

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 мікроклімату**  
**складського комплексу”**

Виконав здобувач вищої освіти  
II курсу, групи КІ-21М-1,4  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Авраменко К.Д.  
« \_\_\_\_ » \_\_\_\_\_ 2022 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Петренюк В.І.  
« \_\_\_\_ » \_\_\_\_\_ 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

Авраменку Кирилу Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи мікроклімату складського комплексу

2. Керівник роботи Петренюк Володимир Ілліч, канд. фіз.-мат. наук, доцент  
(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 10.12.2022 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи мікроклімату складського комплексу

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	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

Петренюк В.І.  
(прізвище та ініціали)Завдання прийнято до виконання  
« 6 » вересня 2022 р.

Підпис здобувача

Авраменко К.Д.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Авраменко К.Д. Дослідження та програмна реалізація системи мікроклімату складського комплексу. 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 мікроклімату складського комплексу.

Метою розробки є дослідження та програмна реалізація системи мікроклімату складського комплексу.

Об'єктом дослідження є процес мікроклімату складського комплексу.

Предметом дослідження є методи мікроклімату складського комплексу.

Методи дослідження базуються на методах інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи мікроклімату складського комплексу.

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 Sydney.

**Ключові слова:** комп'ютерна інженерія, мікроклімат, складський комплекс

## ABSTRACT

**Avramenko K.D. Research and software implementation of the microclimate system of the warehouse complex. 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 microclimate system of the warehouse complex.

The purpose of the development is research and software implementation of the microclimate system of the warehouse complex.

The object of research is the microclimate process of the warehouse complex.

The subject of the study is the microclimate methods of the warehouse complex.

Research methods are based on Internet of Things methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the microclimate system of the warehouse complex.

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 Sydney environment.

**Keywords:** computer engineering, microclimate, warehouse complex

## 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.....	7
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 .....	22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	24
3.1 Опис функціонування системи .....	24
3.2 Розробка структурної схеми.....	35
3.3 Розробка функціональної схеми .....	36
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	42
4.2 Захист розробленого програмного забезпечення.....	58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	61
6 НАУКОВА НОВИЗНА .....	64

						<b>ВКРМ-123.22.0001.00.00.ПЗ</b>		
<b>Вим</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>				
<i>Розроб.</i>	<i>Авраменко К.Д.</i>				<i>Дослідження та програмна реалізація системи мікроклімату складського комплексу</i>	<b>Літ.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<i>Перев.</i>	<i>Петренко В.І.</i>					<b>М</b>	1	104
<i>Н.контр.</i>	<i>Гермак В.С.</i>					<i>ЦНТУ КІ-21М-1,4</i>		
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	65
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	65
7.2 Розрахунок трудомісткості розробки програмної продукції.....	67
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	69
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	74
7.5 Визначення собівартості розробки та ціни програмної продукції.....	78
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	81
7.7 Визначення експлуатаційних витрат.....	81
7.8 Визначення економічної ефективності програмної продукції.....	83
7.9 Висновок.....	85
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	86
8.1 Вступ.....	86
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	87
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК .....	88
8.4 Розробка заходів з умов поліпшення охорони праці.....	91
8.5 Протипожежний захист .....	92
8.6 Розрахункова частина .....	93
8.7 Висновки до розділу.....	94
9 ОСНОВНІ ВИСНОВКИ.....	95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	97

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БП	– блок перетворення
БРІУ	– блок вимірювання, керування та індикації
ЗВЧ	– зверх високі частоти
ІВТМ	– вимірювач вологості та температури.
КПК	– карманий персональний комп'ютер
МК	– мікроконтролер
ПЗ	– програмне забезпечення
ПК	– персональний комп'ютер
ПІ	– перетворювач інтерфейсу
ЯМР	– ядерний магнітний резонанс
GPRS	– стандарт бездротової передачі даних
GSM	– стандарт мобільного зв'язку
IP	– рівень мережної моделі передачі даних
PIC	– мікроконтролер
RAD	– набір візуальних інструментів для швидкісної розробки додатків
RS-232	– протокол обміну даними
RS-485	– протокол обміну даними
USB	– universal serial bus
VCL	– бібліотека візуальних компонентів

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** В Україні, зерно традиційно є стратегічною продукцією. За даними Міністерства агропромислового комплексу, у виробників зерна залишається більше 50% отриманого врожаю. Аналогічні тенденції прогнозуються й у майбутньому. Світова практика так само показує, що в аграрно розвинених державах до 80% урожаю зернових культур зберігається безпосередньо в його виробника. Однак, оснащення наших господарств, незалежно від форм їхньої власності, технологічним устаткуванням для забезпечення якісного післязбирального доведення до кондиції зерна, є вкрай незадовільним. До 70% господарств не мають зерноочисного й сушильного встаткування, що приводить до значних втрат урожаю. Тому в основу створення техніки для післязбиральної обробки зерна необхідно покласти такі напрямки [1]:

- обробка всього зібраного врожаю продовольчого й фуражного зерна на місці виробництва;
- будівництво складських приміщень для зберігання зерна й насіння із повністю укомплектованими засобами механізації вантажно-розвантажувальних операцій.

Таким чином, одним зі шляхів зменшення втрат і підвищення якості зерна, є безумовне забезпечення кожного господарства власним сучасним складським приміщенням для зберігання зерна, у якому існує автоматизована система управління вологістю та температурою. Підсистемою такої системи є мережа датчиків для визначення вологості, з можливістю обробки отриманих значень вологості та реагуванням на її зміну [1-5].

Це дасть можливість зерновиробникам:

- створити власну інфраструктуру якісного збереження врожаю й заощаджувати при цьому до 40-50% енергії на досушуванні й охолодженні зерна;
- зберігання свого зерна у своєму сховищі дозволить зерновласникам

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

незалежно й упевнено розпоряджатися вирощеною продукцією;

– зберігати врожай на власній фермі й не витратити час і засоби на транспортування й простій у чергах на зернозбиральних пунктах;

– не продавати зерно в період збирання, а притримати до часу, коли ціни на нього будуть найбільш сприятливими.

Слід зазначити, що більше 80% енергоносіїв Україна імпортує [6]. Тому усе більше актуальною проблемою є ефективне використання в технологічних процесах альтернативних джерел енергії. Так як одним з найбільш енергоємних процесів у зерновиробництві є досушування зерна, то розробка й впровадження енергозберігаючих технологій доведення його до кондиції, є найважливішим і сучасним завданням.

В зв'язку з тим, що для досушки зерна необхідно періодично визначати значення вологості то впровадження системи виміру вологості є актуальною та вкрай необхідною мірою підвищення строків зберігання та якості зерна. Але для того, що ця система працювала необхідно розробити відповідне програмне забезпечення.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи мікроклімату складського комплексу.

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

- Огляд існуючих систем мікроклімату складського комплексу.
- Дослідження системи мікроклімату складського комплексу.
- Програмна реалізація системи мікроклімату складського комплексу.

*Об'єктом дослідження є процес мікроклімату складського комплексу.*

*Предметом дослідження є методи мікроклімату складського комплексу.*

*Методи дослідження базуються на методах інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.*

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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

- Удосконалено метод мікроклімату складського комплексу.
- Розроблено вітчизняний продукт мікроклімату складського комплексу,

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

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі мікроклімату складського комплексу.

**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 мікроклімату складського комплексу, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Перш ніж перейти до побудови підсистеми виміру вологості й реалізації програмного забезпечення, що може знімати показання з датчиків вологості й подавати сигнали в систему керування вологістю й температурою в складському приміщенні, обґрунтуємо необхідність побудови такої підсистеми.

Втрати зерна при зберіганні дотепер у світі ще досить значні. По даним FAO (Food and Agricultural Organization при ООН) – щорічні втрати зернових становлять більше 10% від загального виробництва з максимумом для деяких менш розвинених країн в 30 – 50%. У більшості цих втрат залежать від вологості й підвищеної температури зерна. На його досушування й охолодження витратиться більше 20% всіх енергоносіїв, необхідних для вирощування й збирання хліба. У країнах, які відчувають дефіцит в енергоносіях, багато зерна не доводять до необхідної кондиції, внаслідок чого воно псується й знецінюється. На сучасному етапі розвитку науки активне вентилявання для досушування й штучного охолодження зерна розглядаються як складні технологічні процеси, при яких властивості зерна, які визначають його якість, повинні бути не тільки збереженими, але й поліпшеними. У такий спосіб забезпечення енергоекономного збереження зерна є одним з найважливіших сучасних завдань у багатьох країнах світу.

## 1.2 Область застосування

Для того, щоб вирішувалося дане завдання, необхідно, щоб ефективно працювала підсистема виміру вологості в складському приміщенні для зберігання зерна. Існує ще ряд причин, по яких необхідно реалізувати вказану підсистему.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Причини з погляду зберігання й збереження – захисту продукції й сировини:

- запобігання розвитку біодеструкторів: цвілі, грибків, забезпечується виконання санітарно – гігієнічних вимог;
- запобігання корозії, що проявляється в більше тривалому ресурсі матеріалу, більше низьких витратах на техобслуговування й т.д.;
- строк зберігання продукції збільшується, тому що стримується розвиток різних процесів, що викликають погіршення її якості;
- устаткування й матеріали, тимчасово виведені з експлуатації або при останові, зберігаються в ідеальних умовах, що перешкоджають зниженню їхньої якості й дозволяють винятково швидко повернутися в робочий режим.

Виробничі причини:

- підтримка заданих значень вологості, відсутність конденсації вологи;
- усуваються втрати й коливання якості продукції внаслідок зміни вологості або температури;
- зменшуються витрати на технічне обслуговування й тривалість простоїв у результаті запобігання утворення речовин, що прилипають до встаткування й засмічують його;
- ефективність і продуктивність охолодних систем збільшується, одночасно знижуються експлуатаційні витрати й потужність необхідного охолодного встаткування;
- зменшуються витрати на техобслуговування електростанцій; збільшується ресурс станцій;
- електричний опір ізоляції електроустаткування підтримується на високому рівні, за допомогою чого вдається уникнути виходу встаткування з ладу при пуску;
- завдяки підвищеній надійності зменшується час простоїв при виробництві.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Економічні причини:

- сукупна вартість зберігання продукції, сировини набагато дешевша, ніж при використанні обігріву;
- поліпшення функціональних характеристик;
- нижче вартість обслуговування;
- повернення інвестицій, у багатьох випадках, відбувається за період менш одного року.

У такий спосіб рішення завдання створення підсистеми виміру вологості в складському приміщенні є актуальним.

Дана підсистема природно буде будуватися з використанням мікроконтролерної техніки.

Будь-який мікроконтролер (МК) призначений для виконання корисних функцій по керуванню або контролю фізичними параметрами конкретних об'єктів керування. Функції керування або контролю зводяться до обробки й наступного використання цифрової двійкової інформації, що надходить від об'єктів керування по лініях зв'язку від різних пристроїв сполучення МК із об'єктом. Як такі пристрої можуть бути датчики різних аналогових фізичних параметрів і пов'язані з ними перетворювачі, що нормують, електричних сигналів, аналого-цифрові перетворювачі, датчики цифрової інформації й ін. У нашому випадку в ролі таких пристроїв будуть виступати датчики вологості. З боку виводу інформації МК взаємодіє із цифровими індикаторами, виконавчими механізмами, дисплеями й іншими засобами запам'ятовування, зберігання й використання результатів обробки інформації.

З'єднання всього різноманіття зовнішніх пристроїв із шинами мікроконтролера здійснюється за допомогою інтерфейсів, які варто розуміти як уніфікований засіб об'єднання різних пристроїв у єдину систему. Будь-який інтерфейс повинен забезпечити рішення наступних двох завдань.

По-перше, інтерфейс у своїй апаратній частині повинен забезпечити електричне з'єднання різних зовнішніх пристроїв з різними електричними й конструктивними параметрами, з єдиною системою шин конкретного мікроконтролера. При цьому повинні бути враховані такі параметри, як кількість

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

ліній зв'язку, рівні й потужності електричних сигналів, довжина й перешкодозахищеність ліній зв'язку.

По-друге, інтерфейс повинен забезпечити гнучке програмне керування всіма підключеними зовнішніми пристроями. У цій частині інтерфейс повинен забезпечити не тільки працездатність зовнішніх пристроїв, але й узгодження по швидкодії різних зовнішніх пристроїв і центрального мікропроцесора. Таким чином, під інтерфейсом варто розуміти уніфікований програмно-апаратний пристрій, призначений для організації обміну інформацією між мікропроцесором і зовнішніми пристроями, об'єднаними в єдину систему.

Таким чином, виникає необхідність програмного забезпечення системи призначеної для виміру вологості в складському приміщенні. При цьому необхідно враховувати що вона, як правило, є підсистемою системи керування температурою й вологістю. Тобто в даній підсистемі необхідно передбачити можливість передачі параметрів, у якомусь стандартизованому виді, вищевказаній системі.

Отже, необхідно реалізувати блок програм, що виконують наступні функції:

- зняття сигналів з датчиків вологості;
- перетворення отриманих сигналів у цифровий вид;
- проведення оцінки отриманих значень, з погляду, їхнього знаходження в рамках заданих параметрів;
- у випадку виходу за рамки цих параметрів передати системі керування температурою й вологістю сигнал про те, що параметри не відповідають, і вказати наскільки;
- стежити за вологістю в складському приміщенні подаючи сигнали системі, доти, поки вологість не ввійде в задані рамки.

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

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	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

Методи виміру вологості прийнято ділити на прямі й непрямі. У прямих методах виробляється безпосередній поділ матеріалу на суху речовину й вологу. У непрямих методах вимірюється величина, функціонально пов'язана з вологістю матеріалу. Непрямі методи вимагають попереднього калібрування з метою встановлення залежності між вологістю матеріалу й вимірюваною величиною.

#### Прямі методи

Найпоширенішим методом є метод висушування (термогідралічний), що полягає в повітряно-тепловому сушінні зразка матеріалу до досягнення рівноваги з навколишнім середовищем; ця рівновага умовно вважається рівнозначним повному видаленню вологи. На практиці застосовується висушування до постійної ваги; найчастіше застосовують так звані прискорені методи сушіння. У першому методі сушіння закінчують, якщо два послідовних зважування досліджуваного зрізця дають однакові або досить близькі результати. Так як швидкість сушіння поступово зменшується, передбачається, що при цьому видаляється майже вся волога, що втримується в зразку. Тривалість визначення цим методом складає звичайно від декількох годин до доби й більше. У прискорених методах сушіння ведеться протягом певного, значно більше короткого проміжку часу при підвищеній температурі (наприклад, стандартний метод визначення вологості зерна сушінням розмеленого навішення при +130 градусах протягом 40 минут).

Визначенню вологості твердих матеріалів висушуванням властиві наступні методичні погрішності:

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

а) при висушуванні органічних матеріалів поряд із втратами гігроскопічної вологи відбувається втрата летучих; одночасно при сушінні в повітрі має місце поглинання кисню внаслідок окислювання речовини, а іноді й термічне розкладання проби;

б) припинення сушіння відповідає не повному видаленню вологи, а рівновазі між тиском водяних пар у матеріалі й тиском водяних пар у повітрі;

в) видалення зв'язаної вологи в колоїдних матеріалах неможливо без руйнування колоїдальної частки й не досягається при висушуванні;

г) у деяких речовинах у ході сушіння утвориться водонепроникна кірка, що перешкоджає подальшому видаленню вологи.

Деякі із зазначених погрішностей можна зменшити сушінням у вакуумі при зниженій температурі або в потоці інертного газу. Однак для вакуумного сушіння потрібно більше громіздка й складна апаратура, ніж для повітряно-теплової.

При найпоширенішому методі сушіння (у сушильних шафах) є погрішності, що залежать від застосовуваної апаратури й техніки висушування. Так, наприклад, результати визначення вологості залежать від тривалості сушіння, від температури й атмосферного тиску, при яких протікало сушіння. Температура має особливо велике значення при використанні прискорених методів, коли зниження температури сильно впливає на кількість вилученої вологи. На результати висушування впливають також форма й розміри бюксів і сушильної шафи, розподіл температури в сушильній шафі, швидкість руху повітря в ньому, можливість віднесення пилу або дрібних часток зразка й т.д. Для матеріалів, що піддаються перед визначенням вологості здрібнюванню, велике значення має зменшення вологи в зразку в процесі здрібнювання. Цей збиток особливо великий, якщо при розмелі має місце нагрівання зразка.

Таким чином висушування являє собою чисто емпіричний метод, яким визначається не дійсна величина вологості, а якась умовна величина, більш-менш близька до неї. Визначення вологості, виконані в неоднакових умовах, дають

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

погано порівнянні результати. Значно більш точні результати дає вакуумне сушіння, виконуване звичайно в камері при тиску 25 мм рт. ст. і нижче до постійної ваги.

У дистиляційних методах зразок підігрівається в посудині з певною кількістю рідини, що не змішується з водою. Пари води, що виділяються, разом з парами рідини піддаються відгону й, проходячи через холодильник, конденсуються у вимірвальній посудині, у якій вимірюється обсяг або маса води.

Екстракційні методи засновані на добуванні вологи з досліджуваного зразка водовбирною рідиною й визначенні характеристик рідкого екстракту, що залежать від його вологовмісту – щільності, показника переломлення, температури кипіння або замерзання й т.д.

Основою хімічних методів є обробка зразка реагентом, що вступає в хімічну реакцію тільки з вологою, що втримується в зразку. Кількість води в зразку визначається по кількості рідкого або газоподібного продукту реакції. Так для зерна можна використовувати титрування К. Фішера

### **Непрямі методи**

У цих методах оцінка вологості виробляється по зміні різних його властивостей.

Механічні методи засновані на вимірі механічних характеристик, що змінюються з вологістю, твердих матеріалів (опір роздавлюванню зерна).

Методи, засновані на вимірі неелектричних властивостей матеріалів.

Радіометричні методи базуються в основному на сучасних способах дослідження состава, структури й властивостей речовини, що використовують взаємодію різних видів електромагнітних коливань і ядерних випромінювань із досліджуваною речовиною. У радіометричних (ядерно-фізичних) методах використовуються різні види ядерних випромінювань (гамма-промені, бета-частинки, швидкі нейтрони) і взаємодій (поглинання й розсіювання гама- і бета-випромінювання, пружне розсіювання швидких нейтронів). Так, наприклад, в

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

основі гамма-методів лежить ослаблення інтенсивності гама- випромінювання твердою фазою й вологою зерна в результаті розсіювання й поглинання атомами речовини.

В основі методу ядерного магнітного резонансу (ЯМР) лежить резонансне поглинання радіочастотної енергії ядрами атомів водню (протонами) води при приміщенні вологого матеріалу а постійне магнітне поле. Явище ЯМР пов'язане із квантовими переходами між зєємановськими енергетичними рівнями атомних ядер, що виникають у результаті взаємодії ядерного магнітного моменту із зовнішнім магнітним полем.

Оптичні методи засновані на залежності оптичних властивостей матеріалів від їхнього вологовмісту. Для твердих матеріалів використовуються інфрачервона й видима області спектра.

Теплофізичні методи засновані на залежності від вологості матеріалу його теплофізичних властивостей – коефіцієнта теплопровідності, питомої теплоємності й коефіцієнта температуропровідності.

#### Електричні методи виміру вологості.

Основою електричних методів виміру вологості є залежність від вологості параметрів, що характеризують поведження вологих матеріалів в електричних полях. Кондуктометричні методи засновані на вимірі електричної провідності матеріалу на постійному струмі й змінному струмі промислової або звукової частоти.

Влагоутримуючі матеріали, будучи в сухому виді діелектриками, у результаті зволоження стають напівпровідниками. Питомий опір змінюється, отже, залежно від вологості в надзвичайно широкому діапазоні, що охоплює 12-18 порядків. Неоднорідність діелектрика, наявність у ньому вологи позначаються не тільки на величині питомої провідності, але й на якісних особливостях електропровідності: на її залежності від напруженості електричного поля й температури.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14



методу складається у визначенні постійної поширення в зразку вимірюваного матеріалу шляхом вивчення картини розподілу стоячої хвилі на ділянці лінії, не заповненої діелектриком;

в) метод, заснований на використанні хвиль, відбитих від поверхні вимірюваного зразка. У цьому випадку для визначення діелектричної проникності використовують параметри хвилі, що виникла в результаті взаємодії падаючих і відбитої хвиль;

г) резонансний метод заснований на вимірі параметрів резонатора при внесенні в нього досліджуваного матеріалу. Вимірюючи частоти резонатора, визначають діелектричну проникність, а вимірюючи її добротність, визначають коефіцієнт втрат.

Друга група методів заснована на дослідженні характеристик електромагнітної хвилі, що пройшла через зразок випробуваного матеріалу, шляхом порівняння з характеристиками хвилі, що поширюється по іншому шляху, або хвилі, що поширюється по тому ж шляху, але при відсутності матеріалу. Виміри зводяться до визначення комплексного коефіцієнта передачі ділянки прямої системи, заповненої досліджуваною речовиною (коефіцієнтів поглинання й відбиття, як функції вологовмісту). Такою системою може бути як хвилевід, частково або повністю заповнений матеріалом, так і область вільного простору, у якій поширюються електромагнітні коливання ЗВЧ.

Оптичні методи одержали найбільше поширення, оскільки їхньою характерною рисою є безконтактність вимірів, можливість інтегральної оцінки вологості в великих об'ємах (більша інформаційна ємність методу). Останнє є важливим достоїнством, тому що в реальних виробничих умовах завжди спостерігається нерівномірний розподіл вологи в об'ємі.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

## 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.

– 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.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17





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, SOCI, 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 відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, 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.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20



## 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 стилізуємий FMX компонент TМемо 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.

## 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

забезпечення, яке призначено для системи мікроклімату складського комплексу.

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-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Опис датчиків вологості

Розглянемо більш детально один із затребуваних і актуальних на сьогоднішній день напрямків підрозділу сенсорного контролю – датчики вологості. Основні області застосування: клімат-контроль у промислових, складських і адміністративних приміщеннях, контроль технологічних процесів, екологічних параметрів, метеорологічний контроль і т.д.

#### Загальний опис

В основі побудови датчиків вологості лежить тришарова конденсаторна структура, що складається із платинових електродів і спеціального термореактивного полімерного ізолятора між ними (рисунок 3.1). Вся ця структура розміщена на підложці із кремнію, на якій також виконана інтегральна схема нормалізації й посилення сигналу. Через пори у верхньому електроді й завдяки конструктивній негерметичності корпуса датчика досягається рівноважний вміст води в навколишньому повітрі й міжелектродному просторі. Шар термореактивного полімеру, що покриває пористий платиновий електрод зверху, служить гарним захистом чутливого елемента від забруднення пилом, маслами. У той же час такий захист сприяє збільшенню часу відгуку датчика при зміні вологості.

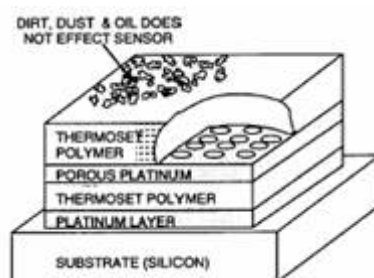


Рисунок 3.1 – Структура чутливого елемента датчика вологості

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Для всіх типів датчиків інтегральна схема формує сигнал, що лінійно змінюється по напрузі, прямо пропорційний напрузі живлення й відносній вологості повітря:

$$V_{out}=V_s (0,0062(\%RH)+0,16) \quad (3.1)$$

де  $V_s$  – напруга живлення, %RH – відносна вологість при 25 °C

Справжнє значення вологості при температурі, що змінюється, можна визначити по формулі:

$$RH=(\%RH)/(1, 0546-0,00216T) \quad (3.2)$$

де T – температура в °C

Для найбільш точного виміру вологості в датчиках серії НІН-3602 у корпус вбудовані датчики температури (рисунок 3.2).

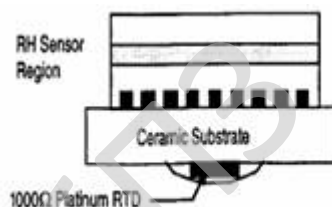


Рисунок 3.2 – Вбудований датчик температури

### Особливості застосування датчиків серії НІН-3602

Для роботи в суворих кліматичних умовах і найбільш точного визначення значення вологості поблизу точки роси фірма *Honeywell* випускає серію НІН-3602. Датчики випускаються в корпусі ТО-5, у який вбудований датчик температури, для моделі НІН-3602-А це NTC термістор номіналом 100 кОм, для НІН-3602-С платиновий датчик температури номіналом 1 кОм. Можливі ситуації, коли при високому значенні відносної вологості (більше 95%) при зниженні температури можлива конденсація пар води, що спричиняє залипання показань на рівні 100% відносної вологості. Для запобігання цього випадку в датчиках серії НІН-3602 передбачений гідрофобний фільтр, виконаний зі спеціально обробленої нержавіючої сталі. Вивести зі сплячого стану також допомагає подача

на датчик трохи підвищеної напруги живлення, однак при цьому можливо деякий зсув калібрування на 2:3 %. Сама по собі конденсація й наступний випар вологи не робить впливу на калібрування датчиків.

#### Основні переваги серії НІН-3602

- наявність вбудованого датчика температури
- корпус із гідрофобним фільтром
- кожний датчик постачений паспортом з індивідуальними даними
- лазерне припасування елементів
- висока стійкість до впливу хімічних речовин (крім пар ацетону або етанолу).

#### Особливості застосування датчиків серії НІН-3610

Датчики серії НІН-3610 є самими масовими по виробництву й застосуванню в схемах до яких не пред'являються особливі вимоги по точності виміру на граничній вологості (більше 95%). Датчики виготовлені в SIP корпусі, причому моделі НІН-3610-001(3) мають формовані виводи із кроком 2,54 мм, у моделей НІН-3610-002(4) крок між виводами 1,27 мм. Також датчики серій НІН-3610-003(4) поставляються з індивідуальними паспортними даними. Вихідний сигнал (напруга) з датчика прямо пропорційний напрузі живлення й відносній вологості повітря.

До переваг даної серії можна віднести:

- високу точність, компактність габаритів;
- простоту у використанні;
- мале енергоспоживання;
- малий час відгуку;
- можливість проведення вимірів у широкому діапазоні тисків.

Недоліки:

- світлочутливі, вимагають екранування від яскравого світла.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

## Вимірювач вологості та температури ІВТМ

Прилад ІВТМ складається із блоку виміру, керування й індикації (БРІУ) і первинних перетворювачів. До приладу може підключатися ІВМ РС-сумісний комп'ютер із принтером. Конструктивно блок керування виконується в пластмасовому або металевому корпусах. Первинний перетворювач також виконується в пластмасовому або металевому корпусі залежно від обраної моделі й складається з вимірювальної камери, у якій розташовуються сенсори й корпуси, у якому розташовується схема попередньої обробки сигналів.

Для виміру вологості використовуються сорбційно-ємнісні мікроелектронні сенсори. Для виміру температури в приладі застосовані сенсори резистивного типу. Сигнал від сенсорів обох типів перетворюється в частотний сигнал за допомогою первинних перетворювачів.

### Опис схеми перетворювача ШВТ-03

Перетворювач виконаний за схемою РС-генератора на таймері типу 555. У якості R елемента в каналі температури використовується терморезистор, а в якості С-елемента каналу вологості використовується ємнісний сенсор вологості.

Підключення датчиків до таймера виробляється за допомогою електронного комутатора. Крім вимірювальних елементів комутатор робить підключення до таймера зразкових РС-елементів (як зразкові елементи застосовуються термостабільні резистори й конденсатори). Застосування подібної вимірювальної схеми дозволяє робити автокомпенсацію перетворювача при зміні температури навколишнього середовища. Керування комутатором, підрахунок частоти з таймера, обчислення температури й вологості здійснюється логічним блоком перетворювача, виконаним на базі ПІС-контролера. По програмі, закладеній в мікроконтролері, здійснюється вимір частоти від сенсорів, зразкових елементів і обчислення значень температури й вологості по індивідуальних калібруваннях, що перебуває в пам'яті обчислювального пристрою перетворювача. Обчислені значення параметрів вологості й температури в послідовному цифровому коді надходять на вихідний пристрій.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Живлення перетворювача здійснюється постійним струмом з напругою від 6 до 9В. Живлення складових частин перетворювача здійснюється за допомогою внутрішнього стабілізатора 5В. Вихідний пристрій необхідно для передачі даних про обмірювані значення в приладі. Передача даних здійснюється по напівдуплексному каналі диференціальним методом. Застосування даного способу передачі дозволяє перетворювачу працювати на довгих лініях при великому рівні електромагнітних перешкод. Відстань, на якій стійко працює перетворювач, становить не менш 300 метрів.

### **Опис БРІУ**

БРІУ виконаний у вигляді мікропроцесорної системи. Він призначений для роботи з перетворювачем вологості й температури типу ІПВТ-03 і його модифікаціями. До приладу може бути підключене до 8 перетворювачів ІПВТ-03. Перетворювачі підключаються до приладу в будь-якому порядку і є взаємозамінними з перетворювачами свого типу. Прилад може бути пов'язаний з комп'ютером по послідовному каналу зв'язку з інтерфейсом RS-232 або RS-485. До приладу може бути підключений блок реле. У приладі передбачений аналоговий вихід.

Робота БРІУ визначається програмою, записаною в постійний запам'ятовувальний пристрій. Внутрішні змінні, а також константи калібрування й інші оперативні параметри (при необхідності) зберігаються в FLASH-пам'яті, що є енергонезалежною й зберігає інформацію при відключеному живленні протягом усього терміну служби приладу. Одним з режимів приладу є режим нагромадження даних. У даному режимі прилад із заданою періодичністю записує дані про обмірювані значення вологості й температури із прив'язкою до реального часу. Інформація зберігається в спеціальній енергонезалежній пам'яті реєстрації. Мінімальний обсяг пам'яті 32 Кб (максимальний 192 Кб), що дозволяє восьмиканальному приладу запам'ятати до 455 (2730) відрахунків інформації. Установка внутрішніх годинників приладу, періодичності запам'ятовування даних і їхній перегляд здійснюються за допомогою комп'ютера.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

БРІУ робить послідовне опитування перетворювачів, і дані про температуру й вологість, обмірювані й розраховані в кожному перетворювачі, відображаються на індикаторі приладу. У тому випадку якщо один або обидва параметри в каналі замасковані, замість значень температури й вологості виводяться символи ---і-. При перевищенні встановленого порога для даного каналу БРІУ видає звуковий сигнал і виводить номер каналу, у якому виявлений вихід за поріг, на індикатор.

Роз'єм **RS232/RS485** призначений для підключення до комп'ютера по інтерфейсу RS232 і об'єднанню приладів у мережу по інтерфейсу RS485.

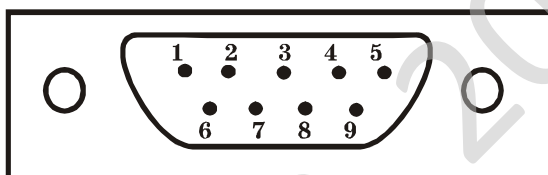


Рисунок 3.3 – Вид вилки з боку монтажу.

- |                           |                                    |
|---------------------------|------------------------------------|
| 1 – сигнал А лінії RS485  | 4, 6, 7, 8 – не використовуються   |
| 2 – сигнал Rx лінії RS232 | 5 – загальний (земля) RS232, RS485 |
| 3 – сигнал Tx лінії RS232 | 9 – сигнал У лінії RS485           |

Роз'єм "ЛІНІЇ КЕРУВАННЯ" для підключення блоку реле призначені для підключення 16-ти каналного блоку реле. У розніманні з маркуванням "Лінії керування 0-F" роз'єм № 1 відповідає лінії керування 0, роз'єм № 2 відповідає лінії керування 1 і так далі до роз'єм № 16, що відповідає лінії F. Роз'єм з № 17 по 25 загальний (-).

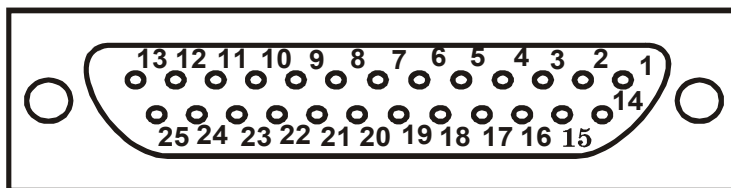


Рисунок 3.4 – Вид розетки для підключення блоку реле з боку монтажу.

**Аналоговий вихід.** При перевищенні максимальних значень вологості й температури або відсутності імпульсів у відповідних каналах вихідний струм залишається рівним максимальному.

При зниженні вологості або температури нижче мінімального значення вихідний струм дорівнює нулю. При маскуванні каналів вологості й температури вихідний струм може приймати довільне значення в межах діапазону його зміни. У режимах настроювання приладу вихідний струм залишається рівним останньому обмірюваному значенню. Вихідний сигнал – струм прямо пропорційний вимірюваній відносній вологості й температурі й може змінюватися залежно від замовлення межах від 0 до 20, від 4 до 20 і від 0 до 5 мА. Значення вологості й температури розраховуються по формулах:

$$H = (I_h - I_{min}) \frac{(H_{max} - H_{min})}{(I_{max} - I_{min})} + H_{min}, \% \quad (3.3)$$

$$T = (I_t - I_{min}) \frac{(T_{max} - T_{min})}{(I_{max} - I_{min})} + T_{min}, ^\circ \quad (3.4)$$

де  $I_h$  – значення струму, що відповідає вимірюваній вологості,  $I_{min}$  – мінімальне значення вихідного струму,  $I_{max}$  – максимальне значення вихідного струму,  $H_{min}$  – мінімальне значення вологості,  $H_{max}$  – максимальне значення вологості,  $I_t$  – значення струму, що відповідає вимірюваній температурі,  $T_{min}$  – мінімальне значення температури,  $T_{max}$  – максимальне значення температури.  $I_{min}$ ,  $I_{max}$ ,  $H_{min}$ ,  $H_{max}$ ,  $T_{min}$ ,  $T_{max}$  – параметри аналогового виходу, що задаються при замовленні.

### **Опис багатоканальної системи визначення вологості**

Запропонована в магістерському проекті система, можлива для реалізації в трьох різних варіантах:

- Багатоканальна система контролю й керування вологістю й температурою.
- Вимірювальна мережа зі збором інформації на FLASH-card.
- Вимірювальна мережа з передачею інформації через GPRS.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Розглянемо їх докладніше

## Багатоканальна система контролю й керування вологістю й температурою

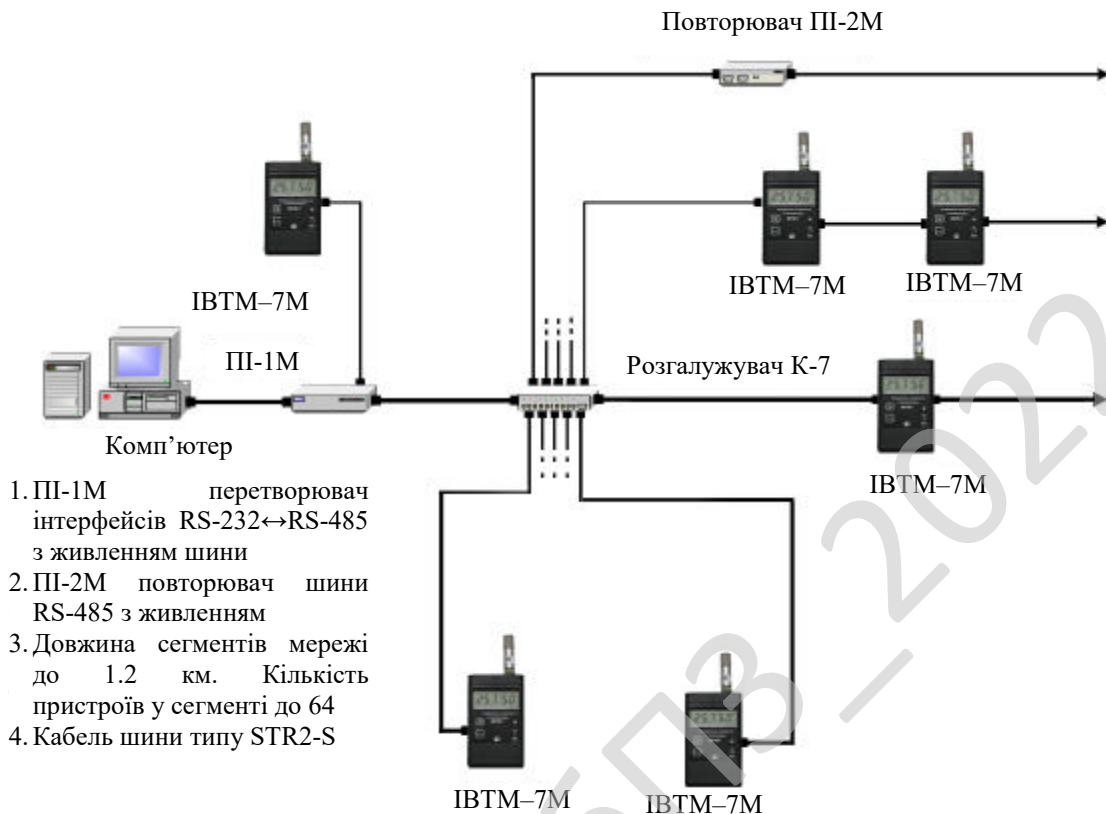


Рисунок 3.5 – Багатоканальна система контролю й управління мікрокліматом

Багатоканальна система контролю й управління мікроклімату. Призначена для контролю (і керування) температурою й відносною вологістю на різних, вилучених друг від друга об'єктах.

У систему можуть поєднуватися прилади ІВТМ-7 МК-3, ІВТМ-7 МК-3-М, ІВТМ-7 Р-МК, ІВТМ-7 Р-МК-М, ІВТМ-7/8 Р-МК, які призначені для виміру вологості й температури в складському приміщенні. Як допоміжні пристрої в системі, при необхідності, використовуються перетворювачі інтерфейсу ПІ-1, ПІ-2, комутатор RS-232, і адаптери. Інформація від всіх приладів виводиться на комп'ютер. З нього ж може здійснюватися керування приладами, що мають функцію регулювання. Для роботи із системою використовується комп'ютерна

програма. Діапазон виміру відносної вологості 0...99%, температури -20...+60 °C (-50...+150 °C по спеціальному замовленню). Прилад має можливість підключення до комп'ютера по інтерфейсу RS232 і RS485.

### **Вимірювальна мережа зі збором інформації на FLASH-card**

#### Можливості мережі

- До складу мережі можуть входити будь-які портативні й мережні одно- і багатоканальні прилади, що підтримують інтерфейс RS-485 (або інтерфейс RS-232 с зовнішнім перетворювачем інтерфейсу ПІ-7). Кількість приладів у мережі обмежується параметрами інтерфейсу RS-485 (64 приладів в мережі без додаткових підсилювачів).

- У вимірювальну мережу можуть входити також виконавчі пристрої (блоки реле й т.д)

- Збір інформації здійснюється КПК, що підключається до вимірювальної мережі через спеціальний пристрій узгодження. Для КПК написане спеціальне програмне забезпечення.

- Програма на КПК здійснює функції сервера, надаючи клієнтам інформацію, зчитувальну із приладів, по запиту із глобальної мережі Інтернет. Доступ серверної частини в Інтернет організується за допомогою встановлюваного в КПК модуля GPRS через операторів мобільного зв'язку GSM, що підтримують цей сервіс. ПЗ дозволяє відслідковувати стан зв'язку й при її обриві автоматично відновлювати з'єднання. Реалізовано функцію реєстрації на сервері динамічної IP адреси, виділюваного КПК у момент з'єднання, що дозволяє підтримувати постійний зв'язок між клієнтами й сервером.

- Для ПК написана спеціальна програма-клієнт, що дозволяє шляхом запиту через мережу Інтернет одержати доступ з будь-якої точки миру до інформації, зчитувальної КПК. Клієнтське ПЗ також підтримує технологію реєстрації на сервері динамічного IP адреси, надаваного КПК.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

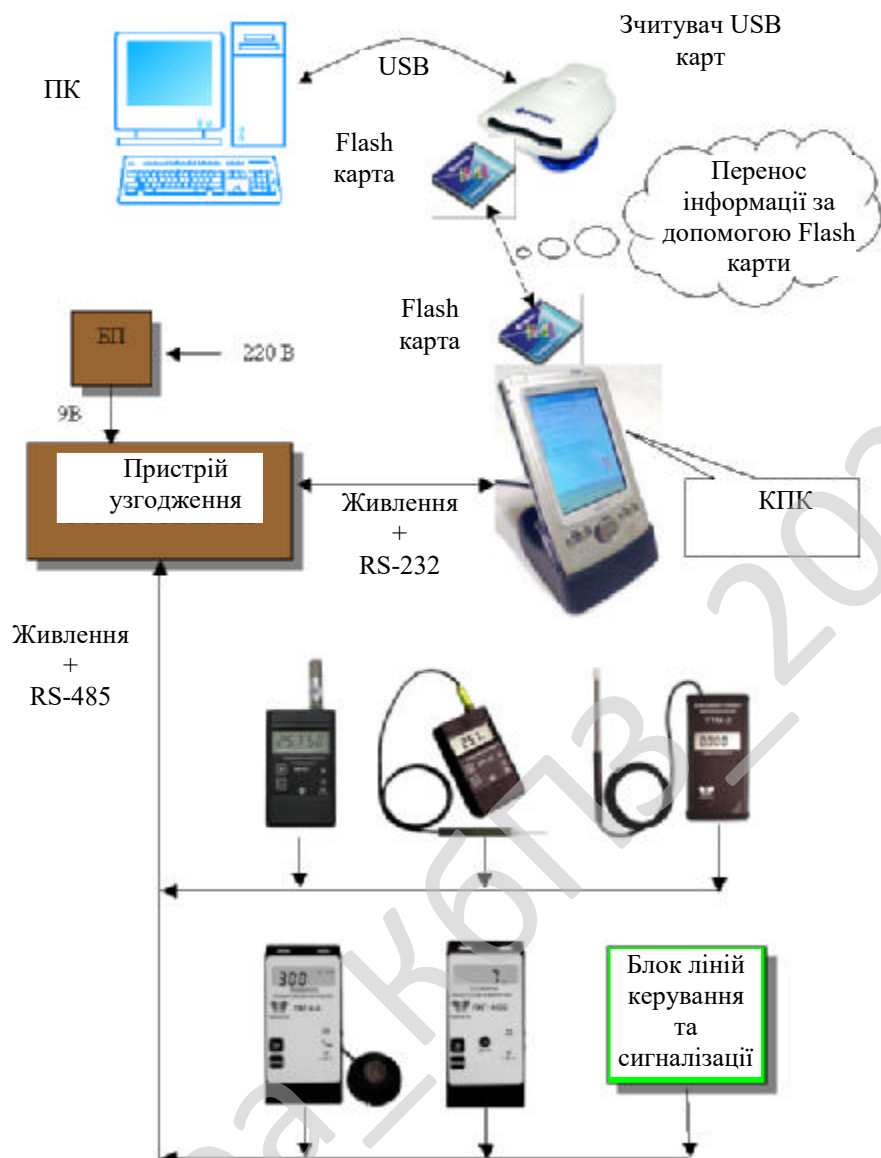


Рисунок 3.6 – Вимірювальна мережа зі збором інформації на FLASH-карту

### Вимірювальна мережа з передачею інформації через GPRS

#### Можливості мережі

- До складу мережі можуть входити будь-які портативні й мережні одно- і багатоканальні прилади, що підтримують інтерфейс RS-485 (або інтерфейс RS-232 с зовнішнім перетворювачем інтерфейсу ПІ-7). Кількість приладів у мережі обмежується параметрами інтерфейсу RS-485 (64 приладів в мережі без додаткових підсилювачів).

- У вимірювальну мережу можуть входити також виконавчі пристрої (блоки реле й т.д)



Рисунок 3.7 – Вимірювальна мережа з передачею інформації через GPRS

- Збір інформації здійснюється КПК, що підключається до вимірювальної мережі через спеціальний пристрій узгодження. Для КПК написано спеціальне програмне забезпечення.

- Програма на КПК здійснює функції сервера, надаючи клієнтам інформацію, зчитувальну із приладів, по запиту із глобальної мережі Інтернет.

Доступ серверної частини в Інтернет організується за допомогою встановлюваного в КПК модуля GPRS через операторів мобільного зв'язку GSM, що підтримують цей сервіс. ПЗ дозволяє відслідковувати стан зв'язку й при її обриві автоматично відновлювати з'єднання. Реалізовано функцію реєстрації на сервері динамічної IP адреси, виділюваного КПК у момент з'єднання, що дозволяє підтримувати постійний зв'язок між клієнтами й сервером.

- Для ПК написана спеціальна програма-клієнт, що дозволяє шляхом запиту через мережу Інтернет одержати доступ з будь-якої точки миру до інформації, зчитувальної КПК. Клієнтська ПЗ також підтримує технологію реєстрації на сервері динамічного IP адреси, надаваного КПК.

### 3.2 Розробка структурної схеми

Розглянемо розроблену структурну схему (рисунок 3.8.) з об'єкту спостереження до кінцевої точки оператора.



Рисунок 3.8 – Структурна схема системи

Взаємодія оператора з об'єктами спостереження в системі мікроклімату складського комплексу відбувається за наступною схемою Спочатку у складському приміщенні у відповідних бункерах встановлюються датчики вимірювання вологості в залежності від моделі бункера зберігання використовуються різні датчики в нашому випадку це RKF/A (детальніше

розглянута взаємодія у пункті 3.5) далі за допомогою 32 каналного АЦП/ЦАП дані обробляються та за допомогою перетворювача інтерфейсів з RS485 – RS232 відбувається взаємодія з персональним комп'ютером та розробленим програмним забезпеченням яким керує оператор.

### 3.3 Розробка функціональної схеми

#### Розробка функціональної схеми апаратної частини

На рисунку 3.9 розглянута функціональна схема апаратної частини, розглянемо її детально. З об'єктів спостереження, яких може бути 32 (при необхідності це число можливо подвоїти), виходить інформація за допомогою датчиків RKF/A зображених на рисунку 3.10.

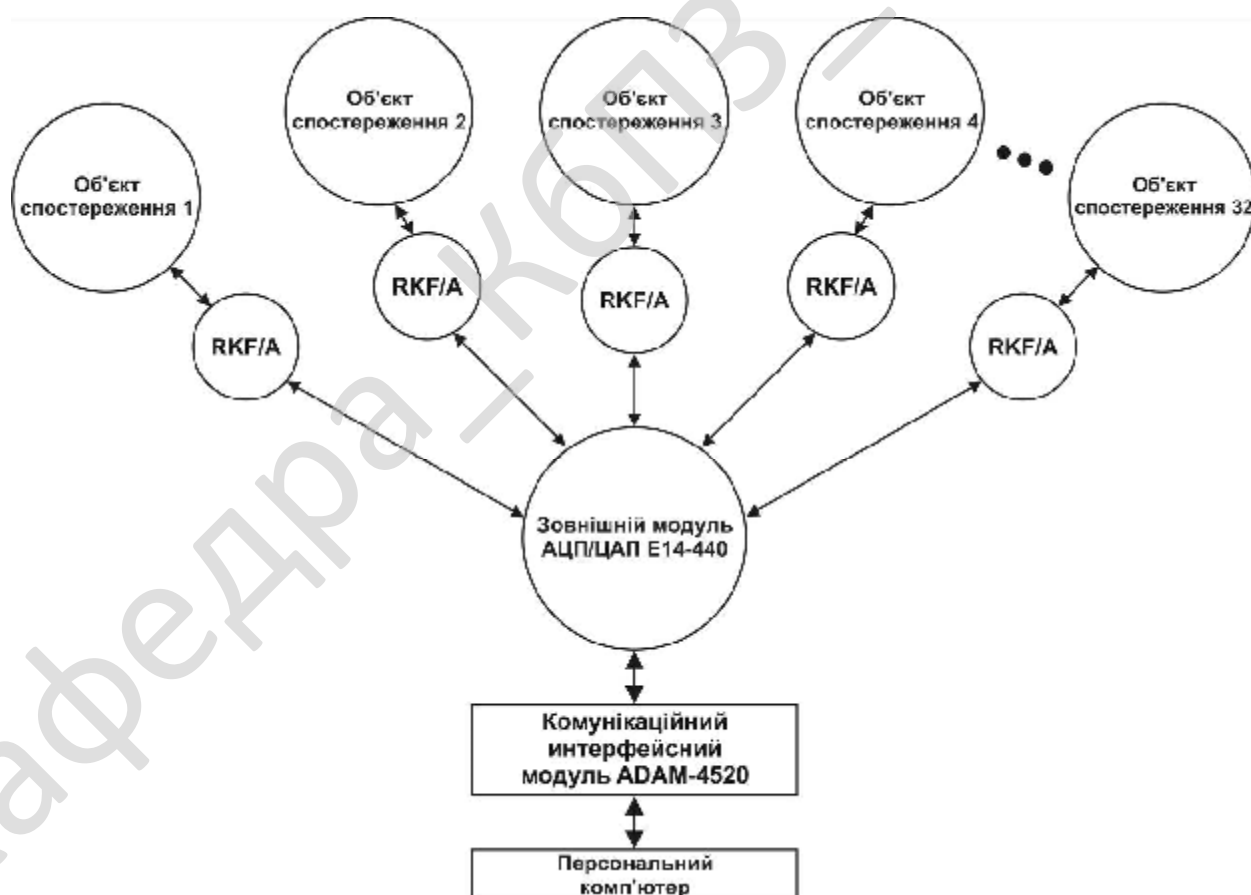


Рисунок 3.9 – Функціональна схема апаратної частини

Вимір відносної вологості стає усе більше важливим в автоматизації промислового керування. Вимірювальний прилад реєструє відносну вологість навколишнього середовища або в каналі бункера за допомогою ємнісного сенсора.



Рисунок 3.10 – Зовнішній вигляд каналного датчика вологості RKF/A

Датчик перетворить ці виміри в стандартизований сигнал виходу, що готовий до подальшої обробки у розробленій магістерській програмі. Сенсорний елемент захищений фільтром. Кожний прилад цієї серії проходить 2-денне випробування. Схема використаного підключення зображена на рисунку 3.11.

Далі інформація передається на зовнішній модуль АЦП/ЦАП E14-440 котрий розвантажує процесор комп'ютера від керування введенням/виведенням і первинною обробкою сигналів. Програмно здійснюється вибір діапазонів виміру, частоти дискретизації й конфігурація входів АЦП як диференціальних із загальною "землею".

Синхронізація АЦП можлива по зовнішньому синхросигналу або за рівнем вхідного сигналу. Передбачено опцію установки двохканального ЦАП. Функціонально модуль ідентичний платі L-1450. Модуль особливо зручний для створення портативних вимірювальних систем.

Для з'єднання з персональним комп'ютером необхідно використання комунікаційного інтерфейсного модуля (RS485 – RS232), в нашому випадку використовується ADAM-4520 (рисунок 3.12).

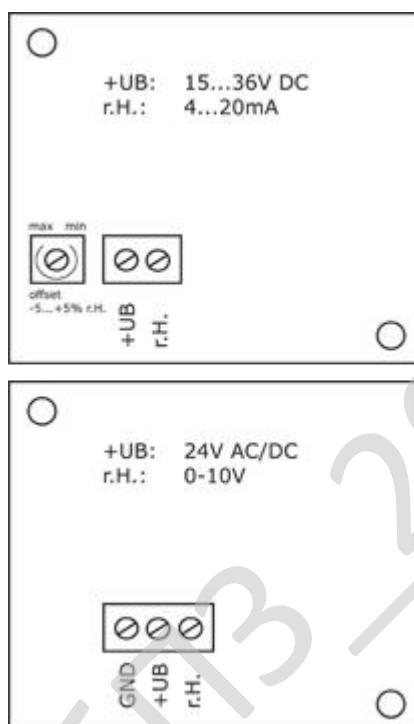


Рисунок 3.11 – Схема підключення каналного датчика вологості RKF/A



Рисунок 3.12 – Зовнішній вигляд комунікаційного інтерфейсного модуля ADAM-4520

## Розробка функціональної схеми програмної частини схеми

На рисунку 3.13 зображена розроблена функціональна схема програмної частини магістерського проекту. З неї видно як розроблена програма взаємодіє з апаратною частиною.

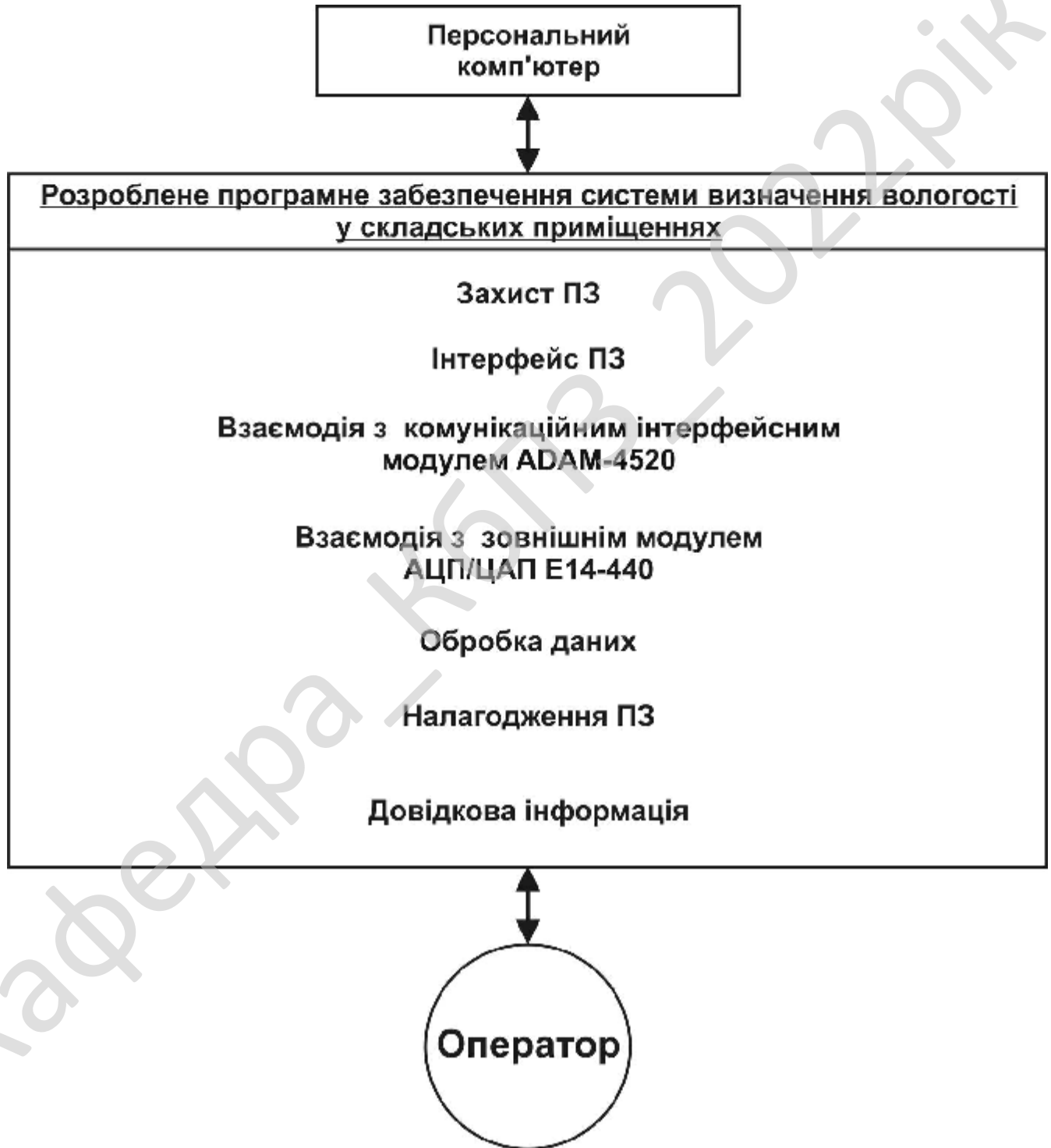


Рисунок 3.13 – Функціональна схема програмної частини

Розроблена магістерська програма розбита на такі частини:

1. Взаємодія з комунікаційним інтерфейсним модулем ADAM-4520.
2. Взаємодія з зовнішнім модулем АЦП/ЦАП Е14-440.
3. Інтерфейс ПЗ.
4. Захист ПЗ.
5. Обробка даних.
6. Налаштування ПЗ.
7. Довідкова інформація.

### 3.4 Розробка діаграми процесів

Розглянемо діаграму процесів рисунок 3.14, на діаграмі зображений процес роботи й взаємодії програми, саме по діаграмі можна зрозуміти як працює розроблена програма в цілому. Починається й закінчується програма в першому блоці котрий є основною крапкою відліку у діаграмі, при переміщенні по стрілках можна побачити загальну схему взаємодії блоків і їх входження друг у друга.

Розглядаючи діаграму можна побачити, що чітко простежується те, що взаємодія програми відбувається через блок захисту програми який взаємодіє з бібліотеками налаштування та взаємодії з комунікаційним інтерфейсним модулем та модулем ЦП/ЦАП також модуль захисту взаємодіє з інтерфейсом ПЗ.

Блок інтерфейсу ПЗ крім налаштування ПЗ проводить взаємодію з комунікаційним інтерфейсним модулем який у свою чергу взаємодіє з зовнішнім модулем і в остаточному підсумку проходить обробка даних.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>

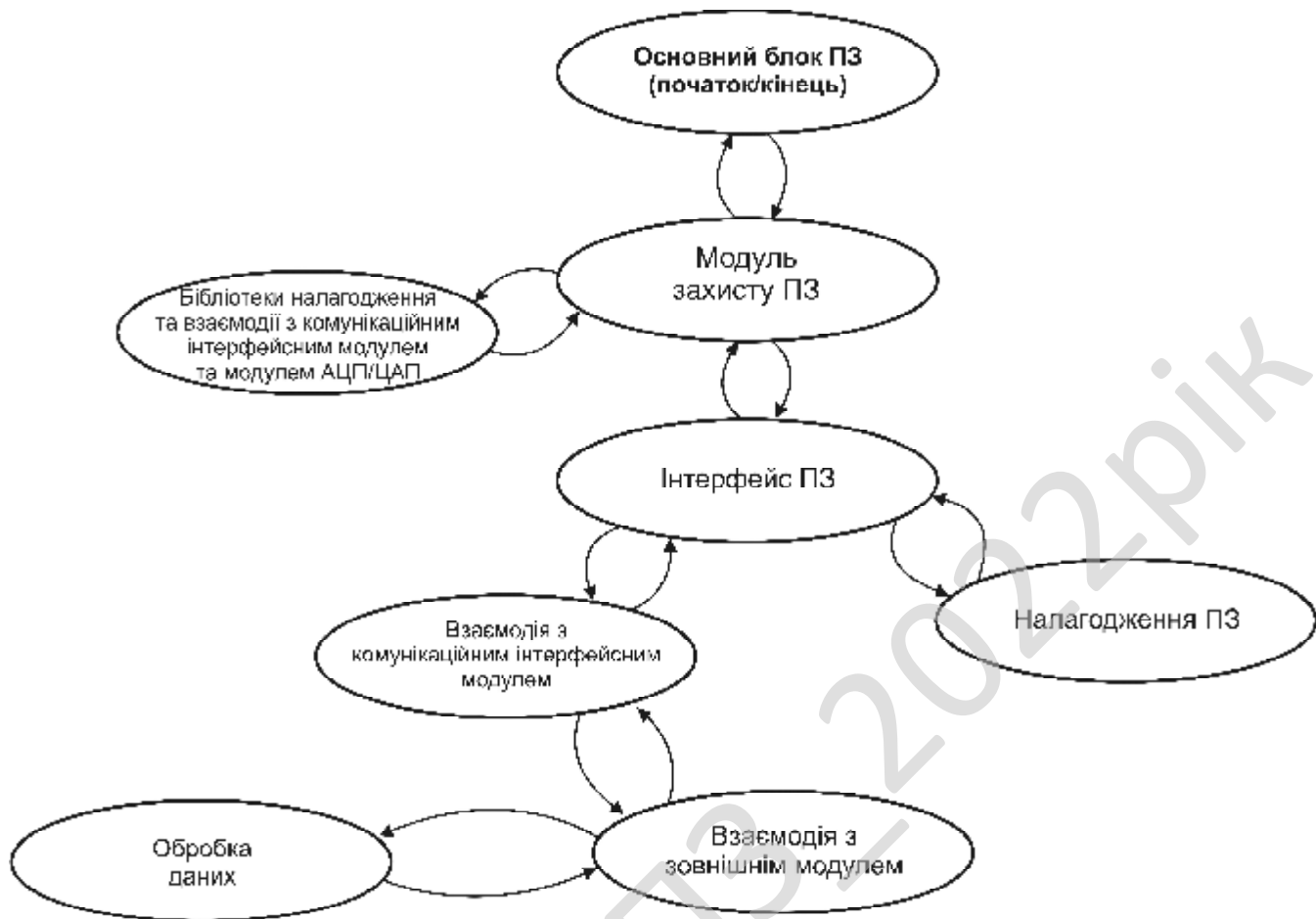


Рисунок 3.14 – Діаграма процесів

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

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

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 зображена основна блок-схема програми з котрої йде два відгалуження підпрограма отримання даних (рисунок 4.2) та підпрограма налагодження граничних значень вологості (рисунок 4.3). При розробці блок-схеми робився упор на впровадження бібліотек взаємодії з обладнанням.

При детальному розгляді магістерська програма розбита на декілька важливих блоків, розглянемо складові елементи блок-схеми та їхній опис.

Початкові блоки програми:

- початкова ініціалізація змінних та виділення пам'яті;
- підключення додаткових модулів;
- пошук файлу, читання та встановлення початкових граничних значень;
- підключення бібліотек взаємодії з RS232/RS485;
- ініціалізація та спроба доступу до RS232/RS485;
- перевірка доступу до RS232/RS485.

Робота програми:

- встановлення інтерфейс них початкових значень у головному вікні ПЗ;
- виведення головного вікна ПЗ на екран;
- підпрограма отримання даних (рисунок 4.2) ;
- підпрограма налагодження граничних значень вологості (рисунок 4.3) ;

Завершення роботи програми:

- виведення вікна повідомлення про завершення роботи ПЗ;
- приховання головного вікна ПЗ на екран;
- очікування кінця поточної взаємодії з RS232/RS485;
- закриття порту та очищення буфера обміну;
- відключення додаткових модулів та подальше звільнення ресурсів програми.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

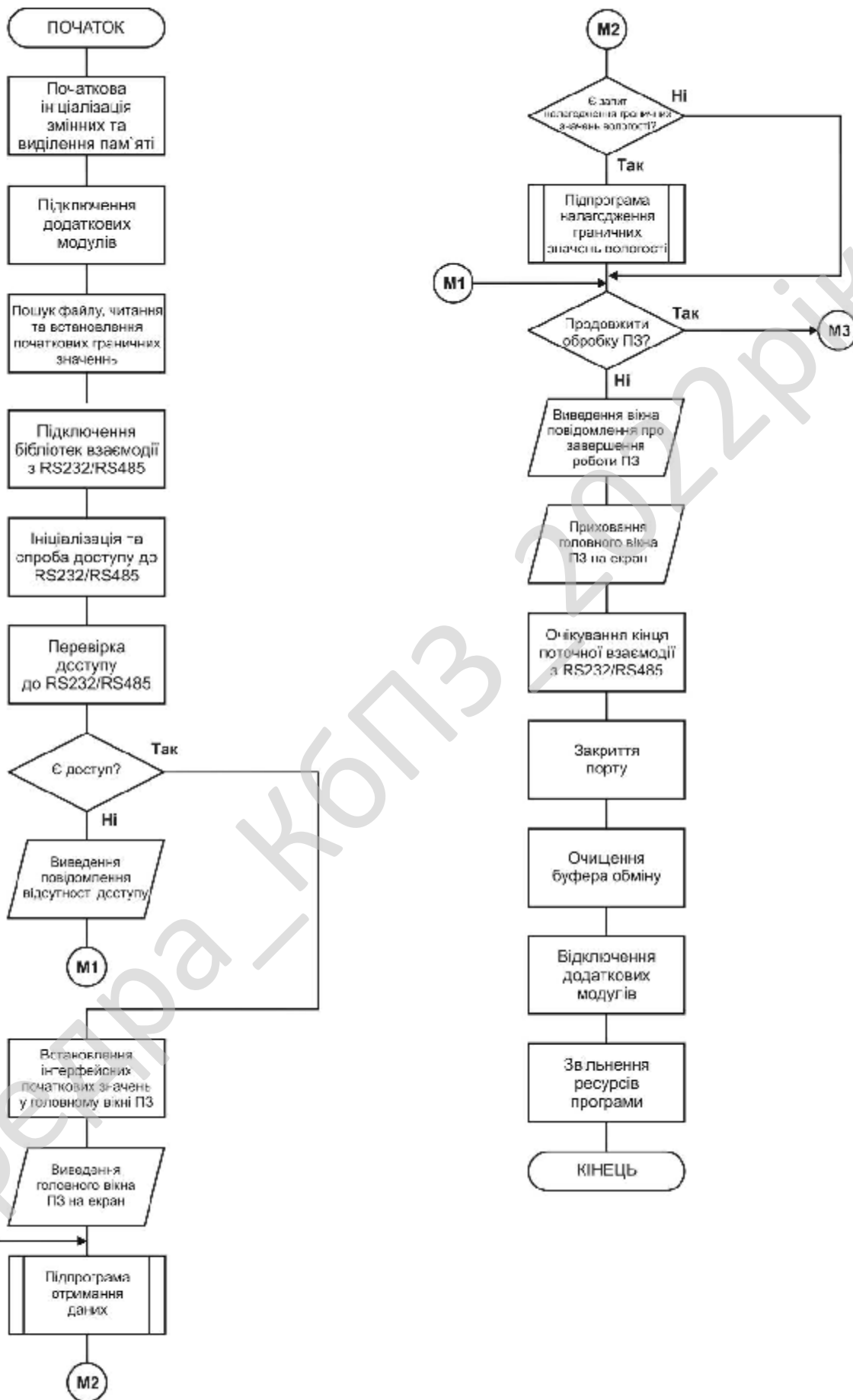


Рисунок 4.1 – Блок схема основної частини ПЗ

Підпрограма отримання даних

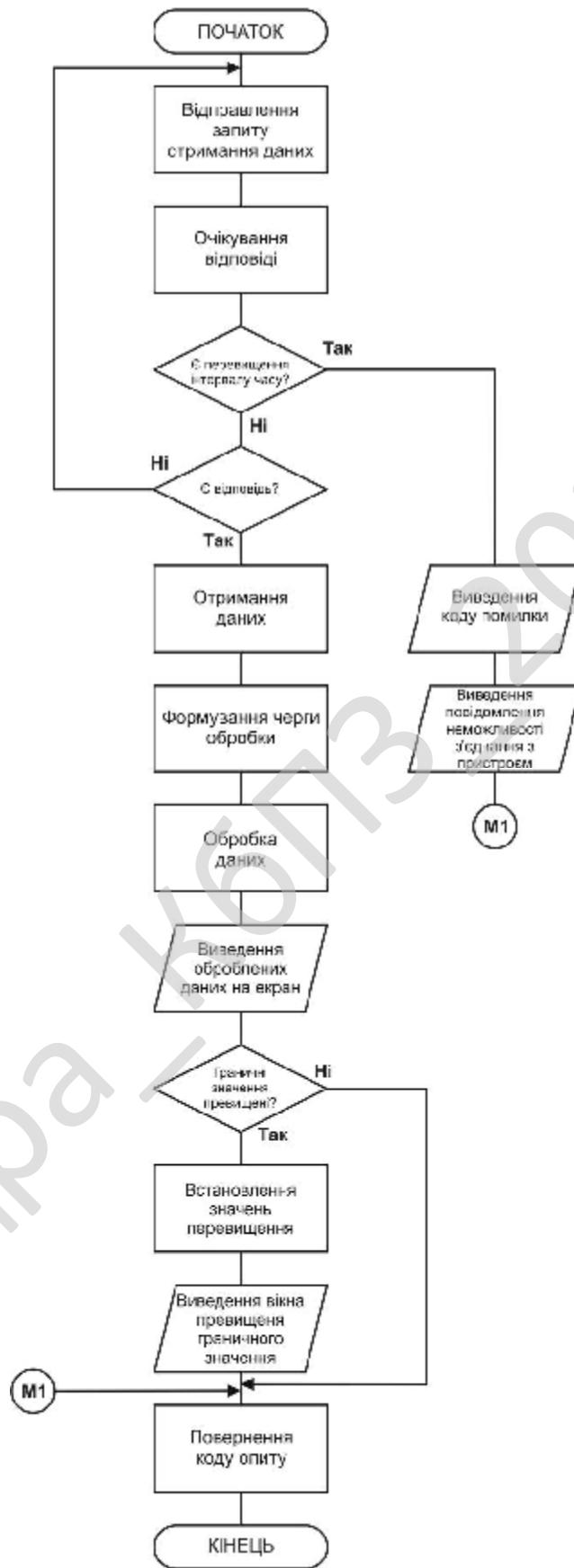


Рисунок 4.2 – Блок-схема підпрограми отримання даних

Підпрограма налагодження граничних значень вологості



Рисунок 4.3 – Блок-схема підпрограми налагодження граничних значень вологості





кожної операції запису дане значення додається до результату множення WriteTotalTimeoutMultiplier на кількість записуваних байт. Нульове значення полів WriteTotalTimeoutMultiplier і WriteTotalTimeoutConstant означає, що загальний тайм-аут для операції запису не використовується.

### 3) Структура TCOMSTAT.

Поля структури TCOMSTAT мають наступне призначення:

- **Flags** : DWORD – набір прапорів;
- **cbInQue** : DWORD – кількість байт у буфері прийому.
- **cbOutQue** : DWORD – кількість байт у буфері передачі.

Розглянемо одинадцять використовуваних функції API для роботи з портом їхні виклики й параметри.

#### 1) API функція CreateFile.

Опис – За допомогою її відкривається порт, тобто одержуємо хендл порту для подальшої роботи.

Оголошення функції:

```
function CreateFile(lpFileName: PChar;  
dwDesiredAccess, dwShareMode: DWORD;  
lpSecurityAttributes: LPSECURITY_ATTRIBUTES;  
dwCreationDistribution, dwFlagsAndAttributes: DWORD;  
hTemplateFile: THandle): THandle,
```

де:

- lpFileName – покажчик на нуль-термінований рядок ('\\.\COM1');
- dwDesiredAccess – режим відкриття порту (запис, читання, запис/читання);
- dwShareMode – режим перегляду порту;
- lpSecurityAttributes – структура атрибутів відкриття;
- dwCreationDistribution – завжди OPEN\_EXISTING для портів;
- dwFlagsAndAttributes – завжди FILE\_FLAG\_OVERLAPPED;
- hTemplateFile – завжди 0 для портів.



Оголошення функції:

```
function G(S)etCommMask(hPort: THandle; Events : DWORD) : boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;
- Events : DWORD – покажчик на маску EVENTS.

#### **5) API функції GetCommTimeouts і SetCommTimeouts.**

Опис – Функції необхідні для читання й запису налаштувань таймаутів порту.

Оголошення функції:

```
function G(S)etCommTimeouts(hPort: THandle;  
                             lpCommTimeouts : TCommTimeouts) : boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;
- lpCommTimeouts : TCommTimeouts – покажчик на TCommTimeouts.

#### **6) API функції GetCommModemStatus.**

Опис – Функція для читання поточного стану регістра стану порту.

Оголошення функції:

```
function GetCommModemStatus(hPort: THandle;  
                             var lpModems :DWORD): boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;
- lpModems : DWORD – змінна зі станами ліній.

Після виклику функції lpModems містить комбінацію наступних констант:

- MS\_CTS\_ON – стан лінії CTS;
- MS\_DSR\_ON – стан лінії DSR;
- MS\_RING\_ON – стан 9 ніжки DB9 порту;
- MS\_RLSD\_ON – наявність/відсутність несучої зв'язку.

#### **7) API функція SetupComm.**

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>50</b>

Опис – Функція необхідна для установки розмірів внутрішніх буферів прийому й передачі драйвера порту.

Оголошення функції:

```
function SetupComm(hPort: THandle; lpIn, lpOut : DWORD) : boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;
- lpIn : DWORD – довжина буфера для читання;
- lpOut : DWORD – довжина буфера для запису.

### **8) API функція ClearCommError.**

Опис – Функція необхідна для скидання помилок, одержання поточного коду помилок і читання структури стану порту.

Оголошення функції:

```
function ClearCommError(hPort: THandle;  
                        lpErrors : DWORD;  
                        lpCommStat : PComStat) : boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;
- lpErrors : DWORD – покажчик на маску помилок;
- lpCommStat : PComStat – покажчик на TComStat.

### **9) API функція WaitCommEvent.**

Опис – Функція повернення (очікування) EVENTS від COM порту.

Оголошення функції:

```
function WaitCommEvent(hPort: THandle;  
                      lpMask : DWORD;  
                      lpOver : POverlapped ) : boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;
- lpMask : DWORD – покажчик на маску EVENTS;
- lpOver : POverlapped – покажчик на TOverlapped.

При вдалому завершенні функції змінна lpMask містить маску отриманих EVENTS. При невдачі й при GetLastError=ERROR\_IO\_PENDING для одержання маски EVENTS необхідно спочатку викликати Wait-Функцію (наприклад WaitForSingleObject) на очікування установки hEvent структури lpOver (параметр WaitCommEvent ) у сигнальний стан.

#### 10) API функції ReadFile і WriteFile.

Опис – Функції необхідні для читання й запису інформації з і в порт.

Оголошення функції:

```
function Read(Write)File(hPort: THandle;  
    lpBuff : pointer;  
    lpCnt : DWORD;  
    var lpCntByte : DWORD;  
    lpOver : POverlapped) : boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;
- lpBuff : pointer – покажчик на буфер для запису прийнятих байт;
- lpCnt : DWORD – кількість байт для читання (запису);
- lpCntByte : DWORD – кількість лічених в lpBuff байт (записаних у порт із lpBuff);
- lpOver : POverlapped – покажчик на TOverlapped.

У випадку вдалого завершення змінна lpCntByte містить кількість байт, лічених в lpBuff у випадку ReadFile або записаних у порт із lpBuff у випадку WriteFile.

#### 11) API функція TransmitCommChar.

Опис – Функція призначена для негайної передачі байта портом.

Оголошення функції:

```
function TransmitCommChar(hPort: THandle; Chr : Char) : boolean;
```

де:

- hPort : THandle – хендл порту від CreateFile;



```

//одержуємо поточне DCB порту
GetCommState(cId,DCB)
Then
begin
    CloseHandle(cId);
    Exit;
end;
//змінюємо DCB
DCB.BaudRate := 9599;
DCB.ByteSize := 8;
DCB.Parity := NoParity;
DCB.StopBits := OneStopBit;
//виконуємо налаштування порту з новим DCB
if not SetCommState(cId,DCB) then begin
CloseHandle(cId);
Exit;
end;
//одержуємо поточні параметри таймаутов
GetCommTimeouts(cId,TimeOuts);
//набудовуємо поточні параметри таймаутов таким чином,
//щоб ReadFile і WriteFile повертали значення негайно
TimeOuts.ReadIntervalTimeout:= MAXDWORD;
TimeOuts.ReadTotalTimeoutMultiplier:= 0;
TimeOuts.ReadTotalTimeoutConstant:= 0;
TimeOuts.WriteTotalTimeoutMultiplier:= 0;
TimeOuts.WriteTotalTimeoutConstant:= 0;
//виконуємо налаштування порту з новими таймаутами
if not SetCommTimeouts(cId,TimeOuts) then begin
CloseHandle(cId);
Exit;
end;
//опускаємо прапор завершення дочірнього потоку
Terminated := False;
//стартуємо дочірній потік (функція потоку - ReadsComm)
//для обробки EVENTS порту й встановлюємо пріоритет
CommThread:=CreateThread(nil,0,@ReadsComm,nil,0,ThreadID);
if CommThread = 0 then begin
CloseHandle(cId);
Exit;
end;
SetThreadPriority(CommThread,8);
end;

```

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>54</b>





```

private
procedure ObrMess(var Msg : TMessage); message cmRxByte;
    .....
end;
    .....
procedure TfmMain.ObrMess(var Msg : TMessage);
var
    i, Cnt : integer;
    S : string;
    Buff : pointer;
begin
//записуємо прийняті байти в S
    Buff := Msg.WParam;
    Cnt := Msg.LParam;
    S := '';
    for i := 0 to Cnt-1 do begin
        if Byte(Buff^) >= $20
            then S := S + Chr(Buff^)
            else S := S + IntToHex(Byte(Buff^),2) + '_';
        inc(Buff);
    end;
//додаємо S в Мемо
    Memo1.Lines.Add(S);
//звільняємо пам'ять, виділену в дочірньому потоці
    FreeMem(Pointer(Msg.WParam));
end;

```

### 5) Вихідний код основної процедури запису в порт.

```

function WriteComm(hComm : THandle; Buff : array of byte; Count : integer; Wait :
boolean) : integer;
var
    Ovr : TOverlapped;
    Code : DWord;
    S : string;
begin
//ініціалізуємо TOverlapped структуру
    FillChar(Ovr, SizeOf(TOverlapped), 0);
    Ovr.hEvent := CreateEvent(nil, TRUE, FALSE, #0);
//намагаємося записати
    if not WriteFile(hComm, Buff, Count, Result, @Ovr) and Wait
        then begin
            if (GetLastError() = ERROR_IO_PENDING) and

```

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>57</b>

```

        (WaitForSingleObject (Ovr.hEvent, INFINITE) = WAIT_OBJECT_0)
    then GetOverlappedResult (cId, Ovr, Result, False);
end;
CloseHandle (Ovr.hEvent);
end;

```

**б) Вихідний код закриття порту.** Оскільки для обробки EVENTS порту використовується дочірній потік, то функція закриття порту повинна мати такий вигляд.

```

function CloseComm : boolean;
begin
    //установлюємо прапор завершення
    Terminated := True;
    //якщо в цей момент потік перебуває в WaitCommEvent
    // примусово змушуємо завершитися
    SetCommMask (cId, 0);
    //передаємо ресурси системі, щоб потік одержав ресурси
    Sleep (10);
    //закриваємо хендл порту
    CloseHandle (cId);
    Result := True;
end;

```

#### 4.2 Захист розробленого програмного забезпечення

Tiny Encryption Algorithm (TEA) [1] – блочний алгоритм шифрування типу «Мережі Фейстеля». Алгоритм був розроблений на факультеті комп'ютерних наук Кембриджського університету Девідом Вілером (David Wheeler) і Роджером Нідгемом (Roger Needham) та вперше представлений в 1994 році [2] на симпозиумі зі швидкими алгоритмами шифрування в Льовені (Бельгія).

Шифр не патентований, широко використовується в ряді криптографічних додатків і широкому спектрі апаратного забезпечення, завдяки вкрай низькими вимогами до пам'яті й простоті реалізації. Алгоритм має як програмну реалізацію на різних мовах програмування, так і апаратну реалізацію на інтегральних схемах типу FPGA.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму TEA, який заснований на бітових операціях з 64-бітним блоком, має 128-бітний ключ шифрування. Стандартна кількість раундів мережі Фейстеля біля 64 (32 циклу), однак, для досягнення найкращої продуктивності або шифрування, число циклів можна варіювати від 8 (16 раундів) до 64 (128 раундів). Мережа Фейстеля несиметрична через використання в якості операції накладення додавання по модулю 232.

Перевагами шифру є його простота в реалізації, невеликий розмір коду й досить висока швидкість виконання, а також можливість оптимізації виконання на стандартних 32-бітних процесорах, так як в якості основних операцій використовуються операції виключна «АБО» (XOR), побітового зсуву й додавання по модулю 232. Оскільки алгоритм не використовує таблиць підстановки і раундова функція досить проста, алгоритму потрібно не менше 16 циклів (32 раундів) для досягнення ефективної дифузії, хоча повна дифузія досягається вже через 6 циклів (12 раундів).

Алгоритм має відмінну стійкість до лінійного криптоаналізу і досить гарну до диференціального криптоаналізу. Головним недоліком цього алгоритму шифрування є його вразливість до атак «на пов'язаних ключах» (англ. Related-key attack). Через простий розклад ключів кожен ключ має 3 еквівалентних ключа. Це означає, що ефективна довжина ключа складає всього 126 біт [3] [4], тому даний алгоритм не слід використовувати в якості геш-функції.

### Опис алгоритму

Вихідний текст розбивається на блоки по 64 біта кожен. 128-бітний ключ  $K$  ділиться на чотири 32-бітних підключа  $K[0]$ ,  $K[1]$ ,  $K[2]$  і  $K[3]$ . На цьому підготовчий процес закінчується, після чого кожен 64-бітний блок шифрується протягом 32 циклів (64 раундів) за нижченаведеним алгоритмом. [5]

Припустимо, що на вхід  $n$ -го раунду ( $1 \leq n \leq 64$ ) надходять права й ліва частини  $(L_n, R_n)$ , тоді на виході  $n$ -го раунду будуть ліва й права частини  $(L_{n+1}, R_{n+1})$ , які обчислюються за такими правилами:





На рисунку 5.2 зображено вікно занесення та зміни параметрів вологості зерна при збиранні та зберіганні.

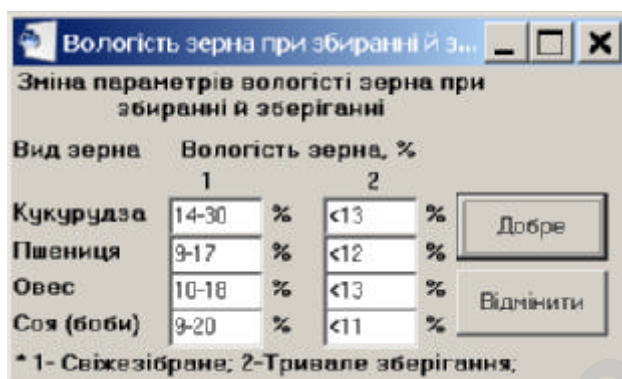


Рисунок 5.2 – Вікно встановлення параметрів зберігання

На рисунку 5.3 наведено вікно перевірки зв'язку з перехідником ADAM-4250, між кінцевою апаратурою та лінією зв'язку.

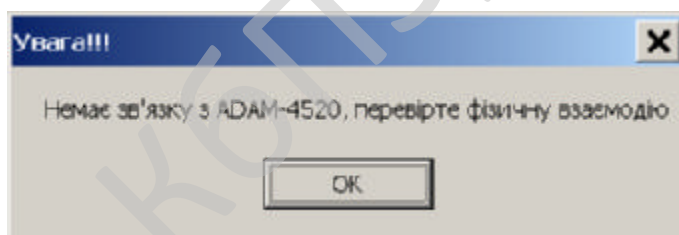


Рисунок 5.3 – Попереджувальне вікно відсутності зв'язку з ADAM-4520

На рисунку 5.4 зображене вікно перевірки зв'язку з пристроєм керування датчиками E14-440. Цей пристрій може одночасно оброблювати дані та керувати до 32 датчиків.

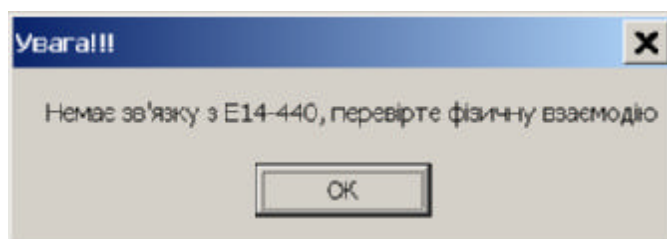


Рисунок 5.4 – Попереджувальне вікно відсутності зв'язку з E14-440

У випадку втрати взаємозв'язку з кінцевими пристроями та програмним забезпеченням на монітор видається форма виду зображеного на рисунку 5.5.

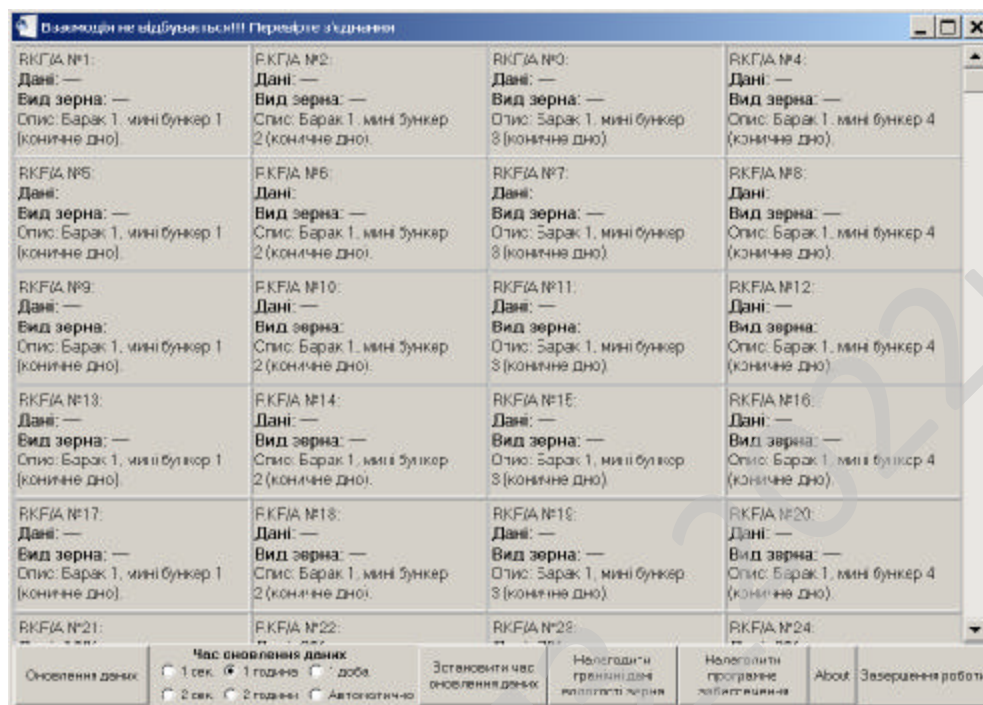


Рисунок 5.5 – Головне вікно програми, немає зв'язку

Форма авторського права наведена на рисунку 5.6.

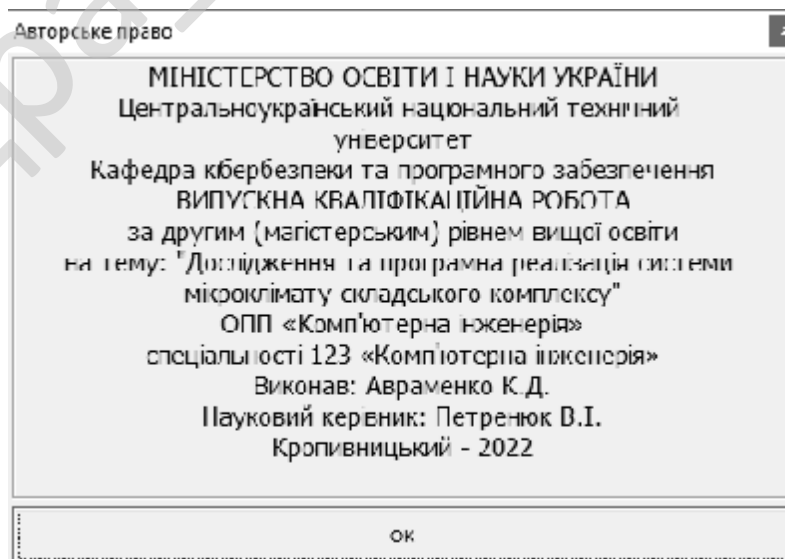


Рисунок 5.6 – Форма авторського права

## 6 НАУКОВА НОВИЗНА

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

*Метою розробки є дослідження та програмна реалізація системи мікроклімату складського комплексу.*

*Об'єктом дослідження є процес мікроклімату складського комплексу.*

*Предметом дослідження є методи мікроклімату складського комплексу.*

*Методи дослідження базуються на методах інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод мікроклімату складського комплексу.
- Розроблено вітчизняний продукт мікроклімату складського комплексу, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

## 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 60 днів (три місяці).

В магістерській роботі було проведене дослідження та виконана програмна реалізація системи мікроклімату складського комплексу.

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	40
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження табл. 7.1

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

Продовження табл. 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	40000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	55
38. Ставка податку на додану вартість, %	Ндв	20

## 7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

де  $A$  – коефіцієнт Боєма,  $A=2,45$ ;  $Size$  – 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.

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

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

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

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

де  $\prod V_j$  – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 75 = 126 \text{ люд/день}$$

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

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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	9	Д7
Робочий проект	126	Ф 7.1-7.4
Впровадження	13	Д13
Всього	167	–

### 7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

де  $F_{рч}$  – плановий фонд робочого часу одного спеціаліста, днів,  
 $T_{пз}$  – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{167 \cdot 1}{60 \cdot 5} = 3 \text{ ставки}$$

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	9	810	13,5
Монітор	60	9	540	9
Клавіатура	30	9	270	4,5
Маніпулятор «мишка»	30	9	270	4,5
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 <sub>ч</sub>	48,49

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:

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

$$\Phi_{\text{др}}^c = \frac{49 \cdot 3}{1,2} = 122,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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



Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців:

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	0,25	10000	7500
Продакт-менеджер	0,25	10000	7500
Інженер-програміст	3	10000	90000
Інженер-електронщик	0,25	10000	7500
Інженер-системотехнік	0,25	10000	7500
Адміністратор мережі	0,5	10000	15000
Системний програміст	0,25	10000	7500
Дизайнер WEB	0,25	10000	7500
Інженер-верстальник	0,25	10000	7500
Бухгалтер-економіст	0,5	10000	15000
Всього за період розробки	$R_{cn}=5,75$	-	$\Phi_{роб}=172500$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{сд} = \frac{172500}{5,75 \cdot 60} = 500 \text{ грн.}$$

#### 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>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

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

де  $R_{cn}^1$  – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць.  $S_y$  – питома площа на одне робоче місце,  $m^2$ ;  $\Pi_{nl}$  – 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./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно 8  $m^2$ . З урахуванням цього:

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

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

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

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

де  $\Pi_m$  – ціна меблів для одного робочого місця, грн.

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет магазину Компбест за 28.10.22 – джерело <https://compbest.com.ua>.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок	Fujitsu P720 Tower	7347
Процесор	Intel Core i3-4130 (2 (4) ядра по 3.40 GHz); Cache Memory 3	-
Системна плата	Fujitsu D3221-A1 Intel Haswell с TDP до 95 Вт	-
Відеокарта	AMD Radeon RX 550 4GB GDDR5 Re Dragon PowerColor (AXRX 550 4GBD5-DH)	-
Жорсткий диск	HDD Seagate Barracuda 750 Gb 7200 32Mb SATAIII ST3750528AS (ST3750528AS)	-
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-10600 CL9 Transcend JetRam, non-Reg., no-ECC , CL 9 (2 модулі)	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	GRESSO GE-7525, 500W (120mm big fan), 2xIDE, full-ATX,БЖ 2xSATA, 1xFDD, Air Duct, 2xUSB 2.0, Mic+Audio, silver/black	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4- B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	-
інше	Клавіатура, мишка	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	Samsung S22R350FHI (LS22R350FHIXCI) / 21.5" (1920x1080) IPS LED / HDMI, VGA	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струменевий	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	114885,1

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5			
4. Вимірювальні пристрої	5190	-	-
5. Господарський інвентар	28000	-	-
Всього по групі	33190	25	8297,5
Нематеріальні активи			
6. Нематеріальні активи	40000	10	4000
Разом	$K_p = 1596075$		$A_p = 140140$

### 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.

$$Z_o = 500 \cdot 167 / 40 = 2090 \text{ грн}$$

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, %

$$Z_d = 2090 \cdot 10 \cdot 0,01 = 209 \text{ грн}$$

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

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

де  $H_c$  – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(2090+209) = 506 \text{ грн}$$

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

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

де  $H_g$  – загальногосподарські витрати, %

$$G_{ocn} = 2090 \cdot 15 \cdot 0,01 = 314 \text{ грн}$$

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 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає  $C_n=210$  грн., визначаємо вартість паперу за період розробки:

$$Z_{M1} = C_n \cdot N_M \cdot n_{міс}. \quad (7.16)$$

$$Z_{M1} = 210 \cdot 1 \cdot 0,5 = 105 \text{ грн.}$$

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 10):

$$Z_{M2} = \sum C_{\delta}, \quad (7.17)$$

де:  $C_{\delta}$  – вартість дисків CD/DVD: CDR box – 23 грн./шт., DVD-R box – 49,2 грн./шт.

$$Z_{M2} = 49,2 \cdot 10 = 492 \text{ грн.}$$

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.

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

$$Z_M = (105 + 492 + 1702) / 40 = 57 \text{ грн.}$$

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, %

$$O_n = 2090 \cdot 15 \cdot 0,01 = 314 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 40$  прим.)

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

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

$$A_m = 140140 \cdot 3 / (40 \cdot 12) = 876 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

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

$$C_n = 2090 + 209 + 506 + 314 + 57 + 314 + 876 = 4366 \text{ грн.}$$

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 55%

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

де  $P_n$  – рівень рентабельності, %

$$P_p = 0,01 \cdot 55 \cdot 4366 = 2401 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1. Основна зарплата виконавців	$Z_o$	2090
2. Додаткова зарплата виконавців	$Z_d$	209
3. Відрахування на соціальні потреби	$C_{oc}$	506
4. Загальногосподарські витрати	$\Gamma_{ocn}$	314
5. Витрати на матеріали	$Z_m$	57
6. Освоєння нових операційних систем, мов програмування	$O_n$	314
7. Амортизація основних фондів	$A_m$	876
8. Повна собівартість програмного забезпечення	$C_n$	4366
9. Плановий прибуток	$P_p$	2401
10. Ціна підприємства $C_n = C_n + P_p$	$C_n$	6767
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{де} \cdot C_n$	$ПДВ$	1353,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	$C$	9168



Витрати на оплату праці:

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

де  $T_p$  – кількість годин обслуговування системи за рік, год.;  $Z_z$  – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 900 годин на рік до 200 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 900 \cdot 100 \cdot 1,1 \cdot 1,22 = 120780 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 200 \cdot 100 \cdot 1,1 \cdot 1,22 = 26840 \text{ грн.}$$

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 ( $C_{ел}$ ).

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

$$Z_{ел \text{ баз}} = 0,475 \cdot 2455 \cdot 1,8 = 2099 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,475 \cdot 1227 \cdot 1,8 = 1049 \text{ грн}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	9168	–	4584
Всього відрахувань	-	–	9168	–	4584



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

$$T_{cn} = \frac{9168}{122879 - 32473} = 0,1 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	40
2. Повна собівартість розробленої програми	Грн	4366
3. Ціна розробленої програми	Грн.	6767
4. Плановий прибуток від реалізації розробленої програми	Грн.	2401
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1596075
7. Загальний прибуток від реалізації програмної продукції	Грн.	96040
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	61005
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	0,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	9168
11. Величина економічного ефекту у користувача програмної продукції	Грн.	85822
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

## 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.

Кафедра \_ КБПЗ \_ 2022 рік

					VKPM-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно-обчислювальних машин (ЕОМ). Сьогодні, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки. Адже саме завдяки їм стала можливою швидка переробка величезних обсягів інформації, проведення необхідних розрахунків, виконання різних видів робіт, пов'язаних обробкою текстових та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом.

Стрімке впровадження комп'ютерів не тільки в сфері управління виробництвом, в банківській системі, бізнесі, системі освіти, але також на транспорті, сфері обслуговування призвело до того, що десятки мільйонів людей у всьому світі виявились втягнутими у взаємодію людини з комп'ютером. Природно виникає запитання: настільки безпечною є ця взаємодія для людини? Адже відома аксіома про те, що будь-яка взаємодія людини та засобів праці двостороння.

Людина впливає на удосконалення засобів праці, а останні – на працюючу людину. Отже, навіть сучасні технології та техніка, до яких безперечно, залежать комп'ютерні технології та ЕОМ несуть у собі певні потенційні небезпеки. У зв'язку з цим набуває актуальності адекватна оцінка конкретних умов і характеру праці, яка сприяє обґрунтованому розробленню та впровадженню комплексу заходів і засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці за рахунок поліпшення параметрів виробничого середовища,

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

зменшення важкості, напруженості трудового процесу та збереження здоров'я працівників на комп'ютеризованих робочих місцях.

Законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці на підприємстві при роботі за комп'ютером., зокрема «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», затверджені наказом Мінсоцполітики від 14.02.2018 № 207 [1], «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98. [2].

Загальні вимоги пожежної безпеки під час експлуатації комп'ютерної техніки визначають «Правила пожежної безпеки в Україні» (затверджені наказом МВС від 30.12.2014 № 1417) [3], комп'ютерних класів – пункт 3 розділу VIII «Правил пожежної безпеки для навчальних закладів та установ системи освіти України» (затверджені наказом МОН від 15.08.2016 № 974). [4] та інші державні стандарти, що регламентують експлуатування комп'ютерної техніки як радіоелектронної апаратури.

## 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Можна виділити наступні основні фактори, що впливають на стан здоров'я людей, які працюють за комп'ютером:

- сидяче положення на протязі тривалого періоду;
- вплив електромагнітного випромінювання монітора;
- втома очей, навантаження на зір;
- перевантаження суглобів кистей;
- стрес при втраті інформації.

У кожному з цих випадків ступінь ризику прямо пропорційний часу, що проводиться за комп'ютером і поблизу нього. В сучасних умовах взаємодія людини з технікою значно ускладнилась, що вимагає комплексного підходу, який

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

передбачає розгляд людини, технічних засобів праці та виробничого середовища, як взаємозв'язаних елементів єдиної системи. Все вищесказане в повній мірі відноситься й до системи «людина–комп'ютер–середовище».

Вагомий вплив на працездатність та здоров'я користувачів комп'ютерів здійснює виробниче середовище. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму, випромінювання.

Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам та запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини необхідно здійснити санітарно-гігієнічну характеристику умов працівника, який працює з програмним продуктом.

### **8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК**

Розглянемо приміщення в якому працює користувач ПК з даним програмним продуктом.

Приміщення має одностороннє природне освітлення і загальне штучне освітлення. Стіни і стеля обклеєні світлими шпалерами, підлога вкрита темним ламінатом. У приміщенні відсутні сильні вібрації та шкідливі речовини. Склад повітря в нормі. У кімнаті знаходиться ПК з 4-ядерним процесором і 23-дюймовим IPS монітором, а також меблі.

Приміщення має довжину 4м, ширину 3,5м, висоту стелі 2,7м. Кількість робочих місць – одне. Площа – 14м<sup>2</sup>, об'єм – 37,8м<sup>3</sup>. Виходячи з цього, отримано дані, наведені в таблиці 8.1.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>88</b>

Таблиця 8.1–Фактичні та нормативні значення параметрів приміщення

Параметр	Норма *	Реальні параметри
Площа, S	не менше 6 м <sup>2</sup>	14 м <sup>2</sup>
Об'єм, V	не менше 20 м <sup>3</sup>	37,8 м <sup>3</sup>

\*Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

За даними, які наведено у табл. 8.1, можна зробити висновок, що отримані показники, площа та об'єм приміщення у розрахунку на одно робоче місце користувача ПК відповідає чинним нормам і вимогам.

Щодо мікроклімату, то згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [5] роботу з ПК можна віднести до категорії легка 1а. Джерелами тепла в цьому приміщенні є люди, електроустаткування, освітлювальні прилади в темний час доби і система опалювання взимку. Оператором виділяється до 120ккал теплової енергії за годину. Оптимальні та фактичні значення параметрів мікроклімату приведені в таблиці 8.2.

Таблиця 8.2 – Значення параметрів мікроклімату

Період року	Параметр	Оптимальний*	Фактичний
Теплий	Температура	23 – 25 <sup>0</sup> С	24 <sup>0</sup> С
	Вологість	40 – 60%	50%
	Швидкість повітря	< 0,1м/с	
Холодний	Температура	22 – 24 <sup>0</sup> С	23 <sup>0</sup> С
	Вологість	40 – 60%	55%
	Швидкість повітря	< 0,1м/с	

\*ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

По отриманим замірам параметрів мікроклімату можна зробити висновок, що усі показники задовольняють вимогам зазначеним для робіт категорії легка 1а і є задовільними для здоров'я людини.

Щодо освітлення, то згідно з ДБН В.2.5-28:2006 «Природне і штучне освітлення» [6] ця робота відноситься до Va розряду зорових робіт. Передбачається використання природного, штучного і змішаного освітлення.

Природне освітлення здійснюється за допомогою вікна, площа якого складає  $S' = 1,8 * 1,5 = 2,7 \text{ м}^2$  і є боковим освітленням. У світильниках місцевого і загального освітлення використовуються світлодіодні лампи потужністю 20Вт із світловим потоком лампи 900 лм. Згідно замірів рівень освітлення в даному приміщенні і на робочому місці складає в межах 350 -500 лк, що відповідає нормованому значенню

Джерелом шуму в приміщенні є комп'ютер. Вентилятори (кулери) системного блоку, процесора, відеокарти і блоку живлення є сучасними і мають низький рівень шуму. Згідно з технічною документацією шум, зумовлений кулером в блоці живлення складає 25 дБ, кулером процесора – 30 дБ, загальний - 34 дБ. Враховуючи незначний рівень шуму від персонального комп'ютера і незначний рівень фонового шуму від іншого устаткування, можна стверджувати, що сумарний рівень шумового забруднення приміщення не перевищує максимально допустимий рівень коригованої звукової потужності і складає не більше 50 дБА, що відповідає рівню шуму для приміщень з комп'ютерною технікою згідно Державних санітарних правил і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

У приміщенні відсутні джерела інфрачервоного, ультрафіолетового і електромагнітного випромінювання, бо монітор ПК вироблений на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновією лампою, що не має сильного електромагнітного випромінювання і сертифіковані в Україні.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Блок живлення є екранованим і не випускає вищезазначених видів випромінювання.

#### **8.4 Розробка заходів з умов поліпшення охорони праці**

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці користувача ПК.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату, рівня звуку і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості і рівня звуку, проведених відділом охорони праці.

Крім рекомендацій щодо конкретного приміщення, де було проведено дослідження умов праці, існують загальні вимоги, які зарекомендовані відповідними нормативними документами.

Правильна організація робочих місць запобігає передчасній втомлюваності користувача і сприяє збереженню здоров'я. Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- вибір ергономічного об'єднаного робочого положення, виробничих меблів з урахуванням характеристик людини;
- раціональне компонування обладнання на робочих місцях;

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

- урахування характеру й особливостей трудової діяльності. Стосовно робочих місць користувача ВДТ, то організація робочого має забезпечуватися відповідно до ДСанПіН 3.3.2-007-98. Для запобігання перевтомленню необхідно виконувати вправи для очей та дотримуватись розпорядку роботи та відпочинку. На робочому місці реалізовувався режим відпочинку: кожні дві години – перерва для виконання фізичних вправ для м'язів очей.

### 8.5 Протипожежний захист

Пожежі в приміщеннях з оргтехнікою становлять особливу небезпеку, бо поєднані з великими матеріальними збитками. Пожежа може виникнути при взаємодії горючих речовин і джерел запалювання. Горючими речовинами є будівельні та опоряджувальні матеріали, пластмасові корпуси техніки, шнури тощо. Джерелами запалювання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв електроживлення, де внаслідок різних порушень виникає перегрівання елементів, утворюються електричні іскри та дуги, здатні спричинити займання горючих матеріалів.

При обслуговуванні, ремонтних та профілактичних роботах використовуються різні легкозаймисті рідини, прокладаються тимчасові електропровідники, здійснюється паяння. Виникає додаткова пожежна небезпека, яка потребує відповідних заходів пожежного захисту. До засобів гасіння пожежі, призначених для локалізації невеликих займань, належать вогнегасники, сухий пісок, азбестові ковдри. Приміщення, в який встановлено комп'ютери і де немає необхідності влаштування систем автоматичного пожежогасіння, необхідно оснащувати переносними вуглекислотними з розрахунку 2 шт. на кожні 20 м<sup>2</sup> в приміщеннях. Звуковбирне облицювання стін, стель приміщень треба виконувати з негорючих та важко горючих матеріалів.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

З метою виявлення початкової стадії займання необхідно використовувати пристрої систем автоматичного пожежогасіння там, де цього вимагають Правила пожежної безпеки.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

## 8.6 Розрахункова частина

В приміщенні (де відсутні джерела виділення шкідливих речовин) працює одна людина. Робота пов'язана з використанням ПЕОМ. Розміри приміщення:  $A = 4$  м,  $B = 3,5$  м,  $H = 2,8$  м, устаткування займає 15% об'єму. Визначити найменшу необхідну кількість повітря для вентиляції.

Для приміщень, в яких відсутні виділення шкідливих речовин у повітрі, розрахунок вентиляції здійснюється залежно від кількості працюючих.

Необхідна кількість повітря ( $\text{м}^3$  /год.), яка забезпечує відповідність параметрів повітря робочої зони нормованим значенням, визначається за наступною формулою:

$$L = L' \cdot N, \quad (8.1)$$

де  $L'$  – нормативна кількість повітря на одного працюючого, яка залежить від питомого об'єму приміщення,  $\text{м}^3$  / (год.-люд.);

$N$  – кількість працюючих.

Питомий об'єм приміщення  $V_p$ , ( $\text{м}^3$  /люд.), визначається за формулою:

$$V_p = V/N, \quad (8.2)$$

де  $V$  – об'єм приміщення,  $\text{м}^3$ .

Визначаємо вільний об'єм приміщення

$$V = A \cdot B \cdot H \cdot 0,85 = 4 \cdot 3,5 \cdot 2,8 \cdot 0,85 = 33,3 \text{ м}^3.$$

Питомий вільний об'єм складає

$$V' = V / N = 33,3 / 1 = 33,3 \text{ м}^3 / \text{люд.} < 20 \text{ м}^3 / \text{люд.}$$

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

Нормована кількість повітря на одну людину при  $V' < 20 \text{ м}^3 / \text{люд.}$  становить  $30 \text{ м}^3 / (\text{год.} \cdot \text{люд.})$ .

### 8.7 Висновки до розділу

У даному розділі магістерської роботи проведено аналіз умов працівника робота якого пов'язана з комп'ютерною технікою. Проведено аналіз основних санітарно-гігієнічних показників в заданому приміщенні, де працівник зайнятий постійною роботою за комп'ютером.. Створені умови повинні забезпечувати комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано найменшу необхідну кількість повітря для вентиляції.

Дотримання умов, що визначають оптимальну організацію робочих місць працівників, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відносінах продуктивність їх праці.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>94</b>

## 9 ОСНОВНІ ВИСНОВКИ

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

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 мікроклімату складського комплексу.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем мікроклімату складського комплексу.
- Досліджена система мікроклімату складського комплексу.
- На основі отриманих результатів досліджень створена програмна реалізація системи мікроклімату складського комплексу.

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

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.

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

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

Програма реалізована на мові високого рівня Delphi 10.4 Sydney. 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 TEA.

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 85822 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

					ВКРМ-123.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Авраменко К.Д. Дослідження та програмна реалізація системи мікроклімату складського комплексу // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.
2. Клюев В.В. Неразрушающий контроль и диагностика. Справочник, 2-е изд., перераб. и доп. / В.В. Клюев – М: Машиностроение, 2003. – 656 с.
3. Клюня В.Л. Основы экономической теории / В.Л. Клюня, Н.В. Черченко. – Минск: Минск, 2006. – 238 с.
4. 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.
5. 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.
6. 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.
7. 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.
8. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко,

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов,

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.

16. Коваленко А.С. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.-

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

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

23. 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.

24. 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.

25. 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.

26. 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.

27. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

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

29. 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.

30. 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.

31. 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.

32. 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. – Вып. 8 (48). – С. 51-54.

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

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

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

35. Лазарев А.А. Выбор показателя затрат для анализа сравнительной экономической эффективности техники конечного потребления / А.А. Лазарев, М.В. Бейлин // Сборник научных трудов ХГПУ.– Х.: ХГПУ, 1999. – Вып. 74. – С. 27-29.

36. 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.

37. 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.

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

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

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

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

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

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

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

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

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

46. 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.

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

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

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

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

51. 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.

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

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-

					<b>ВКРМ-123.22.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103



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

					<b>ВКРМ-123.22.0001.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Авраменко К.Д.				Дослідження та програмна реалізація системи мікроклімату складського комплексу	Літ.	Аркуш	Аркушів
Перевірів	Петренко В.І.					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 мікроклімату складського комплексу.

## 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 мікроклімату складського комплексу.

## 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-123.22.0001.00.00.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 мікроклімату складського комплексу;
- 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.

					<b>ВКРМ-123.22.0001.00.00.ТЗ</b>	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 Sydney.

					ВКРМ-123.22.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

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

## 7 Економічні вимоги

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

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

## 8 Вимоги щодо охорони праці

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

					ВКРМ-123.22.0001.00.00.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 – 104 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 19.12.2022 р.

					<b>ВКРМ-123.22.0001.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

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

ЗАТВЕРДЖУЮ

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

\_\_\_\_\_ Петренюк В.І.

*Дослідження та програмна реалізація  
системи мікроклімату складського комплексу*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 32

Літера: РП

Кропивницький – 2022 року

## Головний файл проекту

```
program LGZTT-500;

{2022 рік
Розробив: студент Авраменко Кирило Дмитрович
Тема: Дослідження та програмна реалізація системи мікроклімату складського
комплексу}

uses
  Forms,
  UN_1 in 'UN_1.pas' {UN_1},
  Visible in 'Visible.pas' {U_Visible},
  TLib in 'TLib.pas' {TLib_1},
  Data in 'Data.pas' {Data_LGZTT-500},
  ABOUT in 'ABOUT.pas' {UN_ABOUT},

{$R *.res}

begin
  Application.Initialize;
  Application.Title:=' LGZTT-500';
  Application.CreateForm(T UN, UN_1);
  Application.CreateForm(T Visible, U_Visible);
  Application.CreateForm(TLib, TLib_1);
  Application.CreateForm(Tdata, Data_LGZTT-500);
  Application.CreateForm(TUN_ABOUT, UN_ABOUT);
  Application.Run;
end.
```

## Файл проекту UN\_1.pas

```
unit UN_1;

{2022 рік
Розробив: студент Авраменко Кирило Дмитрович
Тема: Дослідження та програмна реалізація системи мікроклімату складського
комплексу}

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, ComCtrls, Menus, StdCtrls, jpeg, RXCtrls;

type

  TMbcsByteType = (mbSingleByte, mbLeadByte, mbTrailByte);

  TEraRange = record
    StartDate: Integer;
    EndDate: Integer;
  end;

  TRS485 = packed record
    DSR, CTS, RING, RLSD: boolean;
  end;

  TForm1 = class(TForm)
    Panel1: TPanel;
    OpenDialog1: TOpenDialog;
    Panel2: TPanel;
    StatusBar1: TStatusBar;
    Panel3: TPanel;
    Panel4: TPanel;
    MainMenu1: TMainMenu;
    N11: TMenuItem;
    N21: TMenuItem;
    N31: TMenuItem;
    N41: TMenuItem;
    Animate1: TAnimate;
    Panel5: TPanel;
    Image1: TImage;
    Panel6: TPanel;
    Image2: TImage;
    Label1: TLabel;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    SaveDialog1: TSaveDialog;
    PopupMenu1: TPopupMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
```

```

RichEdit1:TRichEdit;
Button6:TButton;
N4:TMenuItem;
N5:TMenuItem;
N6:TMenuItem;
N7:TMenuItem;
N8:TMenuItem;
RS:TMenuItem;
N9:TMenuItem;
N10:TMenuItem;
Timer1:TTimer;
N12:TMenuItem;
N13:TMenuItem;
Image3:TImage;
Button7:TButton;
Panel7:TPanel;
SecretPanell:TSecretPanel;
procedure N3Click(Sender:TObject);
procedure N1Click(Sender:TObject);
procedure Button2Click(Sender:TObject);
procedure Button1Click(Sender:TObject);
procedure Button3Click(Sender:TObject);
procedure Button4Click(Sender:TObject);
procedure FormCreate(Sender:TObject);
procedure Button6Click(Sender:TObject);
procedure Button5Click(Sender:TObject);
procedure N12Click(Sender:TObject);
procedure Timer1Timer(Sender:TObject);
procedure Button7Click(Sender:TObject);
private
  _TRANSFER:string;
public
  procedure _TEST;
  function ByteType(const S:string; Index:Integer):TMbcsByteType;
  procedure Add_log(s:string);
  function GetProcAddress(Module:HMODULE; Proc:PChar):Pointer;
  function GetPackageModuleHandle(PackageName:PChar):HMODULE;
  function StrByteType(Str:PChar; Index:Cardinal):TMbcsByteType;
  procedure COPYtoRS(A:string;B:string);
  function GetModuleHandle(ModuleName:PChar):HMODULE;
  procedure AUTO(A:integer);
  function LoadLibrary(ModuleName:PChar):HMODULE;
  procedure CLEAR(A:integer);
  function FreeLibrary(Module:HMODULE):LongBool;
  procedure OutOfMemoryError;
published
  property Align;      property Anchors;
  property BiDiMode;  property Constraints;
  property DockSite;  property DragCursor;
  property DragKind;  property DragMode;
  property Enabled;   property Font;
  property HotTrack;  property Images;
  property MultiLine; property MultiSelect;
  property OwnerDraw; property ParentBiDiMode;
  property ParentFont;property ParentShowHint;
  property PopupMenu; property RaggedRight;
  property ScrollOpposite;

```

```

property ShowHint; property Style;
property TabHeight; property TabOrder;
property TabPosition;property Tabs;
property OnDockOver;property OnDragDrop;
property OnDragOver;property OnDrawTab;
property OnEndDock; property OnEndDrag;
property OnEnter; property OnExit;
property OnGetImageIndex;
property OnGetSiteInfo;
property OnMouseDown;
property OnMouseMove;
property OnMouseUp; property OnResize;
property OnStartDock;
property OnStartDrag;
property OnUnDock;
end;

TDrawTabEvent = procedure(Control:TCustomTabControl; TabIndex:Integer;
    const Rect:TRect; Active:Boolean) of object;
TTabGetImageEvent = procedure(Sender:TObject; TabIndex:Integer;
    var ImageIndex:Integer) of object;
TLangRec = packed record
    FName:string;
    FLCID:LCID;
    FExt:string;
end;

TSysLocale = packed record
    DefaultLCID:Integer;
    PriLangID:Integer;
    SubLangID:Integer;
    FarEast:Boolean;
    MiddleEast:Boolean;
end;

var
    cId:THandle;
    DCB:TDCB;
    TimeOuts:TCommTimeouts;
    Stat:TComStat;
    RS485:TRS485;
    RS232:TR232;
    RecivBuff:array[0..255] of byte;
    CntByte:integer;
    Terminated:boolean;
    Form1:TForm1;
implementation

uses Unit2, Unit3;

{$R *.dfm}

function InitsCommRS232_485 (Num:integer):boolean;
var
    ThreadId:Dword;
begin
    Result:=False;

```

```

cId:=CreateFile(PChar('\\.\COM'+ IntToStr(Num),
GENERIC_READ or GENERIC_WRITE,0,nil,OPEN_EXISTING,
FILE_FLAG_OVERLAPPED,0);
if cId = INVALID_HANDLE_VALUE then Exit;
if not (SetCommMask(cId,EV_RXCHAR or EV_TXEMPTY) and
SetupComm(cId,256,256) and
PurgeComm(cId,PURGE_TXABORT or PURGE_RXABORT or
PURGE_TXCLEAR or PURGE_RXCLEAR) and
GetCommState(cId,DCB))
Then
begin
  CloseHandle(cId);
  Exit;
end;
DCB.BaudRate:=9599;
DCB.ByteSize:=8;
DCB.Parity:=NoParity;
DCB.StopBits:=OneStopBit;
if not SetCommState(cId,DCB) then begin
  CloseHandle(cId);
  Exit;
end;
GetCommTimeouts(cId,TimeOuts);
TimeOuts.ReadIntervalTimeout:= MAXDWORD;
TimeOuts.ReadTotalTimeoutMultiplier:= 0;
TimeOuts.ReadTotalTimeoutConstant:= 0;
TimeOuts.WriteTotalTimeoutMultiplier:= 0;
TimeOuts.WriteTotalTimeoutConstant:= 0;
if not SetCommTimeouts(cId,TimeOuts) then begin
  CloseHandle(cId);
  Exit;
end;
Terminated:=False;
CommThread:=CreateThread(nil,0,@ReadsComm,nil,0,ThreadID);
  if CommThread = 0 then begin
    CloseHandle(cId);
    Exit;
  end;
  SetThreadPriority(CommThread,8);
end;
Result:=True;
end;

function ReadsComm:longint; stdcall;
var
  Ovr:TOverlapped;
  Events:array[0..1] of THandle;
  Buff:array[0..255] of byte;
  Trans, Signal, Mask, Kols, ModemState:DWord;
  i:integer;
  Reciv:pointer;
begin
  Result:=0;
  FillChar(Ovr,SizeOf(TOverlapped),0);
  Ovr.hEvent:=CreateEvent(nil,TRUE,FALSE,#0);
  Events[0]:=Ovr.hEvent;
  while not Terminated do begin

```

```

Mask:=0;
if not WaitCommEvent(cId,Mask,@Ovr) then begin
if ERROR_IO_PENDING = GetLastError() then begin
if WaitForSingleObject(Ovr.hEvents,INFINITE)= WAIT_OBJECT_0 then
GetOverlappedResult(cId,Ovr,Trans,False);
end;
end;
if Terminated then break;
ClearCommError(cId,Errs,@Stat);
If (Mask and EV_RXCHAR) = EV_RXCHAR then begin
if Terminated then break;
ReadFile(cId,Buff,256,Kols,@Ovr);
if Terminated then break;
if Kols > 0 then begin
Move(Buff[0],RecvBuff[CntByte],Kols);
Inc(CntByte,Kols);
if CntByte > 48 then begin
GetMem(Recv,sizeof(CntByte));
Move(RecvBuff[0],Byte(Recv^),CntByte);
PostMessage(Handle,cmRxByte,Integer(Recv),Kols);
CntByte:=0;
if ((Mask and EV_DSR) = EV_DSR) or
((Mask and EV_CTS) = EV_CTS) or
((FMask and EV_RING) = EV_RING) or
((FMask and EV_RLSD) = EV_RLSD) then begin
if Terminated then break;
GetCommRSStatus(cId,ModemState);
GetMem(Recv,sizeof(TModems));
TRS(Recv^).DSR:=(RSState and MS_DSR_ON) = MS_DSR_ON;
TRS(Recv^).CTS:=(RSState and MS_CTS_ON) =MS_CTS_ON;
TRS(Recv^).RING:=(RSState and MS_RING_ON) =MS_RING_ON;
TRS(Recv^).RLSD:=(RSState and MS_RLSD_ON) =MS_RLSD_ON;
PostMessage(Handle,cmRS,Integer(Recv),0);
end;
end;
CloseHandle(Ovr.hEvent);
end;

function WriteComm(hComm:THandle; Buff:array of byte; Count:integer;
Wait:boolean):integer;
var
Ovr:TOverlapped;
Code:DWord;
S:string;
begin
FillChar(Ovr,SizeOf(TOverlapped),0);
Ovr.hEvent:=CreateEvent(nil,TRUE,FALSE,#0);
if not WriteFile(hComm,Buff,Count,Result,@Ovr) and Wait
then begin
if (GetLastError() = ERROR_IO_PENDING) and
(WaitForSingleObject(Ovr.hEvent,INFINITE)= WAIT_OBJECT_0)
then GetOverlappedResult(cId,Ovr,Result,False);
end;
CloseHandle(Ovr.hEvent);
end;

function CloseComm:boolean;

```

```

begin
Terminated:=True;
  SetCommMask(cId,0);
  Sleep(10);
  CloseHandle(cId);
  Result:=True;
end;

function TForm1.GetPackageModuleHandle(PackageName:PChar):HMODULE;
var
  PkgName:array[0..MAX_PATH] of Char;

function CheckPackageName(linkmap:plink_map):Boolean;
var
  Name1:PChar;
begin
  Result:=True;
  if linkmap.l_name <> nil then
  begin
  // START
  Name1:=StrRScan(linkmap.l_name, PathDelim);
  if Name1 = nil then
  Name1:=linkmap.l_name
  else Inc(Name1);
    Result:=StrPos(Name1, PkgName) = nil;
    end;
  end;
  procedure MakePkgName(Prefix, Name:PChar);
  begin
    StrCopy(PkgName, Prefix);
    StrLCat(PkgName, Name, sizeof(PkgName)-1);
    PkgName[High(PkgName)]:=#0;
  end;
  begin
  if (PackageName = nil) or (StrScan(PackageName, PathDelim) <> nil) then
    Result:=0
  else
  begin
    MakePkgName('bpl', PackageName);
    Result:=InitModule(ScanLinkMap(@CheckPackageName));
    if Result = 0 then
    begin
      MakePkgName('dcl', PackageName);
      Result:=InitModule(ScanLinkMap(@CheckPackageName));
      if Result = 0 then
      begin
        MakePkgName('', PackageName);
        Result:=InitModule(ScanLinkMap(@CheckPackageName));
      end;
    end;
  end;
end;

function TForm1.GetModuleHandle(ModuleName:PChar):HMODULE;

function CheckModuleName(linkmap:plink_map):Boolean;
var

```

```

    Name1:PChar;
begin
    Result:=True;
    if ((ModuleName = nil) and ((linkmap.l_name = nil) or (linkmap.l_name[0] =
#0))) or
        ((ModuleName[0] = PathDelim) and (StrComp(ModuleName, linkmap.l_name) =
0)) then
    begin
        Result:=False;
        Exit;
    end else
    begin
        Name1:=StrRScan(linkmap.l_name, PathDelim);
        if Name1 = nil then
            Name1:=linkmap.l_name
        else Inc(Name1);
        if StrComp(ModuleName, Name1) = 0 then
        begin
            Result:=False;
            Exit;
        end;
    end;
end;
begin
    Result:=InitModule(ScanLinkMap(@CheckModuleName));
end;

procedure TForm1.N3Click(Sender:TObject);
begin
    Form1.Close;
end;

procedure TForm1.N1Click(Sender:TObject);
begin
    if SaveDialog1.Execute then
    begin
        RichEdit1.Lines.SaveToFile(SaveDialog1.FileName);
    end;
end;

procedure TForm1.Button2Click(Sender:TObject);
begin
    Form1.Hide;
    AboutBox.show;
end;

procedure TForm1.Button1Click(Sender:TObject);
begin
    Form1.close;
end;

procedure TForm1.Button3Click(Sender:TObject);
begin
    Application.ProcessMessages;
end;

procedure TForm1.Button4Click(Sender:TObject);

```

```
begin
    if OpenFileDialog1.Execute then
        begin
            _TRANSFER:=OpenDialog1.FileName;
        end;
    end;

procedure TForm1.Add_log(s:string);
begin
    RichEdit1.Lines.Add(datetimetostr(now)+' '+s);
    RichEdit1.Repaint;
end;

procedure TForm1.Button6Click(Sender:TObject);
begin
    if _TRANSFER<>' ' then
        begin
            if SaveDialog1.Execute then
                begin
                    COPYtoRS(_TRANSFER,SaveDialog1.FileName);
                end;
        end;
    end;

procedure TForm1.COPYtoRS(A, B:string);
var
    ass:boolean;
begin
    dranzaction(true);
    application.ProcessMessages;
    Animat1.Active:=true;
    Animat1.Refresh;
    sleep(2500);
    Animat1.Active:=false;
end;

procedure TForm1.Button5Click(Sender:TObject);
begin
    Application.ProcessMessages;
end;

procedure TForm1.N12Click(Sender:TObject);
begin
    RichEdit1.Lines.Clear;
end;

procedure TForm1.Timer1Timer(Sender:TObject);
begin
    _TEST;
end;

procedure TForm1.Button7Click(Sender:TObject);
begin
    FORM3.SHOW;
    Form1.Hide;
end;
```

```

function TForm1.LoadLibrary(ModuleName:PChar):HMODULE;
begin
    Result:=HMODULE(dlopen(ModuleName, RTLD_LAZY));
end;

function TForm1.FreeLibrary(Module:HMODULE):LongBool;
begin
    Result:=LongBool(dlclose(Pointer(Module)));
end;

function TForm1.GetProcAddress(Module:HMODULE; Proc:PChar):Pointer;
var
    Info:TDLInfo;
    Error:PChar;
    ModHandle:HMODULE;
begin
    dlerror;
    Result:=dlsym(Pointer(Module), Proc);
    Error:=dlerror;
    if Error <> nil then
        Result:=nil
    else if dladdr(Result, Info) <> 0 then
        begin
            if Info.dli_fbase = ExeBaseAddress then
                Info.dli_fname:=nil;
            ModHandle:=HMODULE(dlopen(Info.dli_fname, RTLD_LAZY));
            if ModHandle <> 0 then
                begin
                    dlclose(Pointer(ModHandle));
                    if ModHandle <> Module then
                        Result:=nil;
                    end;
                end else Result:=nil;
            end;
        end;
    type
        plink_map = ^link_map;
        link_map = record
            l_addr:Pointer;
            l_name:PChar;
            l_ld:Pointer;
            l_next, l_prev:plink_map;
        end;
        pr_debug = ^r_debug;
        r_debug = record
            r_version:Integer;
            r_map:plink_map;
            r_brk:Pointer;
            r_state:Integer;
            r_ldbase:Pointer;
        end;
    var
        _r_debug:pr_debug = nil;
    function ScanLinkMap(Func:Pointer):plink_map;
    var
        linkmap:plink_map;
        function Eval(linkmap:plink_map; Func:Pointer):Boolean;
        asm

```

```

        MOV     ECX, [EBP]
        PUSH   EBP
        CALL   EDX
        POP    ECX

    end;
begin
    if _r_debug = nil then
        _r_debug:=dlsym(RTLD_DEFAULT, '_r_debug');
    if _r_debug = nil then
        begin
            Assert(False, 'Unable to locate ''_r_debug'' symbol');
            Result:=nil;
            Exit;
        end;
    linkmap:=_r_debug.r_map;
    while linkmap <> nil do
        begin
            if not Eval(linkmap, Func) then Break;
            linkmap:=linkmap.l_next;
        end;
        Result:=linkmap;
    end;

function InitModule(linkmap:plink_map):HMODULE;
begin
    if linkmap <> nil then
        begin
            Result:=HMODULE(dlopen(linkmap.l_name, RTLD_LAZY));
            if Result <> 0 then
                dlclose(Pointer(Result));
            end else Result:=0;
        end;
    end;

function GetModuleHandle(ModuleName:PChar):HMODULE;

    function CheckModuleName(linkmap:plink_map):Boolean;
    var
        Name1:PChar;
    begin
        Result:=True;
        if ((ModuleName = nil) and ((linkmap.l_name = nil) or (linkmap.l_name[0] = #0)))
        or ((ModuleName[0] = PathDelim) and (StrComp(ModuleName, linkmap.l_name) = 0))
        then
            begin
                Result:=False;
                Exit;
            end else
            begin
                Name1:=StrRScan(linkmap.l_name, PathDelim);
                if Name1 = nil then
                    Name1:=linkmap.l_name
                else Inc(Name1);
                if StrComp(ModuleName, Name1) = 0 then
                    begin
                        Result:=False;
                        Exit;
                    end;
            end;
        end;
end;

```

```
        end;
    end;

begin
    Result:=InitModule(ScanLinkMap(@CheckModuleName));
end;

procedure TForm1.OutOfMemoryError;
begin
    raise OutOfMemory;
end;

function TForm1.ByteType(const S:string; Index:Integer):TMbcsByteType;
begin
    Result:=mbSingleByte;
    if SysLocale.FarEast then
        Result:=ByteTypeTest(PChar(S), Index-1);
end;

function TForm1.StrByteType(Str:PChar; Index:Cardinal):TMbcsByteType;
begin
    Result:=mbSingleByte;
    if SysLocale.FarEast then
        Result:=ByteTypeTest(Str, Index);
end;

end.
```

Кафедра \_ КБПЗ \_ 2022 рік

## Файл проекту Visible.pas

```
unit UN_Visible;

{2022 рік
Розробив: студент Авраменко Кирило Дмитрович
Тема: Дослідження та програмна реалізація системи мікроклімату складського комплексу }

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Image1: TImage;
    Label1: TLabel;
    Panel3: TPanel;
    Image2: TImage;
    Image3: TImage;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Image1Click(Sender: TObject);
  private
    FTracking: Boolean;
    FParentColor: Boolean;
    FVisible: Boolean;
    FSmooth: Boolean;
    FButtonSize: Integer;
    FRange: Integer;
    FSize: Integer;
    FPosition: Integer;
    FThumbSize: Integer;
    FColor: TColor;
    FIncrement: TScrollBarInc;
    FStyle: TScrollBarStyle;
    FMargin: Word;
    function IsIncrementStored: Boolean;
    function IsRangeStored: Boolean;
    procedure SetButtonSize(const Value: Integer);
    procedure SetColor(const Value: TColor);
    procedure SetParentColor(const Value: Boolean);
    procedure SetPosition(const Value: Integer);
    procedure SetRange(const Value: Integer);
    procedure SetSize(const Value: Integer);
    procedure SetStyle(const Value: TScrollBarStyle);
    procedure SetThumbSize(const Value: Integer);
    procedure SetVisible(const Value: Boolean);
    { Private declarations }
  public
    published
    property ButtonSize: Integer read FButtonSize write SetButtonSize default 0;
    property Color: TColor read FColor write SetColor default clBtnHighlight;
    property Increment: TScrollBarInc read FIncrement write FIncrement stored
      IsIncrementStored default 8;
```

```
property Margin:Word read FMargin write FMargin default 0;
property ParentColor:Boolean read FParentColor write SetParentColor default
True;
property Position:Integer read FPosition write SetPosition default 0;
property Range:Integer read FRange write SetRange stored IsRangeStored default
0;
property Smooth:Boolean read FSmooth write FSmooth default False;
property Size:Integer read FSize write SetSize default 0;
property Style:TScrollBarStyle read FStyle write SetStyle default ssRegular;
property ThumbSize:Integer read FThumbSize write SetThumbSize default 0;
property Tracking:Boolean read FTracking write FTracking default False;
  property Visible:Boolean read FVisible write SetVisible default True;
end;

var
  Form3:TForm3;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm3.FormClose(Sender:TObject; var Action:TCloseAction);
begin
  Form3.hide;
  Form1.show;
end;
procedure TForm3.Image1Click(Sender:TObject);
begin
  form3.Hide;
  form1.Show;
end;

procedure TForm3.SetButtonSize(const Value:Integer);
begin
  FButtonSize:=Value;
end;

procedure TForm3.SetColor(const Value:TColor);
begin
  FColor:=Value;
end;

procedure TForm3.SetParentColor(const Value:Boolean);
begin
  FParentColor:=Value;
end;

procedure TForm3.SetPosition(const Value:Integer);
begin
  FPosition:=Value;
end;

procedure TForm3.SetRange(const Value:Integer);
begin
  FRange:=Value;
```

```
end;
```

```
procedure TForm3.SetSize(const Value:Integer);  
begin  
    FSize:=Value;  
end;
```

```
procedure TForm3.SetStyle(const Value:TScrollBarStyle);  
begin  
    FStyle:=Value;  
end;
```

```
procedure TForm3.SetThumbSize(const Value:Integer);  
begin  
    FThumbSize:=Value;  
end;
```

```
procedure TForm3.SetVisible(const Value:Boolean);  
begin  
    FVisible:=Value;  
end;  
end.
```

Кафедра \_ КБПЗ \_ 2022 рік

## Файл проекту Data.pas

```

unit Data_LGZTT-500;

{2022 рік
Розробив: студент Авраменко Кирило Дмитрович
Тема: Дослідження та програмна реалізація системи мікроклімату складського комплексу }

interface

uses Windows, Variants, ActiveX, Classes, Midas, Provider, SysUtils;

type

TDataModule = class(TDataModule, IAppServer)
private
    FProviders:TList;
    FLock:TRTLCriticalSection;
    function GetProviderCount:integer;
protected
    function GetProvider(const ProviderName:string):TCustomProvider; virtual;
    class procedure Updateistry(Register:Boolean; const ClassID, ProgID:string);
    override;
    function AS_GetProviderNames:OleVariant; safecall;
    function AS_ApplyUpdates(const ProviderName:WideString; Delta:OleVariant;
        MaxErrors:Integer; out ErrorCount:Integer;
        var OwnerData:OleVariant):OleVariant; safecall;
function AS_GetRecords(const ProviderName:WideString; Count:Integer;
out RecsOut:Integer; Options:Integer; const CommandText:WideString;
var Params, OwnerData:OleVariant):OleVariant; safecall;
function AS_DataRequest(const ProviderName:WideString;
Data:OleVariant):OleVariant; safecall;
    function AS_GetParams(const ProviderName:WideString; var
OwnerData:OleVariant):OleVariant; safecall;
    function AS_RowRequest(const ProviderName:WideString; Row:OleVariant;
        RequestType:Integer; var OwnerData:OleVariant):OleVariant; safecall;
    procedure AS_Execute(const ProviderName:WideString;
        const CommandText:WideString; var Params, OwnerData:OleVariant); safecall;
public
    constructor Create(AOwner:TComponent); override;
    destructor Destroy; override;
    procedure RegisterProvider(Value:TCustomProvider); virtual;
    procedure UnRegisterProvider(Value:TCustomProvider); virtual;
    procedure Lock; virtual;
    procedure Unlock; virtual;
    property Providers[const ProviderName:string]:TCustomProvider read
    GetProvider;
    property ProviderCount:integer read GetProviderCount;
end;

type
TRemoteDataModule = class(TDataModule)
private
    FProviders:TList;
    FLock:TRTLCriticalSection;
    function GetProviderCount:integer;

```

```

public
  constructor Create(AOwner:TComponent); override;
  destructor Destroy; override;
  procedure RegisterProvider(Value:TCustomProvider); virtual;
  procedure UnRegisterProvider(Value:TCustomProvider); virtual;
  procedure Lock; virtual;
  procedure UnLock; virtual;
  function CRDMGetProviderNames:OleVariant;
  function GetProvider(const ProviderName:string):TCustomProvider; virtual;
  property Providers[const ProviderName:string]:TCustomProvider read
    GetProvider;
  property ProviderCount:integer read GetProviderCount;
end;

procedure RegisterPooled(const ClassID:string; Max, Timeout:Integer;
Singleton:Boolean = False);
procedure UnregisterPooled(const ClassID:string);
procedure EnableTransport(const ClassID:string);
procedure DisableTransport(const ClassID:string);
procedure EnableWebTransport(const ClassID:string);
procedure DisableWebTransport(const ClassID:string);

implementation

uses ComObj, MidConst;

procedure RegisterPooled(const ClassID:string; Max, Timeout:Integer;
Singleton:Boolean = False);
begin
  CreateKey(SClsid + ClassID, SPooled, SFlagOn);
  CreateKey(SClsid + ClassID, SMaxObjects, IntToStr(Max));
  CreateKey(SClsid + ClassID, STimeout, IntToStr(Timeout));
  if Singleton then
    CreateKey(SClsid + ClassID, SSingleton, SFlagOn) else
    CreateKey(SClsid + ClassID, SSingleton, SFlagOff);
end;

procedure DeleteValue(const Key, ValueName:string);
var
  Handle:HKey;
  Status:Integer;
begin
  Status:=RegOpenKey(HKEY_CLASSES_ROOT, PChar(Key), Handle);
  if Status = 0 then
    begin
      RegDeleteValue(Handle, PChar(ValueName));
      RegCloseKey(Handle);
    end;
end;

procedure UnregisterPooled(const ClassID:string);
begin
  DeleteValue(SClsid + ClassID, SPooled);
  DeleteValue(SClsid + ClassID, SMaxObjects);
  DeleteValue(SClsid + ClassID, STimeout);
  DeleteValue(SClsid + ClassID, SSingleton);
end;

```

```
procedure EnableTransport(const ClassID:string);
begin
  CreateKey(SClsid + ClassID, Sockets, SFlagOn);
end;

procedure DisableTransport(const ClassID:string);
begin
  DeleteValue(SClsid + ClassID, Sockets);
end;

procedure EnableWebTransport(const ClassID:string);
begin
  CreateKey(SClsid + ClassID, SWeb, SFlagOn);
end;

procedure DisableWebTransport(const ClassID:string);
begin
  DeleteValue(SClsid + ClassID, SWeb);
end;

constructor TDataModule.Create(AOwner:TComponent);
begin
  InitializeCriticalSection(FLock);
  FProviders:=TList.Create;
  inherited Create(AOwner);
end;

destructor TDataModule.Destroy;
begin
  inherited Destroy;
  FProviders.Free;
  DeleteCriticalSection(FLock);
end;

procedure TDataModule.Lock;
begin
  EnterCriticalSection(FLock);
end;

procedure TDataModule.Unlock;
begin
  LeaveCriticalSection(FLock);
end;

procedure TDataModule.RegisterProvider(Value:TCustomProvider);
begin
  FProviders.Add(Value);
end;

procedure TDataModule.UnRegisterProvider(Value:TCustomProvider);
begin
  FProviders.Remove(Value);
end;

function TDataModule.GetProvider(const ProviderName:string):TCustomProvider;
var
```

```

    i:Integer;
begin
    Result:=nil;
    for i:=0 to FProviders.Count - 1 do
        if AnsiCompareText(TCustomProvider(FProviders[i]).Name, ProviderName) = 0
        then
            begin
                Result:=TCustomProvider(FProviders[i]);
                if not Result.Exported then
                    Result:=nil;
                Exit;
            end;
        if not Assigned(Result) then
            raise Exception.CreateResFmt(@SProviderNotExported, [ProviderName]);
        end;

function TDataModule.AS_GetProviderNames:OleVariant;
var
    List:TStringList;
    i:Integer;
begin
    Lock;
    try
        List:=TStringList.Create;
        try
            for i:=0 to FProviders.Count - 1 do
                if TCustomProvider(FProviders[i]).Exported then
                    List.Add(TCustomProvider(FProviders[i]).Name);
            List.Sort;
            Result:=VarArrayFromStrings(List);
        finally
            List.Free;
        end;
    finally
        UnLock;
    end;
end;

function TDataModule.AS_ApplyUpdates(const ProviderName:WideString;
    Delta:OleVariant; MaxErrors:Integer; out ErrorCount:Integer;
    var OwnerData:OleVariant):OleVariant;
begin
    Lock;
    try
        Result:=Providers[ProviderName].ApplyUpdates(Delta, MaxErrors, ErrorCount,
        OwnerData);
    finally
        UnLock;
    end;
end;

function TDataModule.AS_GetRecords(const ProviderName:WideString; Count:Integer;
    out RecsOut:Integer; Options:Integer; const CommandText:WideString;
    var Params, OwnerData:OleVariant):OleVariant;
begin
    Lock;
    try

```

```

        Result:=Providers[ProviderName].GetRecords(Count, RecsOut, Options,
            CommandText, Params, OwnerData);
    finally
        UnLock;
    end;
end;

function TDataModule.AS_RowRequest(const ProviderName:WideString;
    Row:OleVariant; RequestType:Integer; var OwnerData:OleVariant):OleVariant;
begin
    Lock;
    try
        Result:=Providers[ProviderName].RowRequest(Row, RequestType, OwnerData);
    finally
        UnLock;
    end;
end;

function TDataModule.AS_DataRequest(const ProviderName:WideString;
    Data:OleVariant):OleVariant; safecall;
begin
    Lock;
    try
        Result:=Providers[ProviderName].DataRequest(Data);
    finally
        UnLock;
    end;
end;

function TDataModule.AS_GetParams(const ProviderName:WideString; var
    OwnerData:OleVariant):OleVariant;
begin
    Lock;
    try
        Result:=Providers[ProviderName].GetParams(OwnerData);
    finally
        UnLock;
    end;
end;

procedure TDataModule.AS_Execute(const ProviderName:WideString;
    const CommandText:WideString; var Params, OwnerData:OleVariant);
begin
    Lock;
    try
        Providers[ProviderName].Execute(CommandText, Params, OwnerData);
    finally
        UnLock;
    end;
end;

class procedure TDataModule.Updateistry(Register:Boolean;
    const ClassID, ProgID:string);
var
    CatReg:ICatRegister;
    Rslt:HResult;
    CatInfo:TCATEGORYINFO;

```

```

    Description:string;
begin
    Rslt:=CoCreateInstance(CLSID_StdComponentCategoryMgr, nil,
        CLSCTX_INPROC_SERVER, ICatRegister, CatReg);
    if Succeeded(Rslt) then
    begin
        if Register then
        begin
            CatInfo.catid:=CATID_MIDASAppServer;
            CatInfo.lcid:=$0409;
            StringToWideChar(MIDAS_CatDesc, CatInfo.szDescription,
                Length(MIDAS_CatDesc) + 1);
            OleCheck(CatReg.RegisterCategories(1, @CatInfo));
            OleCheck(CatReg.RegisterClassImplCategories(StringToGUID(ClassID), 1,
                @CATID_MIDASAppServer));
        end else
        begin
            OleCheck(CatReg.UnRegisterClassImplCategories(StringToGUID(ClassID), 1,
                @CATID_MIDASAppServer));
            DeleteKey(Format(SClsid + SCatImplBaseKey, [ClassID]));
        end;
    end else
    begin
        if Register then
        begin
            CreateKey('Component Categories\' + GUIDToString(CATID_MIDASAppServer), '409',
                MIDAS_CatDesc);
            CreateKey(Format(SClsid + SCatImplKey, [ClassID,
                GUIDToString(CATID_MIDASAppServer)]), '', '');
        end else
        begin
            DeleteKey(Format(SClsid + SCatImplKey, [ClassID,
                GUIDToString(CATID_MIDASAppServer)]));
            DeleteKey(Format(SClsid + SCatImplBaseKey, [ClassID]));
        end;
    end;
    if Register then
    begin
        Description:=GetRegStringValue(SClsid + ClassID, '');
        CreateKey('AppID\' + ClassID, '', Description);
        CreateKey(SClsid + ClassID, 'AppID', ClassID);
    end else
        DeleteKey('AppID\' + ClassID);
end;
begin
end;

function TDataModule.GetProviderCount:integer;
begin
    Result:=FProviders.Count;
end;

constructor TCRemoteDataModule.Create(AOwner:TComponent);
begin
    InitializeCriticalSection(FLock);
    FProviders:=TList.Create;
    inherited Create(AOwner);
end;

```

```

destructor TCRemoteDataModule.Destroy;
begin
    inherited Destroy;
    FProviders.Free;
    DeleteCriticalSection (FLock);
end;

procedure TCRemoteDataModule.Lock;
begin
    EnterCriticalSection (FLock);
end;

procedure TCRemoteDataModule.Unlock;
begin
    LeaveCriticalSection (FLock);
end;

procedure TCRemoteDataModule.RegisterProvider (Value:TCustomProvider);
begin
    FProviders.Add (Value);
end;

procedure TCRemoteDataModule.UnRegisterProvider (Value:TCustomProvider);
begin
    FProviders.Remove (Value);
end;

function TCRemoteDataModule.CRDMGetProviderNames:OleVariant;
var
    List:TStringList;
    I, J:Integer;
begin
    Lock;
    try
        List:=TStringList.Create;
        try
            for I:=0 to FProviders.Count - 1 do
                if TCustomProvider (FProviders[I]).Exported then
                    List.Add (TCustomProvider (FProviders[I]).Name);
            List.Sort;
            if List.Count > 0 then
                begin
                    Result:=VarArrayCreate ([0, List.Count -1], varOleStr);
                    for J:=0 to List.Count -1 do Result[J]:=WideString (List[J]);
                end;
            finally
                List.Free;
            end;
        finally
            UnLock;
        end;
    end;
end;

function TCRemoteDataModule.GetProvider (const
ProviderName:string):TCustomProvider;

```

```
var
  i:Integer;
begin
  Result:=nil;
  for i:=0 to FProviders.Count - 1 do
    if AnsiCompareStr(TCustomProvider(FProviders[i]).Name, ProviderName) = 0
    then
      begin
        Result:=TCustomProvider(FProviders[i]);
        if not Result.Exported then
          Result:=nil;
        Exit;
      end;
    if not Assigned(Result) then
      raise Exception.CreateResFmt(@SProviderNotExported, [ProviderName]);
  end;
```

Кафедра \_ КБПЗ \_ 2022 рік

## Файл проекту TLib.pas

```
// Файл бібліотеки TLib_InteR.dll

unit TLib_1;

{2022 рік
Розробив: студент Авраменко Кирило Дмитрович
Тема: Дослідження та програмна реалізація системи мікроклімату складського комплексу }

interface

uses Windows, Types;

function  OpenTlib32 (HW32:THANDLE;
ServiceName:PChar;
EntryPoint:PChar):THANDLE;

function  CloseTlib32 (HW32:THANDLE;):THANDLE;

function  OpenPciDeviceOnSlot (HW32:THANDLE;
BusNum :Word;
DeviceNum:Word;
FuncNum :Word):THANDLE; stdcall;

procedure ClosePciDevice (HW32:THANDLE; PciDevice:THANDLE;); stdcall;
function  ReadPciCommandReg (HW32:THANDLE; PciDevice:THANDLE;):Word; stdcall;
procedure WritePciCommandReg (HW32:THANDLE;
PciDevice:THANDLE;
CommandRegValue:Word); stdcall;
procedure ControlPciCommandRegBits (HW32:THANDLE;
PciDevice:THANDLE;CommandRegBit:Byte;
SetClearFlag:Byte); stdcall;
function  GetActiveHW (HW32:THANDLE;):BOOL; stdcall;
function  GetHardAccess (HW32:THANDLE;):BOOL; stdcall;
procedure SetHardAccess (HW32:THANDLE; bNewValue:BOOL); stdcall;
function  GetPortByte (HW32:THANDLE; PortAddr:DWORD):Byte;
procedure SetPortByte (HW32:THANDLE; PortAddr:DWORD; nNewValue:Byte); stdcall;
function  GetPortWord (HW32:THANDLE; PortAddr:DWORD):Word;
procedure SetPortWord (HW32:THANDLE; PortAddr:DWORD; nNewValue:Word); stdcall;
function  GetPortLong (HW32:THANDLE; PortAddr:DWORD):Longint;
procedure SetPortLong (HW32:THANDLE; PortAddr:DWORD; nNewValue:Longint); stdcall;
function  MapPhysToLinear (HW32:THANDLE; PhAddr:DWORD; PhSize:DWORD):Pointer;
procedure UnmapMemory (HW32:THANDLE; PhAddr:DWORD; PhSize:DWORD); stdcall;
function  GetLockedMemory (HW32:THANDLE; ):Pointer;
function  GetTempVar (HW32:THANDLE; ):Longint;

function  GetMemByte (HW32:THANDLE; MappedAddr:DWORD ; Offset:DWORD):Byte;
stdcall;
procedure SetMemByte (HW32:THANDLE; MappedAddr:DWORD ; Offset:DWORD;
nNewValue:Byte); stdcall;
function  GetMemWord (HW32:THANDLE; MappedAddr:DWORD ; Offset:DWORD):Word;
stdcall;
procedure SetMemWord (HW32:THANDLE; MappedAddr:DWORD ; Offset:DWORD;
nNewValue:Word); stdcall;
function  GetMemLong (HW32:THANDLE; MappedAddr:DWORD ; Offset:DWORD):DWORD;
```

```

procedure SetMemLong (HW32:THANDLE;MappedAddr:DWORD ; Offset:DWORD;
nNewValue:DWORD);stdcall;
function IsIRQMasked (HW32:THANDLE; IRQNumber:WORD):BOOL;
procedure UnmaskIRQ (HW32:THANDLE; IRQNumber:WORD; HWHandler:TOnHWInterrupt);
stdcall;
procedure UnmaskIRQEx (HW32:THANDLE;
IRQNumber:WORD;IrqShared:DWORD;HWHandler:TOnHWInterrupt;
ClearRec:pIrqClearRec); stdcall;
procedure UnmaskDelphiIRQ (HW32:THANDLE; TVic:THANDLE; IRQNumber:WORD;
HWHandler:TOnDelphiInterrupt); stdcall;
procedure UnmaskDelphiIRQEx (HW32:THANDLE;
TVic:THANDLE;IRQNumber:WORD;IrqShared:DWORD;HWHandler:TOnDelphiInterrupt;ClearRe
c:pIrqClearRec); stdcall;

procedure MaskIRQ (HW32:THANDLE; IRQNumber:WORD);
function GetIRQCounter (HW32:THANDLE; IRQNumber:WORD):DWORD;

procedure PutScanCode (HW32:THANDLE; b:Byte);
function GetScanCode (HW32:THANDLE;):Word;
procedure HookKeyboard (HW32:THANDLE; KbdHandler:TOnKeystroke);
procedure HookDelphiKeyboard (HW32:THANDLE; TVic:THANDLE;
KbdHandler:TOnDelphiKeystroke);
procedure UnhookKeyboard (HW32:THANDLE;);
procedure PulseKeyboard (HW32:THANDLE;);
procedure PulseKeyboardLocal (HW32:THANDLE;);

function GetRSNumber (HW32:THANDLE;):Byte;
procedure SetRSNumber (HW32:THANDLE; nNewValue:Byte);
function GetRSNumPorts (HW32:THANDLE;):Byte;
function GetRSBasePort (HW32:THANDLE;):DWORD;

function GetPin (HW32:THANDLE; nPin:Byte):BOOL;
procedure SetPin (HW32:THANDLE; nPin:Byte; bNewValue:BOOL);

function GetRSAckwl (HW32:THANDLE;):BOOL;
function GetRSBusy (HW32:THANDLE;):BOOL;
function GetRSPaperEnd (HW32:THANDLE;):BOOL;
function GetRSSlct (HW32:THANDLE;):BOOL;
function GetRSError (HW32:THANDLE;):BOOL;
procedure RSInit (HW32:THANDLE;);
procedure RSSlctIn (HW32:THANDLE;);
procedure RSStrobe (HW32:THANDLE;);
procedure RSAutofd (HW32:THANDLE; Flag:BOOL);

procedure SetRSReadMode (HW32:THANDLE; );
procedure SetRSWriteMode (HW32:THANDLE; );

procedure ForceIrqRS (HW32:THANDLE; IrqEnable:BOOL);

ReadPortFIFO (HW32:THANDLE; pBuffer:pPortByteFIFO);stdcall;
ReadPortWFIFO (HW32:THANDLE; pBuffer:pPortWordFIFO);stdcall;
ReadPortLFIFO (HW32:THANDLE; pBuffer:pPortLongFIFO);stdcall;
WritePortFIFO (HW32:THANDLE; pBuffer:pPortByteFIFO);stdcall;
WritePortWFIFO (HW32:THANDLE; pBuffer:pPortWordFIFO);stdcall;
WritePortLFIFO (HW32:THANDLE; pBuffer:pPortLongFIFO);stdcall;

GetHDDInfo (HW32:THANDLE;
IdeNumber:Word;

```

```

Master :Word;
Info :pHDDInfo);
GetLastPciBus (HW32:THANDLE;):Word; stdcall;
GetHardwareMechanism (HW32:THANDLE;):Word; stdcall;
GetPciDeviceInfo (HW32:THANDLE;
Bus,Device,Func:Word;
CfgInfo :pPciCfg):BOOL;

GetSysDmaBuffer (HW32:THANDLE;
BufReq:pDmaBufferRequest):BOOL; stdcall;

GetBusmasterDmaBuffer (HW32:THANDLE;
BufReq:pDmaBufferRequest):BOOL; stdcall;

FreeDmaBuffer (HW32:THANDLE;
BufReq:pDmaBufferRequest):BOOL; stdcall;

function AcquireRS (HW32:THANDLE; RSNumber:Word):Word; stdcall;
procedure ReleaseRS (HW32:THANDLE; RSNumber:Word);
function IsRSAcquired (HW32:THANDLE; RSNumber:Word):Word; stdcall;

function RunRing0Function (HW32:THANDLE; Ring0FunctionAddress:TRing0Function;
pParm:Pointer):Longint; stdcall;
function DebugCode (HW32:THANDLE;):Longint; stdcall;
{==} implementation {==}
function OpenTlib32 (HW32:THANDLE;
ServiceName:PChar;
EntryPoint:PChar):THANDLE;
stdcall; external 'Tlib_Inter.dll' name 'OpenTlib32';
function CloseTlib32 (HW32:THANDLE;):THANDLE;
stdcall; external 'Tlib_Inter.dll' name 'CloseTlib32';
function OpenPciDeviceOnSlot (HW32:THANDLE; BusNum :Word; DeviceNum:Word; FuncNum
:Word):THANDLE;
stdcall; external 'Tlib_Inter.dll' name 'OpenPciDeviceOnSlot';
procedure ClosePciDevice (HW32:THANDLE; PciDevice:THANDLE;);stdcall; external
'Tlib_Inter.dll' name 'ClosePciDevice';
function ReadPciCommandReg (HW32:THANDLE; PciDevice:THANDLE;):Word;stdcall;
external 'Tlib_Inter.dll' name 'ReadPciCommandReg';
procedure WritePciCommandReg (HW32:THANDLE;
PciDevice:THANDLE; CommandRegValue:Word);stdcall; external 'Tlib_Inter.dll' name
'WritePciCommandReg';
procedure ControlPciCommandRegBits (HW32:THANDLE;
PciDevice:THANDLE;
CommandRegBit:Byte; SetClearFlag:Byte);stdcall; external 'Tlib_Inter.dll' name
'ControlPciCommandRegBits';
function GetActiveHW (HW32:THANDLE;):BOOL;
stdcall; external 'Tlib_Inter.dll' name 'GetActiveHW';
function GetHardAccess (HW32:THANDLE;):BOOL;
stdcall; external 'Tlib_Inter.dll' name 'GetHardAccess';
procedure SetHardAccess (HW32:THANDLE; bNewValue:BOOL);
stdcall; external 'Tlib_Inter.dll' name 'SetHardAccess';
function GetPortByte (HW32:THANDLE; PortAddr:DWORD):Byte;
stdcall; external 'Tlib_Inter.dll' name 'GetPortByte';
procedure SetPortByte (HW32:THANDLE; PortAddr:DWORD; nNewValue:Byte);
stdcall; external 'Tlib_Inter.dll' name 'SetPortByte';
function GetPortWord (HW32:THANDLE; PortAddr:DWORD):Word;
stdcall; external 'Tlib_Inter.dll' name 'GetPortWord';
procedure SetPortWord (HW32:THANDLE; PortAddr:DWORD; nNewValue:Word);

```

```

stdcall; external 'Tlib_InteR.dll' name 'SetPortWord';
function GetPortLong(HW32:THANDLE; PortAddr:DWORD):LongInt;
stdcall; external 'Tlib_InteR.dll' name 'GetPortLong';
procedure SetPortLong(HW32:THANDLE; PortAddr:DWORD; nNewValue:LongInt);
stdcall; external 'Tlib_InteR.dll' name 'SetPortLong';
function MapPhysToLinear(HW32:THANDLE; PhAddr:DWORD; PhSize:DWORD):Pointer;
stdcall; external 'Tlib_InteR.dll' name 'MapPhysToLinear';
procedure UnmapMemory(HW32:THANDLE; PhAddr:DWORD; PhSize:DWORD);
stdcall; external 'Tlib_InteR.dll' name 'UnmapMemory';
function GetLockedMemory(HW32:THANDLE; ):Pointer;
stdcall; external 'Tlib_InteR.dll' name 'GetLockedMemory';
function GetTempVar(HW32:THANDLE; ):LongInt;
stdcall; external 'Tlib_InteR.dll' name 'GetTempVar';
function GetMemByte(HW32:THANDLE; MappedAddr:DWORD ; Offset:DWORD):Byte;
stdcall; external 'Tlib_InteR.dll' name 'GetMem';
procedure SetMemByte(HW32:THANDLE; MappedAddr:DWORD ; Offset:DWORD;
nNewValue:Byte);
stdcall; external 'Tlib_InteR.dll' name 'SetMem';
function GetMemWord(HW32:THANDLE;MappedAddr:DWORD ; Offset:DWORD):Word;
stdcall; external 'Tlib_InteR.dll' name 'GetMemW';
procedure SetMemWord(HW32:THANDLE;MappedAddr:DWORD ; Offset:DWORD;
nNewValue:Word);
stdcall; external 'Tlib_InteR.dll' name 'SetMemW';
function GetMemLong(HW32:THANDLE;MappedAddr:DWORD ; Offset:DWORD):DWORD;
stdcall; external 'Tlib_InteR.dll' name 'GetMemL';
procedure SetMemLong(HW32:THANDLE;MappedAddr:DWORD ; Offset:DWORD;
nNewValue:DWORD);
stdcall; external 'Tlib_InteR.dll' name 'SetMemL';
function IsIRQMasked(HW32:THANDLE; IRQNumber:WORD):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'IsIRQMasked';
procedure UnmaskIRQ(HW32:THANDLE; IRQNumber:WORD; HWHandler:TOnHWInterrupt);
stdcall; external 'Tlib_InteR.dll' name 'UnmaskIRQ';
procedure UnmaskDelphiIRQ(HW32:THANDLE; TVic:THANDLE; IRQNumber:WORD;
HWHandler:TOnDelphiInterrupt);
stdcall; external 'Tlib_InteR.dll' name 'UnmaskDelphiIRQ';
procedure
UnmaskIRQEx(HW32:THANDLE;IRQNumber:WORD;IrqShared:DWORD;HWHandler:TOnHWInterrupt
;
ClearRec:pIrqClearRec);stdcall; external 'Tlib_InteR.dll' name 'UnmaskIRQEx';
procedure UnmaskDelphiIRQEx(HW32:THANDLE;
TVic:THANDLE;IRQNumber:WORD;IrqShared:DWORD;HWHandler:TOnDelphiInterrupt;ClearRe
c:pIrqClearRec);
stdcall; external 'Tlib_InteR.dll' name 'UnmaskDelphiIRQEx';
procedure MaskIRQ(HW32:THANDLE; IRQNumber:WORD);
stdcall; external 'Tlib_InteR.dll' name 'MaskIRQ';
function GetIRQCounter(HW32:THANDLE; IRQNumber:WORD):DWORD;stdcall; external
'Tlib_InteR.dll' name 'GetIRQCounter';
procedure PutScanCode(HW32:THANDLE; b:Byte);
stdcall; external 'Tlib_InteR.dll' name 'PutScanCode';
function GetScanCode(HW32:THANDLE;):Word;
stdcall; external 'Tlib_InteR.dll' name 'GetScanCode';
procedure HookKeyboard(HW32:THANDLE; KbdHandler:TOnKeystroke);stdcall; external
'Tlib_InteR.dll' name 'HookKeyboard';
procedure HookDelphiKeyboard(HW32:THANDLE; TVic:THANDLE;
KbdHandler:TOnDelphiKeystroke);
stdcall; external 'Tlib_InteR.dll' name 'HookDelphiKeyboard';
procedure UnhookKeyboard(HW32:THANDLE;);
stdcall; external 'Tlib_InteR.dll' name 'UnhookKeyboard';
procedure PulseKeyboard(HW32:THANDLE;);

```

```

stdcall; external 'Tlib_InteR.dll' name 'PulseKeyboard';
procedure PulseKeyboardLocal(HW32:THANDLE;);
stdcall; external 'Tlib_InteR.dll' name 'PulseKeyboardLocal';
function GetRSNumber (HW32:THANDLE;):Byte;
stdcall; external 'Tlib_InteR.dll' name 'GetRSNumber';
procedure SetRSNumber(HW32:THANDLE; nNewValue:Byte);
stdcall; external 'Tlib_InteR.dll' name 'SetRSNumber';
function GetRSNumPorts (HW32:THANDLE;):Byte;
stdcall; external 'Tlib_InteR.dll' name 'GetRSNumPorts';
function GetRSBasePort (HW32:THANDLE;):DWORD;
stdcall; external 'Tlib_InteR.dll' name 'GetRSBasePort';
function GetPin(HW32:THANDLE; nPin:Byte):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'GetPin';
procedure SetPin(HW32:THANDLE; nPin:Byte; bNewValue:BOOL);
stdcall; external 'Tlib_InteR.dll' name 'SetPin';
function GetRSAckwl (HW32:THANDLE;):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'GetRSAckwl';
function GetRSBusy(HW32:THANDLE;):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'GetRSBusy';
function GetRSPaperEnd(HW32:THANDLE;):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'GetRSPaperEnd';
function GetRSSlct(HW32:THANDLE;):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'GetRSSlct';
function GetRSError(HW32:THANDLE;):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'GetRSError';
procedure RSInit(HW32:THANDLE;);
stdcall; external 'Tlib_InteR.dll' name 'RSInit';
procedure RSSlctIn(HW32:THANDLE;);
stdcall; external 'Tlib_InteR.dll' name 'RSSlctIn';
procedure RSStrobe(HW32:THANDLE;);
stdcall; external 'Tlib_InteR.dll' name 'RSStrobe';
procedure RSAutofd(HW32:THANDLE; Flag:BOOL);
stdcall; external 'Tlib_InteR.dll' name 'RSAutofd';
procedure SetRSReadMode(HW32:THANDLE; );
stdcall; external 'Tlib_InteR.dll' name 'SetRSReadMode';
procedure SetRSWriteMode(HW32:THANDLE; );
stdcall; external 'Tlib_InteR.dll' name 'SetRSWriteMode';
procedure ForceIrqRS(HW32:THANDLE; IrqEnable:BOOL);
stdcall; external 'Tlib_InteR.dll' name 'ForceIrqRS';
ReadPortFIFO (HW32:THANDLE; pBuffer:pPortByteFIFO);
stdcall; external 'Tlib_InteR.dll' name 'ReadPortFIFO';
ReadPortWFIFO(HW32:THANDLE; pBuffer:pPortWordFIFO);
stdcall; external 'Tlib_InteR.dll' name 'ReadPortWFIFO';
ReadPortLFIFO(HW32:THANDLE; pBuffer:pPortLongFIFO);
stdcall; external 'Tlib_InteR.dll' name 'ReadPortLFIFO';
WritePortFIFO(HW32:THANDLE; pBuffer:pPortByteFIFO);
stdcall; external 'Tlib_InteR.dll' name 'WritePortFIFO';
WritePortWFIFO(HW32:THANDLE; pBuffer:pPortWordFIFO);
stdcall; external 'Tlib_InteR.dll' name 'WritePortWFIFO';
WritePortLFIFO(HW32:THANDLE; pBuffer:pPortLongFIFO);
stdcall; external 'Tlib_InteR.dll' name 'WritePortLFIFO';
GetHDDInfo(HW32:THANDLE; IdeNumber:Word; Master :Word; Info:pHDDInfo); stdcall;
external 'Tlib_InteR.dll' name 'GetHDDInfo';
GetLastPciBus (HW32:THANDLE;):Word;
stdcall; external 'Tlib_InteR.dll' name 'GetLastPciBus';
GetHardwareMechanism(HW32:THANDLE;):Word;
stdcall; external 'Tlib_InteR.dll' name 'GetHardwareMechanism';

```

```
GetPciDeviceInfo (HW32:THANDLE; Bus, Device, Func:Word;
CfgInfo:pPciCfg):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'GetPciDeviceInfo';
GetSysDmaBuffer (HW32:THANDLE; BufReq:pDmaBufferRequest):BOOL; stdcall; external
'Tlib_InteR.dll' name 'GetSysDmaBuffer';
GetBusmasterDmaBuffer (HW32:THANDLE; BufReq:pDmaBufferRequest):BOOL; stdcall;
external 'Tlib_InteR.dll' name 'GetBusmasterDmaBuffer';
FreeDmaBuffer (HW32:THANDLE; BufReq:pDmaBufferRequest):BOOL;
stdcall; external 'Tlib_InteR.dll' name 'FreeDmaBuffer';
function AcquireRS (HW32:THANDLE; RSNumber:Word):Word;
stdcall; external 'Tlib_InteR.dll' name 'AcquireRS';
procedure ReleaseRS (HW32:THANDLE; RSNumber:Word);
stdcall; external 'Tlib_InteR.dll' name 'ReleaseRS';
function IsRSAcquired (HW32:THANDLE; RSNumber:Word):Word; stdcall; external
'Tlib_InteR.dll' name 'IsRSAcquired';
function RunRing0Function (HW32:THANDLE;
Ring0FunctionAddress:TRing0Function; pParm:Pointer):Longint; stdcall; external
'Tlib_InteR.dll' name 'RunRing0Function';
function DebugCode (HW32:THANDLE):Longint; stdcall; external 'Tlib_InteR.dll'
name 'DebugCode';
end.
```

Кафедра — КБПЗ — 2022 рік

## Файл проекту ABOUT.pas

```
unit UN_ABOUT;

{2022 рік
Розробив: студент Авраменко Кирило Дмитрович
Тема: Дослідження та програмна реалізація системи мікроклімату складського комплексу }

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls, Buttons,
ExtCtrls;

type
  TAboutBox = class(TForm)
    Panell: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Version: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    OKButton: TButton;
    procedure OKButtonClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  TWMDDE_Initiate = packed record
    Msg: Cardinal;
    PostingApp: HWND;
    App: Word;
    Topic: Word;
    Result: Longint;
  end;

  TWMDDE_Poke = packed record
    Msg: Cardinal;
    PostingApp: HWND;
    PackedVal: Longint;
    Result: Longint;
  end;

var
  AboutBox: TAboutBox;

implementation

uses Unit1;

{$R *.dfm}

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
  AboutBox.hide;
```

```
Form1.show;  
end;
```

```
procedure TAboutBox.FormClose(Sender:TObject; var Action:TCloseAction);  
begin  
    AboutBox.hide;  
    Form1.show;  
end;
```

```
end.
```

Кафедра \_ КБПЗ \_ 2022 рік