
  Height = 73
  DataSource = DataSource1
  TabOrder = 10
  TitleFont.Charset = DEFAULT_CHARSET
  TitleFont.Color = clWindowText
  TitleFont.Height = -11
  TitleFont.Name = 'MS Sans Serif'
  TitleFont.Style = []
end
object DBNavigator1: TDBNavigator
  Left = 368
  Top = 104
  Width = 240
  Height = 25
  DataSource = DataSource1
  TabOrder = 11
end
object StringGrid3: TStringGrid
  Left = 280
  Top = 48
  Width = 73
  Height = 25
  TabOrder = 12
  OnSelectCell = StringGrid3SelectCell
end
object Button6: TButton
  Left = 192
  Top = 120
  Width = 81
  Height = 25
  Caption = 'Button6'
  TabOrder = 13
  OnClick = Button6Click
end
object Edit3: TEdit
  Left = 144
  Top = 32
  Width = 41
```

```
Height = 21
TabOrder = 14
Text = 'Edit3'
end
object Button7: TButton
Left = 192
Top = 144
Width = 81
Height = 25
Caption = 'Button7'
TabOrder = 15
OnClick = Button7Click
end
object DBGrid2: TDBGrid
Left = 616
Top = 32
Width = 233
Height = 57
DataSource = DataSource2
TabOrder = 16
TitleFont.Charset = DEFAULT_CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnClick = DBGrid2CellClick
Columns = <
  item
    Expanded = False
    FieldName = 'UDK'
    Visible = True
  end
  item
    Expanded = False
    FieldName = 'OP'
    Visible = True
  end
  item
    Expanded = False
    FieldName = 'UDK'
    Visible = True
  end
  item
```

```
        Expanded = False
        FieldName = 'OP'
        Visible = True
    end>
end
object DBNavigator2: TDBNavigator
    Left = 616
    Top = 104
    Width = 240
    Height = 25
    DataSource = DataSource2
    TabOrder = 17
    OnDblClick = BK
end
object CheckBox1: TCheckBox
    Left = 0
    Top = 48
    Width = 97
    Height = 17
    Caption = 'CheckBox1'
    TabOrder = 18
    OnClick = CheckBox1Click
end
object CheckBox2: TCheckBox
    Left = 0
    Top = 72
    Width = 97
    Height = 17
    Caption = 'CheckBox2'
    TabOrder = 19
    OnClick = CheckBox2Click
end
object Button8: TButton
    Left = 192
    Top = 168
    Width = 81
    Height = 25
    Caption = 'Button8'
    TabOrder = 20
    OnClick = Button8Click
end
object Edit4: TEdit
    Left = 144
```

```
Top = 48
Width = 41
Height = 21
TabOrder = 21
Text = 'Edit4'
end
object Edit5: TEdit
Left = 144
Top = 64
Width = 41
Height = 21
TabOrder = 22
Text = 'Edit5'
end
object MainMenu1: TMainMenu
object N11: TMenuItem
Caption = 'Адміністрування'
object N21: TMenuItem
Caption = 'Введення користувача'
OnClick = N21Click
end
object N1: TMenuItem
Caption = 'Зміна пароля'
OnClick = N1Click
end
object N2: TMenuItem
Caption = 'Змінити користувача'
OnClick = N2Click
end
object N9: TMenuItem
Caption = 'Видалити користувача'
OnClick = N9Click
end
end
object N31: TMenuItem
Caption = 'Підготовка метаданих'
object N41: TMenuItem
Caption = 'Введення метаданих'
OnClick = N41Click
end
object N3: TMenuItem
Caption = 'Корекція УДК'
OnClick = N3Click
```

```
        end
    end
    object N4: TMenuItem
        Caption = 'Пошук джерела'
        object N5: TMenuItem
            Caption = 'По назві'
            OnClick = N5Click
        end
        object N6: TMenuItem
            Caption = 'Поавтору'
            OnClick = N6Click
        end
        object N7: TMenuItem
            Caption = 'По УДК'
            OnClick = N7Click
        end
    end
end
end
object Table1: TTable
    Active = True
    DatabaseName = 'Z'
    TableName = 'B'
    Left = 368
end
object DataSource1: TDataSource
    DataSet = Table1
    Left = 400
end
object Table2: TTable
    Active = True
    DatabaseName = 'Z'
    TableName = 'Y'
    Left = 600
end
object DataSource2: TDataSource
    DataSet = Table2
    Left = 632
end
end
```