

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
« ____ » _____ 2022 р.

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

Виконав здобувач вищої освіти
II курсу, групи КІ-21М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Іщак Я.С.
« ____ » _____ 2022 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2022 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
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

« 6 » вересня 2022 року

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

Іщаку Ярославу Сергійовичу

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|---|---|---------------------|--|--|---|---|---------------------------------|--------------|---|--|
| 1. Тема роботи | <u>Дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства</u> | | | | | | | | | | |
| 2. Керівник роботи | <u>Смірнова Тетяна Віталіївна, канд. техн. наук</u>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання) | | | | | | | | | | |
| затверджені наказом вищого навчального закладу № 19-13 від 17.08.2022 року | | | | | | | | | | | |
| 3. Строк подання студентом роботи до захисту | <u>10.12.2022 р.</u> | | | | | | | | | | |
| 4. Мета та завдання випускної кваліфікаційної роботи: | <u>Метою розробки є дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства</u> | | | | | | | | | | |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td>1. Призначення та область використання.</td><td>6. Наукова новизна.</td></tr><tr><td>2. Перегляд аналогічних існуючих систем.</td><td>7. Економічна ефективність розробленої програми.</td></tr><tr><td>3. Опис і обґрунтування проектних рішень.</td><td>8. Заходи з охорони праці та техніки безпеки.</td></tr><tr><td>4. Етапи програмування системи.</td><td>9. Висновки.</td></tr><tr><td>5. Впровадження системи в промислову експлуатацію</td><td></td></tr></table> | 1. Призначення та область використання. | 6. Наукова новизна. | 2. Перегляд аналогічних існуючих систем. | 7. Економічна ефективність розробленої програми. | 3. Опис і обґрунтування проектних рішень. | 8. Заходи з охорони праці та техніки безпеки. | 4. Етапи програмування системи. | 9. Висновки. | 5. Впровадження системи в промислову експлуатацію | |
| 1. Призначення та область використання. | 6. Наукова новизна. | | | | | | | | | | |
| 2. Перегляд аналогічних існуючих систем. | 7. Економічна ефективність розробленої програми. | | | | | | | | | | |
| 3. Опис і обґрунтування проектних рішень. | 8. Заходи з охорони праці та техніки безпеки. | | | | | | | | | | |
| 4. Етапи програмування системи. | 9. Висновки. | | | | | | | | | | |
| 5. Впровадження системи в промислову експлуатацію | | | | | | | | | | | |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) | | | | | | | | | | | |
| <u>Наукова новизна</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Структурна схема системи</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Функціональна схема системи</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Діаграма процесів</u> | <u>1 аркуш</u> | | | | | | | | | | |
| <u>Блок-схема алгоритму роботи додатку</u> | <u>2 аркуша</u> | | | | | | | | | | |
| <u>Показники економічної ефективності</u> | <u>1 аркуш</u> | | | | | | | | | | |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	14.11.2022
Охорона праці	Оришака О.В.	06.10.2022	16.11.2022

7. Дата видачі завдання « 6 » вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника

Смірнова Т.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача

Іщак Я.С.
(прізвище та ініціали)

АНОТАЦІЯ

Ишак Я.С. Дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

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

Метою розробки є дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства.

Об'єктом дослідження є процес складського аудиту автотранспортного підприємства.

Предметом дослідження є методи складського аудиту автотранспортного підприємства.

Методи дослідження базуються на методах теорії аудиту, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи складського аудиту автотранспортного підприємства.

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

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

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

Програму розроблено в середовищі Delphi 10.4.1.

Ключові слова: комп'ютерна інженерія, складський аудит

ABSTRACT

Ishchak Ya.S. Research and software implementation of the warehouse audit system of the motor vehicle enterprise. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the warehouse audit system of a motor vehicle enterprise.

The purpose of the development is the research and software implementation of the warehouse audit system of the motor vehicle enterprise.

The object of the study is the warehouse audit process of the motor vehicle enterprise.

The subject of the study is the methods of warehouse audit of the motor vehicle enterprise.

Research methods are based on audit theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the warehouse audit system of the motor vehicle enterprise.

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

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

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

The program was developed in the Delphi 10.4.1 environment.

Keywords: computer engineering, warehouse audit

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	24
3.1 Опис функціонування системи	24
3.2 Розробка структурної схеми.....	40
3.3 Розробка функціональної схеми	46
3.4 Розробка діаграми процесів.....	49
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	51
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	51
4.2 Захист розробленого програмного забезпечення.....	60
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	63
6 НАУКОВА НОВИЗНА	66

						ВКРМ-123.22.0012.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
<i>Розроб.</i>	Ицак Я.С.				<i>Дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства</i>	Літ.	Аркуш	Аркушів
<i>Перев.</i>	Смірнова Т.В.					М	1	106
<i>Н.контр.</i>	Гермак В.С.					<i>ЦНТУ КІ-21М-1,4</i>		
<i>Затв.</i>	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	67
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	67
7.2 Розрахунок трудомісткості розробки програмної продукції.....	69
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	71
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	76
7.5 Визначення собівартості розробки та ціни програмної продукції.....	80
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	83
7.7 Визначення експлуатаційних витрат.....	83
7.8 Визначення економічної ефективності програмної продукції.....	85
7.9 Висновок.....	87
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	88
8.1 Вступ.....	88
8.2 Пожежна безпека.....	90
8.3 Пропозиції щодо підвищення працездатності ІТ-фахівця.....	91
8.4 Розробка заходів з умов поліпшення охорони праці.....	93
8.5 Розрахункова частина	94
8.6 Висновки до розділу.....	96
9 ОСНОВНІ ВИСНОВКИ.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- БД – база даних
- ЛОМ – локальна обчислювальна мережа
- ПММ – паливно-мастильні матеріали
- СУБД – система управління базою даних
- IP – Internet Protocol – міжмережевий протокол
- SMTP – Simple Mail Transfer Protocol – простий протокол передачі пошти
- TCP – Transmission Control Protocol – протокол управління передачею
- UDP – User Datagram Protocol – протокол користувальницьких датаграм
- URL – уніфікований показник інформаційного ресурсу

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Сучасне життя немислиме без ефективного управління. Важливою категорією є автоматизовані системи обробки інформації, від яких багато в чому залежить ефективність роботи будь-якого підприємства або установи. Така система повинна:

- забезпечувати одержання загальних і/або деталізованих звітів за підсумками роботи;
- дозволяти легко визначати тенденції зміни найважливіших показників;
- забезпечувати одержання інформації, критичної за часом, без істотних затримок;
- виконувати точний і повний аналіз даних.

Серед найбільш яскравих представників систем управління базами даних можна відзначити: Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual FoxPro, Microsoft Visual Basic, а також баз даних Microsoft SQL Server і Oracle, використовувані в додатках, побудованих за технологією «клієнт-сервер». Фактично, у будь-якій сучасній СУБД існує аналог, що випускається іншою компанією, що має аналогічну область застосування й можливості, будь-який додаток здатний працювати з багатьма форматами подання даних, здійснювати експорт і імпорт даних завдяки наявності великої кількості конвертерів. Загальноприйнятими, також, є технології, що дозволяють використовувати можливості інших додатків, наприклад, текстових процесорів, пакетів побудови графіків і т.п., і убудовані версії мов високого рівня (частіше – діалекти SQL і/або VBA) та засоби візуального програмування інтерфейсів розроблювальних додатків. Тому вже не має істотного значення на якій мові, й на основі якого пакета, написаний конкретний додаток, і який формат даних у ньому використовується. Більше того, стандартом «де-факто» стала «швидка розробка додатків» або RAD (від англійського Rapid Application Development), заснована

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

на широко декларуємому у літературі «відкритому підході», тобто необхідність і можливість використання різних прикладних програм і технологій для розробки більш гнучких і потужних систем обробки даних. Тому в одному ряді з «класичними» СУБД все частіше згадуються мови програмування Visual Basic 4.0 і Visual C++, які дозволяють швидко створювати необхідні компоненти додатків, критичні по швидкості роботи, які важко, а іноді неможливо розробити засобами «класичних» СУБД. Сучасний підхід до управління базами даних має на увазі також широке використання технології «клієнт-сервер».

Таким чином, на сьогоднішній день розроблювач не зв'язаний рамками якого-небудь конкретного пакета, а залежно від поставленої задачі може використовувати самі різні додатки. Тому, більш важливим представляється загальний напрямок розвитку СУБД і інших засобів розробки додатків у цей час.

У багатьох суб'єктах господарювання використовуються транспорт, або якісь механізми. Логічно, що ці механізми повинні працювати на якихось паливно-мастильних матеріалах та використовувати запасні частини, які встановлюються замість тих, що зносилися. Це більш логічним є те, що треба зберігати десь дані про наявність цих матеріалів та запчастин, а також необхідно контролювати їх рух.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства.

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

- Огляд існуючих систем складського аудиту автотранспортного підприємства.
- Дослідження системи складського аудиту автотранспортного підприємства.
- Програмна реалізація системи складського аудиту автотранспортного підприємства.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Об'єктом дослідження є процес складського аудиту автотранспортного підприємства.

Предметом дослідження є методи складського аудиту автотранспортного підприємства.

Методи дослідження базуються на методах теорії аудиту, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод складського аудиту автотранспортного підприємства.

– Розроблено вітчизняний продукт складського аудиту автотранспортного підприємства, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі складського аудиту автотранспортного підприємства.

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

Робота апробована на LVI Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2022, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №13.

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для автоматизації обліку запасних частин та паливно-мастильних матеріалів (ПММ) для механізованої колони підприємства. У програмі повинен вестися облік номерних запасних частин і агрегатів, а також реалізована реєстрація операцій по сервісному обслуговуванню й ремонтам транспортних засобів. Ці функції дозволяють планувати як ресурси ремонтної зони, так і ресурси вільних автомобілів. Використання довідника часових нормативів підвищує продуктивність і ефективність роботи: простіше призначати й дотримувати час ремонтів транспортних засобів, можна більш раціонально використовувати ресурси й уникати виконання непотрібної роботи.

Облік заявок на ремонт транспортних засобів, реєстрації несправностей, формування дефектної відомості й планування ремонтів ведеться за допомогою документа «Попередні замовлення на ремонт». Документ «Ремонтний аркуш» застосовується для обліку виконаних ремонтів ТЗ, відбиття фактів установки/зняття шин і акумуляторів, а також є підставою для списання запасних частин і матеріалів.

Програма зобов'язана дозволяти вести облік установлених шин і акумуляторів, автоматично розраховувати пробіги й наробіток транспортних засобів, агрегатів, спецобладнання.

Використання цієї інформації істотно спрощує планування технічного обслуговування парку.

Крім того, у програмі повинен вестися облік пально-мастильних матеріалів (ПММ), застосовуваних на підприємстві. Необхідно реалізувати облік палива, придбаного різними способами – за готівку або по картах безготівкової

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

оплати, отриманого по талонах, виданого зі складу підприємства або отриманого у стороннього постачальника.

Програма повинна дозволяти вести облік витрат палива для автомобілів з необмеженою кількістю устаткування й причепів. Налаштування таблиці залежності нормативів витрати ПММ від температури полегшують розрахунок шляхових аркушів у регіонах із часто мінливими сезонними температурами. Конфігурація повинна містити заповнений довідник моделей транспортних засобів (більше 500 моделей) із установленими нормами витрати ПММ. Розрахунок витрати ПММ передбачений, у тому числі для автомобілів з газодизельними й газобензиновими двигунами, що працюють на двох видах палива. Можливість налаштування взаємозамінних аналогів ПММ допомагає швидко замінити вид палива якщо буде потреба.

У системі повинна бути реалізовано завантаження даних процесингових центрів по деталізації заправлень ПММ. Отримані дані не впливають на залишки ПММ на транспортних засобах і призначені тільки для звірення з даними, отриманими від водіїв. Можливість завантаження даних процесингових центрів по деталізації зроблених заправлень ПММ полегшує контроль помилок і допомагає виконувати звірення з АЗС.

1.2 Область застосування

Областю застосування програми є автоматизація спеціалізованого обліку транспортного підрозділу (стан транспортних засобів, експлуатаційні показники на основі шляхових аркушів, облік покришок і акумуляторів, ТО, руху палива й ПММ).

Переваги рішення:

- автоматизація ключових бізнес-процесів транспортного підрозділу на базі єдиної інформаційної системи;
- ефективне керування інформацією про транспортні засоби;

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- автоматизація всього циклу експлуатації транспортного засобу від надходження до списання з балансу;
- ефективне керування видатком палива, пально-мастильних матеріалів, запасних частин на транспортний засіб і планування майбутніх видатків;
- оперативне одержання звітності в різних розрізах.
- Користувачі системи: керівники середньої й вищої ланки, співробітники автобази всіх рівнів і бухгалтера.

Відмінні риси:

Рішення дозволяє автоматизувати всі ключові бізнес-процеси служб спеціального транспорту на підприємствах:

- Облік транспортних засобів;
- Облік акумуляторів і автопокришок;
- Облік ремонтів, виданих запасних частин і встаткування по транспортному засобі;
- Облік приходу й видатку палива, пально-мастильних матеріалів і спецрідин.

Компоненти рішення:

- Облік транспортних засобів:
 - а) контроль автомобілів і реєстрація нових за допомогою довідників транспортних засобів, марок, призначень і т.д.
 - б) відстеження поточний стан автомобіля (експлуатація, оренда, ремонт, списане).
- Облік покришок і акумуляторів:
 - а) історія списання й установки покришок і акумуляторів на автомобіль;
 - б) відстеження вироблення ресурсу покришок і акумуляторів.
- Облік ремонтів:
 - а) історія ремонтів і технічних оглядів кожного автомобіля;
 - б) заборона виписки шляхових аркушів для автомобілів у ремонті.
- Облік руху палива й пально-мастильних матеріалів:

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

- а) перевірка видатку палива на підставі норм із урахуванням сезону (літо/зима);
- б) відбиття заправлення різних видів палива в один бак;
- в) реєстрація даних про рух пально-мастильних матеріалів і палива на гаражні потреби без виписки шляхових аркушів;
- г) автоматичне створення документів на покупку й списання палива, врахованого в шляхових аркушах;
- д) облік палива для додаткового устаткування.

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Ціль будь-якої інформаційної системи – обробка даних про об'єкти реального миру. У широкому змісті слова база даних – це сукупність відомостей про конкретні об'єкти реального миру в якій-небудь предметній області. Під предметною областю прийнято розуміти частину реального миру, що підлягає вивченню для організації управління їй, в остаточному підсумку, автоматизації, наприклад підприємство, ЗВО і т.ін.

Створюючи базу даних, користувач прагне впорядкувати інформацію з різних ознак і швидко витягати вибірку з довільним сполученням ознак. Зробити це можливо, тільки якщо дані структуровані.

Структурування – це введення угод про способи подання даних. Неструктурованими називають дані, записані, наприклад, у текстовому файлі.

Користувачами бази даних можуть бути різні прикладні програми, програмні комплекси, а також фахівці предметної області, що виступають у ролі споживачів або джерел даних, називані кінцевими користувачами.

У сучасній технології баз даних передбачається, що створення бази даних, її підтримка й забезпечення доступу користувачів до неї здійснюються централізовано за допомогою спеціального програмного інструментарію – системи управління базами даних.

База даних (БД) – це поійменована сукупність структурованих даних, що відносяться до певної предметної області.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Система управління базами даних (СУБД) – це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримки їх в актуальному стані й організації пошуку в них необхідної інформації.

Централізований характер управління даними в базі даних припускає необхідність існування деякої особи (групи осіб), на яких покладають функції адміністрування даними, збереженими в базі.

Класифікація баз даних

За технологією обробки дані бази даних підрозділяються на централізовані й розподілені.

Централізована база даних зберігається в пам'яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом мережі ЕОМ, можливий розподілений доступ до такої бази. Такий спосіб використання баз даних часто застосовують у локальних мережах ПК.

Розподілена база даних складається з декількох, можливо пересічних або навіть дублюючих одна одну частин, збережених у різних ЕОМ обчислювальній мережі. Робота з такою базою здійснюється за допомогою системи управління розподіленою базою даних (СУРБД).

За способом доступу до даних бази даних розділяються на бази даних з локальним доступом і бази даних з віддаленим (мережевим) доступом.

Системи централізованих баз даних з мережевим доступом припускають різні архітектури подібних систем:

- файл-сервер;
- клієнт-сервер.

Файл-Сервер

Архітектура систем БД із мережевим доступом припускає виділення однієї з машин мережі в якості центральної (сервер, файлів). На такій машині зберігається спільно використовувана централізована БД. Всі інші машини мережі виконують функції робочих станцій, за допомогою яких підтримується доступ користувальницької системи до централізованої бази даних. Файли бази

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

даних відповідно до користувальницьких запитів передаються на робочі станції, де в основному й відбувається обробка. При великій інтенсивності доступу до тих самих даних продуктивність інформаційної системи падає. Користувачі можуть створювати також на робочих станціях локальні БД, які використовуються ними монополярно.

Клієнт-Сервер

У цій концепції мається на увазі, що крім зберігання централізованої бази даних, центральна машина (сервер бази даних) повинна забезпечувати виконання основного об'єму обробки даних. Запит на дані, видаваний клієнтом (робочою станцією), породжує пошук і добування даних на сервері. Витягнуті дані (але не файли) транспортуються по мережі від сервера до клієнта. Специфікою архітектури клієнт-сервер є використання мови запитів SQL.

Структурні елементи бази даних

Поняття бази даних тісно пов'язане з такими поняттями структурних елементів, як поле, запис, файл (таблиця).

Поле – елементарна одиниця логічної організації даних, що відповідає неподільній одиниці інформації – реквізиту. Для опису поля використовуються наступні характеристики:

- ім'я (Прізвище, Ім'я, По батькові, Дата народження);
- тип (символьний, числовий, календарний);
- довжина (наприклад, 15 байт, причому буде визначатися максимально можливою кількістю символів);
- точність для числових даних, наприклад два десяткових знаки для відображення дробової частини числа.

Запис – сукупність логічно зв'язаних полів. Екземпляр запису – окрема реалізація запису, що містить конкретні значення її полів.

Файл (таблиця) – сукупність екземплярів записів однієї структури.

У структурі запису файлу вказуються поля, значення яких є ключами первинними (ПК), які ідентифікують екземпляр запису, і вторинними (ВК), які

							ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.	Підпис	Дата				13

виконують роль пошукових або групових ознак (за значенням вторинного ключа можна знайти кілька записів).

Поняття інформаційного об'єкта

Інформаційний об'єкт – це опис деякої сутності (реального об'єкта, явища, процесу, події) у вигляді сукупності логічно зв'язаних реквізитів (інформаційних елементів). Такими сутностями для інформаційних об'єктів можуть служити: цех, склад, матеріал, вуз, студент, задача іспитів і т.д.

Інформаційний об'єкт певного реквізитного складу й структури утворить клас (тип), якому привласнюється унікальне ім'я (символьне позначення), наприклад Студент, Сесія, Стипендія.

Інформаційний об'єкт має множину реалізацій – екземплярів, кожний з яких представлений сукупністю конкретних значень реквізитів і ідентифікується значенням ключа (простого – один реквізит або складеного – кілька реквізитів). Інші реквізити інформаційного об'єкта є описовими. При цьому ті самі реквізити в одних інформаційних об'єктах можуть бути ключовими, а в інші – описовими. Інформаційний об'єкт може мати кілька ключів.

Нормалізація відносин

Одні й ті самі дані можуть групуватися в таблиці (відносини) різними способами, тобто можлива організація різних наборів відносин взаємозалежних інформаційних об'єктів. Угрупування атрибутів у відносинах повинна бути раціональною, тобто мінімізуючою дублювання даних процедури, що спрощує, їх обробку й відновлення.

Певний набір відносин має кращі властивості при включенні, модифікації, видаленні даних, чим всі інші можливі набори відносин, якщо він відповідає вимогам нормалізації відносин.

Нормалізація відносин – формальний апарат обмежень на формування відносин (таблиць), що дозволяє усунути дублювання, забезпечує несуперечність збережених у базі даних, зменшує трудовитрати на ведення (введення, коректування) бази даних.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Виділено три нормальні форми відносин і запропонований механізм, що дозволяє будь-яке відношення перетворити до третьої (самої зробленої) нормальної форми.

Перша нормальна форма

Відношення називається нормалізованим або наведеним до першої нормальної форми, якщо всі його атрибути прості (далі неподільні). Перетворення відносини до першої нормальної форми може привести до збільшення кількості реквізитів (полів) відносини й зміни ключа.

Наприклад, відношення Студент = (Номер, Прізвище, Ім'я, По батькові, Дата, Група) наводиться в першій нормальній формі.

Друга нормальна форма

Щоб розглянути питання приведення відносин до другої нормальної форми, необхідно дати пояснення до таких понять, як функціональна залежність і повна функціональна залежність.

Описові реквізити інформаційного об'єкта логічно пов'язані із загальним для них ключем, цей зв'язок носить характер функціональної залежності реквізитів.

Функціональна залежність реквізитів – залежність, при якій екземпляри інформаційного об'єкта певному значенню ключового реквізиту відповідає тільки одне значення описового реквізиту.

Таке визначення функціональної залежності дозволяє при аналізі всіх взаємозв'язків реквізитів предметної області виділити самостійні інформаційні об'єкти.

У випадку складеного ключа вводиться поняття функціональне повної залежності.

Функціонально повна залежність не ключових атрибутів полягає в тім, що кожний не ключовий атрибут функціонально залежить від ключа, але не перебуває у функціональній залежності ні від якої частини складеного ключа.

					ВКРМ-123.22.0012.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		15

Відношення буде перебувати в другій нормальній формі, якщо воно перебуває в першій нормальній формі, і кожний не ключовий атрибут функціонально повно залежить від складеного ключа.

Третя нормальна форма

Поняття третьої нормальної форми ґрунтується на понятті нетранзитивної залежності.

Транзитивна залежність спостерігається в тому випадку, якщо один із двох описових реквізитів залежить від ключа, а інший описовий реквізит залежить від першого описового реквізиту.

Відношення буде перебувати в третій нормальній формі, якщо воно перебуває в другій нормальній формі, і кожний неключовий атрибут нетранзитивно залежить від первинного ключа.

Для усунення транзитивної залежності описових реквізитів необхідно провести «розщеплення» вихідного інформаційного об'єкта. У результаті розщеплення частина реквізитів віддаляється з вихідного інформаційного об'єкта й включається до складу інших (можливо, знову створених) інформаційних об'єктів.

Типи зв'язків

Всі інформаційні об'єкти предметної області зв'язані між собою. Розрізняються зв'язки декількох типів, для яких уведені наступні позначення:

- один до одного (1:1);
- один до багатьох (1 : Б);
- багато до багатьох (Б : Б).

Зв'язок *один до одного* (1:1) припускає, що в кожний момент часу одному екземпляру інформаційного об'єкта А відповідає не більше одного екземпляра інформаційного об'єкта В и навпаки.

При зв'язку *один до багатьох* (1 : Б) одну екземпляра інформаційного об'єкта А відповідає 0, 1 або більше екземплярів об'єкта В, але кожний екземпляр об'єкта В зв'язаний не більш ніж з 1 екземпляром об'єкта А.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Зв'язок *багато до багатьох* (Б : Б) припускає, що в кожний момент часу одному екземпляру інформаційного об'єкта А відповідає 0, 1 або більше екземплярів об'єкта В и навпаки.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4к моніторах High DPI за

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільнюються з допомоги вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCl, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відлагодочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL,

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

включаючи `std::vector`, `std::map` і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в `debug-time`.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка `Snake`.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємі FMX компонент TМето на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

2.3 Розгорнута постановка завдання

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

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Опис моделей даних

Відомості про моделі даних

Інфологічна модель відображає реальний мир у деякій зрозумілій людині концепції, повністю незалежній від параметрів середовища зберігання даних. Існує множина підходів до побудови таких моделей: графові моделі, семантичні мережі, модель «сутність-зв'язок» і т.д. Найбільш популярною з них виявилася модель «сутність-зв'язок».

Інфологічна модель повинна бути відображена в комп'ютерно-орієнтовану даталогічну модель, «зрозумілу» СУБД. У процесі розвитку теорії й практичного використання баз даних, а також засобів обчислювальної техніки створювалися СУБД, що підтримують різні даталогічні моделі.

Спочатку стали використовувати ієрархічні даталогічні моделі. Простота організації, наявність заздалегідь заданих зв'язків між сутностями, подібність із фізичними моделями даних дозволяли домагатися прийнятної продуктивності ієрархічних СУБД на повільних ЕОМ з досить обмеженими об'ємами пам'яті. Але, якщо дані не мали деревоподібної структури, то виникала маса складностей при побудові ієрархічної моделі й бажанні домогтися потрібної продуктивності.

Мережеві моделі також створювалися для малоресурсних ЕОМ. Це досить складні структури, що складаються з «наборів» – пойменованих дворівневих дерев. «Набори» з'єднуються за допомогою «записів-зв'язувань», утворюючи ланцюжки й т.д. При розробці мережевих моделей було вигадано багато «маленьких хитрощів», що дозволяють збільшити продуктивність СУБД. Але істотно ускладнили останні. Прикладний програміст повинен знати масу термінів, вивчити кілька внутрішніх мов СУБД, детально представляти логічну

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

структуру бази даних для здійснення навігації серед різних екземплярів, наборів, записів і т.п. Один з розроблювачів операційної системи UNIX сказав: «Мережева база – це найкращий спосіб втратити дані».

Складність практичного використання ієрархічних і мережевих СУБД змушувала шукати інші способи подання даних. Наприкінці 60-х років з'явилися СУБД на основі інвертованих файлів, що відрізняються простотою організації й наявністю досить зручних мов маніпулювання даними. Однак такі СУБД володіють рядом обмежень на кількість файлів для зберігання даних, кількість зв'язків між ними, довжину запису й кількість її полів.

Фізична організація даних впливає на експлуатаційні характеристики БД. Розроблювачі СУБД намагаються створити найбільш продуктивні фізичні моделі даних, пропонуючи користувачам той або інший інструментарій для підналаштування моделі під конкретну БД. Розмаїтість способів коректування фізичних моделей сучасних промислових СУБД не дозволяє розглянути їх у цьому розділі.

Види моделей даних

Ядром будь-якої бази даних є модель даних. Модель даних являє собою множину структур даних, обмежень цілісності й операцій маніпулювання даними. За допомогою моделі даних можуть бути представлені об'єкти предметної області й взаємозв'язку між ними.

Модель даних – сукупність структур даних і операцій їхньої обробки.

СУБД ґрунтується на використанні ієрархічної, мережевої або реляційної моделі, на комбінації цих моделей або на деякій їхній підмножині.

Розглянемо три основних типи моделей даних: ієрархічну, мережеву й реляційну.

Ієрархічна модель даних

Ієрархічна структура представляє сукупність елементів, зв'язаних між собою за певними правилами. Об'єкти, зв'язані ієрархічними відносинами, утворюють орієнтований граф (перевернене дерево).

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

До основних понять ієрархічної структури відносяться: рівень, елемент (вузол), зв'язок. Вузол – це сукупність атрибутів даних, що описують деякий об'єкт. На схемі ієрархічного дерева вузли представляються вершинами графа. Кожний вузол на більш низькому рівні зв'язаний тільки з одним вузлом, що перебуває на більш високому рівні. Ієрархічне дерево має тільки одну вершину (корінь дерева), не підлегла ніякій іншій вершині й що знаходиться на самому верхньому (першому) рівні. Залежні (підлеглі) вузли перебувають на другому, третьому й т.д. рівнях. Кількість дерев у базі даних визначається числом кореневих записів.

До кожних записів бази даних існує тільки один (ієрархічний) шлях від кореневого запису.

Мережева модель даних

У мережевій структурі при тих же основних поняттях (рівень, вузол, зв'язок) кожний елемент може бути пов'язаний з будь-яким іншим елементом.

Реляційна модель даних

Поняття реляційний (англ. relation – відношення) пов'язане з розробками відомого американського фахівця в області систем баз даних Е. Кодда.

Ці моделі характеризуються простотою структури даних, зручним для користувача табличним поданням і можливістю використання формального апарата алгебри відносин і реляційного вираховування для обробки даних.

Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

- кожний елемент таблиці – один елемент даних;
- всі стовпці в таблиці однорідні, тобто всі елементи в стовпці мають однаковий тип (числовий, символічний і т.д.) і довжину;
- кожний стовпець має унікальне ім'я;
- однакові рядки в таблиці відсутні;
- порядок проходження рядків і стовпців може бути довільним.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Відносини представлені у вигляді таблиць, рядка яких відповідають кортежам або записам, а стовпці – атрибутам відносин, доменам, полям.

Поле, кожне значення якого однозначно визначає відповідний запис, називається простим ключем (ключовим полем). Якщо записи однозначно визначаються значеннями декількох полів, то така таблиця бази даних має складений ключ.

Щоб зв'язати дві реляційні таблиці, необхідно ключ першої таблиці увести до складу ключа другої таблиці (можливий збіг ключів); у протилежному випадку потрібно ввести в структуру першої таблиці зовнішній ключ – ключ другої таблиці.

Проектування моделі даних

Предметна область – частина реального миру, що підлягає вивченню з метою організації керування й, в остаточному підсумку, автоматизації. Предметна область представляється множиною фрагментів, наприклад, підприємство – цехами, дирекцією, бухгалтерією й т.д. Кожний фрагмент предметної області характеризується множиною об'єктів і процесів, що використовують об'єкти, а також множиною користувачів, характеризуємих різними поглядами на предметну область.

У теорії проектування інформаційних систем предметну область (або, якщо завгодно, увесь реальний світ у цілому) прийнято розглядати у вигляді трьох подань:

- подання предметної області в тім виді, як вона реально існує
- як її сприймає людина (мається на увазі проектувальник бази даних)
- як вона може бути описана за допомогою символів.

Тобто кажуть, що ми маємо справу з реальністю, описом (поданням) реальності й з даними, які відбивають це подання.

Дані, використовувані для опису предметної області, представляються у вигляді трирівневої схеми (рисунок 3.1).

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27



Рисунок 3.1 – Модель ANSI/SPARC

Зовнішнє подання (зовнішня рисунки) даних є сукупністю вимог до даних з боку деякої конкретної функції, виконуваної користувачем. Концептуальна рисунки є повною сукупністю всіх вимог до даних, отриманої з користувальницьких подань про реальний світ. Внутрішня рисунки – це сама база даних.

Звідси впливають основні етапи, на які розбивається процес проектування бази даних інформаційної системи:

Концептуальне проектування – збір, аналіз і редагування вимог до даних. Для цього здійснюються наступні заходи:

- обстеження предметної області, вивчення її інформаційної структури;
- виявлення всіх фрагментів, кожний з яких характеризується користувальницьким поданням, інформаційними об'єктами й зв'язками між ними, процесами над інформаційними об'єктами;
- моделювання й інтеграція всіх подань.

По закінченні даного етапу одержуємо концептуальну модель, інваріантну до структури бази даних. Часто вона представляється у вигляді моделі «сутність-зв'язок».

Логічне проектування – перетворення вимог до даних у структури даних. На виході одержуємо СУБД-орієнтовану структуру бази даних і специфікації прикладних програм. На цьому етапі часто моделюють бази даних стосовно до різних СУБД і проводять порівняльний аналіз моделей.

Фізичне проектування – визначення особливостей зберігання даних, методів доступу й т.д.

Подання даних за допомогою моделі «сутність-зв'язок»

Перш, ніж приступати до створення системи автоматизованої обробки інформації, розроблювач повинен сформулювати поняття про предмети, факти й події, якими буде оперувати дана система. Для того, щоб привести ці поняття до тієї або іншої моделі даних, необхідно замінити їхніми інформаційними поданнями. Одним з найбільш зручних інструментів уніфікованого подання даних, незалежного від його програмного забезпечення, що реалізує, є модель «сутність-зв'язок» (entity-relationship model, ER-model).

Модель «сутність-зв'язок» ґрунтується на якійсь важливій семантичній інформації про реальний світ і призначена для логічного подання даних. Вона визначає значення даних у контексті їхнього взаємозв'язку з іншими даними. Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найбільш загальною.

Відзначимо, що модель «сутність-зв'язок» не є моделлю даних у точному значенні, оскільки не визначає операцій над даними й обмежується описом тільки їхньої логічної структури.

Модель «сутність-зв'язок» була запропонована в 1976 р. Пітером Пін - Шен Ченом.

Елементи моделі

Будь-який фрагмент предметної області може бути представлений як *множина сутностей*, між якими існує деяка *множина зв'язків*. Дано визначення:

Сутність (entity) – це об'єкт, що може бути ідентифікований якимось способом, що відрізняє його від інших об'єктів. Приклади: конкретна людина, підприємство, подія й т.д.

Набір сутностей (entity set) – множина сутностей одного типу (володіючих однаковими властивостями). Приклади: всі люди, підприємства,

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

свята й т.д. Набори сутностей не обов'язково повинні бути непересічними. Наприклад, сутність, що належить до набору БЕНЗИН, також належить набору ПАЛИВО-МАСТИЛЬНІ.

Сутність фактично представляє із себе множину атрибутів, які описують властивості всіх членів даного набору сутностей.

Надалі для визначення сутності і її атрибутів будемо використовувати позначення виду:

БЕНЗИН (МАРКА, ОКТАНОВЕ ЧИСЛО, ДАТА ЗАВОЗУ).

Наприклад, паливно мастильні матеріали, можна описати як ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ (ПММ) (НОМЕР_ПММ, НАЙМЕНУВАННЯ).

Множина значень (область визначення) атрибута називається доменом. Наприклад, для атрибута ДАТА ЗАВОЗУ домен (назвемо його ЧИСЛО_РОКІВ) задається інтервалом цілих чисел більших нуля, оскільки людей з негативним віком не буває.

У згаданій статті П. Чена атрибут визначається як функція, що відображає набір сутностей у набір значень або в декартовий добуток наборів значень. Так атрибут ДАТА ЗАВОЗУ робить відображення в набір значень (домен) ЧИСЛО_РОКІВ. Атрибут ОКТАНОВЕ ЧИСЛО робить відображення в декартово добуток наборів значень ОКТАНОВЕ ЧИСЛО й ВИРОБНИК.

Звідси визначається ключ сутності – група атрибутів, така, що відображення набору сутностей у відповідну групу наборів значень є взаємооднозначним відображенням. Іншими словами: ключ сутності – це один або більше атрибутів, що унікально визначають дана сутність. У нашій прикладі ключем сутності БЕНЗИН є атрибут МАРКА.

Зв'язок (relationship) – це асоціація, встановлена між декількома сутностями.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Наприклад:

– оскільки бензин відноситься до ПММ, між сутностями БЕНЗИН і ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ існує зв'язок «відноситься до» або ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ-БЕНЗИН;

– так як бензин може бути якоюсь маркою, то між сутностями БЕНЗИН і ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ є зв'язок «вид марки» або ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ-БЕНЗИН;

– можуть існувати й зв'язку між сутностями одного типу, наприклад зв'язок ДИЗЕЛЬ – БЕНЗИН між двома сутностями ПММ.

Слід зазначити, що в методиці проектування даних є своєрідне правило гарного тону, відповідно до якого сутності позначаються за допомогою імен іменників, а зв'язку – дієслівними формами. Дане правило, однак, не є обов'язковим.

Зв'язок також може мати атрибути. Наприклад, для зв'язку ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ-БЕНЗИН можна задати атрибут ЯКІСТЬ БЕНЗИНУ.

Роль сутності у зв'язку – функція, що виконує сутність у даному зв'язку. Наприклад, у зв'язку БЕНЗИН-ДИЗЕЛЬ сутності ПММ можуть мати ролі «бензин» і «дизель». Вказівка ролей у моделі «сутність-зв'язок» не є обов'язковим і служить для уточнення семантики зв'язку.

Набір зв'язків (relationship set) – це відношення між n (причому n не менше 2) сутностями, кожна з яких ставиться до деякого набору сутностей.

Хоча, строго говорячи, поняття «зв'язок» і «набір зв'язків» різні (перша є елементом другого), їх, проте, дуже часто змішують. Тому, надалі також будемо часто користуватися термінами «зв'язок» маючи на увазі «набір зв'язків» і «сутність» маючи на увазі «набір сутностей».

У випадку $n=2$, тобто коли зв'язок поєднує дві сутності, вона називається бінарною. Доведено, що n -арний набір зв'язків ($n>2$) завжди можна замінити множиною бінарних, однак перші краще відображають семантику предметної області.

Те число сутностей, що може бути асоційоване через набір зв'язків з іншою сутністю, називають ступенем зв'язку. Розгляд ступенів особливо корисно для бінарних зв'язків. Можуть існувати наступні ступені бінарних зв'язків:

Один до одного (позначається 1:1). Це означає, що в такому зв'язку сутності з однією роллю завжди відповідає не більше одній сутності з іншою роллю. У розглянутому нами прикладі це зв'язок «керує», оскільки в кожному відділі може бути тільки один начальник, а співробітник може керувати тільки в одному відділі. Даний факт представлений на наступному рисунку 3.2, де прямокутники позначають сутності, а ромб – зв'язок. Тому що ступінь зв'язку для кожної сутності дорівнює 1, то вони з'єднуються однією лінією (рисунок 3.2).



Рисунок 3.2 – Зв'язок один до одного

Іншою важливою характеристикою зв'язку крім її ступеня є клас приналежності вхідних у неї сутностей або кардинальність зв'язку. Так як бензин обов'язково повиний бути ПММ, то кожній сутності «ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ» неодмінно повинна відповідати сутність «БЕНЗИН».

Таким чином, говорять, що сутність «БЕНЗИН» має обов'язковий клас приналежності (цей факт позначається також вказівкою інтервалу числа можливих входжень сутності у зв'язок, у цьому випадку це 1,1), а сутність «ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ» має необов'язковий клас приналежності (0,1). Тепер даний зв'язок ми можемо описати як 0,1:1,1. Надалі кардинальність бінарних зв'язків ступеня 1 будемо позначати в такий спосіб (рисунок 3.3):

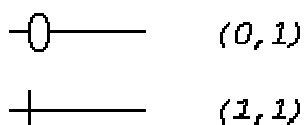


Рисунок 3.3 – Бінарні зв'язки першого ступеня

Один до багатьох (1:n). У цьому випадку сутності з однією роллю може відповідати будь-яке число сутностей з іншою роллю. Такий зв'язок ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ-БЕНЗИН. У кожному відділі може працювати довільне число співробітників, але співробітник може працювати тільки в одному відділі. Графічно ступінь зв'язку n відображається «деревоподібною» лінією, так це зроблено на наступній схемі (рисунок 3.4).

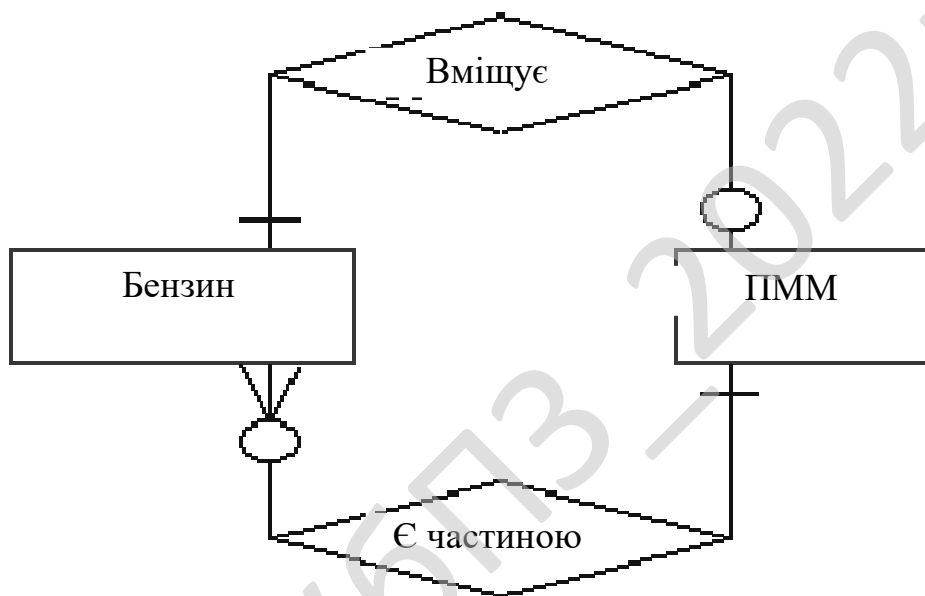


Рисунок 3.4 – Відображення зв'язку ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ – БЕНЗИН

Даний рисунок додатково ілюструє той факт, що між двома сутностями може бути визначено кілька наборів зв'язків.

Тут також необхідно враховувати клас приналежності сутностей. Кожний бензин повинен знаходитися на складі, але не кожний склад (наприклад, знову сформований) повинен включати наявність бензину. Тому сутність «ПАЛИВО-МАСТИЛЬНІ МАТЕРІАЛИ» має обов'язковий, а сутність «БЕНЗИН» необов'язковий класи приналежності. Кардинальність бінарних зв'язків ступеня n будемо позначати так (рисунок 3.5):

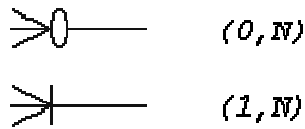


Рисунок 3.5 – Бінарні зв'язки n-ой ступеня

Багато до одного (n : 1). Цей зв'язок аналогічний відображенню 1 : n. Припустимо, що розглянуте нами підприємство буде свою діяльність на підставі контрактів, що містяться із замовниками. Цей факт відображається в моделі «сутність-зв'язок» за допомогою зв'язку КОНТРАКТ-ЗАМОВНИК, що поєднує сутності КОНТРАКТ (НОМЕР, СТРОК_ВИКОНАННЯ, СУМА) і ЗАМОВНИК(НАЙМЕНУВАННЯ, АДРЕСА). Так як з одним замовником може бути укладене більше одного контракту, то зв'язок КОНТРАКТ-ЗАМОВНИК між цими сутностями буде мати ступінь n : 1 (рисунок 3.6).



Рисунок 3.6 – Відображення зв'язку КОНТРАКТ – ЗАМОВНИК

У цьому випадку, по зовсім очевидних міркуваннях (кожний контракт укладений з конкретним замовником, а кожний замовник має хоча б один контракт, інакше він не був би таким), кожна сутність має обов'язковий клас приналежності.

Багато до багатьох (n : n). У цьому випадку кожна з асоційованих сутностей може бути представлена будь-якою кількістю екземплярів. Нехай на розглянутому нами підприємстві для виконання кожного контракту створюється робоча група, у яку входять співробітники різних відділів. Оскільки кожний співробітник може входити в трохи (у тому числі й у жодну) робочих груп, а

кожна група повинна включати не менш одного співробітника, то зв'язок між сутностями СПІВРОБІТНИК і РОБОЧА_ГРУПА має ступінь $n : n$ (рисунок 3.7).



Рисунок 3.7 – Відображення зв'язку БЕНЗИН – РОБОЧА_ГРУПА

Якщо існування сутності x залежить від існування сутності y , то x називається залежною сутністю (іноді сутність x називають «слабкою», а «сутність» y – сильною).

Як приклад розглянемо зв'язок між раніше описаними сутностями РОБОЧА_ГРУПА й КОНТРАКТ. Робоча група створюється тільки після того, як буде підписаний контракт із замовником, і припиняє своє існування по виконанню контракту. Таким чином, сутність РОБОЧА_ГРУПА є залежною від сутності КОНТРАКТ. Залежну сутність будемо позначати подвійним прямокутником, а її зв'язок із сильною сутністю лінією зі стрілкою (рисунок 3.8).



Рисунок 3.8 – Відображення зв'язку РОБОЧА_ГРУПА – КОНТРАКТ

Помітимо, що кардинальність зв'язку для сильної сутності завжди буде (1,1). Клас приналежності й ступінь зв'язку для залежної сутності можуть бути будь-якими. Припустимо, наприклад, що розглянуте нами підприємство користується декількома банківськими кредитами, які представляються набором сутностей: КРЕДИТ(НОМЕР_ДОГОВОРУ, СУМА, СТРОК_ПОГАШЕННЯ, БАНК). По кожному кредиті повинні здійснюватися виплати відсотків і платежі в рахунок його погашення. Цей факт представляється набором сутностей

ПЛАТІЖ(ДАТА, СУМА) і набором зв'язків «здійснюється по». У тому випадку, коли одержання запланованого кредиту відміняється, інформація про нього повинна бути віддалена з бази даних. Відповідно, повинні бути віддалені й всієї відомості про планові платежі по цьому кредиті. Таким чином, сутність ПЛАТІЖ залежить від сутності КРЕДИТ (рисунок 3.9).

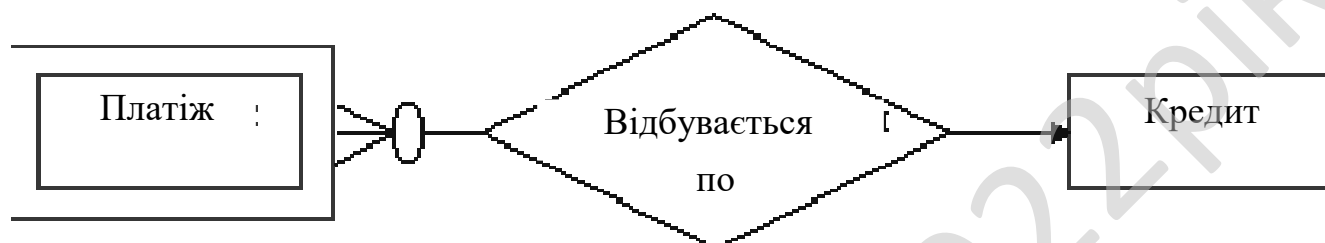


Рисунок 3.9 – Відображення зв'язку ПЛАТІЖ – КРЕДИТ

Опис створення бази даних

Визначення моделі даних

Визначення моделі даних передбачає вказівку множини припустимих інформаційних конструкцій, множини припустимих операцій над даними й множини обмежень для збережених значень даних.

Модель даних, з одного боку, являє собою формальний апарат для опису інформаційних потреб користувачів, а з іншого боку – більшість СУБД орієнтовані на конкретну модель даних, і, таким чином, якщо інформаційні потреби вдається точно виразити засобами однієї з моделей даних, те відповідна СУБД дозволяє відносно швидко створити працездатний фрагмент ІС.

Інформаційні конструкції, операції й обмеження моделей даних вибираються з досить невеликої множини варіантів, що характеризує "великі" інформаційні об'єкти й операції. Зокрема, не допускається розгляд окремих символів даних, операцій додавання атрибутів, обмеження на відповідність типів даних і т.п., що характерно для мов програмування.

Інформаційні об'єкти послужили основою для об'єктно-орієнтованого проектування систем, коли фіксується множина інформаційних об'єктів і дій над

об'єктами. Типовий список дій містить у собі створення/знищення об'єкта, редагування об'єкта, фіксацію одного об'єкта як частина іншого об'єкта, зв'язування об'єктів, синхронізацію дій над об'єктами.

Таки часто всі названі об'єкти вбудовуються в структуру відносин, які можна вважати найпростішими універсальними об'єктами.

Кількість істотно, що розрізняються моделей, даних визначається наявністю різних множин інформаційних конструкцій.

Збережені в базі дані мають певну логічну структуру, тобто, представлені деякою моделлю, підтримуваної СУБД. До числа найважливіших відносяться наступні **моделі даних**:

- інфологічна;
- ієрархічна;
- мережева;
- реляційна;
- об'єктно-орієнтована.

Інфологічне моделювання предметної області

Інфологічна модель займає особливе положення стосовно інших моделей. Вона відповідає четвертому етапу побудови складної системи й дає формалізований опис проблемної області незалежно від структур даних. Інфологічна область моделювання даних охоплює природні для людини концепції відображення реального миру.

Створення цієї моделі є першим кроком процесу формалізації. На відміну від подання природною мовою вона в основному виключає неоднозначність за рахунок використання засобів формальної логіки.

Одне з головних понять інфологічної моделі – об'єкт. Це поняття пов'язане з подіями: виникнення, зникнення й зміна.

Об'єкти можуть бути атомарними або складовими.

Атомарний об'єкт- це об'єкт певного типу, подальше розкладання якого на більше дрібні об'єкти усередині даного типу неможливо.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Складені об'єкти містять у собі множини об'єктів, кортежі об'єктів. Застосовуючи це визначення, рекурсивно можна одержати довільну структуру складених об'єктів.

Звичайно об'єкт має деяка властивість або взаємозв'язок (зв'язок) з іншими об'єктами. Властивість може бути не визначене формально, а лише охарактеризоване як деяке твердження із приводу множини об'єктів.

Інфологічна модель дозволяє виділити три категорії фактів: справжні, значимі й помилкові.

З одного боку, це забезпечує моделі додаткову гнучкість, з іншого боку – створює певні складності.

Розходження між традиційними й інфологічними моделями даних аналогічні розходженню між думкою й істиною. У багатьох моделях більшість повідомлень ставиться до однієї із двох категорій: істина або неправда. Інфологічна модель припускає можливість подання будь-якого повідомлення з якоюсь часткою ймовірності, тобто у вигляді аналога думки. Аналіз такого повідомлення можливий при обліку конкретного контексту. У правильному контексті повідомлення істинно. Але й помилкове твердження може розглядатися як думка.

Ціль інфологічного моделювання – формалізація об'єктів реального миру предметної області й методів обробки інформації відповідно до поставлених задач обробки й вимогами подання даних природними для людини способами збору й подання інформації.

Інфологічні моделі дозволяють одержувати довільні подання простих подій. На їхній основі можуть бути сконструйовані також типи моделей, подібні підтримувані сильно типізованими моделями.

У таких моделях посилання на об'єкти й самі об'єкти розділені, а повідомлення інтерпретуються з урахуванням контексту. Це дозволяє реалізувати множинність посилань і забезпечити розмаїтість інтерпретації.

Інфологічна модель може містити в собі ряд компонентів. Принциповою

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

особливістю цієї моделі є можливість відображення як формалізуємих засобами формальної логіки процесів і об'єктів, так і не формалізуємих надалі процесів.

Основними компонентами інфологічної моделі є:

- опис предметної області;
- опис методів обробки;
- опис інформаційних потреб користувача.

Інфологічна модель носить описовий характер. У силу деякої довільності форм опису в цей час не існує загальноприйнятих способів її побудови. Використовують аналітичні методи, методи графічного опису, системний підхід.

Логічна структура бази даних визначає:

- таблиці і їхні імена, також називані *сутностями* (entities);
- імена полів, також називані *атрибутами* (attributes) кожної таблиці;
- характеристики полів, наприклад унікальність їхнього значення й допустимість значень NULL, а також тип даних, збережених у полі;
- первинний ключ кожної таблиці – поле (кілька полів) зі значеннями, що унікально ідентифікують кожний запис у таблиці. У таблиці також можуть існувати інші унікальні поля, але тільки одне з них розглядається як унікальний ключ доступу для пошуку записів – первинний ключ. У таблиці не обов'язково повинен існувати первинний ключ, однак рекомендується визначати його для кожної таблиці;
- зв'язки між таблицями. Записи в таблиці можуть залежати від одного або декількох записів іншої таблиці. Такі відносини між таблицями називаються *зв'язками*. Зв'язок визначається в такий спосіб: поле або кілька полів однієї таблиці, називаний *зовнішнім ключем*, посилається на первинний ключ іншої таблиці.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

3.2 Розробка структурної схеми

Структурна схема розробленої системи наведена на рисунку 3.10.

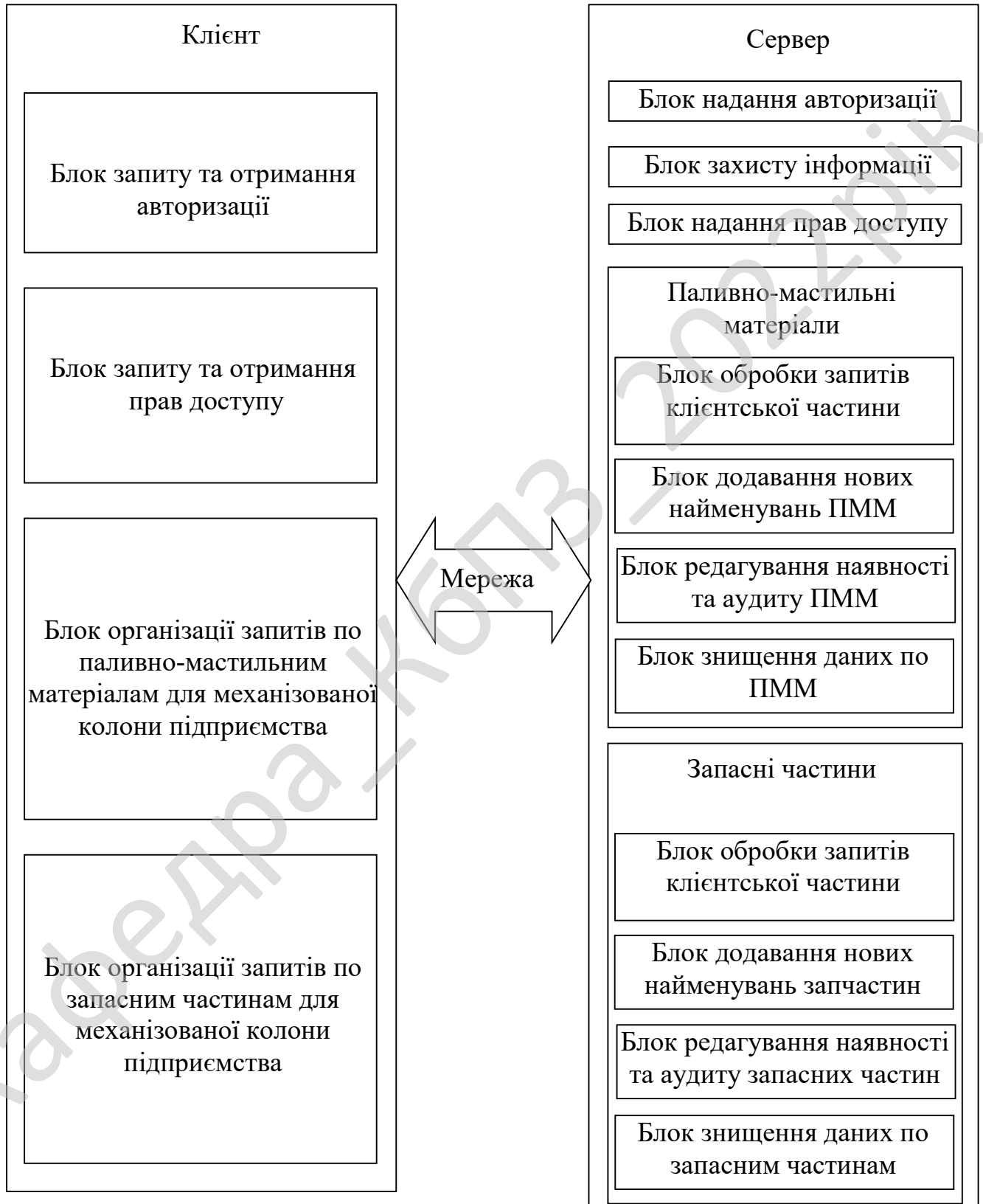


Рисунок 3.10 – Структурна схема розробленої системи

З структурної схеми ми бачимо, що у рамках магістерського проекту реалізовано клієнт-серверну архітектуру доступу до бази даних, де зберігаються дані по паливо-мастильним матеріалам та по запасним частинам для техніки механізованої колони підприємства.

Архітектура клієнт-сервер сьогодні являє собою домінуючу концепцію у створенні розподілених мережевих застосувань і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як “клієнт”. Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є більш складною; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються броузерами; найпоширенішим броузером є Internet Explorer.

Але, крім броузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма; назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит; відповідь пересилається по мережі клієнтові.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна виокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку застосування і на

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

якому здійснюється необхідна обробка інформації;

- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють :

- модель тонкого клієнта, в рамках якої вся логіка застосування та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта.

Тонкими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Виокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка застосування. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів застосувань, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає). Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї; форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними

Браузер формує запит та персилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які

забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

Таким чином, розглянувши загальні поняття клієнт-серверної архітектури, перейдемо до розробленої, у результаті виконання магістерського проекту, програми.

На клієнтській стороні реалізуються наступні функції:

– Блок запиту та отримання авторизації. У цьому блоці реалізується авторизація користувача. Також у цьому блоці реалізовано можливість встановлення та зміни логіну.

– Блок запиту та отримання прав доступу. У цьому блоці реалізується процедура запиту на права доступу, й відповідно отримання прав користувача або адміністратора, з наданими їм повноваженнями.

– Блок організації запитів по паливно-мастильним матеріалам для механізованої колони підприємства. Запити можуть організовуватися або користувачем, у цьому випадку він має право доступу тільки до кількості матеріалів, або адміністратором, у цьому випадку адміністратор має право редагування складу та кількості паливно-мастильних матеріалів.

– Блок організації запитів по запасним частинам для авто техніки механізованої колони підприємства. Запити можуть організовуватися або користувачем, у цьому випадку він має право доступу тільки до кількості запасних частин, або адміністратором, у цьому випадку адміністратор має право редагування складу та кількості запасних части для автотехніки.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Перейдемо до розгляду серверної частини. Доступ до серверної частини має тільки адміністратор та відповідний програміст, який веде цю розподілену базу даних.

На серверній частині реалізовані наступні функції:

– Блок надання авторизації. У цьому блоці на запит зі сторони сервера відбувається перевірка чи є авторизованим користувач. У випадку, якщо відбувається нова авторизація, то перевіряється те, хто проводить авторизацію. Якщо її проводить адміністратор, то нова авторизація дозволяється у іншому випадку, вона заборонена.

– Блок захисту інформації. Цей блок призначен для захисту інформації, яка зберігається у базі даних, та недопущенню відносно неї несанкціонованих дій. Для цього відбувається шифрування усіх даних які зберігаються у базі даних, та перевіряється цифровий підпис при кожному зверненні до бази даних, для встановлення авторства користувача при тій, або іншій дії.

– Блок надання прав доступу. Цей блок надає відповідні права доступу при запиті зі сторони адміністратора. Також у цьому блоці перевіряються права доступу користувача та адміністратора й у випадку порушення переривається робота з базою даних.

Також серверна частина містить два великі блоки. Один з яких обробляє усі запити, які відносяться до паливо-мастильних матеріалів, а другий обробляє усі запити, які відносяться до запасних частин автотехніки механізованої колони підприємства.

Розглянемо їх більш детально.

Блок паливно-мастильних матеріалів підрозділяється на наступні блоки:

– Блок обробки запитів клієнтської частини. У цьому блоці обробляються усі запити які мають відношення до дій з рухом паливо-мастильних матеріалів.

– Блок додавання нових найменувань паливно-мастильних матеріалів. У цьому блоці розширюється діапазон тих паливно-мастильних матеріалів, з якими мають справу у механізованій колонні підприємства.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

– Блок редагування наявності та аудиту ПММ. У цьому блоці адміністратором редагується наявність кількості тих або інших видів паливно-мастильних матеріалів, та реалізована можливість проведення аудиту наявності паливно-мастильних матеріалів на складі.

– Блок знищення даних по паливно-мастильним матеріалам. У цьому блоці відбувається знищення тих, або інших видів паливно-мастильних матеріалів.

Блок запасних частин підрозділяється на такі блоки:

– Блок обробки запитів клієнтської частини. У цьому блоці обробляються усі запити які мають відношення до дій з рухом запасних частин.

– Блок додавання нових найменувань запасних частин. У цьому блоці розширюється діапазон тих запасних частин, з якими мають справу у механізованій колонні підприємства.

– Блок редагування наявності та аудиту запасних частин. У цьому блоці адміністратором редагується наявність кількості тих або інших видів запасних частин, та реалізована можливість проведення аудиту наявності запасних частин на складі.

– Блок знищення даних по запасним частинам. У цьому блоці відбувається знищення тих, або інших видів запасних частин.

3.3 Розробка функціональної схеми

Функціональна схема розробленого програмного забезпечення зображена на рисунку 3.11.

З функціональної схеми ми бачимо, що спершу завантажується головна форма на якій виводиться вікно авторизації. Далі у відповідності з введеним логіном та паролем відбувається перехід або на сторінку адміністратора, або на сторінку користувача системи.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46



Рисунок 3.11 – Функціональна схема розробленого програмного забезпечення

При отриманні прав адміністратора можливе виконання наступних дій над розробленою системою:

– Відкриття доступу користувачам до бази даних паливно-мастильних матеріалів та автозапчастин згідно відповідних прав доступу. Це дозволяє розмежовувати доступ до бази даних згідно отриманих повноважень.

– Шифрування даних у базі в режимі реального часу. Це дозволяє забезпечувати безпеку зберігаємих у базі даних значень.

– Створення нових типів паливно-мастильних матеріалів. Цією функцією треба користуватися, якщо виникла необхідність використовувати типи пального та мастильних матеріалів, які не використовувалися раніше у цій механізованій колоні підприємства.

– Створення нових типів запасних автомобільних частин. Цією функцією треба користуватися, якщо виникла необхідність використовувати типи запасних автомобільних частин, які не використовувалися раніше у цій механізованій колоні підприємства. Наприклад при закупівлі автомобілів марок та типів. Яких не було раніше в автоколоні.

– Аудит роботи з базою даних. Ця функція призначена для того, щоб відстежувати хто і як працював з базою даних. Тобто, коли мав доступ до бази даних, й з якими даними у ній працював.

– Формування звіту. Ця функція дозволяє формувати звіт як по аудиту доступу до бази даних, так й по аудиту наявності паливно-мастильних матеріалів та запасних частин.

Перейдемо до розгляду функцій які можна реалізувати у сторінці користувача. До них відносяться наступні:

– Отримання доступу користувачем до бази даних паливно-мастильних матеріалів та автозапчастин згідно відповідних прав доступу. Це дозволяє мати певні повноваження при роботі з базою даних у якій зберігаються відповідні дані.

Після виконання авторизації, користувач може працювати з двома

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

великими блоками: блок роботи з паливно-мастильними матеріалами та блок роботи з запчастинами.

При праці з блоком роботи з ПММ, користувач може виконувати наступні дії:

- Вибір типу масла.
- Вибір типу та марки бензину. Наприклад А-76 або А-98.
- Вибір типу дизельного пального.
- Перегляд наявності того або іншого виду та марки ПММ.
- Редагування наявності того або іншого виду та типу ПММ.

При праці з блоком роботи з автозапчастинами, користувач може виконувати наступні дії:

- Вибір марки автомобіля для якого потрібна запасна частина, після якого вибирається модель автомобіля для якого потрібна запасна частина.
- Перегляд наявності тієї або іншої запасної частини.
- Редагування наявності тієї або іншої запасної частини.

Розглянувши функціональну схему перейдемо до діаграми процесів, які відбуваються в мережі.

3.4 Розробка діаграми процесів

У системі відбуваються наступні процеси, які відображені на відповідній діаграмі (рисунок 3.12).

Робота програми починається з активізації процесу авторизації, після якого система запускає або процеси, які відносяться до сторінки користувача, або процеси, які відносяться до сторінки адміністратора.

Процесами, які відносяться до сторінки адміністратора є наступні:

- процес аудиту роботи з БД;
- процес управління обліковими записами;
- процес додавання нових записів;

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

- процес формування звітів;
- процес перегляду бази даних, який включає до себе процес формування запитів до БД.

Сторінка користувача надає можливість працювати з наступними процесами:

- процес вибору запчастин та масел;
- процес редагування наявності запису;
- процес пошуку запчастин та масел;
- процес перегляду бази даних, який включає до себе процес формування запитів до БД.

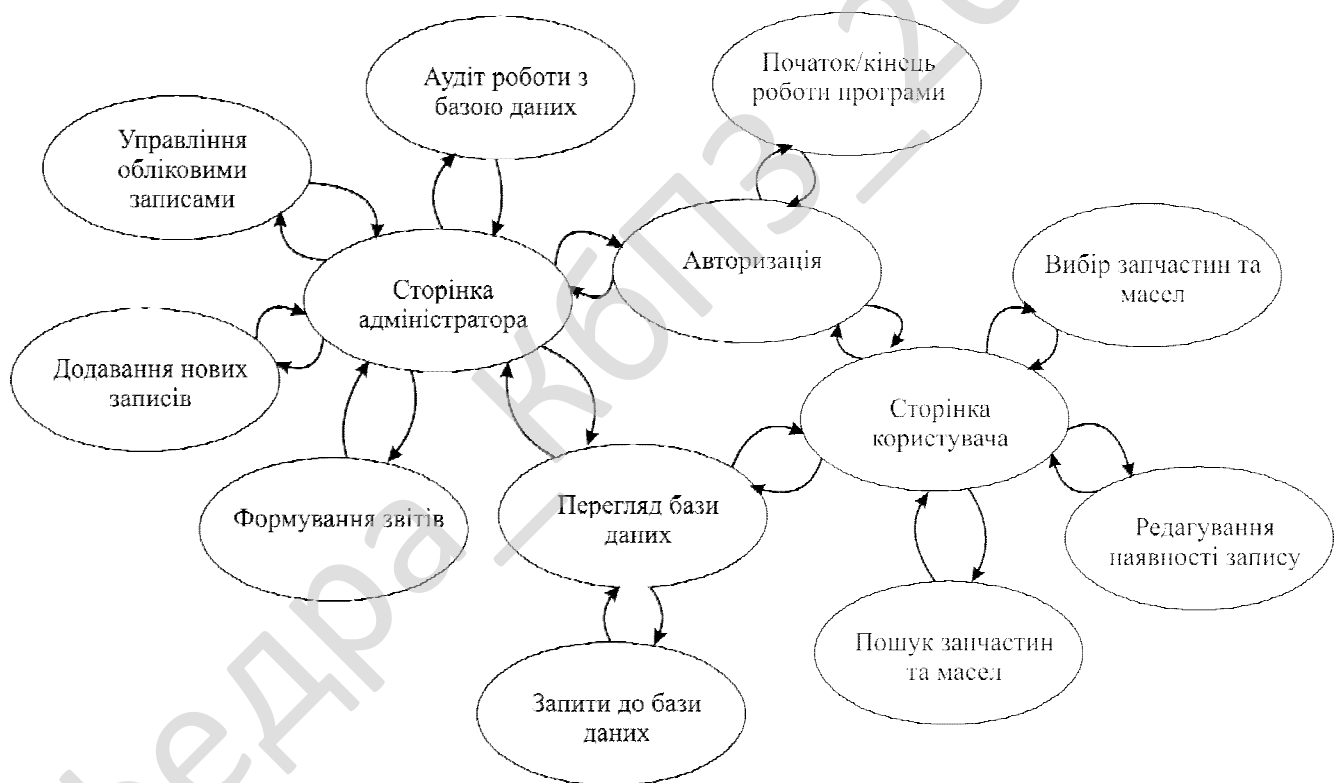


Рисунок 3.12 – Діаграма процесів розробленого програмного забезпечення

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 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 зображена блок-схема основної програми. З неї видно, що програма працює наступним чином.

Спершу виводиться вікно авторизації. У цьому вікні виводиться запит на надання прав доступу.

Відбувається перевірка реєстрації. Якщо реєстрація відбулася успішно то створюється обліковий запис.

У випадку коли реєстрація не являється потрібною, вводиться логін та пароль користувача.

Після цього перевіряється пароль та логін й відповідно перевіряється наявність облікового запису користувача системи.

Якщо обліковий запис не існує то виводиться повідомлення про помилку, та відбувається перехід на початок роботи програми.

Якщо обліковий запис існує, то перевіряється, права доступу – це обліковий запис з правами адміністратора, чи правами користувача.

Якщо є права адміністратора, то виводиться сторінка адміністратора, на якій розблоковуються усі функції.

Це надає змогу, окрім інших функцій проводити аудит роботи з базою даних та управляти обліковимим записами.

Якщо обліковий запис не має прав адміністратора, то перевіряється, чи має він права користувача.

Якщо він цих прав не має, то обліковий запис блокується й з'являється вікно авторизації.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

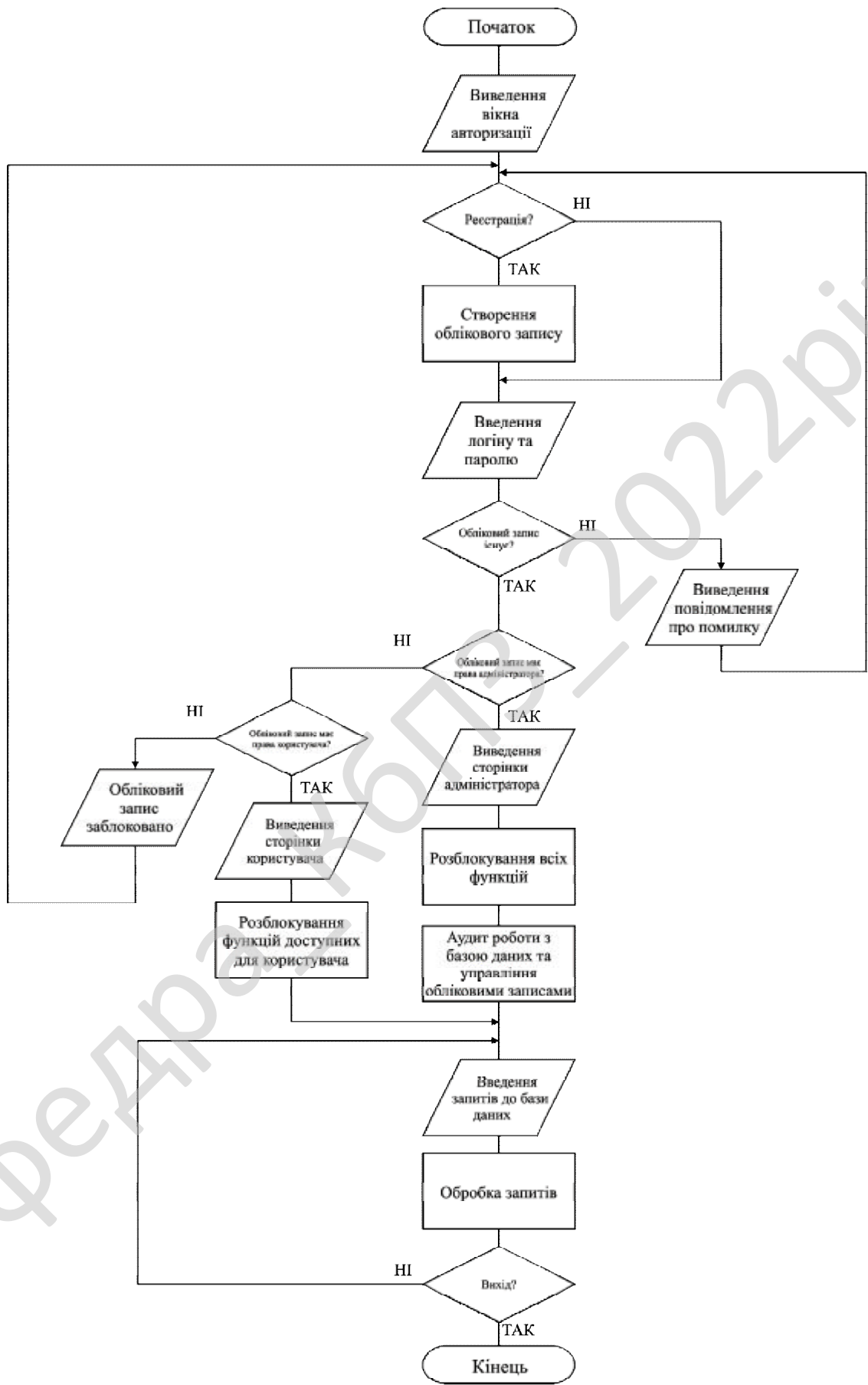


Рисунок 4.1 – Блок-схема основної програми

Якщо ж обліковий запис має права користувача, то виводиться сторінка користувача, й розблоковуються усі функції, які може використовувати користувач.

Після цього відбувається введення запитів до БД, та їх обробка.

По закінченню роботи з базою даних існує можливість завершення програми, або її продовження.

У випадку продовження роботи з програмою, відбувається перехід до вікна введення запитів.

На рисунку 4.2 наведено блок-схему підпрограми роботи з базою даних. Ця робота складається з виконання наступних етапів.

Спершу відбувається підключення до бази даних, з відповідними правами доступу.

Перша дія, яку можливо виконати – перегляд бази даних. Якщо він відбувається то спершу вибирається марка автомобіля, та виводиться список запчастин та ПММ для цієї марки.

Якщо немає потреби переглядати БД, то відбувається перехід до додавання запису у БД.

При необхідності додавання запису у БД виконуються наступні дії:

- перехід у режим редагування;
- вказання групи товару та марки машини;
- привласнення коду запису;
- введення назви запису;
- введення дати випуску та надходження на склад;
- введення кількості автозапчастин та ПММ, для конкретного автомобіля;
- введення ціни закупки;
- додавання запису у БД.

У випадку, якщо немає необхідності додавати запис, то можливо перейти до видалення запису.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Видалення запису відбувається наступним чином:

- спершу відбувається перехід в режим редагування;
- потім вибирається марка машини;
- після цього виводиться список запчастин та масел для даної марки;
- за цією дією слідує вибір запису для видалення;
- відповідно цей запис видаляється.

Якщо немає необхідності видаляти запис з БД, то існує можливість редагування записів у БД.

Для цього виконуються наступні кроки:

- перехід у режим редагування;
- вибір марки машини;
- виведення списку запчастин та ПММ для даної марки;
- вибір запису для редагування;
- виведення змін;
- внесення змін у БД.

Якщо немає потреби редагувати БД то тоді є можливість упорядкування бази даних по тому, або іншому полюта признаку.

Для цього виконуються наступні дії:

- вибирається поле, по якому відбувається сортування;
- відбувається сутньо сортування записів;
- виводяться відсортовані записи.

Остання функція, яку можливо використувати при роботі з БД є функція пошуку.

Вона реалізується наступним чином:

- визначаються параметри пошуку;
- вводиться фраза для пошуку;
- відбувається пошук по ключовій фразі;
- відбувається виведення результатів пошуку.

Це усі основні дії, які можливо виконати при роботі з БД.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

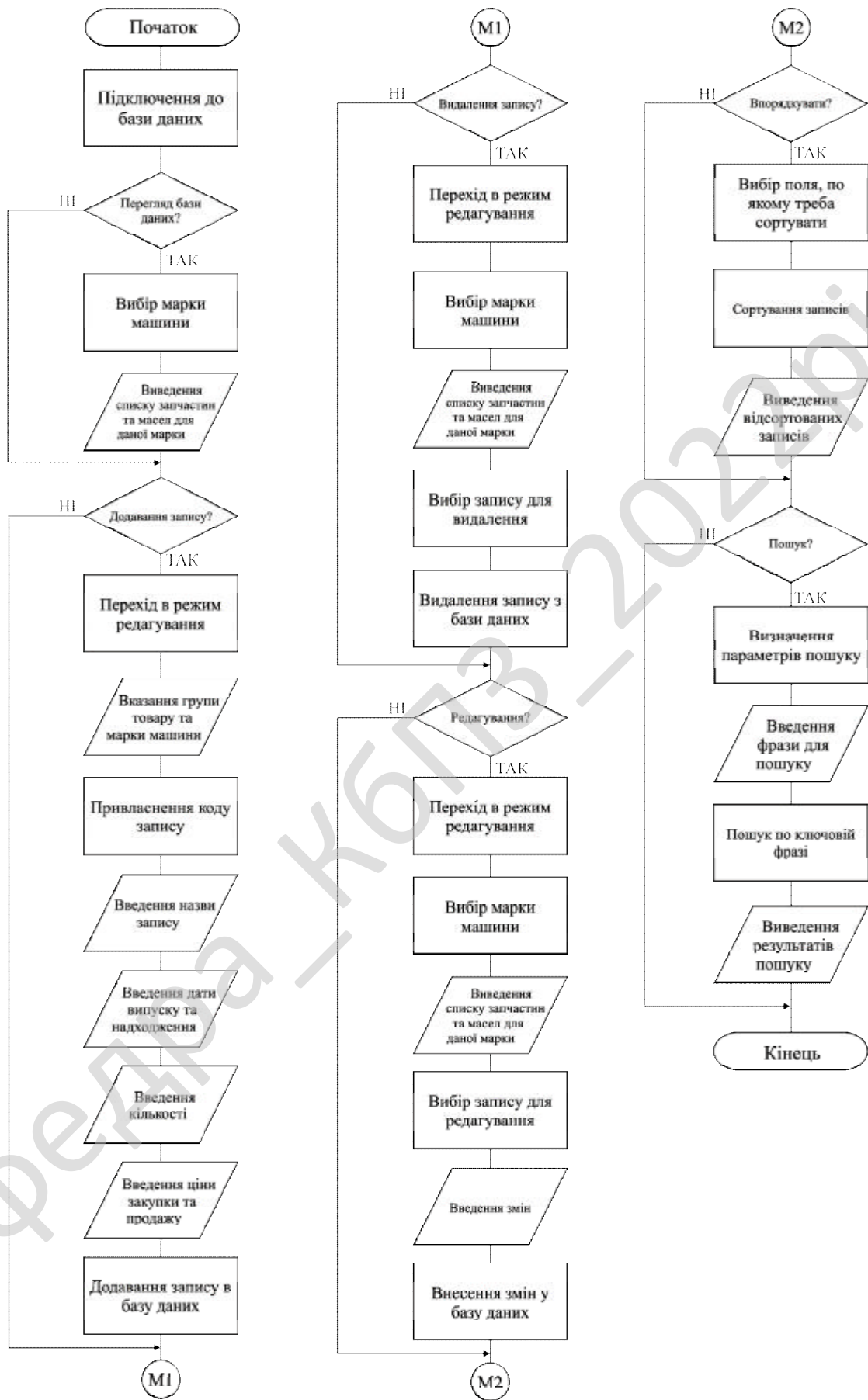


Рисунок 4.2 – Блок-схема підпрограми роботи з базою даних

Нижче наведемо частини коду програми, які призначені для роботи з БД.

```
// підпрограма купівлі товару (запчастин або ПММ)
//виклик процедури
procedure TForm1.Button2Click(Sender: TObject);
begin
mainform.warehouse.Delete;
Form1.Close;
end;
// процедура закупки товару
procedure TForm1.Button1Click(Sender: TObject);
begin
    mainform.warehouse.InsertRecord([edit2.text,edit1.text,maskedit1.text,
        spinedit2.text,spinedit1.text,',',combobox1.text,datetimepicker1.date])
        ;
    mainform.Memo1.Lines.Add('Здійснена купівля'+timetostr(time));
form1.Close;
end;
//процедура введення ціни товару
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    mainform.warehouse.InsertRecord([edit2.text,edit1.text,maskedit1.text,
        spinedit2.text,spinedit1.text,',',combobox1.text,datetimepicker1.date])
        ;
    mainform.Memo1.Lines.Add(timetostr(time)+' Здійснена купівля
        '+edit1.Text+' [ '+spinedit2.Text+' по ціні '+spinedit1.text+'р. ]');
edit1.Clear;
edit2.Clear;
combobox1.Clear;
maskedit1.Clear;
spinedit1.Clear;
spinedit2.Clear;
form1.Close;
end;
// закриття форм
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
edit1.Clear;
edit2.Clear;
combobox1.Clear;
maskedit1.Clear;
spinedit1.Clear;
spinedit2.Clear;
Form1.Close;
end;
//підпрограма пошуку запису у базі даних
//виклик процедури
procedure TForm2.DBEdit1Change(Sender: TObject);
begin
mainform.warehouse.FindField('Код');
end;
//задавання поля пошуку
procedure TForm2.Edit1Change(Sender: TObject);
var lan,i,mark: integer;
            strmark: string;

begin
lan:=mainform.warehouse.RecordCount;
mainform.warehouse.First;
//організація пошуку
if edit1.text='' then begin mark:=0; end
else begin
            if p=0 then begin mark:=strtoint(edit1.text);
```

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```

        for i:=0 to lan do begin
            if mainform.warehouse.Fields[p].AsInteger=mark then
                else mainform.warehouse.Next;
            end; end;
        if ((p=1) or (p=6)) then begin strmark:=edit1.Text;
        for i:=0 to lan do begin
            mainform.warehouse.setkey;

mainform.warehouse.IndexFieldNames:=mainform.warehouse.Fields[p].Field
Name;
            mainform.warehouse.FindNearest([edit1.text]);
            end; end;

end;
end;
procedure TForm2.ComboBox1Change(Sender: TObject);
begin
case combobox1.ItemIndex of
0: p:=0;
1: p:=1;
2: p:=6; end;
end;
procedure TForm2.ComboBox2Change(Sender: TObject);
begin
case combobox2.ItemIndex of
0: pd:=2;
1: pd:=7; end;
end;
procedure TForm2.Button1Click(Sender: TObject);
begin
if pd=0 then pd:=2;
if maskedit1.Text='' then else
begin mainform.warehouse.SetKey;

mainform.warehouse.IndexFieldNames:=mainform.warehouse.Fields[pd].Fiel
dName;

mainform.warehouse.FindNearest([maskedit1.text]);
end;

end;
//закриття форми
procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
edit1.Text:=''; maskedit1.Text:='';
end;
//підпрограма вивантаження/додавання на склад
//виклик процедури
procedure TForm3.Button2Click(Sender: TObject);
begin
form3.Close;
end;
procedure TForm3.Edit1Change(Sender: TObject);
var lan,i,mark: integer;
strmark: string;
begin
lan:=mainform.warehouse.RecordCount;
mainform.warehouse.First;
if edit1.text='' then begin mark:=0; end
else begin
if p=0 then begin mark:=strtoint(edit1.text);
for i:=0 to lan do begin
if mainform.warehouse.Fields[p].AsInteger=mark then
else mainform.warehouse.Next;
end; end;

```

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

```

        if ((p=1) or (p=6)) then begin strmark:=edit1.Text;
        for i:=0 to lan do begin
            mainform.warehouse.setkey;

            mainform.warehouse.IndexFieldNames:=mainform.warehouse.Fields[p].Field
            Name;
                mainform.warehouse.FindNearest([edit1.text]);
                    end;    end;

            end;
        end;
    procedure TForm3.ComboBox1Change(Sender: TObject);
    begin
        case combobox1.ItemIndex of
            0: p:=0;
            1: p:=1;
            2: p:=6;
        end;
    end;
    //процедура загрузки данных на склад
    procedure TForm3.Button3Click(Sender: TObject);
    begin
        mainform.Query1.Active:=false;
        mainform.Query1.Close;
        mainform.Query1.SQL.Clear;
        mainform.Query1.SQL.Add('select '+'Код, Назва, "Дата випуску",
        "Кількість", "Ціна закупки", "Група" '+'from
        '+'mainform.ComboBox1.Text+' where '+'combobox1.Text+'='+edit1.text);
        mainform.Query1.Active:=true;
        dbedit1.DataField:='Кол-во';
    end;
    //процедура знищення даних з БД
    procedure TForm3.BitBtn1Click(Sender: TObject);
    var N,RN: integer;
    begin
        if radiobutton1.Checked=true then begin
            N:=strtoint(edit2.Text);
            RN:=strtoint(dbedit1.Text);
            RN:=RN-N;
            memol.Lines.Add('Вигружено '+edit2.Text+' одиниць
            '+mainform.warehouse.FieldValues['Назва']);
            mainform.memol.Lines.Add(Timetostr(time)+' Вигружено '+edit2.Text+'
            одиниць '+mainform.warehouse.FieldValues['Назва']);
            end;
            if radiobutton2.Checked=true then begin
                N:=strtoint(edit2.Text);
                RN:=strtoint(dbedit1.Text);
                RN:=RN+N;
                memol.Lines.Add('Додано '+edit2.Text+' одиниць
                '+mainform.warehouse.FieldValues['Назва']);
                mainform.memol.Lines.Add(Timetostr(time)+' Додано '+edit2.Text+'
                одиниць '+mainform.warehouse.FieldValues['Назва']);
                end;
            mainform.Query1.Close;
            datasourcel.Enabled:=false;
            mainform.warehouse.edit;
            mainform.warehouse.FieldName('Кількість').AsInteger:=RN;
            mainform.warehouse.Post;
            mainform.Query1.Open;
            datasourcel.Enabled:=true;
            edit2.Text:='';
        end;
    procedure TForm3.BitBtn2Click(Sender: TObject);
    begin

```

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

```

        mainform.Query1.Active:=false;
        edit1.Clear;
        edit2.Clear;
        dbedit1.DataField:='';
        form3.close;
    end;
    procedure TForm3.Edit2Change(Sender: TObject);
    var N,RN: integer;
    begin
    end;
    procedure TForm3.Button1Click(Sender: TObject);
    begin
        dbgrid1.DataSource:=datasourcel;
        mainform.Query1.Active:=false;
        mainform.Query1.Close;
        mainform.Query1.SQL.Clear;
        mainform.Query1.SQL.Add('select '+'Код, Назва, Група '+'from
        "../data/'+mainform.ComboBox1.Text+'" where
        '+combobox1.Text+'="'+edit1.text+'"');
        mainform.Query1.Active:=true;
        dbedit1.DataField:='Кількість';
    end;
    procedure TForm3.FormCreate(Sender: TObject);
    begin
    dbgrid1.DataSource:=mainform.DataSource1;
    end;
    procedure TForm3.ComboBox2Change(Sender: TObject);
    begin
        case combobox1.ItemIndex of
        0: p:=0;
        1: p:=1;
        2: p:=6; end;
    end;
    //процедура знищення форми
    procedure TForm3.FormDestroy(Sender: TObject);
    begin
        mainform.Query1.Active:=false;
        edit1.Text:='';
        edit2.Text:='';
        dbedit1.DataField:='';
    end;
    procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
    begin
        mainform.Query1.Active:=false;
        edit1.Text:='';
        edit2.Text:='';
        dbedit1.DataField:='';
    end;
    //підпрограма аудиту роботи з базою даних
    //виклик процедури перегляду даних
    procedure TForm4.Button1Click(Sender: TObject);
    var i: integer;
    begin
        //підключення до БД
        mainform.Query1.close;
        mainform.Query1.SQL.Clear;
        //Організація запиту на SQL
        mainform.Query1.SQL.Add('Select * from
        "../data/'+mainform.ComboBox1.Text+'" where
        Група="'+combobox1.text+'"');
        mainform.Query1.Open;
        mainform.Query1.Active:=true;
        datasourcel.DataSet:=mainform.Query1;
    end;

```

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

```

end;
procedure TForm4.Button2Click(Sender: TObject);
begin
form4.close;
end;
procedure TForm4.Button3Click(Sender: TObject);
var i:integer;
begin
mainform.warehouse.First;
for i:=1 to mainform.warehouse.recordcount do begin
if mainform.warehouse.FieldByName('Дата випуску').value='' then
datasourcel.DataSet:=mainform.warehouse.Fields.DataSet;
mainform.warehouse.Next;
end;
end;
end;

```

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ДСТУ 8845:2019 – алгоритм симетричного потокового перетворення. В основі ДСТУ 8845:2019 лежить класична схема підсумовуючого генератора подібна генератору SNOW 2.0, SNOW 3.0 та SNOW V. В основі ДСТУ 8845:2019 збережені всі базові операції шифрів сімейства SNOW, а раунд шифрування AES замінений на функцію нелінійної підстановки T, що реалізовує перестановку елементів скінченного поля $GF(2^{64})$ за допомогою компонентів національного стандарту симетричного криптоперетворення ДСТУ 7624:2014.

ДСТУ 8845:2019 використовує 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K і забезпечує високий та надвисокий рівень стійкості із врахуванням можливого застосування квантового криптографічного аналізу. При розробці алгоритму ДСТУ 8845:2019 орієнтувалися на сучасні 64-бітні обчислювальні системи, тому розмір слова обрано рівним 8 байт. запису байтів застосовують подання від старшого до молодшого. Генератор ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру схематично приведено у стандарті.

Як впливає з генератора ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру основними компонентами генератору є регістр зсуву з лінійним зворотнім зв'язком та скінчений автомат на базі якого виконується

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

нелінійне перетворення T . Вхідні дані (ключ шифрування K та вектор ініціалізації IV) використовуються для ініціалізації змінної стану $S_i (i \geq 0)$, яка складається із двох компонент до складу яких входять [12]:

- 16 змінних $s^{(i)}$ – комірок регістра зсуву з лінійним зворотнім зв'язком: $s^{(i)} = (s_{15}^{(i)}, s_{14}^{(i)}, s_{13}^{(i)}, s_{12}^{(i)}, s_{11}^{(i)}, s_{10}^{(i)}, s_9^{(i)}, s_8^{(i)}, s_7^{(i)}, s_6^{(i)}, s_5^{(i)}, s_4^{(i)}, s_3^{(i)}, s_2^{(i)}, s_1^{(i)}, s_0^{(i)})$;

- Два регістри скінченного автомату $r^{(i)} : r^{(i)} = (r_2^{(i)}, r_1^{(i)})$. На виході отримуємо ключовий потік (гамма), який формується з 8-байтних слів Z_i .

З рисунку слідує, що відводи регістра зсуву з лінійним оберненим зв'язком побудовані за примітивним над полем $GF(2^{64})$ поліномом: $f(x) = x^{16} + x^{13} + \alpha^{-1}x^{11} + \alpha$, де $\alpha \in$ коренем примітивного над полем $GF(2^8)$ поліному $gz(z) = z^8 + \beta^{170}z^7 + \beta^{166}z^6 + \beta^2 z^5 + \beta^{224} z^4 + \beta^{70}z^3 + \beta^2$. Поле $GF(2^8)$ як і в ДСТУ 7624:2014 побудовано за примітивним на полем $GF(2)$ поліномом $p(y) = x^8 + x^4 + x^3 + x^2 + 1$, а коефіцієнти $g(z)$ подаються через ступінь примітивного елемента β поля $GF(2^8)$, тобто β корінь поліному $p(y)$. Тобто, у нас є вежа полів: $GF(2) \subset GF(2^8) \subset GF(2^{64}) \subset GF(2^{1024})$, де:

- поле $GF(2^{1024})$ задається відводами зворотного зв'язку як фактор кільце $GF(2^{64})[x]/(f(x))$;

- поле $GF(2^{16424})$ задається як фактор кільце $GF(2^8)[z]/(g(z))$;

- поле $GF(2^{1024})$ задається як фактор кільце $GF(2)[y]/(p(y))$.

З вищезазначеного, слідує що період вихідної послідовності становить 2^{1024} .

Структурно в алгоритмі симетричного потокового перетворення ДСТУ 8845:2019 виділяють три основні функції:

- функція ініціалізації, яка приймає в якості вхідних даних 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K , і виробляє початкове значення змінної стану $S_0 = (s^{(0)}, r^{(0)})$;

- функція наступного стану $Next$, яка приймає на вхід змінну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє наступне значення змінної стану $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$;

– функція ключового потоку $Strm$, що приймає на вході змінну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє на виході 64-бітний ключовий потік Z^i .

Також функція $Next$ може виконуватися в двох режимах, в залежності від способу виконання ітерації, як частина реалізації ініціалізації алгоритму ДСТУ 8845:2019 або як частина функції ключового потоку $Strm$.

Кафедра _ КБПЗ _ 2022 рік

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Робота з програмою починається з вікна авторизації (рисунк 5.1).

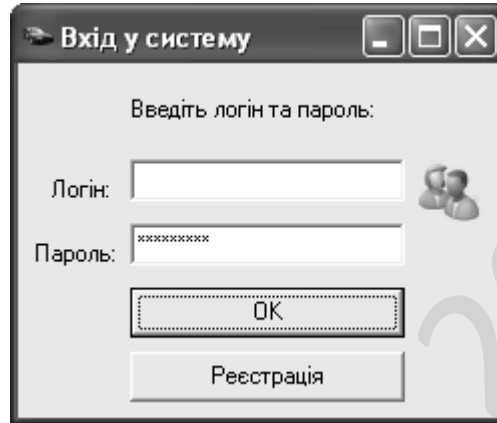


Рисунок 5.1 – Вікно авторизації

Після авторизації маємо можливість перейти до головного вікна програми (рисунк 5.2).

Код	Назва	Дата випуску	Кількість	Ціна закупки	Ціна продажу	Група	Дата народження
24357	Пчиль	24.01	2	300	1100	07 - Числове керування	
46687	Динам	27.02	3	700	800	06 - Пульт	
7689879	Аккумулятор	01.02	5	300	500	10 - Електрообладнання	
0700	Бампер	22.07	1	1250	1300	00 - Підвісь	

Рисунок 5.2 – Головне вікно програми

У головному вікні існує можливість організації швидкого пошуку, форма якого наведена на рисунку 5.3.

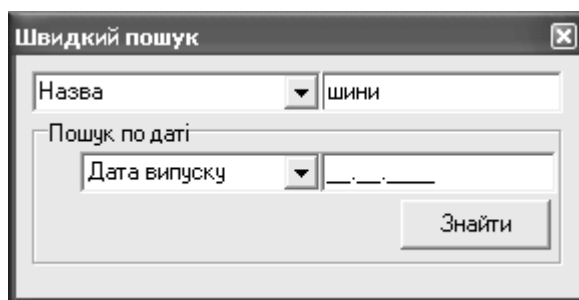


Рисунок 5.3 – Швидкий пошук

Форма купівлі наведена на рисунку 5.4.

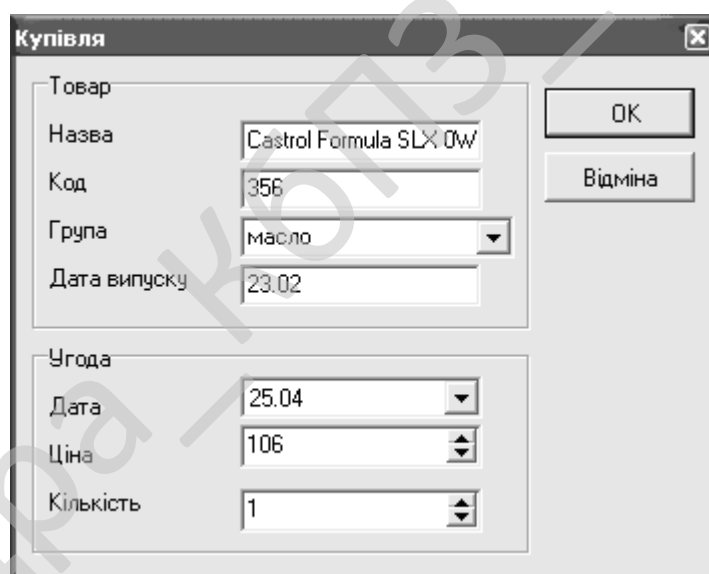


Рисунок 5.4 – Купівля

У цьому розділі наводяться не усі скріншоти, у зв'язку з тим, що їх дуже багато, тому тут наведені основні скріншоти.

До таких відноситься й вікно у якому наведені дані про автора, наукового керівника, тему програми та місце її виконання (рисунку 5.5).

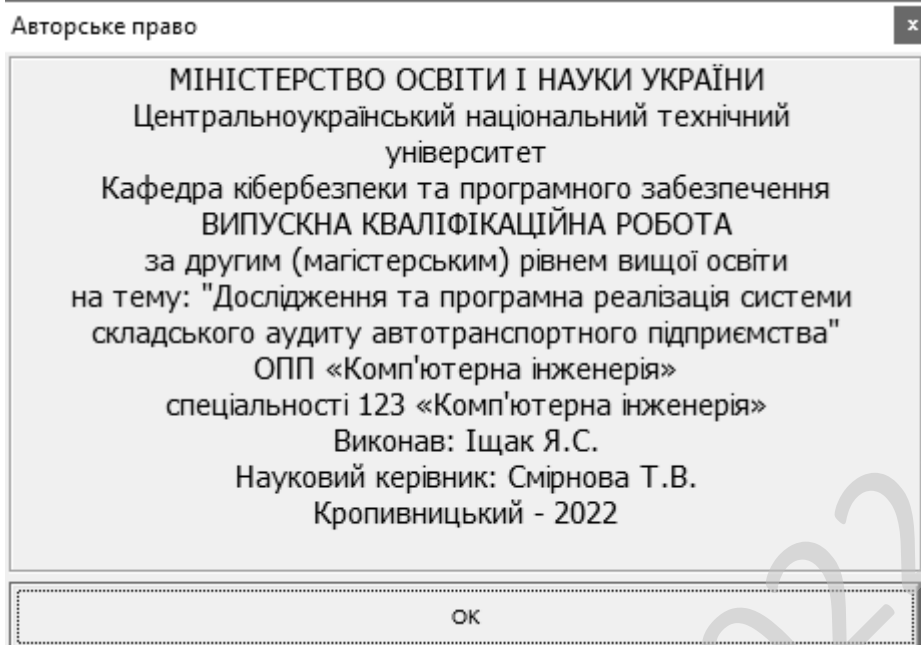


Рисунок 5.5 – Довідка про програму

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

6 НАУКОВА НОВИЗНА

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

Метою розробки є дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства.

Об'єктом дослідження є процес складського аудиту автотранспортного підприємства.

Предметом дослідження є методи складського аудиту автотранспортного підприємства.

Методи дослідження базуються на методах теорії аудиту, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод складського аудиту автотранспортного підприємства.

– Розроблено вітчизняний продукт складського аудиту автотранспортного підприємства, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 24 днів (один місяць).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи складського аудиту автотранспортного підприємства.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	30
3. Запланований термін розробки, днів	Frq	24 (1 місяць)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Г
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	8
8. Кількість форм вихідної інформації.	–	6
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	1
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	3
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	4
17. Складність кінцевого програмного продукту (1-6)	–	5
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	3
20. Вимоги до швидкодії ПП (1-6)	–	3
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	4
23. Професійний рівень аналітиків (1-6)	–	3
24. Професійний рівень програмістів (1-6)	–	4
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	1
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	30000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	40
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

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

B – показник ступеня, що визначається співвідношенням:

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

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

$$B = 1,01 + 0,001(2,43 + 3,64 + 4,22 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,2^{1,027} = 5,5 \text{ люд-міс.}$$

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

$$T_{уточн} = T_{ном} PV_j, \quad (7.3)$$

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

$$T_{уточн} = 5,5 \cdot (1 \cdot 1,09 \cdot 1,30 \cdot 0,91 \cdot 1 \cdot 1 \cdot 1 \cdot 1,15 \cdot 1 \cdot 0,87 \cdot 1,10 \cdot 1,22 \cdot 1,12 \cdot 1,10 \cdot 1 \cdot 1 \cdot 1,10) = 12,9 \text{ люд-міс.}$$

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

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

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

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 12,9^{0,33+0,2(1,027-1,01)} \cdot 60 = 112 \text{ люд/день.}$$

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	15	Д7
Робочий проект	112	Ф 7.1-7.4
Впровадження	15	Д13
Всього	161	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{nz} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{161 \cdot 1}{24 \cdot 3} = 7,7 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

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

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	10	900	15
Монітор	60	10	600	10
Клавіатура	30	10	300	5
Маніпулятор «мишка»	30	10	300	5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м. п.	2,5	200	500	8,33
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	51,99

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{\text{ор}}^c = \frac{52 \cdot 1}{1,2} = 43 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{\text{ел}} = \frac{\Phi_{\text{ор}}^c}{F_{\text{ор}} \cdot T_{\text{зм}}}, \quad (7.7)$$

$$Ч_{ел} = 43 / (24 \cdot 8) = 0,2 \text{ ставки.}$$

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

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2016, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	1	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	1	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	1	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

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

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	0,5	11112	5556
Продакт-менеджер	0,5	8500	4250
Інженер-програміст	7,7	8500	65450
Інженер-електронщик	0,2	8000	1600
Інженер-системотехнік	0,5	8000	4000
Адміністратор мережі	0,5	8000	4000
Дизайнер WEB	0,5	10000	5000
Всього за період розробки	$R_{cn} = 10,4$	-	$\Phi_{роб} = 89856$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$Z_{cd} = \frac{89856}{10,4 \cdot 24} = 360 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

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

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 13 робочих місць;

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

$C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./м². Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./м². На кожне робоче місце у середньому потрібно 8 м². З урахуванням цього:

$$B_{уд} = 13 \cdot 8 \cdot 20000 = 2080000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 208000 грн. Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{сн}^1 \cdot C_{м}, \quad (7.10)$$

де: $C_{м}$ – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 13 \cdot 3500 = 45500 \text{ грн.}$$

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Brain за 14.10.22 – джерело <http://brain.com.ua>.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11457
Системний блок		7509
Процесор	Intel Core i7-4790 (4(8) ядра по 3.6 - 4. GHz); Cache Memory 8 MB	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Системна плата	1st Player ATX NEW	-
Відеокарта	PCIeX: ATI HD5670 SAPPHIRE 1024MB/128bit/DDR3/TV/DualDVI	-
Жорсткий диск	SSD: 480 Gb	-
Оперативна пам'ять	Kingston DDR3 2GB (KVR1333D3N9/2GB) Intel/AMD – 2 шт	-
DVD-привод	-	-
Корпус	ATX Middle Tower GIGABYTE GZ-X4 Silver 500W (GZ-X4 Silver)	-
Кулер	-	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: A Type Cards, black	-
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1 170/160, D-SUB, Wide)	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37 Photo	2970
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	UPS APC BACK-UPS ES 525VA 230V RUSSIA (BE525-RS)	1348

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

					БКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	13	11457	14894,1	163835,1
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2970	297	3267
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	185653,6

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	2080000	-	-
2. Передавальні пристрої	208000	-	-
Всього по групі	2288000	5	114400

Продовження таблиці 7.8

1	2	3	4
Група 4			
3. Обчислювальна техніка	185654	-	-
Всього по групі	185654	50	92827
Група 5, 6			
4. Вимірювальні пристрої	3999	25	-
5. Транспортні засоби	0	20	-
6. Господарський інвентар	45500	25	-
Всього по групі	49499	-	12374,75
7. Нематеріальні активи	30000	10	3000
Разом	$K_p = 2553153$		$A_p = 22260$ 2

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 360 \cdot 161 / 30 = 1932 \text{ грн.}$$

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

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

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 1932 \cdot 10 \cdot 0,01 = 193 \text{ грн.}$$

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(1932+193) = 468 \text{ грн.}$$

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

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

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 1932 \cdot 15 \cdot 0,01 = 290 \text{ грн.}$$

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

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

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджей, тонеру, грн.;

N_e – кількість екземплярів програм, шт.

Згідно прийнятих норм на підприємстві $n_{вум}$ приймаємо 0,2 пачек паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 200$ грн., визначаємо вартість паперу за період розробки:

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

$$Z_{M1} = 200 \cdot 0,2 = 40 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 3):

$$Z_{M2} = \sum Ц_d, \quad (7.17)$$

де: $Ц_d$ – вартість дисків CD/DVD: CDR box – 23 грн./шт., DVD-R box – 35 грн./шт.

$$Z_{M2} = 35 \cdot 2 + 23 = 93 \text{ грн.}$$

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_3, \quad (7.18)$$

де: C_3 – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

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

$$Z_M = (40 + 93 + 1702) / 30 = 61 \text{ грн.}$$

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

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

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

$$O_n = 1932 \cdot 15 \cdot 0,01 = 290 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 30$ прим.):

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

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 222602 \cdot 1 / (30 \cdot 12) = 618 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

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

$$C_n = 1932 + 193 + 468 + 290 + 61 + 290 + 618 = 3852 \text{ грн.}$$

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	1932
2. Додаткова зарплата виконавців	Z_d	193
3. Відрахування на соціальні потреби	C_{oc}	468
4. Загальногосподарські витрати	G_{ocn}	290
5. Витрати на матеріали	Z_M	61
6. Освоєння нових операційних систем, мов програмування	O_n	290
7. Амортизація основних фондів	A_M	618
8. Повна собівартість програмного забезпечення	C_n	3852
9. Плановий прибуток	Π_p	1541
10. Ціна підприємства $C_n = C_n + \Pi_p$	C_n	5393
11. Податок на додану вартість $\Pi_{ДВ} = 0.01 \cdot H_{ов} \cdot C_n$	$\Pi_{ДВ}$	1077
12. Відпускна ціна програмної продукції $C = C_n + \Pi_{ДВ}$	C	6470

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

Для даного програмного забезпечення рівень рентабельності складає 40%.

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

де: P_n – рівень рентабельності, %.

$$\Pi_p = 0,01 \cdot 40 \cdot 3852 = 1541 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	6470
Всього капітальних витрат	–	6470

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Витрати на профілактичні роботи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	Z_p	32208	20130
2. Витрати на електроенергію	$Z_{ел}$	277	116
3. Витрати на амортизацію	$Z_{ам}$	0	3235
Всього витрат за рік	I	32485	23481

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 100 \cdot 1,1 \cdot 1,22 = 32208 \text{ грн,}$$

до:

$$Z_{p \text{ нов}} = 150 \cdot 100 \cdot 1,1 \cdot 1,22 = 20130 \text{ грн.}$$

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

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

$$Z_{ел \text{ баз}} = 0,55 \cdot 240 \cdot 2,1 = 277 \text{ грн.}$$

$$Z_{ел \text{ нов}} = 0,55 \cdot 100 \cdot 2,1 = 116 \text{ грн.}$$

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

					БКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	6470	–	3235
Всього відрахувань	-	–	6470	–	3235

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (5393 - 3852) \cdot 30 - (0,05 \cdot 2288000 + 0,5 \cdot 185654 + 0,25 \cdot 49499 + 0,1 \cdot 30000) \cdot 1/12 = 27680 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_e = \frac{265153}{(5393 - 3852) \cdot 30 \cdot 12 / 1} = 0,5 \text{ року.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	30
2. Повна собівартість розробленої програми	Грн.	3852
3. Ціна розробленої програми	Грн.	5393
4. Плановий прибуток від реалізації розробленої програми	Грн.	1541
5. Рентабельність програмної продукції	%	40
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2553153
7. Загальний прибуток від реалізації програмної продукції	Грн.	46230
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	27680
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Рік	0,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	6470
11. Величина економічного ефекту у користувача програмної продукції	Грн.	5769
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Рік	0,72

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\sigma} - I_n) - E_n (K_n - K_{\sigma}), \quad (7.27)$$

де: I_{σ} , I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

K_{σ} , K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (32485 - 23481) - 0,5 \cdot 6470 = 5769 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_0}{I_0 - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{6470}{32485 - 23481} = 0,72 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

7.9 Висновки

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

В охорону праці включають санітарно-гігієнічні, лікувально-профілактичні та організаційно-технічні системи правових і соціально-економічних заходів.

В кожній ІТ компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

8.2 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4]

8.3 Пропозиції щодо підвищення працездатності ІТ – фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців іт-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців іт-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців іт-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців іт-індустрії

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Пропозиції щодо підвищення працездатності it-фахівців, розіб'ємо на декілька категорій:

Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням it-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють it-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження it-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці it-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність it-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві it-галузі. Тому нами пропонується закупівля тільки меблів, які пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тимбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9].

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

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

8.5 Розрахункова частина

Для захисного штучного заземлення будемо застосовувати вертикальні електроди з сталевого прокату круглого перерізу діаметром 45 мм., довжиною $L=2$ м., та горизонтальний електрод – металева полоса з перетином 45·5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру (прямокутником).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,55$ м. Відстань між вертикальними заземлювачами (електродами) $A=3$ м. Глибина закладення горизонтального контуру заземлення $t=0,75$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,75 + 2/2=1,75 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м.}$$

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багатошаровому ґрунті [11]; $\rho_2 = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунта (глина) [11].

Діаметр вертикального електрода (задан):

$$D_v = 45 \text{ мм.} = 0,045 \text{ м.}$$

Відношення $A/L=3/2=1,5$.

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [11]:

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

$$R_0 = 0,366(\rho/L)[\lg(2L/D_0) + (1/2)\lg((4T+L)/(4T-L))] =$$

$$= 0,366(54,5/2)[(\lg(2 \cdot 2/0,045) + (1/2)\lg((4 \cdot 1,75+2)/(4 \cdot 1,75-2)))] =$$

$$= 20,6 \text{ Ом.}$$

Визначаємо коефіцієнт екранування вертикальних електродів $K_{ев} = 0,53$ при орієнтовній кількості вертикальних електродів, яке дорівнює 5 [11].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при $R_{3Н} = 4 \text{ Ом}$:

$$N = R_0 / (K_{ев} R_{3Н}) = 20,6 / (0,53 \cdot 4) = 9,75 \approx 10 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{\Pi} = 1,05 \cdot A \cdot N = 1,05 \cdot 3 \cdot 10 = 30,7 \approx 31 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунта K_{Π} [11]:

$$R_{\Pi} = 0,366(\rho \cdot K_{\Pi} / L_{\Pi}) \lg(2(L_{\Pi} \cdot L_{\Pi}) / (B \cdot t)) =$$

$$= 0,366(40 \cdot 5 / 40) \cdot \lg((2 \cdot 40^2) / (0,045 \cdot 0,75)) = 11,57 \text{ Ом.}$$

де $K_{\Pi} = 5$ – табличне значення кліматичного коефіцієнта питомого опору ґрунта для відповідної кліматичної зони для з'єднуючої полоси [11]; $B = 45 \text{ мм.} = 0,045 \text{ м.}$ – ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [11]:

$$R = (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев}) =$$

$$= (20,6 \cdot 11,57) / (20,6 \cdot 0,55 + 10 \cdot 11,57 \cdot 0,53) = 3,3 \text{ Ом.}$$

де $\eta_{\Pi} = 0,55$ – табличне значення коефіцієнта екранування з'єднуючої полоси [11].

Умова $R \leq R_{3Н}$ виконується ($3,3 \leq 4$).

Так як R суттєво більше $R_{3Н}$, зменшимо кількість вертикальних електродів до 8 і виконаємо перерахунок. У результаті остаточно отримали: кількість вертикальних електродів дорівнює 8 при $R = 3,9 \text{ Ом}$.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

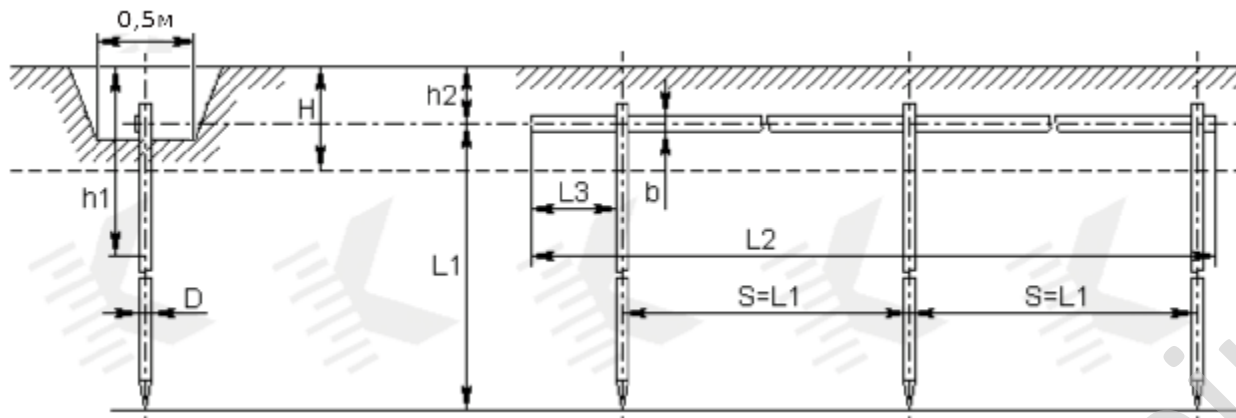


Рисунок 8.1 – Схема штучного заземлення .

8.6 Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд питань пожежної безпеки, небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів о питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

9 ОСНОВНІ ВИСНОВКИ

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

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

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

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем складського аудиту автотранспортного підприємства.

– Досліджена система складського аудиту автотранспортного підприємства.

– На основі отриманих результатів досліджень створена програмна реалізація системи складського аудиту автотранспортного підприємства.

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

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

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

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

Програма реалізована на мові високого рівня Delphi 10.4.1. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 8845:2019.

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

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 5769 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,72 роки.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іщак Я.С. Дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.

2. Коваленко А.С. Разработка структуры базы данных интегрированной информационной системы / А.С. Коваленко, А.В. Коваленко // Информационные технологии и защита информации в информационно-коммуникационных системах: монографія / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – С. 54-64.

3. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.

4. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.

5. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.

6. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

7. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

8. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

9. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

10. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.

11. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.

12. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.

13. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.

14. Коваленко А.С. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов,

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.

15. Коваленко А.С. Метод визначення оптимального комплексу робіт з відновлення працездатності інтегрованої системи технічної діагностики в умовах ресурсних обмежень / А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2016. – Вип. 3(140). – С. 69-72.

16. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.

17. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.

18. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.

19. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

20. Коваленко А.С. Задачи распознавания ситуаций в системах организационной стратегии интеграции производства и операций / А.С.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

Коваленко, А.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVI міжнар. наук.-практ. сем., 11-12 квіт. 2014 р., м. Кіровоград: зб. тез. – Кіровоград: КНТУ, 2014. – С. 53-55.

21. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.

22. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.

23. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.

24. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.

25. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

26. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у промисловості і

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

освіті: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

27. Коваленко А.С. Обґрунтування необхідності створення розподіленої бази даних для забезпечення захисту рухомих повітряних об'єктів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Перспективні напрями захисту інформації: I всеукр. наук.-практ. конф., 07 вер. 2015 р., м. Одеса: зб. тез доп. – Одеса: ОНАЗ, 2015. – С. 35-39.

28. Коваленко А.С. Розробка інформаційної моделі автоматизованої оцінки технічного стану інтегральної інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформаційні технології та взаємодії (IT & I): II між нар. наук.-практ. конф., 3-5 лист. 2015 р., м. Київ: тези доп. – Київ: КНУ ім. Т. Шевченка, 2015. – С. 41-42.

29. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Информационные и телекоммуникационные технологии: образование, наука, практика: II междунар. научн.-практ. конф., 3-4 дек. 2015 г., г. Алматы, Казахстан: сб. труд. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – Т.2. – С. 423-427.

30. Королюк Н.А. Оценка временных интервалов работы лица, принимающего решение, на автоматизированном командном пункте / Н.А. Королюк, А.И. Тимочко // Системы обработки информации. – Х.: ХУПС, 2005. – Вип. 8 (48). – С. 51-54.

31. Костерев В.В. Надёжность технических систем и управление риском: учебн. пособ. / В.В. Костерев. – М.: МИФИ, 2008. – 280 с.

32. Костюков А.В. Підвищення операційної ефективності підприємств на основі моніторингу в реальному часі. / А.В. Костюков, В.М. Костюков. – М.: Машинобудування, 2009. – 192 с.

33. Лазарев А.А. Выбор показателя затрат для анализа сравнительной экономической эффективности техники конечного потребления / А.А. Лазарев,

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

М.В. Бейлин // Сборник научных трудов ХГПУ.– Х.: ХГПУ, 1999. – Вып. 74. – С. 27-29.

34. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 1. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 308 с.

35. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 2. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 208 с.

36. Лапсарь А.П. Метод оценки состояния сложных технических объектов для синтеза быстродействующих прогнозирующих систем / А.П. Лапсарь // Измерительная техника. – 2004. – № 2. – С. 7-10.

37. Линейные задачи оптимизации: Учеб. пособие / С.В. Лутманов. – Пермь: ЛИТЕР-А, 2004. – Ч.1. – Линейное программирование. – 128 с.

38. Литвак Б.Г. Экспертные технологии в управлении. Учебное пособие / Б.Г. Литвак. – М.: Дело, 2014. – 318 с.

39. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: Посібн. / В.М. Локазюк, Ю.Г. Савченко. – К.: Видавничий центр «Академія», 2004. – 376 с.

40. Лопатников Л. И. Экономико-математический словарь: Словарь современной экономической науки / Л.И. Лопатников. – М.: Дело, 2003. – 520 с.

41. Манухина С.Ю. Инженерная психология и эргономика: хрестоматия / С.Ю Манухина. – М.: Изд. центр ЕАОИ, 2009. –224 с.

42. Мартыненко М.В. Человекомашинные процедуры поддержки организационно–управленческих решений: учеб. пособие СПбГЭТУ / М.В. Мартыненко, О.И. Шеховцов. – СПб, 2012. – 250 с.

43. Мунипов О.В. Эргономика: человекоориентированное проектирование техники, программных средств и среды: Учебник / О.В. Мунипов, В.П. Зинченко. – М.: Логос, 2001. – 356 с.

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

44. Надеев А.И. Математическая модель эксплуатационной надежности интеллектуальных датчиков / А.И. Надеев, Р.А. Юсупов, Ю.К. Свечников, Д.Р. Юсупов // Измерительная техника. – М: Стандартинформ, 2004. – № 1. – С. 8-11.

45. Надійність техніки. Аналіз надійності. Основні положення: ДСТУ 2861-94 – [Чинний від 1997-01-01]. – Київ: Держстандарт України, 1995. – 33 с. – (Національний стандарт України).

46. Надійність техніки. Терміни та визначення: ДСТУ 2860-94 – [Чинний від 1996-01-01]. – Київ: Держстандарт України, 1994. – 36 с. – (Національний стандарт України).

47. Нейлор К. Как построить свою экспертную систему / К. Нейлор. – М.: Энергоатомиздат, 2007. – 242 с.

48. Николаева И. П. Экономический словарь / И.П. Николаева. – Проспект, 2015. – 399 с.

49. Онищук А.Г. Радиоприемные устройства: Учебн. пособ. – 2-е изд., испр. / А.Г. Онищук, И.И. Забеньков, А.М. Амелин. – Минск: Новое знание, 2007. – 240 с.

50. Осипов В. Базы данных и Delphi. Теория и практика / В. Осипов. – БХВ-Петербург, 2011. – 752 с.

51. Павленко М.А. Метод кольорового кодування інформаційних елементів при розробці інформаційних моделей в перспективних АСУ / М.А. Павленко, П.Г. Берднік, Д.В. Прибильнов // Наукова весна – 2008: Матеріали міжнародної наук. – практ. конф. – Х.: МСУ, 2008. – С. 25-27.

52. Павленко М.А. Метод разработки модели деятельности оператора АСУ в системах управления сложными динамическими объектами / М.А. Павленко, О.С. Бодяк, М.Ю. Гусак, С.И. Симонов // Системы обработки информации. – Харьков: ХУПС, 2012. – Вып. 9(107). – С. 196-200.

53. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>

54. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

55. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>

56. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

57. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>

58. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>

59. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>

60. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

61. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://spo.stu.cn.ua>

62. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

					ВКРМ-123.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.22.0012.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Ицак Я.С.				<i>Дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-21М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи складського аудиту автотранспортного підприємства.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 17.08.2022 року).

3 Мета та призначення розробки

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

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-123.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

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

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4.1.

					ВКРМ-123.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута пожежна безпека.

					ВКРМ-123.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 106 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 23.12.2022 р.

					ВКРМ-123.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Смірнова Т.В.

*Дослідження та програмна реалізація
системи складського аудиту автотранспортного підприємства*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 21

Літера: РП

Кропивницький – 2022 року

MyEdit.dpr - головний файл проекту

```
program MyEdit;

uses
  Forms,
  MainUnit in 'MainUnit.pas' {MainForm},
  ChildUnit in 'ChildUnit.pas' {ChildForm},
  Unit1 in 'Unit1.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit4 in 'Unit4.pas' {Form4},
  Unit5 in 'Unit5.pas' {Form5},

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TMainForm, MainForm);
  Application.Run;
end.
```

Кафедра _ КБПЗ _ 2022 рік

MainUnit.pas - основна програма

```
unit MainUnit;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, StdActns, ActnList, DB, Grids, DBGrids, ExtCtrls, DBCtrls, DBTables,
  ComCtrls, ToolWin, StdCtrls, Buttons, Mask, ImgList;

type
  TMainForm = class(TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N3: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N14: TMenuItem;
    N15: TMenuItem;
    WindowMenu: TMenuItem;
    OpenDialog1: TOpenDialog;
    ActionList1: TActionList;
    Arrang1: TWindowArrange;
    Cascade1: TWindowCascade;
    Close1: TWindowClose;
    MinimizeAll1: TWindowMinimizeAll;
    TileHorizontally1: TWindowTileHorizontal;
    TileVertically1: TWindowTileVertical;
    N13: TMenuItem;
    N16: TMenuItem;
    Copy1: TEditCopy;
    Cut1: TEditCut;
    Paste1: TEditPaste;
    StatusBar1: TStatusBar;
    DataSource1: TDataSource;
    ToolBar1: TToolBar;
    ComboBox1: TComboBox;
    Timer1: TTimer;
    warehouse: TTable;
    DBGrid1: TDBGrid;
    N2: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N9: TMenuItem;
    N11: TMenuItem;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Memo1: TMemo;
    N10: TMenuItem;
    N12: TMenuItem;
    N17: TMenuItem;
    N18: TMenuItem;
    N19: TMenuItem;
    N20: TMenuItem;
    N21: TMenuItem;
  end;
end;
```

```

N22: TMenuItem;
Label1: TLabel;
Splitter1: TSplitter;
N23: TMenuItem;
N24: TMenuItem;
Query1: TQuery;
N25: TMenuItem;
ImageList1: TImageList;

```

```

procedure ComboBox1Change(Sender: TObject);
procedure N8Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure N3Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);

procedure SpeedButton4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);

procedure SpeedButton7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton8MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure ToolBar1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBGrid1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure N13Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure N18Click(Sender: TObject);
procedure N19Click(Sender: TObject);
procedure N20Click(Sender: TObject);
procedure N21Click(Sender: TObject);
procedure N22Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure N24Click(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure N25Click(Sender: TObject);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  MainForm: TMainForm;
  editmark: boolean;
  stDay : array[1..7] of string[11] = ('неділя','понеділок','вівторок',
                                       'серeda','четвер','п'ятниця','субота');

implementation

  uses Unit1, Unit2, Unit3, Unit4, Unit5;

{$R *.DFM}

procedure TMainForm.ComboBox1Change(Sender: TObject);
//процедура формування доступу до баз даних по автотехніці
begin
warehouse.Active := false;
if combobox1.ItemIndex = 0 then begin
warehouse.TableName:='../data/Audi.db';
end;
if combobox1.ItemIndex = 1 then begin
warehouse.TableName:= '../data/Volkswagen.db';
end;
if combobox1.ItemIndex = 2 then begin
warehouse.TableName := '../data/Skoda.db';
end;
warehouse.ReadOnly:=false;
//dbgrid1.ReadOnly:=true;
warehouse.Active := true;
end;

procedure TMainForm.N8Click(Sender: TObject);
begin
application.Terminate;
end;

//створення форми введення даних по дням тижня
procedure TMainForm.FormCreate(Sender: TObject);
var Present: TDateTime;
    year, month, day : word;
begin
present:=now;
decodedate(present,year,month,day);
statusbar1.Panels[2].Text:=inttostr(day)+'.'+inttostr(month)+'.'+inttostr(year)+
', '+stDay[DayOfWeek(Present)];
end;

procedure TMainForm.N3Click(Sender: TObject);
begin
warehouse.Active := false;
if opendialog1.Execute then begin
warehouse.TableName:=opendialog1.FileName;
warehouse.Active:=true;
end;
end;

procedure TMainForm.Timer1Timer(Sender: TObject);
var date: Tdatetime;

```

```
begin
statusbar1.Panels[1].Text:=timetostr(time);
end;

procedure TMainForm.N6Click(Sender: TObject);
begin
combobox1.ItemIndex := -1;
warehouse.Active :=false;
warehouse.TableName := '';
end;

procedure TMainForm.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
Statusbar1.Panels[0].Text:='Додати';
end;

procedure TMainForm.SpeedButton2MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
Statusbar1.Panels[0].Text:='Редагувати';
end;

// процедура формування швидкого пошуку по БД
procedure TMainForm.SpeedButton4MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
Statusbar1.Panels[0].Text:='Швидкий пошук';
end;

//процедура виводу панелі для закупівні недостаючих запасних
// частин та паливо-мастильних матеріалів
procedure TMainForm.SpeedButton5MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
Statusbar1.Panels[0].Text:='Купівля';
end;

//формування вінка виаоду даних про автора програми

procedure TMainForm.SpeedButton7MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
Statusbar1.Panels[0].Text:='Про програму...';
end;

//процедура виходу з програми
procedure TMainForm.SpeedButton8MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
Statusbar1.Panels[0].Text:='Вихід';
end;

//процедура видалення даних з БД
procedure TMainForm.SpeedButton3MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
Statusbar1.Panels[0].Text:='Видалити';
end;

// Процедура формування виклику завантаження/додавання на склад механізованої
колоні
procedure TMainForm.SpeedButton6MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
```

```

StatusBar1.Panels[0].Text:='Вигрузити/Додати на склад';
end;

procedure TMainForm.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
StatusBar1.Panels[0].Text:='';
end;

procedure TMainForm.ToolBar1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
StatusBar1.Panels[0].Text:='';
end;

procedure TMainForm.DBGrid1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
StatusBar1.Panels[0].Text:='';
end;

//процедура виклику редагування даних у БД
procedure TMainForm.SpeedButton1Click(Sender: TObject);
begin
if editmark=false then begin
label1.Caption:='Редагування';
dbgrid1.Color:=clteal;
//dbgrid1.ReadOnly:=false;
warehouse.insert;
editmark:=true end
  else begin
    // dbgrid1.ReadOnly:=true;
    editmark:=false;
    label1.Caption:='';
    dbgrid1.Color:=clwindow; end;
end;

//Процедура виклику видалення даних з БД
procedure TMainForm.SpeedButton3Click(Sender: TObject);
var k:integer;
begin
k:=messagebox(0,'Видалити запис?', 'Видалення', 33);
if k=1 then begin
    memo1.Lines.Add(Тimetostr(time)+' Запис був видалений з таблиці
[ '+inttostr(warehouse.fieldbyname('Код').Value)+'
'+warehouse.fieldbyname('Назва').Value+' ]');
    warehouse.Delete;
    end;
end;

procedure TMainForm.SpeedButton5Click(Sender: TObject);
begin
Application.CreateForm(TForm1, Form1);
Form1.Show;
end;

procedure TMainForm.SpeedButton8Click(Sender: TObject);
begin
application.Terminate;
end;

procedure TMainForm.SpeedButton2Click(Sender: TObject);
begin
if editmark=false then begin

```

```
label1.Caption:='Редагування';
dbgrid1.Color:=clteal;
//dbgrid1.ReadOnly:=false;
warehouse.Edit;
editmark:=true end
    else begin
        // dbgrid1.ReadOnly:=true;
        editmark:=false;
        label1.Caption:='';
        memo1.Lines.Add(Timetostr(time)+' Таблиця була відредагована');
        dbgrid1.Color:=clwindow; end;
end;

procedure TMainForm.N13Click(Sender: TObject);
begin
speedbutton5.Click;
end;

procedure TMainForm.N4Click(Sender: TObject);
begin
speedbutton1.Click;
end;

procedure TMainForm.N5Click(Sender: TObject);
begin
speedbutton2.Click;
end;

procedure TMainForm.N9Click(Sender: TObject);
begin
speedbutton3.Click;
end;

procedure TMainForm.SpeedButton4Click(Sender: TObject);
begin
Application.CreateForm(TForm2, Form2);

Form2.show;
end;

procedure TMainForm.N18Click(Sender: TObject);
begin
warehouse.IndexFieldNames:='Код';
end;

procedure TMainForm.N19Click(Sender: TObject);
begin
warehouse.IndexFieldNames:='Найзва';
end;

procedure TMainForm.N20Click(Sender: TObject);
begin
warehouse.IndexFieldNames:='Дата випуску';
end;

procedure TMainForm.N21Click(Sender: TObject);
begin
warehouse.IndexFieldNames:='Дата надходження';
end;

procedure TMainForm.N22Click(Sender: TObject);
begin
warehouse.IndexFieldNames:='Група';
end;

procedure TMainForm.SpeedButton6Click(Sender: TObject);
begin
Application.CreateForm(TForm3, Form3);
```

```

form3.Show;
form3.dbgrid1.DataSource:=datasource1;
end;

procedure TMainForm.N24Click(Sender: TObject);
var i,R,P,SR,SP: integer;
begin

    Application.CreateForm(TForm4, Form4);

SR:=0; SP:=0;
form4.Show;
R:=0;
P:=0;

//Закупівля запасних частин

warehouse.Active:=false;
warehouse.TableName='../data/Audi.db';
warehouse.Active:=true;
warehouse.First;
for i:=1 to warehouse.RecordCount do begin
R:=R+mainform.warehouse.fieldbyname('Кількість').Value;
P:=P+mainform.warehouse.fieldbyname('Кількість').Value*mainform.warehouse.fieldb
yname('Ціна закупки').Value;
warehouse.Next;
end;
form4.label2.Caption:=inttostr(R);
form4.Label4.Caption:=inttostr(P); SR:=R; Sp:=P;

                {second table}
R:=0;
P:=0;
warehouse.Active:=false;
warehouse.TableName='../data/Volkswagen.db';
warehouse.Active:=true;
warehouse.First;
for i:=1 to warehouse.RecordCount do begin
R:=R+mainform.warehouse.fieldbyname('Кількість').Value;
P:=P+mainform.warehouse.fieldbyname('Кількість').Value*mainform.warehouse.fieldb
yname('Ціна закупки').Value;
warehouse.Next;
end;
form4.label6.Caption:=inttostr(R);
form4.Label7.Caption:=inttostr(P);      SR:=SR+R; SP:=SP+P;

                {third table}
R:=0;
P:=0;
warehouse.Active:=false;
warehouse.TableName='../data/Skoda.db';
warehouse.Active:=true;
warehouse.First;
for i:=1 to warehouse.RecordCount do begin
R:=R+mainform.warehouse.fieldbyname('Кількість').Value;
P:=P+mainform.warehouse.fieldbyname('Кількість').Value*mainform.warehouse.fieldb
yname('Ціна закупки').Value;
warehouse.Next;
end;
form4.label8.Caption:=inttostr(R);
form4.Label9.Caption:=inttostr(P); SR:=SR+R; SP:=SP+P;

form4.Label17.Caption:=inttostr(SR);
form4.Label19.caption:=inttostr(SP);

warehouse.Active:=false;
warehouse.TableName='../data/'+combobox1.text+'.db';
warehouse.Active:=true;

```

```
end;

procedure TMainForm.SpeedButton7Click(Sender: TObject);
begin
  Application.CreateForm(TForm5, Form5);
  Form5.Show;
end;

procedure TMainForm.N25Click(Sender: TObject);
begin
  warehouse.edit; warehouse.Post;
  memo1.Lines.Add(Timetostr(time)+' Всі зміни збережені');
end;

end.
```

Кафедра _ КБПЗ _ 2022 рік

Unit1.pas - купівля товару (запчасти або ПММ)

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Mask, DBCtrls, DB, DBTables, Spin, Buttons;

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    GroupBox2: TGroupBox;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Edit2: TEdit;
    ComboBox1: TComboBox;
    MaskEdit1: TMaskEdit;
    DateTimePicker1: TDateTimePicker;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
implementation
  uses Mainunit;
  {$R *.dfm}

  procedure TForm1.Button2Click(Sender: TObject);
  begin
    mainform.warehouse.Delete;
    Form1.Close;
  end;

  procedure TForm1.Button1Click(Sender: TObject);
  begin
    mainform.warehouse.InsertRecord([edit2.text,edit1.text,maskedit1.text,spinedit2.
    text,spinedit1.text,',',comboBox1.text,datetimepicker1.date]);
    mainform.Memo1.Lines.Add('Здійснена купівля'+timetostr(time));
    form1.Close;
  end;

  procedure TForm1.BitBtn1Click(Sender: TObject);
  begin
    mainform.warehouse.InsertRecord([edit2.text,edit1.text,maskedit1.text,spinedit2.
    text,spinedit1.text,',',comboBox1.text,datetimepicker1.date]);
    mainform.Memo1.Lines.Add(timetostr(time)+' Здійснена купівля '+edit1.Text+' [
    '+spinedit2.Text+' по ціні '+spinedit1.text+'р. ]');
    edit1.Clear;
  end;

```

```
edit2.Clear;  
combobox1.Clear;  
maskedit1.Clear;  
spinedit1.Clear;  
spinedit2.Clear;  
form1.Close;  
end;  
  
procedure TForm1.BitBtn2Click(Sender: TObject);  
begin  
edit1.Clear;  
edit2.Clear;  
combobox1.Clear;  
maskedit1.Clear;  
spinedit1.Clear;  
spinedit2.Clear;  
Form1.Close;  
end;  
  
end.
```

Кафедра _ КБПЗ _ 2022 рік

Unit2.pas - пошук запису у базі даних

```

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, ComCtrls;

type
  TForm2 = class(TForm)
    Edit1: TEdit;
    ComboBox1: TComboBox;
    GroupBox1: TGroupBox;
    ComboBox2: TComboBox;
    Button1: TButton;
    MaskEdit1: TMaskEdit;
    procedure DBEdit1Change(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);

    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;
  p,pd: integer;
implementation
  uses MainUnit;
  {$R *.dfm}

  procedure TForm2.DBEdit1Change(Sender: TObject);
  begin
    mainform.warehouse.FindField('Код');
  end;

  procedure TForm2.Edit1Change(Sender: TObject);
  var lan,i,mark: integer;
      strmark: string;
  begin
    lan:=mainform.warehouse.RecordCount;
    mainform.warehouse.First;

    if edit1.text='' then begin mark:=0; end
    else begin
      if p=0 then begin mark:=strtoint(edit1.text);
        for i:=0 to lan do begin
          if mainform.warehouse.Fields[p].AsInteger=mark then
            else mainform.warehouse.Next;
          end; end;
        if ((p=1) or (p=6)) then begin strmark:=edit1.Text;
          for i:=0 to lan do begin
            mainform.warehouse.setkey;
          end;
        mainform.warehouse.IndexFieldNames:=mainform.warehouse.Fields[p].FieldName;
        mainform.warehouse.FindNearest([edit1.text]);
        end; end;

```

```
end;
```

```
end;
```

```
procedure TForm2.ComboBox1Change(Sender: TObject);
```

```
begin
```

```
case combobox1.ItemIndex of
```

```
0: p:=0;
```

```
1: p:=1;
```

```
2: p:=6; end;
```

```
end;
```

```
procedure TForm2.ComboBox2Change(Sender: TObject);
```

```
begin
```

```
case combobox2.ItemIndex of
```

```
0: pd:=2;
```

```
1: pd:=7; end;
```

```
end;
```

```
procedure TForm2.Button1Click(Sender: TObject);
```

```
begin
```

```
if pd=0 then pd:=2;
```

```
    if maskedit1.Text='' then else
```

```
        begin mainform.warehouse.SetKey;
```

```
mainform.warehouse.IndexFieldNames:=mainform.warehouse.Fields[pd].FieldName;
```

```
mainform.warehouse.FindNearest([maskedit1.text]);
```

```
end;
```

```
end;
```

```
procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
```

```
begin
```

```
edit1.Text:=''; maskedit1.Text:='';
```

```
end;
```

```
end.
```

Кафедра — КОГЗ 2022 рік

Unit3.pas - вивантажити/додати на склад

```

unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, DBTables, Buttons, Mask, DBCtrls,
  ExtCtrls;

type
  TForm3 = class(TForm)
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    Edit2: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    DBEdit1: TDBEdit;
    GroupBox1: TGroupBox;
    ComboBox1: TComboBox;
    Edit1: TEdit;
    Button1: TButton;
    GroupBox2: TGroupBox;
    Label2: TLabel;
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Label1: TLabel;
    Memo1: TMemo;
    procedure Button2Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;
  p, pd: integer;
implementation
  uses mainunit;
  {$R *.dfm}

  procedure TForm3.Button2Click(Sender: TObject);
  begin
    form3.Close;
  end;

  procedure TForm3.Edit1Change(Sender: TObject);
  var lan, i, mark: integer;
      strmark: string;
  begin
    lan:=mainform.warehouse.RecordCount;
    mainform.warehouse.First;

```

```

if edit1.text='' then begin mark:=0; end
else begin
    if p=0 then begin mark:=strtoint(edit1.text);
    for i:=0 to lan do begin
        if mainform.warehouse.Fields[p].AsInteger=mark then
            else mainform.warehouse.Next;
            end;    end;
    if ((p=1) or (p=6)) then begin strmark:=edit1.Text;
    for i:=0 to lan do begin
        mainform.warehouse.setkey;

mainform.warehouse.IndexFieldNames:=mainform.warehouse.Fields[p].FieldName;
        mainform.warehouse.FindNearest([edit1.text]);
            end;    end;

end;

end;

procedure TForm3.ComboBox1Change(Sender: TObject);
begin
case combobox1.ItemIndex of
0: p:=0;
1: p:=1;
2: p:=6; end;
end;

procedure TForm3.Button3Click(Sender: TObject);
begin
mainform.Query1.Active:=false;
mainform.Query1.Close;
mainform.Query1.SQL.Clear;
mainform.Query1.SQL.Add('select '+Код, Назва, "Дата випуску", "Кількість",
"Ціна закупки", "Група" +'from '+mainform.ComboBox1.Text+' where
'+combobox1.Text+'='+edit1.text);
mainform.Query1.Active:=true;
dbedit1.DataField:='Кол-во';
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
var N,RN: integer;
begin
if radiobutton1.Checked=true then begin
N:=strtoint(edit2.Text);
RN:=strtoint(dbedit1.Text);
RN:=RN-N;
mem1.Lines.Add('Вигружено '+edit2.Text+' одиниць
'+mainform.warehouse.FieldValues['Назва']);
mainform.mem1.Lines.Add(Timetostr(time)+' Вигружено '+edit2.Text+' одиниць
'+mainform.warehouse.FieldValues['Назва']);
end;

if radiobutton2.Checked=true then begin
N:=strtoint(edit2.Text);
RN:=strtoint(dbedit1.Text);
RN:=RN+N;
mem1.Lines.Add('Додано '+edit2.Text+' одиниць
'+mainform.warehouse.FieldValues['Назва']);
mainform.mem1.Lines.Add(Timetostr(time)+' Додано '+edit2.Text+' одиниць
'+mainform.warehouse.FieldValues['Назва']);
end;

mainform.Query1.Close;
datasource1.Enabled:=false;
mainform.warehouse.edit;
mainform.warehouse.FieldByName('Кількість').AsInteger:=RN;
mainform.warehouse.Post;

```

```
mainform.Query1.Open;
datasource1.Enabled:=true;

edit2.Text:='';
end;

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
mainform.Query1.Active:=false;
edit1.Clear;
edit2.Clear;
dbedit1.DataField:='';
form3.close;
end;

procedure TForm3.Edit2Change(Sender: TObject);
var N,RN: integer;
begin

end;

procedure TForm3.Button1Click(Sender: TObject);
begin
dbgrid1.DataSource:=datasource1;
mainform.Query1.Active:=false;
mainform.Query1.Close;
mainform.Query1.SQL.Clear;
mainform.Query1.SQL.Add('select '+'Код, Назва, Група '+'from
"./data/'+mainform.ComboBox1.Text+'" where
'+combobox1.Text+'="'+edit1.text+'"');
mainform.Query1.Active:=true;
dbedit1.DataField:='Кількість';
end;

procedure TForm3.FormCreate(Sender: TObject);
begin
dbgrid1.DataSource:=mainform.DataSource1;
end;

procedure TForm3.ComboBox2Change(Sender: TObject);
begin
case combobox1.ItemIndex of
0: p:=0;
1: p:=1;
2: p:=6; end;
end;

procedure TForm3.FormDestroy(Sender: TObject);
begin
mainform.Query1.Active:=false;
edit1.Text:='';
edit2.Text:='';
dbedit1.DataField:='';
end;

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
mainform.Query1.Active:=false;
edit1.Text:='';
edit2.Text:='';
dbedit1.DataField:='';
end;

end.
```

Кафедра _ КБПЗ _ 2022рік

Unit4.pas - аудит роботи з базою даних

```

unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, Mask, QuickRpt, QRCtrls, ExtCtrls;

type
  TForm4 = class(TForm)
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Label18: TLabel;
    Label19: TLabel;
    Label21: TLabel;
    Button1: TButton;
    ComboBox1: TComboBox;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation
  uses Mainunit;
  {$R *.dfm}

  procedure TForm4.Button1Click(Sender: TObject);
  var i: integer;
  begin

    mainform.Query1.close;
    mainform.Query1.SQL.Clear;
    mainform.Query1.SQL.Add('Select * from "../data/'+mainform.ComboBox1.Text+'"
    where Група="'+combobox1.text+'");
    mainform.Query1.Open;
    mainform.Query1.Active:=true;
    datasources1.DataSet:=mainform.Query1;

```

```
end;
```

```
procedure TForm4.Button2Click(Sender: TObject);  
begin  
form4.close;  
end;
```

```
procedure TForm4.Button3Click(Sender: TObject);  
var i:integer;  
begin  
mainform.warehouse.First;  
for i:=1 to mainform.warehouse.recordcount do begin  
if mainform.warehouse.FieldByName('Дата випуску').value='' then  
datasource1.DataSet:=mainform.warehouse.Fields.DataSet;  
mainform.warehouse.Next;
```

```
end;
```

```
end;
```

```
end.
```

Кафедра _ КБПЗ _ 2022 рік

Unit5.pas - довідка про програму

```
unit Unit5;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm5 = class(TForm)
    Image1: TImage;
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation
  uses Mainunit;
  {$R *.dfm}

  procedure TForm5.Button1Click(Sender: TObject);
  begin
    Form5.Close;
  end;

  procedure TForm5.FormCreate(Sender: TObject);
  begin
    Memo1.Clear;
    Memo1.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
    Memo1.Lines.Add('');
    Memo1.Lines.Add('на тему:');
    Memo1.Lines.Add('');
    Memo1.Lines.Add('Дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства');
    Memo1.Lines.Add('');
    Memo1.Lines.Add('');
    Memo1.Lines.Add('Керівник: Смірнова Т.В');
    Memo1.Lines.Add('');
    Memo1.Lines.Add('Розробив: студент Іщак Ярослав Сергійович');
    Memo1.Lines.Add('                гр. КІ-21М-1,4');
    Memo1.Lines.Add('');
    Memo1.Lines.Add('');
    Memo1.Lines.Add('м. Кропивницький 2022');
  end;
end.
```