

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

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

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи ідентифікації**  
**професійних знань операторів автоматизованих систем**  
**управління”**

Виконав здобувач вищої освіти  
II курсу, групи КІ-22МЗ  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Гордієнко Ю.Ю.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник проекту  
доктор технічних наук, професор  
\_\_\_\_\_ Смірнов О.А.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Центр *Заочної та дистанційної освіти*  
Кафедра *Кібербезпеки та програмного забезпечення*  
Рівень вищої освіти *магістр*  
Галузь знань 12 *“Інформаційні технології”*  
Спеціальність 123 *“Комп’ютерна інженерія”*  
Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерна інженерія”*

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.т.н., проф.  
Олексій СМІРНОВ  
« 6 » вересня 2023 року

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

*Гордієнку Юрію Юрійовичу*

(прізвище, ім’я, по батькові)

- |  |  |
|--|--|
| 1. Тема роботи   | <i>Дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління</i>                  |
| 2. Керівник роботи   | <i>Смірнов Олексій Анатолійович, докт. техн. наук, професор</i><br>(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)                 |
| затверджені наказом вищого навчального закладу № 36-13 від 04.08.2023 року           |  |
| 3. Строк подання студентом роботи до захисту   | <i>10.12.2023 р.</i>   |
| 4. Мета та завдання випускної кваліфікаційної роботи:                                | <i>Метою розробки є дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління</i> |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) |  |
| <i>1. Призначення та область використання.</i>                                       | <i>6. Наукова новизна.</i>   |
| <i>2. Перегляд аналогічних існуючих систем.</i>                                      | <i>7. Економічна ефективність розробленої програми.</i>  |
| <i>3. Опис і обґрунтування проектних рішень.</i>                                     | <i>8. Заходи з охорони праці та техніки безпеки.</i>   |
| <i>4. Етапи програмування системи.</i>   | <i>9. Висновки.</i>  |
| <i>5. Впровадження системи в промислову експлуатацію</i>                             |  |
| 6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)         |  |
| <i>Наукова новизна</i>   | <i>1 аркуш</i>   |
| <i>Структурна схема системи</i>  | <i>1 аркуш</i>   |
| <i>Функціональна схема системи</i>   | <i>1 аркуш</i>   |
| <i>Діаграма процесів</i>   | <i>1 аркуш</i>   |
| <i>Блок-схема алгоритму роботи додатку</i>   | <i>2 аркуша</i>  |
| <i>Показники економічної ефективності</i>  | <i>1 аркуш</i>   |

## 6. Консультанти розділів роботи

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

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

## КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання  
« 6 » вересня 2023 р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
« 6 » вересня 2023 р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Гордієнко Ю.Ю. Дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи ідентифікації професійних знань операторів автоматизованих систем управління.

Метою розробки є дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

Об'єктом дослідження є процес ідентифікації професійних знань операторів автоматизованих систем управління.

Предметом дослідження є методи ідентифікації професійних знань операторів автоматизованих систем управління.

Методи дослідження базуються на методах теорії комп'ютерного тестування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

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.

**Ключові слова:** комп'ютерна інженерія, тестування

## ABSTRACT

**Hordiienko Yu.Yu. Research and software implementation of the system of identification of professional knowledge of operators of automated control systems. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.**

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the system of identification of professional knowledge of operators of automated control systems.

The purpose of the development is research and software implementation of a system for identifying the professional knowledge of operators of automated control systems.

The object of the study is the process of identifying the professional knowledge of operators of automated control systems.

The subject of the study is methods of identifying the professional knowledge of operators of automated control systems.

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

The result of the work is the software implementation of the system for identifying the professional knowledge of operators of automated control systems.

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 environment.

**Keywords:** computer engineering, testing

## 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.....	10
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.....	19
2.3 Розгорнута постановка завдання .....	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	27
3.1 Опис функціонування системи .....	27
3.2 Розробка структурної схеми.....	33
3.3 Розробка функціональної схеми .....	39
3.4 Розробка діаграми процесів.....	52
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	54
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	54
4.2 Захист розробленого програмного забезпечення.....	65
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	68
6 НАУКОВА НОВИЗНА .....	72

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>			
<b>Вим</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	<i>Дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління</i>	<b>Літ.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<i>Розроб.</i>	<i>Гордієнко Ю.Ю.</i>					<b>М</b>	1	113
<i>Перев.</i>	<i>Смірнов О.А.</i>							
<b>Н.контр.</b>	<i>Коваленко А.С.</i>					<b>ЦНТУ КІ-22МЗ</b>		
<b>Затв.</b>	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	73
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	73
7.2 Розрахунок трудомісткості розробки програмної продукції.....	75
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	77
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	81
7.5 Визначення собівартості розробки та ціни програмної продукції.....	86
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	89
7.7 Визначення експлуатаційних витрат.....	89
7.8 Визначення економічної ефективності програмної продукції.....	91
7.9 Висновок.....	93
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	94
8.1 Вступ.....	94
8.2 Пожежна безпека.....	96
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	98
8.4 Розробка заходів з умов поліпшення охорони праці.....	101
8.5 Розрахункова частина .....	101
9 ОСНОВНІ ВИСНОВКИ.....	105
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	107

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- АСУ – Автоматизована система управління  
СН – Система навчання

КБПЗ – 2023

					ВКРМ-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Сучасні великомасштабні підприємства, які характеризуються великими обсягами виробництва, використанням значних людських і матеріальних ресурсів, функціонуванням в умовах невизначеності зовнішнього середовища та ризику, потребують нових методів і технологій управління на основі впровадження автоматизованих систем управління (АСУ). До таких підприємств, в першу чергу, належать гірничорудні кар'єри, авіаційні підприємства, багатомономенклатурні прокатні стани металургійних комбінатів енергетичні системи тощо. Значне місце в підвищенні надійності, живучості, ефективності автоматизованих систем управління та організаційно-технологічних процесів надається операторам АСУ. Оператор АСУ з позиції системного підходу, який базується на інформаційному поєднанні організаційно-технологічного управління, повинен відповідати наступним вимогам: мати глибокі професійні знання щодо предметної галузі виробництва, вміти системно користуватися комп'ютерною технікою та проблемно-орієнтованим програмним забезпеченням; бути здібним формувати і аналізувати організаційно-технологічні моделі реальних виробничих ситуацій; вміти оперативно в реальному вимірі часу супроводжувати технологічний процес відповідними організаційними документами; володіти методологією, методикою та інструментальними засобами прийняття багатокритеріальних та багатоальтернативних управлінських рішень, в тому числі і в умовах невизначеності зовнішнього середовища; мати навички своєчасно в реальному вимірі часу супроводжувати технологічний процес відповідними організаційними документами.

Вітчизняна та зарубіжна практика підготовки та перепідготовки операторів АСУ включає такі етапи:

- технологічну підготовку;
- підготовку системного володіння проблемно-орієнтованим програмно-апаратним забезпеченням;

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– тренажерну підготовку із застосуванням комп'ютерної техніки при моделюванні виробничих ситуацій.

В такій підготовці оцінка знань оператора визначається повнотою та якістю його дій в умовах тренажерного моделювання, тобто здійснюється лише технічна підготовка операторів, що фактично не дозволяє в процесі навчання ідентифікувати їх знання щодо вміння моделювати виробничі ситуації на організаційно-технологічному рівні та приймати ефективні управлінські рішення. Ідентифікація та оцінювання таких знань потребує використання нових мінімально збиткових методів та комп'ютерних технологій. Отже, виникає проблема щодо подальшого розвитку теорії і практики інтелектуального забезпечення ергатичних систем, дослідження та розробки методології та інструментарію системної ідентифікації професійних проблемно-орієнтованих знань операторів АСУ.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

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

- Огляд існуючих систем ідентифікації професійних знань операторів автоматизованих систем управління.
- Дослідження системи ідентифікації професійних знань операторів автоматизованих систем управління.
- Програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

*Об'єктом дослідження* є процес ідентифікації професійних знань операторів автоматизованих систем управління.

*Предметом дослідження* є методи ідентифікації професійних знань операторів автоматизованих систем управління.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

*Методи дослідження* базуються на методах теорії комп'ютерного тестування, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод ідентифікації професійних знань операторів автоматизованих систем управління.

– Розроблено вітчизняний продукт ідентифікації професійних знань операторів автоматизованих систем управління, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі ідентифікації професійних знань операторів автоматизованих систем управління.

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

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Система призначена для реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

Розглянемо аналіз проблем ідентифікації систем навчання та контролю знань операторів АСУ. Наведемо огляд існуючих моделей, методів і засобів щодо вирішення проблем ідентифікації систем навчання та контролю знань операторів АСУ.

При формальному описі процесів АСУ використана методологія системного аналізу. При цьому з позиції функціональної єдності формалізована система прийняття рішень. У залежності від мети і рівня абстрагування під системою розуміємо множину однорідних або різнорідних альтернатив  $X$ , на яких реалізована множина відношень  $R$ . Ці відношення упорядковують альтернативи в структуру, яка має деякий набір властивостей та дозволяє досягти заданої мети. Таким чином, упорядковані альтернативи і відношення між ними створюють певну структуру  $S = \{X, R\}$  яка може бути інтерпретована як цілеспрямована система, для якої розв'язується завдання цілеспрямованого синтезу.

Це завдання полягає у визначенні підмножин альтернатив  $X$  і відношень,  $P \rightarrow X, P \rightarrow R$ . Визначено категорію цілі  $G$  як якийсь (у рамках конкретної предметної області) компроміс між дійсним і бажаним, засобами і результатами, витратами і корисністю, ризиками і невизначеністю. Носіями цілі визначені оператори АСУ.

Модель прийняття рішень  $S$  представлена у вигляді:

$$S = \langle \{X, R\}, I \rangle,$$

де

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

$R$  – відношення слабкої переваги на множині  $X$ ;

$I$  – інформація про цілі функціонування виробничої системи, вираженої у вигляді переваг операторів.

У задачі тестового контролю професійних знань оператора основу множини  $I$  складає розподіл ймовірностей (частот)  $P = \{P_j\}_m$  на класах  $X = \{X_j\}_m$ , які можна одержати при статистичній оцінці інформативності систем тестового контролю і вірогідності результатів професійних знань оператора. Кожна альтернатива  $X_j$  або приводить до єдиного результату (оптимальний варіант прийняття рішень оператором), або може з визначеною ймовірністю привести до одного з декількох наслідків (допустимим варіантом прийняття рішень оператором).

Для побудови математичної моделі зовнішнього впливу використовувався метод ідентифікації моделей зовнішнього впливу, що має низку очевидних переваг. По-перше, він ґрунтується на об'єктивній інформації; по-друге, результати математичного моделювання із застосуванням такої моделі збігаються з експериментом із максимальною точністю. Ідентифікація дії моделей зовнішнього середовища на виробничий процес здійснюється на основі гіпотез (яких в магістерській роботі п'ять) про характер дії зовнішнього середовища. При формуванні гіпотез розглянуті випадки, коли зовнішнє середовище поводить ся стосовно системи агресивно або сприятливо. У даній роботі прийнята концепція нормативної моделі зовнішнього впливу, що заснована на загальному принципі еквівалентності і яка орієнтована на збіг тільки кінцевого результату. У цьому випадку математичний опис моделі й об'єкту хоча б частково, але не збігаються. Вирішимо задачу структурної ідентифікації моделі зовнішнього впливу за результатами експериментального виміру відгуку системи  $u_\delta: A_p = u_\delta$ , де  $A_p$  є заданий оператор, що здійснює зв'язок зовнішнього впливу  $z$  і відгуку математичної моделі і залежних від вектора – параметрів  $\bar{p}$ ;  $z \in Z$ ;  $u_\delta \in U$  ( $Z, U$  – деякі функціональні нормовані простори). Позначивши через  $Q_{\delta,p}$  множину можливих рішень зворотної задачі ідентифікації моделі зовнішнього впливу при

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

фіксованому операторі  $A_p$ ,  $Q_{\delta, \bar{p}} = \{z : \|A_{\bar{p}}z - u_{\delta}\|_U \leq \delta\}$ , будь-яка функція з множини  $Q_{\delta, \bar{p}}$  є моделлю зовнішнього впливу. Задача знаходження  $z \in Q_{\delta, \bar{p}}$  є задачею ідентифікації зовнішнього впливу.

Класифікація методів математичної підтримки прийняття рішень представлена базовими економіко-математичними завданнями і базовими економіко-імітаційними задачами. Кожна з базових задач дає повне уявлення про можливі методи і моделі задач оперативного управління АСУ.

Проведемо порівняльний аналіз системи ідентифікації математичної моделі системи навчання. Розроблено вираз для передатної функції об'єкта:

$W = \frac{S_{xy}}{S_{xx}}$  або в розгорнутій формі:

$$\begin{aligned} S_{xy} &= (Y_{tr} + Y_{nn})(X_{tr}^* + X_{nn}^*) = X_{tr}^* Y_{tr} + Y_{nn} (X_{tr}^* + X_{nn}^*) + X_{nn}^* Y_{tr}; \\ S_{xx} &= (\text{Re}(X_{tr} + X_{nn}))^2 + (\text{Im}(X_{tr} + X_{nn}))^2 = (X_{tr} + X_{nn})(X_{tr}^* + X_{nn}^*) = \\ &= X_{tr} X_{tr}^* + X_{nn} (X_{tr}^* + X_{nn}^*) + X_{tr} X_{nn}^*; \end{aligned}$$

Визначення передатної функції дозволяє моделювати процеси на виході об'єкта по входу об'єкта. Процес на виході об'єкта  $y(t)$  можна представити у вигляді згортки  $h(\alpha, t)$  і  $x(t)$ :

$$y(t) = \int_0^{\infty} h(\alpha, t - \tau) x(\tau) d\tau$$

Теоретичним фундаментом розглянутих класів методів вважають роботи Грона Д., Ейкхоффа П., Павлова О.А., Петрова Е.Г., Ротштейна А.П., Теленик С.Ф., Шабанова-Кушнарєнка Ю.П., Шаронової Н.В.. На основі критичного аналізу розглянутих методів сформульовано завдання дослідження щодо узагальнення і розвитку системи ідентифікації професійних знань операторів автоматизованих систем управління.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## 1.2 Область застосування

Областю застосування є системи навчання. Системою називається будь-яке складне явище, до якого входять численні елементи, котрі утворюють певну сукупність завдяки наявності між ними внутрішніх зв'язків.

На першому рівні система навчання розглядається із загальних позицій як система, що створює взаємодію методики як науки та всієї об'єктно-предметної галузі. На цьому рівні у процесі взаємодії двох підсистем методики як науки та її об'єктно-предметної галузі відбувається збільшення обсягу теоретичних знань з методики навчання і посилення управління практикою навчання з боку теорії.

На другому рівні система навчання співвідноситься зі сферою навчання конкретної галузі. На цьому рівні навчання розглядається як складна ієрархічна система, що наповнена конкретним змістом, спрямованим на навчання конкретної галузі.

На третьому рівні система навчання розглядається як реальний навчально-виховний процес, головними компонентами / підсистемами якого є вчитель, учні та матеріальні засоби навчання.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.23.0088.00.00.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

У сучасних освітніх установах активно використовуються нові ефективні форми й методи організації навчального процесу з використанням інформаційних і комунікаційних технологій. Об'єктивний контроль знань, умінь і навичок того, якого навчають – одне з актуальних завдань теперішнього часу. На сьогоднішній день тестування як форма контролю знань учнів широко застосовна на всіх щаблях різних форм утворення, а проведення єдиного іспиту й організації наступності в утворенні в цілому, без тестування представити неможливо.

Саме тому в системі сучасного утворення надається особливе значення організації процесу тестування й розробці банків тестових завдань.

Комп'ютерне тестування як особливий інноваційний вид тестування є найбільш ефективною формою контролю, перевірки й самоперевірки знань учнів. Тому важливо, щоб процес організації комп'ютерного тестування мав грамотно спроектовану й реалізовану інформаційну систему підтримки.

Перевагами комп'ютерного тестування є:

- об'єктивність тестування: ЕОМ «безстороння» при пред'явленні тестових завдань і підрахунку результатів їхнього виконання;
- зручність фіксації, зберігання й подання результатів тестування, а також можливість їхньої автоматизованої обробки, включаючи ведення баз даних і статистичний аналіз;
- зручність реалізації процедур індивідуально-орієнтованого тестування;
- можливість створення таких тестових завдань, які не можуть бути представлені без комп'ютера, при цьому можливо використання графічних,

					ВКРМ-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11



Продовження таблиці 2.1

№	Групи критеріїв порівняння	Критерії порівняння
3	Вставка об'єктів у тест	Використання різних форматів комп'ютерної графіки й мультимедійних файлів, використання редактора формул, вставка символу, таблиці, блок-схеми, кнопок переходу й ін.
4	Налаштування параметрів тесту	Захист редагування тесту паролем, обмеження дати тестування, часу тестування, кількості тестувань і ін.
5	Додаткові можливості при тестуванні	Виклик убудованого калькулятора, вставка підказки, можливість переходу до попереднього тестового завдання й ін.
6	Авторство й вартість	Авторство й вартість програми.

При пошуку в Інтернет були виділені наступні оболонки:

- SuperTest v2.4;
- Test-Master 99;
- TestBuilder v3.4;
- SunRav TestOfficePro v5.1.

Вибір даних середовищ заснований на тім, що вони дозволяють використовувати основні можливості, необхідні для створення комп'ютерних тестів по різних дисциплінах на основі загальних принципів тестології – теорії й практики створення тестів для системи утворення.

Так само основним критерієм вибору даних середовищ є доступність для викладачів, що не володіють навичками програмування.

Представимо результати проведеного порівняльного аналізу даних середовищ за розробленими критеріями (див. таблиці 2.2-2.7).

Таблиця 2.2 – Порівняльний аналіз середовищ за параметром «Адміністрування й обробка результатів тестування»

Критерій порівняння	SuperTest v2.4	Test-Master 99	TestBuilder v3.4	SunRav TestOfficePro v5.1
1. Реєстрація/ідентифікація тестуємих за паролем	+	–	+	+
2. Ведення протоколу тестування	+	–	+	+(2 види)
3. Можливість заборони на перемикання між вікнами при тестуванні	+	–	+	–
4. Ведення статистики (тобто порівняльного аналізу проведеного тестування між учнями, групами учнів, розділами тесту й ін.)	–	–	–	+

Таблиця 2.3 – Порівняльний аналіз середовищ за параметром «Формування тесту»

Критерій порівняння	SuperTest v2.4	Test-Master 99	TestBuilder v3.4	SunRav TestOfficePro v5.1
1. Імпорт тестів з текстових і табличних файлів	–	–	+(.txt)	+(.txt, .xls, .rtf)

Продовження таблиці 2.3

Критерій порівняння	SuperTest v2.4	Test-Master 99	TestBuilder v3.4	SunRav TestOfficePro v5.1
2. Експорт тестів у текстові редактори	+	-	+	+
3. Розбивка тестових завдань на групи (розділи)	+	-	+	+
4. Перевірка правопису тексту	-	-	-	+
5. Створення Web-тестів	+ (інтеграція з Microsoft Edge)	-	+ (інтеграція з Microsoft Edge)	+
6. Завдання вагової характеристики тестового завдання	+	-	+	+
7. Форматування тексту	-	-	+	+

Створення різних типів тестів одне з головних завдань, що повинна вирішувати тестова оболонка. У порівнюваних оболонках дане завдання вирішується в більшому або меншому ступені. Так у кожній з порівнюваних тестових оболонок передбачено 2 типи тестів: контрольний і психологічний. Далі відбувається розподіл по типах питань і типам їхнього надання. Програми SuperTest v2.4 і TestBuilder v3.4 надають на вибір три системи послідовності

питань: прямий порядок, зворотний порядок і випадковий порядок проходження питань. Test-Master 99 надає на вибір три структури проходження питань: деревоподібна (відповідь спричиняється вибір наступного питання), вибірна (прямий порядок проходження питань) і змішана структури тесту. Більш повно даному завданню відповідає тестовий комплекс SunRav TestOfficePro v5.1, у якому крім структури проходження питань (випадкова й пряма) передбачений вибір типу питання: одиночний вибір, множинний вибір, відповідність, упорядкована список, а так само питання з відкритим варіантом відповіді (відповідь уводиться маскою).

Таблиця 2.4 – Порівняльний аналіз середовищ за параметром «Вставка об'єктів у тест»

Критерій порівняння	SuperTest v2.4	Test-Master 99	TestBuilder v3.4	SunRav TestOfficePro v5.1
1. Графічний формат	bmp, ico, wmf, emf; jpg	–	bmp, ico, wmf, emf; jpg, jpeg	bmp, jpeg, jpg, png, gif, emf, wmf, ico
2. Звуковий формат	–	–	–	wav, midi, mp3
3. Відео формат	–	–	–	avi, mpeg, mpg, mov
4. Підтримка анімації	–	–	–	+
5. Використання редактора формул	–	–	–	+
6. Вставка символів	–	–	–	+
7. Вставка таблиць	–	–	–	+
8. Вставка діаграм	–	–	–	+

Таблиця 2.5 – Порівняльний аналіз середовищ за параметром «Налаштування параметрів тесту»

Критерій порівняння	SuperTest v2.4	Test- Master 99	TestBuilder v3.4	SunRav TestOfficePro v5.1
1. Обмеження дати тестування	–	–	+	+
2. Обмеження часу тестування	+	–	+	+
3. Обмеження кількості тестувань	±	–	+	+
4. Обмеження часу відповіді на кожне тестове завдання	–	–	+	+
5. Перемішування тестових завдань	–	–	–	+
6. Перемішування варіантів відповідей	+	–	+	+
7. Випадковий вибір тестових завдань із тестової бази	+	–	+	+
8. Альтернативне оцінювання	+	–	+	+
9. Можливість завдання шкали оцінювання	+	–	+	+
10. Упорядкувати питання по ступені складності	+	–	+	+
11. Відкритість/закритість тестування	+	–	+	+

У кожній з порівнюваних програм у передбачена функція захисту редагування тесту паролем, що забезпечує схоронність даних тестових завдань.

Таблиця 2.6 – Порівняльний аналіз середовищ за параметром «Додаткові можливості при тестуванні»

Критерій порівняння	SuperTest v2.4	Test- Master 99	TestBuilder v3.4	SunRav TestOfficePro v5.1
1. Виклик убудованого калькулятора	+	–	+	+
2. Вставка повідомлення про правильність відповіді	–	–	+	+
3. Вставка підказки	+	–	+	+
4. Можливість переходу до попереднього тестового завдання	–	–	+	+

У результаті проведеного аналізу можна зробити вивід, що не всі розглянуті програми мають досить широкий інструментарій для розробки й проведення комп'ютерного тестування. Тому, як показав детальний аналіз представлених інструментальних середовищ і, саме головне, апробування програм комп'ютерного тестування в нашій навчальній установі, найбільш прийнятним середовищем для створення комп'ютерних тестів є програмний продукт SunRav TestOfficePro v5.1.

Інструментальне середовище SunRav TestOfficePro v5.1 відповідає більшості запропонованих критеріїв до розробки комп'ютерних тестів. Дані продукти можуть бути використані для проведення тестування в рамках як освітніх установ (вузи, коледжі, школи), а також інших установ (відділи кадрів підприємств і т.п.).

Таблиця 2.7 – Порівняльний аналіз середовищ за параметром «Авторство й вартість»

Критерій порівняння	SuperTest v2.4	Test-Master 99	TestBuilder v3.4	SunRav TestOfficePro v5.1
1. Вартість програми	Дана версія програми є безкоштовної	Програма зовсім безкоштовна	Вартість програми (1 ліцензія): 500 грн. – для приватних осіб; 800 грн. – для навчальних закладів і підприємств. (при покупці 2-х і більше ліцензій ціна договірна)	Демо-версія безкоштовна (сказати про обмеження) 5 900 грн. ( Web-версія- 14 тис. грн.)

## 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





базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

### **Істотне поліпшення Delphi Code Insight**

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

### **Delphi Custom Managed Records**

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

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

### **Єдине керування пам'яттю**

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

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

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

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

## **Розширена підтримка бібліотек C++**

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

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

## **Win 64-відладник і збирач для C++**

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими 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.

## **Підвищення якості й швидкодії інструментів**

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

## **Змінені стилі VCL для High DPI**

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

масштабується під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

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

### **Нові High DPI стилі й стилізація окремих VCL компонент**

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

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

### **Поліпшена кроссплатформеність**

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємі FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- 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.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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 забезпечення, яке призначено для системи ідентифікації професійних знань операторів автоматизованих систем управління.

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

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

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.

КБПЗ\_2023

					ВКРМ-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

З метою розробки методів структурної ідентифікації математичної моделі навчання операторів АСУ розглянемо проблеми ефективності системи професійного навчання операторів АСУ.

Побудову моделі ідентифікації процесу навчання засновано на уявленні останнього процесу навчання у вигляді цілеспрямованого функціонування деякої «складної системи».

Виділено в структурі системи навчання три основні підсистеми:  $\{S_v\}, v = 1, N_s$ ; операторів  $\{\Pi_i\}$  загальною чисельністю  $N_s$ ;  $\{\Pi_i\}, i = 1, N_\Pi$ , навчальних предметів загальною чисельністю  $N_\Pi$ ;  $\{O_j\}, j = 1, N_o$ , способів організації навчання загальною чисельністю  $N_o$ . Ентропію системи навчання позначено виразом:

$$H_{CO} = -\varphi \cdot \log_2 \varphi - (1 - \varphi) \cdot \log_2 (1 - \varphi). \quad (3.1)$$

Введені поняття моделі ефективності дозволяють трансформувати постановку задачі побудови моделі навчання, системи навчання, тобто пошуку функціональної залежності  $\dot{Y}_{CO} = f(\dot{Y}_S; \dot{Y}_I; \dot{Y}_W)$ . Іншими словами, для одного періоду розгляду діяльності системи навчання або для одного такту навчання можна розглядати ймовірність результатів цього такту у вигляді:

$$\begin{cases} \varphi^{(l)} = \varphi^{(0)} \cdot p_{11}^{(l)} + (1 - \varphi^{(0)}) \cdot p_{01}^{(l)}; \\ 1 - \varphi^{(l)} = \varphi^{(0)} \cdot p_{10}^{(l)} + (1 - \varphi^{(0)}) \cdot p_{00}^{(l)}; \end{cases} \quad (3.2)$$

ефективність системи навчання визначається через ймовірність успіху при нескінченному числі тактів навчання:

$$\dot{Y}_{CO} = \dot{Y}_{CO}^{(\infty)} = \ln \frac{\varphi^{(\infty)}}{1 - \varphi^{(\infty)}}.$$

					ВКРМ-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Безумовна ймовірність успіху системи навчання  $[CH]$ , тобто рівняння моделі представлено у вигляді:

$$\varphi = \frac{p}{p+q} = \frac{z \cdot \lambda \cdot \gamma}{z \cdot \lambda \cdot \gamma + (1-z) \cdot (1-\lambda) \cdot (1-\gamma)}. \quad (3.3)$$

Якщо тепер знайти математичну модель ефективності системи

$$\begin{aligned} \dot{Y}_{CO} &= \ln \frac{\varphi}{1-\varphi} = \ln \frac{p/(q+p)}{q/(q+p)} = \ln \frac{p}{q} = \\ &= \ln \frac{z \cdot \lambda \cdot \gamma}{(1-z) \cdot (1-\lambda) \cdot (1-\gamma)} = \dot{Y}_S + \dot{Y}_I + \dot{Y}_\gamma, \end{aligned} \quad (3.4)$$

то ефективність усієї системи навчання буде визначатися сумою ефективності її елементів. Встановлено, що необхідне врахування ще однієї властивості системи навчання, її автономності, обумовленої коефіцієнтом регресії рангу успішності після будь-якого такту навчання щодо попереднього рангу  $W = 1 - p - q$ . Автономність системи навчання може бути оцінена через параметри елементів як навчальної програми  $\Pi_i$  так і для всієї кількості респондентів  $\{S_v\}$ :

$$W = z \cdot (1-\lambda) + \lambda \cdot (1-\gamma) + \gamma \cdot (1-z); \lambda = p(\Pi_i); z = p(S_v); \gamma = p(O_j).$$

У цьому випадку ми можемо стверджувати, що число випадків спостережень  $R_{vi} = 1$  (або  $R_{vi} = 0$ ) навіть за один такт функціонування системи навчання несе деяку інформацію про властивості системи навчання і її елементів.

Описано новий метод аналізу первинної інформації на основі її згортки. Для кожного респондента,  $S_v$ ,  $v=1, N_s$ , передбачається одна з трьох категорій успішності:  $N, U, K$ . Доведено теорему про аперіодичний розподіл бала тесту.

Доведення асимптотичної нормальності розподілу бала респондента цінний ще і тому, що він може бути поширеним і на слабо корельовані тести; функція розподілу нормованої суми слабо зв'язаних випадкових величин із зростом числа доданків прямує до нормального закону. Доведено, що мірою розсіювання бала тесту, тобто вихідного його параметра, є сума математичного очікування міри розсіювання, зумовленого властивостями респондентів. На підставі цього висновку у роботі стверджується, що вірогідний розподіл властивостей респондентів у групі  $N$  є немінучим. Тільки лише у випадку



чисельності респондентів категорії «U», а при аналізі якості успішності – чисельності респондентів категорії «K». Очевидно, що  $U_{CO} = C - N_{CO}$ . Для оцінки успішності в категорії «U»

$$C = N_S; N_{S2} = N_{CO}; N_{IR} = N_S - N_{CO} = C - N_{CO} = N_S - N_{S2} \rightarrow U_{CO}.$$

Для оцінки якості навчання в категорії «U»

$$N_{CO} = U_3; C = N_{IR}; U_{CO} \rightarrow N_{IR} - U_3 = C - N_{CO} = U_4;$$

де –  $N_{CO}$  число респондентів нижчої категорії. Обчислюється відносно значення числа  $U_{CO}$ , тобто  $U = \frac{U_{CO}}{C} = 1 - \frac{N_{CO}}{C}$ , де  $C$  прирівнюється обліковому числу респондентів при аналізі абсолютної успішності або числу всіх «встигаючих» при аналізі якості успішності. Очевидно, що значення  $U$  і відповідні  $U_i$  використовуються для оцінки ефективності навчання (з. за даним критерієм) для безлічі  $\{S_v\}$  у цілому і для окремих навчальних програм.

По якості навчання для категорії «U»

$$C = N_{IR} = N_S - N_{S2}; U \rightarrow U_{R(U)} = 1 - \frac{\sum d_i}{mN_{IR}}; U_i \rightarrow U_{Ri(U)} = 1 - \frac{d_i}{N_{IR}}.$$

Для категорії «U + N» по загальній успішності:

$$C = N_S; \bar{d}_i - \text{кількість незадовільних оцінок}$$

$$d_i \rightarrow \bar{d}_i \left( d_{CO} \rightarrow \bar{d}_{CO} = \sum_{i=1}^m \bar{d}_i \right); U \rightarrow U_K = 1 - \frac{\bar{d}_{CO}}{mN_S}; U_i \rightarrow U_{iK} = 1 - \frac{\bar{d}_i}{N_S}.$$

Ефективність навчання даного контингенту оцінюється за кінцевим результатом, тобто за ступенем навченості респондента кожного  $S_v$ .

Разом з тим існуючі методи аналізу мають серйозний недолік – неможливість оцінки взаємодії основних елементів ( $S, P$  і  $W$ ) системи навчання, тобто неможливість з'ясування хоча б причин відхилень ходу навчання від норми і вироблення, принаймні, напрямку керуючих впливів.

Далі розглянемо задачі синтезу математичної моделі взаємодії оператора і системи навчання.

Використання лінгвістичних правил «IF – THEN» дозволяє значно знизити обсяг експериментальних даних, необхідних для якісної ідентифікації. Для задач,

де більш важливим є обґрунтування прийнятого рішення, мають перевагу нечіткі моделі типу Мамдани :

$$\bigcup_{p=1}^{k_j} \bigcap_{i=1}^n (x_i = a_{i,jp}) \rightarrow y = d_j, j = \overline{1, m}, \quad (3.5)$$

де  $a_{i,jp}$  – лінгвістичний терм, яким оцінюється перемінна  $x_i$  в рядку з номером  $jp$  ( $p = \overline{1, k_j}$ );  $k_j$  – кількість строк-кон'юнкцій, у яких вихід  $y$  оцінюється лінгвістичним термом  $d_j$ ;  $m$  – кількість термів, які використовують для лінгвістичної оцінки вихідної перемінної  $y$ .

Представлені правила віднесення оператора до деякого когнітивного типу. Для цього вводяться додаткові змінні  $P$  і  $L$ , що описують темп і рівень складності представлення інформації. Змінні  $L$  і  $P$  визначаються параметрами  $N_1 \dots N_5$ , що є результатами тестування продуктивних характеристик користувача  $L = f_L(N_1, N_2, N_3)$ ;  $P = f_P(N_4, N_5)$  з урахуванням правила віднесення оператора до деякого когнітивного типу. Завдання ідентифікації полягає в перебуванні нечіткої моделі  $F$ , що забезпечує мінімальне значення середньоквадратичної нев'язки:

$$R = \frac{1}{M} \sum_{j=1}^M (y_r - F(X_r))^2 \rightarrow \min,$$

де  $F(X_r)$  – значення виходу нечіткої моделі при значенні входів, заданих вектором  $X_r$ . Перебування структури і параметрів нечіткої моделі, що забезпечують мінімальне значення критерію, і є задачею ідентифікації.

Зіставити рівень знань («незнань») з позначеними в них об'єктами предметної області можна, оцінивши ступінь симетрії (асиметрії) цієї інформації і дійсності.

У підсумку ентропію знань пропонується оцінювати як

$$E = f(R, S_w),$$

де  $R$  – безліч зв'язків категорій бази знань;  $S_w$  - ентропія для предметної області  $W$ , визначеної як ступінь дезорганізації фантома дійсності, тобто його ентропію максимального значення  $x$  у вибірці. Партнери, що ведуть тематичний діалог, як і



При  $N \rightarrow \infty$  і  $x_0 = I$  маємо:  $\lim_{\substack{I \rightarrow \infty \\ N \rightarrow \infty}} S = -\frac{I}{\ln 2} (\ln \alpha^{-I} + \alpha^{-I} + I)$ .

Розкриємо  $n(x)$  в загальному випадку одержимо:

$$S = -\frac{I}{\ln 2} \left[ \ln \alpha^{-I} + \alpha^{-I} + I + \ln(x_0^{-\alpha} - I^{-\alpha}) + \frac{I + \alpha}{\alpha} \cdot \frac{\ln(x_0^\alpha) / x_0^\alpha - \ln(I^\alpha) / I^\alpha}{x_0^{-\alpha} - I^{-\alpha}} \right], \quad (3.8)$$

Так як  $x_0^\alpha \square I$ , а  $I^{-\alpha} \square 1$ , то

$$S = -\frac{I}{\ln 2} \left[ \ln \alpha^{-I} + \alpha^{-I} + I - \frac{I + \alpha}{\alpha} \times \frac{\ln(I^\alpha)}{I^\alpha} \right] \quad (3.9)$$

Вираз дозволяє оцінити ступінь дезорганізації фантома дійсності через його ентропію, тобто через одну змінну  $x = I$ . А це, у свою чергу, через максимальне значення  $x$  у вибірці, число елементів якої потенційно дорівнює  $N$ .

### 3.2 Розробка структурної схеми

Розглянемо методи параметричної ідентифікації математичних моделей оцінки знань операторів АСУ. Запропонуємо підхід, що припускає проведення з оператором серії активних експериментів, у продовж яких реєструються кінцеві результати процесу оцінювання.

З безлічі математичних моделей для опису ймовірності правильної відповіді в залежності від рівня знань  $\theta_v$ ,  $v$  – того респондента та  $i$  труднощів  $i$  – того завдання обране однопараметричне нелінійне рівняння Георга Раша:

$$P_{vi} = \frac{e^{D(\theta_v - \beta_i)}}{1 + e^{D(\theta_v - \beta_i)}}, \quad (3.10)$$

де параметр  $D$  введений для того, щоб стандартизувати шкали ймовірностей для різних математичних моделей. Для  $v$  – того респондента функція Раша має вигляд:

$$P_v = \frac{e^{D(\theta_v - \beta)}}{1 + e^{D(\theta_v - \beta)}}. \quad (3.11)$$

Для одного  $i$  – го завдання функція Раша записується в такий спосіб:

$$P_i = \frac{e^{D(\theta - \beta_i)}}{1 + e^{D(\theta - \beta_i)}} \text{ при } \beta_i \rightarrow +\infty . \quad (3.12)$$

Загалом задача зводиться до визначення  $\theta$  і  $\beta$  шляхом алгоритмічних дій і послідовних ітерацій, виходячи з експериментальних даних тестування:

$$X_v \rightarrow \theta_v, R_i \rightarrow \beta_i,$$

де  $X_v$  – сумарний бал  $v$  – того тестованого, а  $R_i$  – сума правильних відповідей для  $i$  – го завдання. Для виміру цих двох латентних змінних використовують логіт – модель або модель бінарного (3.дихотомічного) вибору  $U$  і ту саму одиницю виміру – логіт:

$$\ell_{\theta_v} = \ln \frac{p_v}{q_v}; p_v = \frac{X_v}{N}; q_v = 1 - p_v \text{ логіт рівня знань } v \text{ – го респондента,}$$

$$\ell_{\beta_i} = \ln \frac{q_i}{p_i}; p_i = \frac{R_i}{k}; q_i = 1 - p_i \text{ логіт рівня труднощів } i \text{ – і задачі,}$$

де  $\ell_{\theta_v}$  і  $\ell_{\beta_i}$  – логіти рівня професійних знань  $v$  – того респондента і труднощі  $i$  – го завдання відповідно,  $p_v$  і  $q_v$  – частка правильних і неправильних відповідей для  $v$  – того респондента,  $p_i$  і  $q_i$  – частка правильних і неправильних відповідей по всіх респондентах для  $i$  – го завдання відповідно. Алгоритми обчислень розбиваються на ряд етапів: упорядковується матриця даних тестування; виробляється розрахунок початкових значень  $\beta_i^0$ . Для всіх завдань по рівню

обчислюються логіти труднощів завдань, потім їхнє середнє значення  $\bar{\ell}_{\beta} = \frac{\sum_{i=1}^k \ell_{\beta_i}}{k}$ .

У наступному етапі переносять центр розподілу логітів труднощів на завдання:  $\beta_i^0 = \ell_{\beta_i} - \bar{\ell}_{\beta}$ .

При формуванні тесту в нього потрібно включати завдання з різними значеннями  $\theta_v$  і  $\beta_i$ . Якщо тестована група однорідна за рівнем професійних знань, то в тест необхідно брати завдання з великою крутістю характеристики. Якщо ж група неоднорідна, то в тест включаються завдання з малою крутістю, але при

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>34</b>

цьому треба намагатися, щоб криві завдань не перетиналися і були розташовані по всьому простору. Величина  $\alpha_i$  обчислюється за допомогою бісеріального коефіцієнта кореляції між балами  $i$ -го завдання і сумою індивідуальних балів усіх респондентів:

$$\alpha_i = \frac{(r_{bis})_i}{\sqrt{1 - (r_{bis})_i^2}}.$$

Для системи тестового контролю професійних знань бісеріальний коефіцієнт кореляції описується залежністю:

$$(r_{bis})_i = \frac{(\bar{X}_1)_i - (\bar{X}_0)_i}{S_x} \cdot \frac{(n_1)_i (n_0)_i}{N_i \cdot N \sqrt{N(N-1)}}, \quad (3.13)$$

де  $(\bar{x}_i)_i$  – середній сумарний бал тих  $N_i$  респондентів, що на  $i$ -те завдання дали правильну відповідь ( $X_{vi} = 1$ ),  $(\bar{X}_0)_i$  – теж саме для тих респондентів, що на  $i$ -те завдання дали неправильну відповідь ( $X_{vi} = 0$ );  $n_{1i}$  і  $n_{0i}$  – кількість правильних і неправильних відповідей на  $i$ -те завдання;  $u_i$  – ордината функції нормального розподілу в точці;  $S_x$  – вибіркова дисперсія.

Розробка математичної моделі рівняння зв'язку для системи навчання починається з після оптимізаційного етапу. Після оцінки погодженості і конформності питання виявилися достатніми для віднесення тесту до класу однорідних, з'являється можливість використання такого оптимального тесту в якості системного при програмованому контролі знань. Основою для такої можливості є визначені функціональні зв'язки параметрів тесту (тобто деяких констант, що задають модель тесту) з тестовими показниками (тобто величинами, що спостерігаються в процесі тестування: числами вірних відповідей на питання тесту). Зрозуміло, що ці зв'язки є детермінованими тільки стосовно деякої генеральної сукупності респондентів, чисельність якої  $N$  достатня для застосування граничних теорем, тобто оперування поняттями ймовірностей, математичних сподівань і т.п. замість частот, середніх значень і т.д. Як модель тесту використовується цілком визначений функціональний зв'язок ймовірностей

для випадкової події  $X_{vi} = 1$  ( $v$ -й респондент дав вірну відповідь на  $i$ -те питання тесту):

$$\begin{cases} P\left[X_{vi} = 1 / A_v\right] = S; & P\left[X_{vi} = 1 / \bar{A}_v\right] = P_i, \\ P\left[X_{vi} = 1\right] = Z_v * S + (1 - Z_v)P_i = \hat{O}_{vi}, \end{cases} \quad (3.14)$$

де параметри  $A$ -тесту  $S$  і  $P_i$  є константами і визначають  $S$  – ймовірність вірної відповіді за умови знання;  $P_i$  – ймовірність вірної відповіді за умови незнання, тобто ймовірність угадування. Обидва ці показники є випадковими величинами і мають асимптотично нормальні розподіли. Розглянемо спочатку розподіл ваги питання  $B_i$  з урахуванням однорідності тесту, для будь-якого  $B_i$  маємо

$$\begin{cases} \hat{M}(B_i) = (S - P_i) \sum_{v=1}^N Z_v + NP_i; \\ D(B_i) = (S - P_i)(1 - 2P_i) \sum_{v=1}^N Z_v - (S - P_i)^2 \cdot \sum_{v=1}^N Z_v^2 + NP_i(1 - P_i). \end{cases} \quad (3.15)$$

Вхідні у виразі (3.15) для  $M(B_i)$  і  $D(B_i)$  суми  $\sum_v Z_v$  і  $\sum_v Z_v^2$  розглядаються як параметри розподілу рівнів знань у генеральній сукупності  $N$ . Дійсно, кожний з респондентів у цій сукупності володіє деяким  $Z_v$  (тобто визначеною ймовірністю знання). Звернемося далі до другого тестового показника, так званому бала тесту  $K$ . Кожна з умовних ймовірностей  $k_v = j$  у зв'язку з однорідністю тесту відшукується у вигляді:

$$\begin{cases} P\left[k_v = j / A_v\right] = C_m^j S^j (1 - S)^{m-j}; \\ P\left[k_v = j / \bar{A}_v\right] = C_m^j P_i^j (1 - P_i)^{m-j}, \end{cases} \quad (3.16)$$

що дозволяє знайти ймовірність  $P[k = j]$  для бала тесту у вигляді  $P[k = j] = C_m^j S^j (1 - S)^{m-j} M(Z) + C_m^j P_i^j (1 - P_i)^{m-j} [1 - M(Z)]$ . Отже, при відомих параметрах розподілів ваги питання  $M(B_i)$  і  $D(B_i)$ , а також бала тесту  $M(K)$  і  $D(K)$  є можливість скласти рівняння зв'язку цих величин з параметрами тесту  $S$  і  $P_i$ , а також параметрами розподілу рівнів знань  $M(Z)$  і  $D(Z)$  у вигляді системи рівнянь:

$$\begin{cases} M(K) = M(S - P_i)M(Z) + MP_i; \\ M(B_i) = N(S - P_i)M(Z) + NP_i; \\ D(K) = [1 + (M - 1)(S + P_i) - 2MP_i]M(S - P_i)M(Z) - \\ - M^2(S - P_i)^2 M(Z) + MP_i(1 - P_i); \\ D(B_i) = M(Z)(S - P_i)(1 - 2P_i) + NP_i(1 - P_i) - \\ - N(S - P_i)^2 D(Z) - N(S - P_i)^2 M^2(Z). \end{cases} \quad (3.17)$$

Перші два рівняння в системі не є незалежними, тому що з теорії дихотомічних тестів випливає очевидне співвідношення:

$$\frac{M(K)}{M} = \frac{M(B_i)}{N} = \frac{M(B)}{MN} = \alpha, \quad (3.18)$$

де  $B_i = jN_j = \sum_{v=1}^N X_{vi}$ ;  $B = \sum_{i=1}^m B_i$  – вага тесту. Таким чином, система дозволяє розв'язок по відношенню тільки трьох невідомих  $S; P_i; M(Z); D(Z)$ , і то за умови, що попередньо знайдено математичні сподівання і дисперсії тестових показників  $K$  і  $B_i$ . Крім того, необхідно врахувати, що питання оцінок довірчих інтервалів вирішені у теоретичній статистиці лише для моментів двох перших порядків, хоч оцінки максимальної правдоподібності можна знайти для моментів будь-якого порядку.

У цих умовах переважнішим виявляється шлях використання цієї системи рівнянь, що дозволяє при відомому  $P_i$  відшукати параметр тесту  $S$  у вигляді

$$S = \frac{D(K) + M^2(K)(1 - P_i + MP_i)}{(M - 1)[M(K) - MP_i]}.$$

Що ж стосується ймовірності угадування  $P_i$ , то цей параметр діагностичного тесту можна вважати рівним  $P_i = \frac{1}{H}$ , де  $H = 2, 3, \dots, n$  – число відповідей до питань тесту. За умови підвищених вимог до вірогідності оцінки  $S$  можна знаходити параметр  $P_i$  методом рандомізації підсумків тесту. Неминучість використання, що відзначалася  $M(K)$  вище,  $D(K)$  замість їхніх оцінок приводить до необхідності розгляду  $D(K) = f[M(K)]$  для дослідження  $S = const$  функції  $P_i = const$ .

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Для границь довірчих інтервалів дисперсії  $D'_K$  і  $D''_K > D'_K$ ;  $D'_B$  і  $D''_B > D'_B$  і відповідних коефіцієнтів довіри  $Q_K = 1 - \beta_K$ ,  $Q_B = 1 - \beta_B$  можна скласти рівняння зв'язку у вигляді:

$$\begin{cases} P_{m-1}\{X''_B\} = 1 - \frac{\beta_B}{2}; & P_{m-1}\{X'_B\} = \frac{\beta_B}{2}; \\ X''_B = \frac{S_B^2}{D_B''}(M-1); & X'_B = \frac{S_B^2}{D_B'}(M-1); \\ \hat{O}_0\left\{\frac{X''_K - N + 1}{\sqrt{2N-2}}\right\} = \frac{\beta_K}{2}; & \hat{O}_0\left\{\frac{X'_K - N + 1}{\sqrt{2N-2}}\right\} = 1 - \frac{\beta_K}{2}; \\ X''_K = \frac{S_K^2}{D_K''}(N-1); & X'_K = \frac{S_K^2}{D_K'}(N-1). \end{cases} \quad (3.19)$$

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано структуру системи ідентифікації професійних знань операторів автоматизованих систем управління.

Сервіс тестувальника професійних знань оператора автоматизованих систем управління, містить у собі редактор тестів, що дозволяє створювати нові завдання, з'ясовувати наявність повторень питань у темах, відсутність правильних і неправильних варіантів відповіді.

Формування завдання. Відповідно до вказівок тестувальника цей блок створює сценарій перевіркової роботи для кожного оператора автоматизованих систем управління, випадковим образом вибираючи з банку питань обумовлене викладачем кількість завдань по кожній темі.

Банк питань. База тестових завдань, що на даний момент складається з питань двох типів:

- Відкриті тестові завдання.
- Багатоальтернативні тестові завдання.

Банк відповідей містить правильні відповіді до кожного завдання.

Блок управління забезпечує звірення даного оператором автоматизованих систем управління відповіді зі змістом банку відповідей.

Блок формування оцінок. Розраховує підсумкову оцінку й виводить її на екран.



всі можливі положення респондента рівноймовірні. Так чи інакше завжди є можливість оцінити ступінь невизначеності наших представлень про  $\{Z_v\}$ , причому мірою цієї невизначеності є ентропія. Звичайно, після тестування залишається деяка невизначеність (менша або, у крайньому випадку, рівна  $\dot{Y}(Z_v)$  уявлень про безліч  $\{Z_v\}$ ). Ця так названа залишкова ентропія  $\dot{Y}(Z_v/R_v)$  і дозволяє відтворити кількість інформації  $\{Z_v\}$ , що утримується в реакціях респондента ( $R_v$ ) на всі  $m$  питання тесту:  $J(z_v/R_v) = \dot{Y}(z_v) - \dot{Y}(z_v/R_v)$ . Визначення залишкової або умовної ентропії  $\dot{Y}(Z_v/R_v)$  задає певних труднощів. Відповідно до визначення кількості інформації про подію (з. величину)  $A_v$ , що утримується в події  $B_j$ , можна знайти  $J(A_v/B_j) = \log \frac{p(A_v/B_j)}{p(A_v)} = \log \frac{p(B_j/A_v)}{p(B_j)}$ , і тоді в середньому в безлічі  $\{B_j\}$  утримується інформація про  $\{A_v\}$ , чисельно рівна  $J(A/B) = \sum_v \sum_j p(A_v \cap B_j) \cdot J(A_v/B_j)$ , де  $p(A_v/B_j)$  – ймовірність  $A_v$  за умови  $B_j$ ;  $p(A_v) = \sum_j p(A_v/B_j) \cdot p(B_j)$  – безумовна ймовірність  $A_v$ ;  $p(A_v \cap B_j)$  – ймовірність спільної події  $A_v$  і  $B_j$ .

Використання відомих положень теорії ймовірностей дозволяє представити  $J(A/B)$  у вигляді

$$\begin{aligned} J(A/B) &= \dot{Y}(B) - \dot{Y}(B/A) = \\ &= -\sum_j p(B_j) \log p(B_j) + \sum_v p(A_v) \sum_j p(B_j/A_v) \times \log p(B_j/A_v). \end{aligned} \quad (3.20)$$

Відповідно до прийнятих в даній роботі позначень ймовірність вірної відповіді (подія  $X_{vi} = 1$ )  $v$ -го респондента на  $i$ -те питання тесту є функцією латентних ознак респондента ( $Z_v$ ) і питання ( $t_i, p_i$ ), тому  $\varphi_{vi} = f(z_v; t_i; p_i) = p(x_{vi} = 1/z_v)$ . Ця умовна ймовірність однієї з можливих реакцій на  $i$ -те питання тесту дозволяє знайти вираз для кількості інформації, що подається одним питанням тесту.

$$\begin{cases} \hat{O}_{vi} = p(x_{vi} = 1) = \sum_{z_v} \varphi_{vi} \cdot p(z_v); \\ 1 - \hat{O}_{vi} = p(x_{vi} = 0) = \sum_{z_v} (1 - \varphi_{vi}) \cdot p(z_v). \end{cases} \quad (3.21)$$

Тоді  $J(z_v/x_{vi}) = \dot{Y}(z_v) - \dot{Y}(z_v/x_{vi})$  причому  $\dot{Y}(z_v/x_{vi})$  – суть залишковий рівень невизначеності суджень про  $Z_v$  по реакціях на питання  $i$ . Знайдемо цю ентропію як:

$$\begin{aligned} \dot{Y}(z_v/x_{vi}) = & -\hat{O}_{vi} \sum_{z_v} \frac{p(z_v) \cdot \varphi_{vi}}{\hat{O}_{vi}} [\log p(z_v) + \log \varphi_{vi} - \log(\hat{O}_{vi})] - \\ & -(1 - \hat{O}_{vi}) \cdot \sum_{z_v} \frac{p(z_v) \cdot (1 - \varphi_{vi})}{1 - \hat{O}_{vi}} [\log p(z_v) + \log(1 - \varphi_{vi}) - \log(1 - \hat{O}_{vi})]. \end{aligned}$$

Очевидно, що найбільшою інформативністю буде володіння таким питанням тесту, для якого досягається  $\text{Inf}_{\varphi_{vi}} \{\dot{Y}(z_v/x_{vi})\}$ , однак у відношенні тестів з розривною моделлю. Залишкова ентропія для кожного з питань такого тесту  $i$  не дорівнює нулеві, тобто одне питання тесту не передає всю інформацію про  $\{Z_v\}$ , тому що  $J(z_v/x_{vi}) = \dot{Y}(z_v) - \dot{Y}(z_v/x_{vi}) = -p(z_v \geq t_i) \times \log p(z_v \geq t_i) - p(z_v < t_i) \cdot \log p(z_v < t_i) < \dot{Y}(z_v)$  і подібний тест повинен бути обов'язково неоднорідним.

Усе сказане вище дозволяє вважати тести з розривними моделями питань (обов'язково неоднорідними) – еталонними тестами, але з істотним застереженням: результат тестування повинний розглядатися з  $m$  букв двоїстого коду, тобто необхідно враховувати топологію вірних відповідей. Запропонована модель однорідного діагностичного тесту дозволяє розглядати процес тестування як процес передачі інформації з деякого каналу зв'язку (тестові) від респондента до того, хто перевіряє. У такому каналі зв'язку кількість інформації про діагноз, поставлений тестом обсягом  $m$ , визначиться як:

$$J_z^m(p;t) = \sum_{j=0}^m \left[ z \cdot C_m^j (1-t)^j \cdot t^{m-j} \cdot \log \frac{1}{z + (1-z) \cdot E_j} + (1-z) \cdot C_m^j \cdot p^j \cdot (1-p)^{m-j} \cdot \log \frac{E_j}{z + (1-z) \cdot E_j} \right],$$

тут  $E_j = \frac{p^j (1-p)^{m-j}}{(1-t)^j \cdot t^{m-j}}$ .

Найбільша ж кількість інформації, тобто початкова (апріорна) ентропія діагнозу складає  $\dot{Y}(A) = -z \log Z - (1-z) \log(1-z)$ . В теорії інформації цим поняттям

визначають найбільшу кількість інформації, що здатна передати канал зв'язку в одиницю часу, тобто

$$\text{Sup}_{z;m} \left\{ \frac{J_z^m}{m} \right\} = \max_z J_z^1 = C,$$

де  $J_z^1$  – кількість інформації, переданої за один крок (у нашому випадку утримується в одній відповіді на питання тесту),

$$C = \max_z J_z^1 = \ln \frac{w+1}{wp} - \dot{Y}(p) \quad (3.22)$$

причому  $w = \exp \left[ \frac{\dot{Y}(p) - \dot{Y}(t)}{U} \right]$ .

Доказана теорема про інформативність тесту.

Знайдемо оцінку достатнього обсягу однорідного лінійного метричного тесту.

Звертаючись до принципової відмінності розподілів  $p(k^*)$  і  $R(k^*)$ , слід акцентувати увагу на можливість грубої оцінки інформативності однорідного лінійного метричного тесту. Кількість інформації, що утримується в цих розподілах, складає

$$\begin{cases} \dot{Y}(k^*) = -\sum_{g=0}^m p(k^*=g) \cdot \log p(k^*=g); \\ \dot{Y}^\circ(k^*) = -\sum_{g=0}^m R(k^*=g) \cdot \log R(k^*=g). \end{cases} \quad (3.23)$$

Найбільше значення  $\dot{Y}^T(k^*)$  визначається виразом

$$\max_z \left\{ \dot{Y}_m^\circ(k^*) \right\} = m - \frac{\sum_{g=0}^m C_m^g \cdot \log_2 C_m^g}{2^m} \quad (3.24)$$

Середня кількість інформації про «рівні знань»  $k_o^* = g$ , що утримується в підсумках однорідного лінійного метричного тесту, складає

$$\begin{aligned} J_m^z(k^*/k) &= \dot{Y}(k^*) - \dot{Y}(k^*/k) = \\ &= m \left[ \dot{Y}(\hat{O}) - (1-z) \dot{Y}(p) \right] + \delta(m; \hat{O}; z) - \rho(m; z) = \\ &= m \left[ \dot{Y}(\hat{O}) - (1-z) \dot{Y}(p) \right] + \sum_{j=0}^m C_m^j \hat{O}^j (1-\hat{O})^{m-j} \sum_{g=0}^j C_j^g \left( \frac{z}{\hat{O}} \right)^{j-g} \cdot \log \frac{C_j^g}{C_m^g} \end{aligned}$$

Розглянемо логічні передумови оцінки адекватності моделі компараторної ідентифікації рівня знань. У модель тесту, не визначену поки що функціонально, вводиться одна ознака респондента – «рівень знань»  $z_v$  і одна ознака питання – «трудність»  $t_i$ . Тоді з логічних передумов

$$\begin{cases} \frac{\partial \varphi_{vi}}{\partial z_v} > 0; & z_{min} \leq z_v \leq z_{max}; \\ \frac{\partial \varphi_{vi}}{\partial t_i} < 0; & t_{min} \leq t_i \leq t_{max}, \end{cases} \quad (3.25)$$

впливає, що ймовірність вірної відповіді  $\varphi_{vi}(z_v; t_i)$  повинна рости в міру росту рівня знань, зменшуватися в міру росту труднощів питань тесту. Логічні передумови дозволяють оцінити адекватність моделі тесту в граничних умовах, при  $z_v$  і  $t_i$  нуль і нескінченність, тобто

$$\begin{cases} \varphi_{vi}(z_v = 0; t_i) = 1; \\ \varphi_{vi}(z_v; t_i = \infty) = 1. \end{cases} \quad (3.26)$$

Якщо ж тест допускає угадування, то його ймовірність повинна входити в рівняння моделі, причому

$$\begin{cases} \frac{\partial \varphi_{vi}}{\partial p_i} > 0; \\ \varphi_{vi}(z_v = 0; t_i; p_i) = p_i; \\ \varphi_{vi}(z_v; t_i = \infty; p_i) = p_i. \end{cases} \quad (3.27)$$

Досліджено евристичні методи оцінки адекватності моделі рівня знань. Хоч ці дослідження ще не можна визнати вичерпними, проте можна сформулювати деякі практично важливі положення, що впливають із властивостей однорідного тесту.

Розглянемо задачі ідентифікації методів оцінки математичної моделі вірогідності системи навчання операторів.

Параметри тестового контролю змінюються: нестационарні процеси і їх вимір протікає на фоні нестационарних, випадкових перешкод; при цьому досліджуємо модель процесу, що спостерігається. Надалі будемо використовувати спеціальну модель  $a_0$  контрольованого процесу



2. Ймовірність помилок другого роду, тобто ймовірність настання помилкової події, що полягає в тому, що непрацездатний респондент визнається при тестовому контролі працездатним, рівні:

$$\beta(t) = \int_{\infty}^{a_1} f_1[x(t)] \left[ \int_{a_2-x}^{b_2-x} f_2(\zeta) d\zeta \right] dx + \int_{b_1}^{\infty} f_1[x(t)] \left[ \int_{a_2-x}^{b_2-x} f_2(\zeta) d\zeta \right] dx. \quad (3.32)$$

Умовні ймовірності помилок першого роду  $P_e$  і другого роду  $P_H$ , що характеризують точність тестових вимірів який використовують для тестового контролю, розраховують за формулами:

$$D_e(t) = \frac{\alpha(t)}{P(t)}, \quad P_H(t) = \frac{\beta(t)}{1-P(t)}, \quad (3.33)$$

де апіорі-ймовірність того, що контрольований параметр знаходиться в полі довірчих інтервалів, знаходиться за формулою

$$P(t) = \int_{a_1}^{b_1} f[x(t)] dx. \quad (3.34)$$

Розроблено алгоритм методики визначення помилок першого і другого роду.

Ймовірність помилки першого роду в мажоритарному контролі

$$\alpha(k, m) = \sum_{i=k}^m C_m^i \alpha_1^i (1 - \alpha_1)^{m-i}.$$

Розглянемо задачу ідентифікації моделей оптимізації систем оцінки знань.

Нехай  $Z(N)$  – випадкові витрати виробництва на виготовлення  $N$  одиниць товару (з.сировини, виробів, послуг і т.д.), що включає в себе  $Z_0(N)$  – постійну складову (технічну вартість системи навчання) і  $Z_1(N)$  – перемінну складову (з.витрати на організацію системи навчання і  $I$  контролю професійних знань операторів). Знайдемо оптимальні витрати  $Z_{i_{ro}}(N)$ , що забезпечуються так  $\mu_k$  ( $k = 2, 3, \dots$ )-им управлінням, при якому відхилення інформативності професійних знань від «шаблонних» оптимальних альтернатив на безлічі буде множинним. У такій постановці реалізується оптимальне технологічне управління виробничою системою. Допустимо, що питома вага складових  $Z_0(N)$  і

$Z_1(N)$  рівні відповідно  $q$  і  $1-q$ . Величини  $q$  і  $1-q$  зручно трактувати як ймовірності появи відповідних витрат у загальних витратах:

$$Z(N) = qZ_0(N) + (1-q)Z_1(N). \quad (3.35)$$

Розглядаючи  $Z_0(N)$  і  $Z_1(N)$  як реалізації випадкової величини витрат  $Z_0(N)$ , що з'являються з ймовірностями  $q$  і  $1-q$ , неважко визначити початкові  $m_k$  ( $k=1, 2, \dots$ ) і центральні моменти сумарних витрат  $Z(N)$ :

$$m_k [Z(N)] = Z_0^k(N)q + Z_1^k(N)(1-q), \quad k=1, 2, \dots; \quad (3.36)$$

$$\mu_k [Z(N)] = \{Z_0(N) - m_1 [Z(N)]\}^k q + \{Z_1(N) - m_1 [Z(N)]\}^k (1-q) \quad (3.37)$$

По цих моментах, використовуючи ортогональний розклад функції розподілу в ряд Грамма-Шарльє, неважко синтезувати функцію розподілу  $F[Z(N)]$ . Використання цього розподілу дозволяє розв'язувати ряд задач оптимального управління, визначати умови існування й єдності оптимальних рішень.

Виберемо в ролі критерію оптимальності перший початковий момент сумарних витрат – середні сумарні витрати

$$m_1 [Z(N)] = Z_0(N)q + Z_1(N)(1-q) \quad (3.38)$$

і визначимо необхідні умови існування й єдності оптимального рішення. Диференціюючи функцію (3.38) по  $N$  і дорівнюючи результат нулеві, одержимо рівняння оптимізації у вигляді:

$$\frac{\frac{\partial Z_1(N)}{\partial N}}{\frac{\partial Z_0(N)}{\partial N}} = \frac{q}{1-q}. \quad (3.39)$$

Для того, щоб  $N_{opt}$ , отримане з розв'язку рівняння (3.39), доставляло мінімум функції (3.38), необхідно виконання умови:

$$\frac{\partial^2 Z_0(N)}{\partial N^2} q > -\frac{\partial^2 Z_1(N)}{\partial N^2} (1-q). \quad (3.40)$$

З аналізу умови (3.39) можна зробити висновок про те, що екстремум досягається в точці, де досягається рівність похідних  $Z'_0(N)$  і  $Z'_1(N)$ , взятих з

вагами  $q$  і  $1-q$ . При  $q=1-q=\frac{1}{2}$  досягається рівність похідних за абсолютним значенням.

Розроблено метод ідентифікації математичних моделей оптимізації за критерієм максимального прибутку.

Критерій оптимальності представимо в наступному вигляді

$$\pi(d) = N(d)d - N(d)Z_1 - Z_0 = N(d)(d - Z_1) - Z_0, \quad (3.41)$$

де  $\pi(d)$  – прибуток,  $Z_0$  - постійна складових витрат,  $Z_1$  - перемінна складових витрат на одиницю товару.

Застосовуючи класичний метод визначення екстремуму функції однієї змінної, диференціюється (3.41) по  $d$  і прирівнюючи результат нулеві, одержимо наступне рівняння оптимізації

$$N'(d)(d - Z_1) + N(d) = 0, \quad (3.42)$$

де  $N'(d) = \frac{\partial N}{\partial d}$  – похідна від  $N$  по  $d$  в точці екстремуму.

Розв'язавши рівняння (3.42), одержимо функціональне рівняння для визначення оптимальної ціни

$$d_{opt} = Z_1 - \frac{N(d_{opt})}{N'(d_{opt})}, \quad (3.43)$$

$$\text{де величини } Z_1 \text{ і } \Delta Z = -\frac{N(d_{opt})}{N'(d_{opt})} \quad (3.44)$$

можна розглядати відповідно як нульове наближення для значення  $d_{opt}$  і деяке виправлення оптимального рішення на наступний крок ітерації.

Доказана теорема про існування оптимального рішення. Необхідні умови оптимізації режиму термінального управління, що відстежують викладені в доведеній теоремі рішення про оптимальне термінальне управління, доведені.

Задача оптимізації розглядалася в наступній постановці. Перемінні  $N_0$ ,  $d$ ,  $d_0$ ,  $z_0$ ,  $z_1$  розглядаються як випадкові гауссові величини, математичні сподівання і дисперсії яких відомі. Функції  $N(d)$ ,  $\pi(d)$ ,  $F(d, \pi)$  розглядаються як нелінійні функції цих випадкових аргументів. Визначення дисперсії  $D[F(g, d, \pi)]$  дозволяє

ставити і розв'язувати задачу оптимального вибору вагових коефіцієнтів, мінімізуючи дисперсію по критерію оптимізації

$$D\{F_1[\pi(d), \pi_{max}; d, d_{opt}; g]\} = g^2 D[F_{11}] + (1-g)^2 D[F_{12}] \quad (3.45)$$

Задачу оптимізації вагових коефіцієнтів  $g$  і  $1-g$  розглядали в наступній постановці:

$$g_{opt} = \arg \min D\{F_1[\pi(d), \pi_{max}; d, d_{opt}; g]\}, \quad (3.46)$$

тоді

$$D_{\min}\{F_1(g_{opt})\} = \min_g D\{F_1(g)\}. \quad (3.47)$$

Диференціюючи дисперсію по  $g$  і дорівнюючи результат нулеві, одержимо:

$$g_{opt} = D(F_{12}) / [D(F_{11}) + D(F_{12})]. \quad (3.48)$$

Визначимо дисперсії  $N(d)$  і  $N(d_{opt})$  як дисперсії функції трьох аргументів  $N_0$ ,  $d_0$ ,  $d$  і, підставляючи це значення в рівняння (3.45), знайдемо:

$$\begin{aligned} D[N(d)] &\approx \left(\frac{\partial N}{\partial N_0}\right)^2 D_{N_0} + \left(\frac{\partial N}{\partial d}\right)^2 D_d + \left(\frac{\partial N}{\partial d_0}\right)^2 D_{d_0} \approx \\ &\approx D_{N_0} e^{-2\left(1 + \frac{z_{10} + \square d}{d_{00}}\right)} \left\{ 1 + \frac{N_{00}^2}{D_{N_0}} \left[ \frac{\square d^2}{d_{00}^2} + \frac{D_{d_0} + D_{z_1}}{d_{00}^2} \left[ 1 + \left(1 + \frac{z_{10} + \square d}{d_{00}}\right)^2 \right] \right] \right\}, \end{aligned} \quad (3.49)$$

$$\begin{aligned} D[N(d_{opt})] &\approx \left(\frac{\partial N}{\partial N_0}\right)^2 D_{N_0} + \left(\frac{\partial N}{\partial d_0}\right)^2 D_{d_0} + \left(\frac{\partial N}{\partial z_1}\right)^2 D_{z_1} \approx \\ &\approx D_{N_0} e^{-2\left(1 + \frac{z_{10}}{d_{00}}\right)} \left\{ 1 + \frac{N_{00}^2}{D_{N_0}} \left[ \frac{\square d^2}{d_{00}^2} + \frac{D_{d_0} + D_{z_1}}{d_{00}^2} \left[ 1 + \left(1 + \frac{z_{10}}{d_{00}}\right)^2 \right] \right] \right\}. \end{aligned} \quad (3.50)$$

Функціональна схема розробленої системи зображена на рисунку 3.2.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48



## **Блок ідентифікації професійних знань операторів автоматизованих систем управління**

- Реєстрація операторів автоматизованих систем управління при вході в програму.
- Вибір тесту зі списку тестів заданого курсу, у тому числі й випадковий.
- Режим випадкового вибору питань, наприклад, 10-і питань тесту з 100 можливих.
- Ще один режим випадкового вибору питань: задане число питань по кожній темі.
- Режим тестування: задане число спроб виконання тесту.
- Відображення результатів пройдених даним оператором автоматизованих систем управління тестів.
- Проходження оператором автоматизованих систем управління тесту із записом результатів у загальну базу даних.
- Відповідь на питання може здійснюватися шляхом вибору одного або декількох правильних відповідей або шляхом введення відповіді із клавіатури.
- Під час проходження тесту можливий пропуск окремих питань із наступним поверненням до них.
- Під час тестування користувачеві доступна інформація про поточні підсумки виконання тесту й прогнозованої оцінці.
- Обчислення оцінки й відображення результатів проходження тесту.
- Є режим навчання, у якому дозволене проходити тест кілька разів без занесення результату в базу даних.
- У режимі навчання після кожної відповіді можливий вивід докладного опису ("Чому").
- Під час відповіді на питання можливий запуск додаткових завдань: файлів Word, Excel, аудіо-відео й т.п.
- Автоматичне створення резервних копій бази даних.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>50</b>

## **Блок тестувальника професійних знань операторів автоматизованих систем управління**

– Реєстрація тестувальника професійних знань операторів автоматизованих систем управління (з обов'язковим уведенням пароля).

– Редагування довідників операторів автоматизованих систем управління, тестувальників професійних знань операторів автоматизованих систем управління, комп'ютерів.

– Редагування вмісту тестів, описів відповідей і критеріїв оцінок.

– Налаштування режимів випадок випадкового вибору питань: приналежність питань до тем.

– Вивід і аналіз результатів проходження операторами автоматизованих систем управління тестів.

– Керування тестуванням і навчанням у комп'ютерних класах з робочого місця тестувальників професійних знань операторів автоматизованих систем управління, комп'ютерів.

– Друк різних списків і звітів.

– Експорт і импорт даних через текстові файли й файли Excel.

– Кожне питання може мати до 8-і відповідей, декілька з яких можуть бути правильними.

– Можливо автоматичне формування тестів шляхом випадкового вибору, наприклад, 10-і з 100 можливих питань.

– З кожним питанням може бути зв'язане додаткове завдання (файли Word, Excel, HTML або будь-які інші), що задається під час тестування або навчання, і докладне пояснення відповіді, що доступно після відповіді на питання.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51



Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

КБПЗ - 2023

					ВКРМ-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

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

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

Перед розглядом подробиць схеми роботи програми розглянемо виконані основні напрацювання.

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 та 4.3 зображено роботу підпрограм. З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограм та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограм виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання. Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи ідентифікації професійних знань операторів автоматизованих систем управління. При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>54</b>

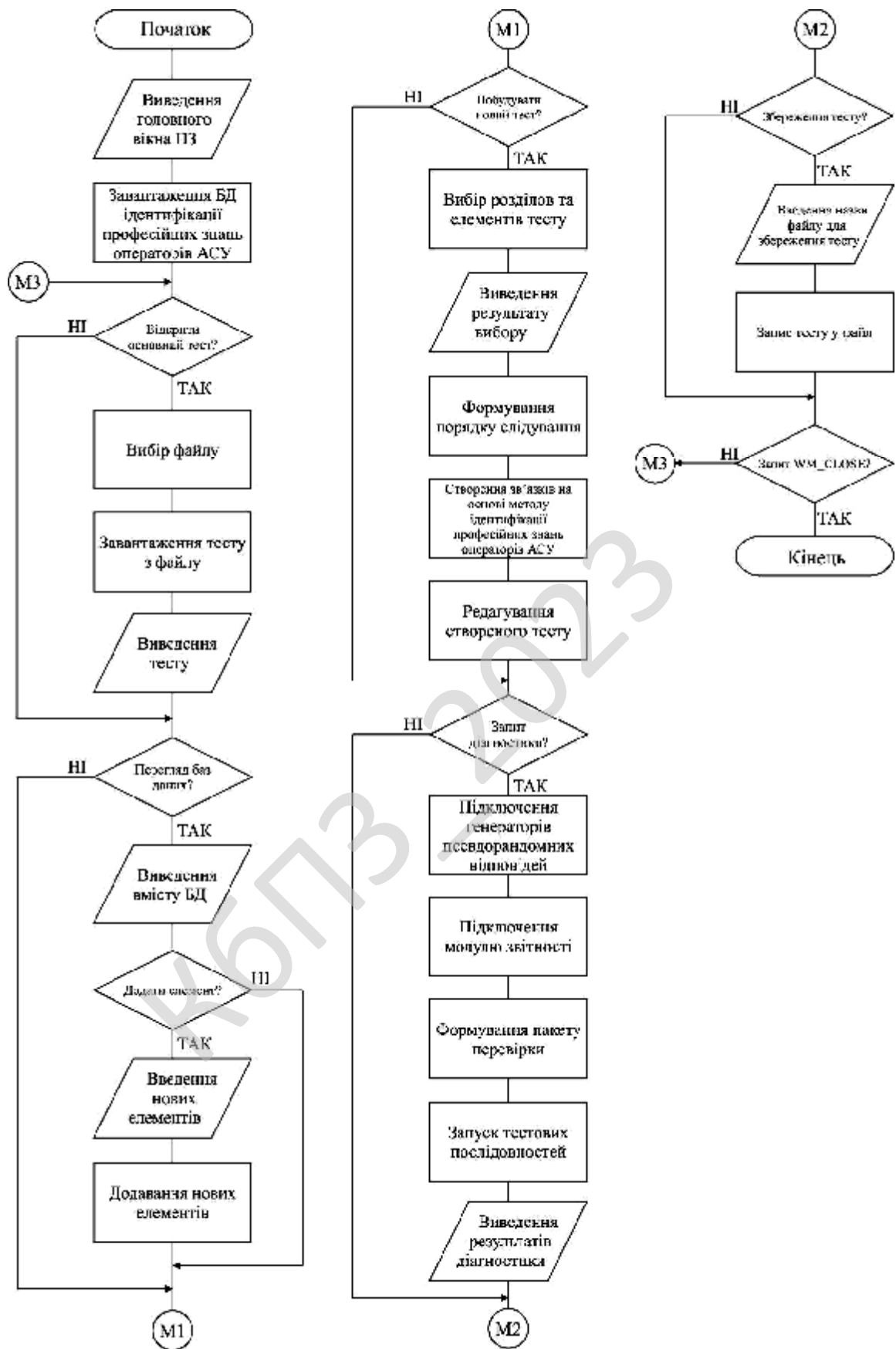


Рисунок 4.1 – Блок-схема основної програми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML.

Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

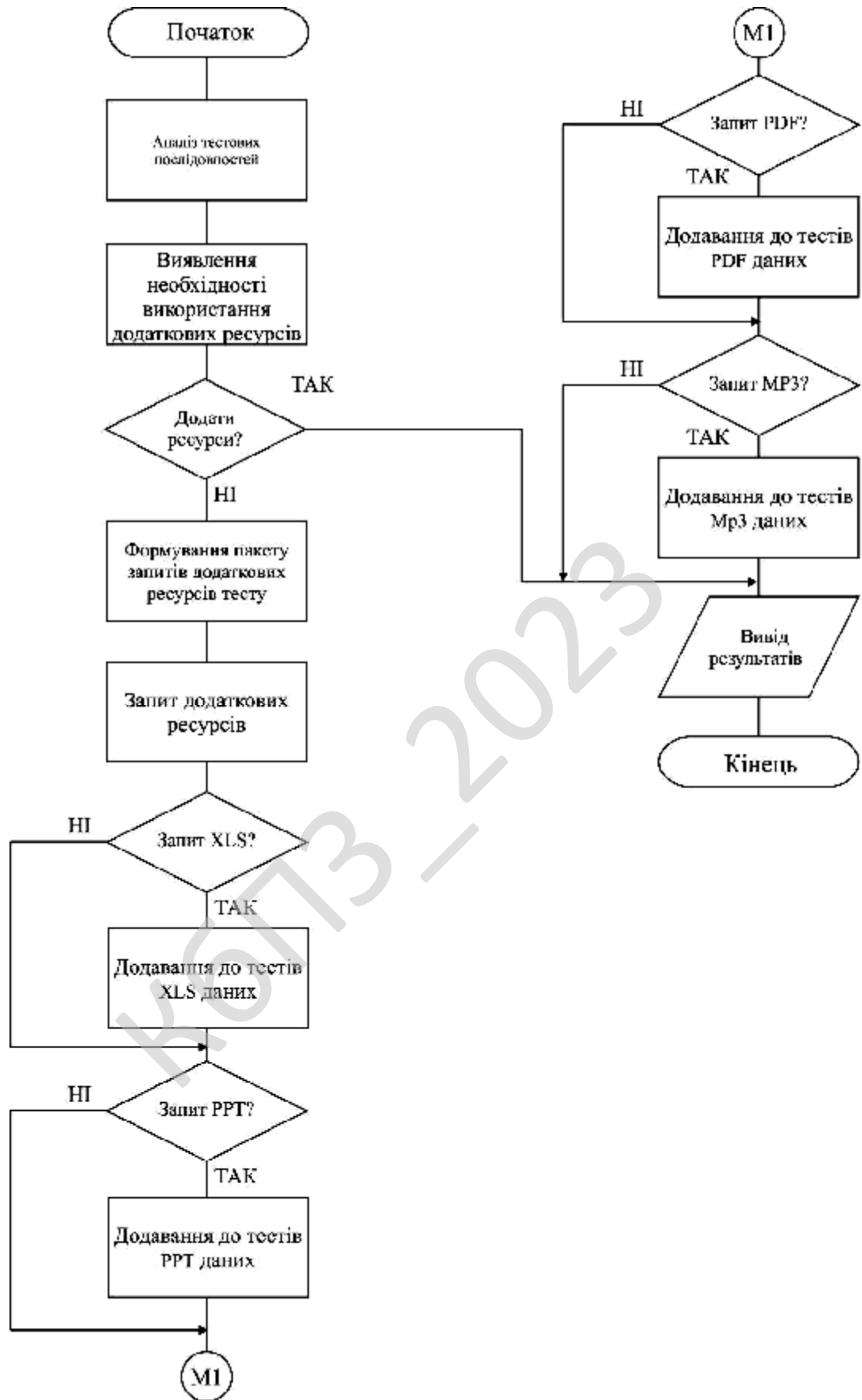


Рисунок 4.2 – Блок-схема роботи підпрограми

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

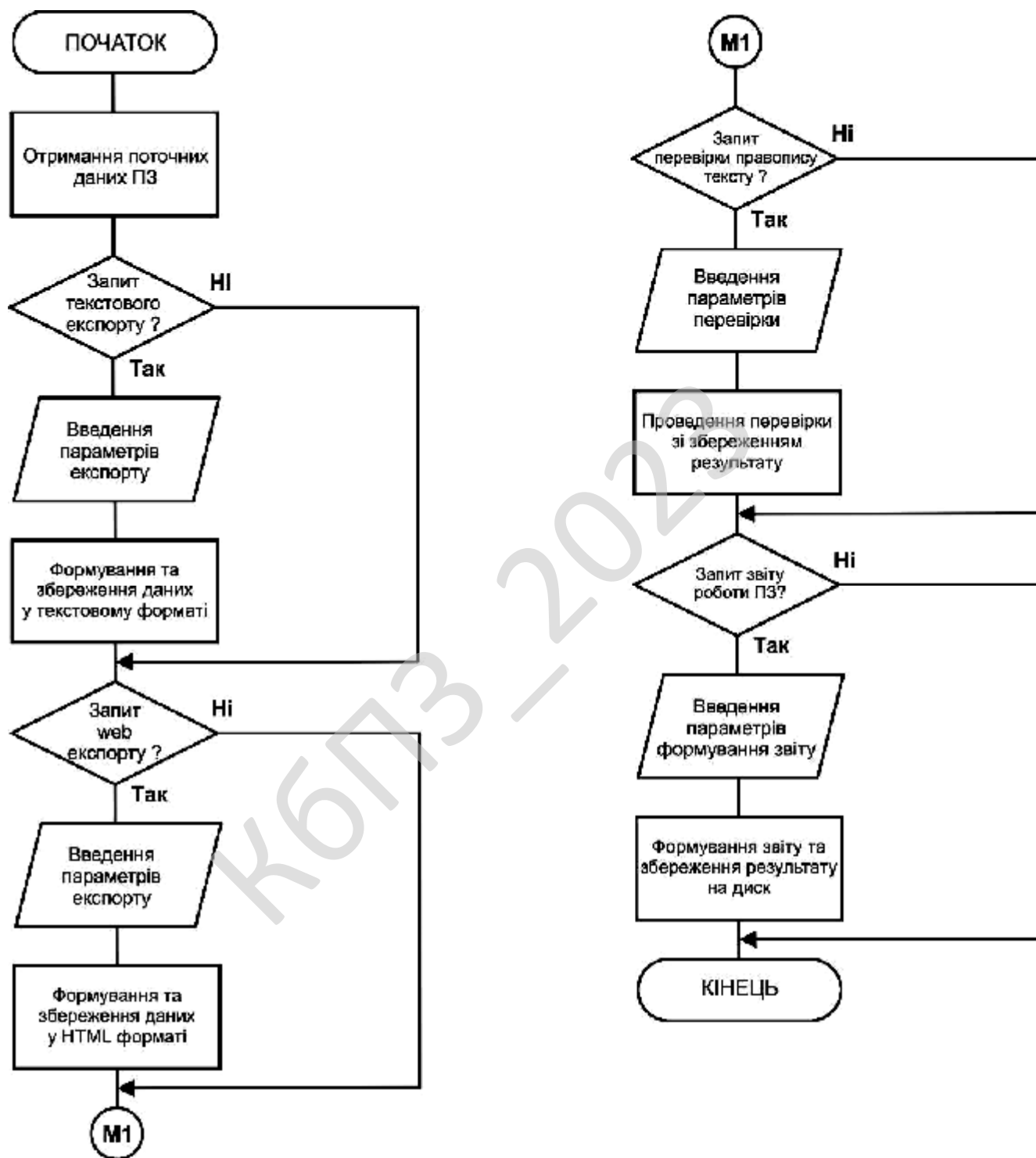


Рисунок 4.3 – Блок-схема роботи підпрограми



```

end;
end;
//збереження параметрів програми
Procedure TForm1.SaveCfg;
begin
  IniFile.WriteInteger('Colors','Desktop',Desktop_Color);
  IniFile.WriteInteger('Colors','Grid',Grid_Color);
  IniFile.WriteBool('Selection','Full',Selection_Metod);
IniFile.WriteBool('Creation','Selection',Select_on_creation);
IniFile.WriteBool('Selection','Draw_conn',Draw_connections);
IniFile.WriteBool('Selection','Sel_conn',Select_connections);
end;
Procedure TForm1.DesktopSetup;
var i, j : integer;
begin
  for i:=0 to 200 do for j:=0 to 200 do Field[i,j]:=0;
  Frame21.Image1.Canvas.Brush.Style := bsSolid;
  Frame21.Image1.Canvas.Pen.Mode := pmCopy;
  Frame21.Image1.Canvas.Pen.Color := Desktop_Color;
  Frame21.Image1.Canvas.Brush.Color := Desktop_Color;
  Frame21.Panel1.Width:=2000*Koeff_Mashtab div 10;
  Frame21.Panel1.Height:=2000*Koeff_Mashtab div 10;
  Frame21.Image1.Canvas.Rectangle(0,0,2000*Koeff_Mashtab div 10,2000*Koeff_Mashtab
div 10);
  for i:=0 to 200 do
    for j:=0 to 200 do
Frame21.Image1.Canvas.Pixels[i*Koeff_Mashtab,j*Koeff_Mashtab]:=Grid_Color;
// Button4Click(nil);
end;
//видалення виділених елементів
procedure DeleteSelectedElements;
var i, j, k, l : integer;
begin
  i := 1;
  while i <= Last_Elem do
    begin
      if Elements[i].Selected then
        begin
          Elements[i].destroy;
          for j:=1 to 5 do
            if Oscill[i]=i then Oscill[i]:=0;
          if Form17.Visible then Form17.CheckListBox1ClickCheck(nil);

```

```

for j:=0 to High(Conns) do
begin
k:=0;
while k<=High(Conns[j]) do
begin
if Conns[j][k].Elem=i then
begin
for l:=k to High(Conns[j])-1 do
Conns[j][l]:=Conns[j][l+1];
k:=k-1;
SetLength(Conns[j],High(Conns[j]));
end;
k:=k+1;
end;
end;
end;
j:=0;
while j<=High(Conns) do
begin
if High(Conns[j])<1 then
begin
for l:=j to High(Conns)-1 do
Conns[l]:=Conns[l+1];
SetLength(Conns,High(Conns));
end;
j:=j+1;
end;
for j:=0 to High(Conns) do
for k:=0 to High(Conns[j]) do
if Conns[j][k].Elem>=i then Conns[j][k].Elem:=Conns[j][k].Elem-1;
for j:= i to Last_Elem-1 do
begin
Elements[j] := Elements[j+1];
Elements[j].Image.Tag := Elements[j].Image.Tag -1;
end;
dec(Last_Elem);
dec(i);
end;
inc(i);
end;
end;
//видалення елементів
procedure DeselectElements;

```

						<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			<b>61</b>

```

var i :integer;
begin
  for i:=1 to Last_Elem do
    Elements[i].Deselect;
  end;
procedure SelectAllElements;
var i : integer;
begin
  for i:=1 to Last_Elem do
    Elements[i].Select;
  end;
procedure InversElementSelection;
var i: integer;
begin
  for i:=1 to Last_Elem do
    Elements[i].Swich_Selection;
  end;

//промальовування виділення елементів мишкою
Procedure TForm1.DrawSelectionRectangle (SP : TPoint);
begin
  Frame21.Image1.Canvas.Brush.Style := bsClear;
  Frame21.Image1.Canvas.Pen.Color := clWhite;
  Frame21.Image1.Canvas.Pen.Style := psDash;
  Frame21.Image1.Canvas.Pen.Mode := pmXor;
  Frame21.Image1.Canvas.Pen.Width := 1;
  Frame21.Image1.Canvas.Rectangle(Base_Selection_Point.X,Base_Selection_Point.Y,SP.X
, SP.Y);
end;
procedure Draw_Leg(Sender:TObject; i:integer);
begin
  TImage(Sender).Canvas.Pen.Mode:=pmXor;
  TImage(Sender).Canvas.Pen.Width:=10;
  TImage(Sender).Canvas.Pen.Color:=clRed xor
Form1.Frame21.Image1.Canvas.Pixels[1,1];
  if Elements[TImage(Sender).Tag].Legs[i].x=0 then begin
TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Elements[
TImage(Sender).Tag].Legs[i].y*10);
TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Elements[
TImage(Sender).Tag].Legs[i].y*10);
  end else begin

```

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

```

TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);
TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);
    end;
end;

//з'єднання елементів
function IsConnected(E,L: Integer): Boolean;
var i,j: Integer;
begin
    Result:=true;
    for i:=0 to High(Conns) do
        for j:=0 to High(Conns[i]) do
            if (Conns[i][j].Elem=E) and (Conns[i][j].Leg=L) then exit;
        Result:=false;
    end;
procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
var selcon: Boolean;
    i: Integer;
begin
    if (Button <> mbLeft) and (Button <> mbRight) then exit;
    if Connect_Status then exit;
    selcon:= false;
    i:=1;
    if TImage(Sender).tag=0 then exit;
    if (x<Koeff_Mashtab) or (x>(Elements[TImage(Sender).tag].Size.x-
1)*Koeff_Mashtab) then
        for i:=0 to High(Elements[TImage(Sender).tag].Legs) do
            if (x>(Elements[TImage(Sender).tag].Legs[i].x-1)*Koeff_Mashtab)
                and (x<(Elements[TImage(Sender).tag].Legs[i].x+1)*Koeff_Mashtab)
                and (y>(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
                    and
(y<(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab+Koeff_Mashtab div 2)
then begin selcon:=true; break; end;
    if Button = mbRight then
        begin
            if selcon and IsConnected(TImage(Sender).tag,i) then
                begin
                    TImage(Sender).PopupMenu:=PopupMenu3;

```

```

        Connect_Start.Elem:=TImage(Sender).tag;
        Connect_Start.Leg:=i;
    end else TImage(Sender).PopupMenu:=nil;
    exit;
end;
if selcon then begin
    Connect_Status:=true;
    Connect_Start.Elem:=TImage(Sender).tag;
    Connect_Start.Leg:=i;
    Legs_sel2:=false;
end;

M_x:=X;
M_y:=y;
if not (ssCtrl in Shift) then Mouse_Status:=True;
end;
//додавання з'єднання
procedure AddConnect(E1,L1,E2,L2: integer);
label 1,2;
var i,j,k:integer;
begin
    for i:=0 to High(Conns) do
        for j:= 0 to High(Conns[i]) do
            if (Conns[i][j].Elem=E1) and (Conns[i][j].Leg=L1) then goto 1;
1:
        for k:=0 to High(Conns) do
            for j:= 0 to High(Conns[k]) do
                if (Conns[k][j].Elem=E2) and (Conns[k][j].Leg=L2) then goto 2;
2:
            if ((i=High(Conns)+1) and (k=High(Conns)+1)) or (High(Conns)=-1) then
                begin
                    SetLength(Conns,High(Conns)+2);
                    SetLength(Conns[High(Conns)],2);
                    j:=0;
                    Conns[High(Conns)][j].Elem:=E1;
                    Conns[High(Conns)][j].Leg:=L1;
                    j:=1;
                    Conns[High(Conns)][j].Elem:=E2;
                    Conns[High(Conns)][j].Leg:=L2;
                    InitTrace;
                    Exit;
                end;
            if k=i then exit;

```

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

```

if i=High(Conns)+1 then
  begin
    SetLength(Conns[k],High(Conns[k])+2);
    j:=High(Conns[k]);
    Conns[k][j].Elem:=E1;
    Conns[k][j].Leg:=L1;
    InitTrace;
    Exit;
  end;
if k=High(Conns)+1 then
  begin
    SetLength(Conns[i],High(Conns[i])+2);
    j:=High(Conns[i]);
    Conns[i][j].Elem:=E2;
    Conns[i][j].Leg:=L2;
    InitTrace;
    Exit;
  end;
SetLength(Conns[i],High(Conns[i])+2+High(Conns[k]));
for j:=High(Conns[i])-High(Conns[k]) to High(Conns[i]) do
  begin
    Conns[i][j].Elem:=Conns[k][j-High(Conns[i])+High(Conns[k])].Elem;
    Conns[i][j].Leg:=Conns[k][j-High(Conns[i])+High(Conns[k])].Leg;
  end;
SetLength(Conns[k],0);
for j:=k to High(Conns)-1 do
  Conns[j]:=Conns[j+1];
SetLength(Conns[j],0);
SetLength(Conns,High(Conns));
InitTrace;
end;

```

## 4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою CRYPTON – алгоритм симетричного блочного шифрування (розмір блоку 128 біт, ключ довжиною до 256 біт), розроблений південнокорейським криптологом Чьо Лім Хун з південнокорейської компанії Future Systems, яка з кінця 1980-х років працює на ринку забезпечення мереж і захисту інформації.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>65</b>

Алгоритм був розроблений в 1998 році в якості шифру – учасника конкурсу AES. Як зізнавався автор, конструкція алгоритму спирається на алгоритм SQUARE[1]. В алгоритмі Crypton немає традиційних для блочних шифрів мережі Фейстеля. Основу даного шифру становить так звана SP-мережа (повторювана циклова функція, що складається із замін-перестановок, орієнтована на розпаралелену нелінійну обробку всього блоку даних). Крім високої швидкості, перевагами таких алгоритмів є полегшення дослідження стійкості шифру до методів диференціального та лінійного криптоаналізу, що є на сьогодні основними інструментами розтину блочних шифрів. На конкурс AES була представлена версія алгоритму Crypton v0.5. Однак, як казав Чьо Лім Хун, йому не вистачало часу для розробки повної версії. І вже на першому етапі конкурсу AES в ході аналізу алгоритмів, версія Crypton v0.5 була замінена на версію Crypton v1.0. Відмінність нової версії від первинної полягала в зміні таблиці замін та в модифікації процесу розширення ключа.

Як і інші учасники конкурсу AES, Crypton призначений для шифрування 128-бітових блоків даних[2]. При шифруванні використовуються ключі шифрування для декількох фіксованих розмірів – від 0 до 256 біт з кратністю 8 бітів. Структура алгоритму Crypton – структура «Квадрата» – багато в чому схожа на структуру алгоритму Square, створеного в 1997 році. Криптографічні перетворення для алгоритмів з даною структурою можуть бути виконані як для цілих рядків і стовпців масиву, так і над окремими його байтами. (Варто зазначити, що алгоритм Square був розроблений авторами майбутнього переможця конкурсу AES – авторами алгоритму Rijndael – Вінсентом Ріджменом і Джоан Дейменом.)

### **Шифрування**

Алгоритм Crypton являє 128-бітовий блок шифруємих даних у вигляді байтового масиву  $4 \times 4$ , над якими в процесі шифрування проводиться кілька раундів перетворень. У кожному раунді передбачається послідовне виконання наступних операцій:

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

- Таблична заміна  $\gamma$ ;
- Лінійне перетворення  $\pi$ ;
- Байтова перестановка  $\tau$ ;
- Операція  $\sigma$ .

### **Таблична заміна $\gamma$**

Алгоритм Scurpton використовує 4 таблиці замін. Кожна з яких заміщає 8-бітне вхідне значення на вихідне такого ж розміру.

### **Лінійне перетворення $\pi$**

Тут використовується 4 спеціальні константи. Ці константи об'єднані в маскуючі послідовності

### **Байтова перестановка $\tau$**

Дана перестановка перетворює найпростішим чином рядок даних у стовпець.

### **Операція $\sigma$**

Дана операція є побітовим складанням всього масиву даних з ключем раунду. Зауважимо, саме 12 раундів шифрування рекомендується автором алгоритму Чьо Хун Лімом, проте сувора кількість раундів не встановлена.

КБПЗ 2023

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>67</b>

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської роботи.

Розроблене програмне забезпечення системи ідентифікації професійних знань операторів автоматизованих систем управління, складається з наступних функціональних блоків:

– Навігаційне меню: Файл; Шаблон тестів; Оператор; налаштування; Довідка.

– Функції представлені у графічному вигляді (іконки).

– Вікна інформації яке розподіляється на три вкладки: Тести; Формат; Виключення.

– Вікна виведення календарю планування.

– Функціональних кнопок ПЗ.

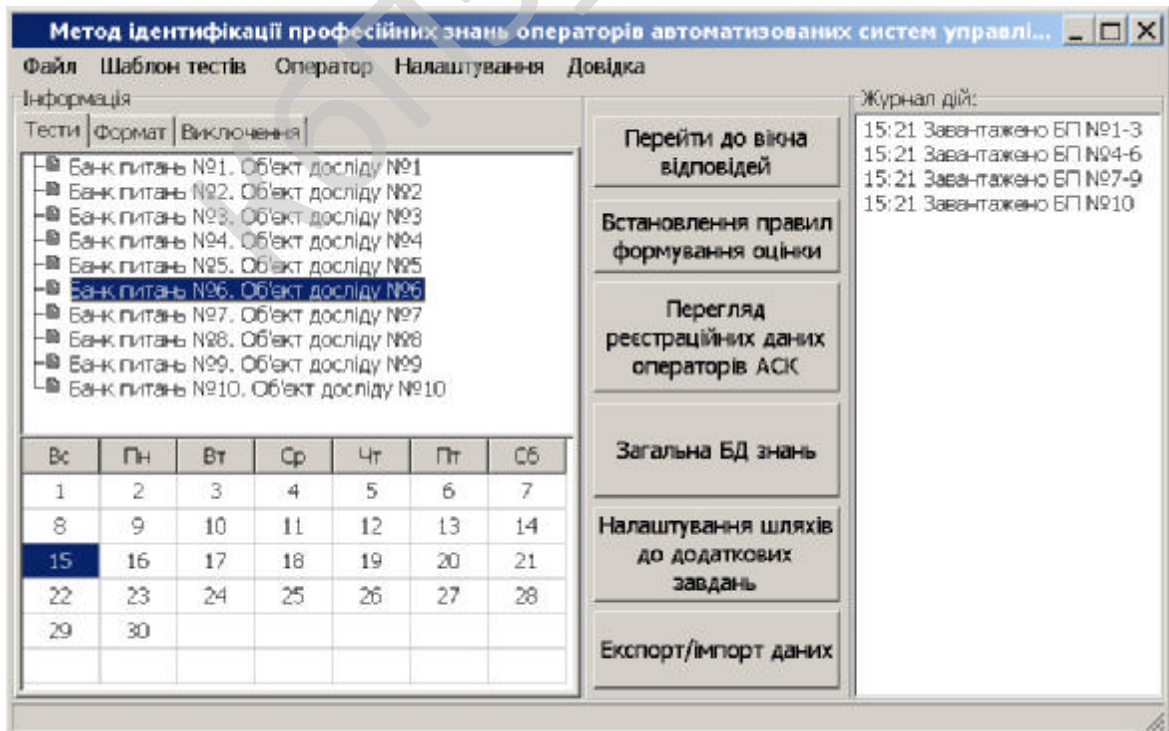


Рисунок 5.1 – Головне вікно розробленого ПЗ

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

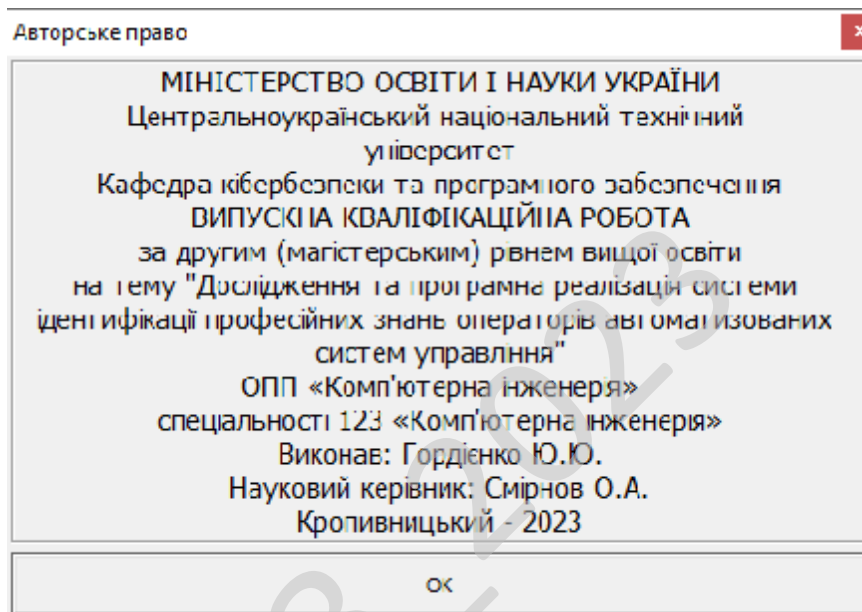


Рисунок 5.2 – Авторське право

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69



– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєструватися), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи ідентифікації професійних знань операторів автоматизованих систем управління.

*Метою розробки є дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.*

*Об'єктом дослідження є процес ідентифікації професійних знань операторів автоматизованих систем управління.*

*Предметом дослідження є методи ідентифікації професійних знань операторів автоматизованих систем управління.*

*Методи дослідження базуються на методах теорії комп'ютерного тестування, методах математичної статистики, методах розробки програмного забезпечення.*

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

– Удосконалено метод ідентифікації професійних знань операторів автоматизованих систем управління.

– Розроблено вітчизняний продукт ідентифікації професійних знань операторів автоматизованих систем управління, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

## 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 48 днів (два місяці). В магістерській роботі була досліджена та розроблена програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

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	20
3. Запланований термін розробки, днів	Fpq	48 (2 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

Продовження таблиці 7.1

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

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	20000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

## 7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де:  $A$  – коефіцієнт Боема,  $A = 2,45$ ;

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

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

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.

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

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

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

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

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

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

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

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

### 7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки.}$$

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

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

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

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	10	900	15
Монітор	60	10	600	10
Клавіатура	30	10	300	5
Маніпулятор «мишка»	30	10	300	5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	3	90	1,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	400	1000	16,67
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 <sub>ч</sub>	60,83

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:

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

$$\Phi_{\text{др}}^c = \frac{60,83 \cdot 2}{1,2} = 101,4 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 101,4 / (48 \cdot 8) = 0,26 \text{ ставка.}$$

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

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2022, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	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.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	16000	48000
Продакт-менеджер	0,25	14000	10500
Інженер-програміст	3,8	16000	182400
Інженер-електронщик	0,26	14500	11310
Інженер-системотехнік	0,25	14500	10875
Адміністратор мережі	0,5	14500	21750
Системний програміст	0,25	14500	10875
Дизайнер WEB	0,25	14000	10500
Інженер-верстальник	0,25	13700	10275
Бухгалтер-економіст	0,5	14500	21750
Всього за період розробки	$R_{cn} = 7,31$	-	$\Phi_{роб} = 338235$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{338235}{7,31 \cdot 48} = 964 \text{ грн.}$$

#### 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-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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

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

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 *у.о./м<sup>2</sup>*. Враховуючи, що курс складає 1 *у.о.* = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно 8  $m^2$ . З урахуванням цього:

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

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

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

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

де:  $C_m$  – ціна меблів для одного робочого місця, грн.

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

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 28.10.23 – джерело <https://compbest.com.ua>.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82



Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D ( 5ms, 300/3000: 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	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	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат (МФУ)	1	5965	596,5	6561,5
Всього	—	—	—	199177

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	199177	-	-
Всього по групі	199177	50	99588,5
Нематеріальні активи			
4. Нематеріальні активи	20000	10	2000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	5000
Разом	$K_p = 1807208$		$A_p = 176988,5$

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



Згідно виданих норм приймаємо 0,25 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає  $C_n = 210$  грн., визначаємо вартість паперу за період розробки  $N_m = 2$  міс:

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

$$Z_{M1} = 210 \cdot 2 \cdot 0,25 = 105 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 4 примірників.

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

де:  $C_d$  – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 28 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 28 грн./шт.

$$Z_{M2} = 28 \cdot 4 = 112 \text{ грн.}$$

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

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

де:  $C_z$  – 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.

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

$$Z_M = (105 + 112 + 1702) / 20 = 96 \text{ грн.}$$

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, %.

$$O_n = 10074 \cdot 15 \cdot 0,01 = 1511 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 20$  прим.):

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

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

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

$$A_m = 176989 \cdot 2 / (20 \cdot 12) = 1475 \text{ грн.}$$

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

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

$$C_n = 10074 + 1007,4 + 2438 + 1511 + 96 + 1511 + 1475 = 18112,4 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	$Z_o$	10074
2. Додаткова зарплата виконавців	$Z_d$	1007,4
3. Відрахування на соціальні потреби	$C_{oc}$	2438
4. Загальногосподарські витрати	$\Gamma_{ocn}$	1511
5. Витрати на матеріали	$Z_m$	96
6. Освоєння нових операційних систем, мов програмування	$O_n$	1511
7. Амортизація основних фондів	$A_m$	1475
8. Повна собівартість програмного забезпечення	$C_n$	18112,4
9. Плановий прибуток	$P_p$	9056,2
10. Ціна підприємства $C_n = C_n + P_p$	$C_n$	27168,6
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{дв} \cdot C_n$	$ПДВ$	5433,7
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	$C$	32602,3

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 50%.

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

де:  $P_n$  – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 18112,4 = 9056,2 \text{ грн.}$$

## 7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

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

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	32602
Всього капітальних витрат	–	32602

## 7.7 Визначення експлуатаційних витрат

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



Витрати 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	25	–	32602	–	8150,5
Всього відрахувань	-	–	32602	–	8150,5

### 7.8 Визначення економічної ефективності програмної продукції

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

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

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

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

$$E_e = (27168,6 - 18112,4) \cdot 20 - (0,05 \cdot 1408000 + 0,5 \cdot 199177 + 0,2 \cdot 143000 + 0,25 \cdot 37031 + 0,1 \cdot 20000) \cdot 2/12 = 146149 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

де:  $K_p^*$  – балансова вартість основних фондів розробника.

$$T_e = \frac{1807208}{(27168,6 - 18112,4) \cdot 20 \cdot 12 / 2} = 1,7 \text{ роки.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	20
2. Повна собівартість розробленої програми	Грн.	18112,4
3. Ціна розробленої програми	Грн.	27168,6
4. Плановий прибуток від реалізації розробленої програми	Грн.	9056,2
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1807208
7. Загальний прибуток від реалізації програмної продукції	Грн.	181120
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	146149
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,7
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	32602
11. Величина економічного ефекту у користувача програмної продукції	Грн.	13233
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Роки	1,5

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

$$E_{cn} = (44286 - 22912,5) - 0,25 \cdot 32602 = 13223 \text{ грн.}$$

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

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{32602}{44286 - 22912,5} = 1,5 \text{ роки.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

## 7.9 Висновки

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

					VKPM-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Забезпечення охорони праці та техніки безпеки є надзвичайно важливою складовою будь-якого підприємства, якщо мова йде про впровадження нових технологічних рішень та програмного забезпечення. У зв'язку зі швидкими змінами в галузі інформаційних технологій та підвищеною складністю систем, пов'язаних з управлінням інвентарем та сервісним обслуговуванням, забезпечення безпеки праці та техніки стає ще актуальнішим завданням.

Техніка безпеки – це система правил і заходів, які допомагають запобігти травмам, хворобам і аваріям на робочому місці або в повсякденному житті. Знання техніки безпеки дуже важливе, бо воно рятує життя і здоров'я людей. Наприклад, якщо людина працює на шахті, то вона повинна знати правила безпеки у вугільних шахтах, щоб не потрапити під обвал або не підірватися на міні. Або якщо людина хоче навчитися програмувати, то вона повинна дотримуватися гігієнічних вимог і не сидіти за комп'ютером занадто довго, щоб не зашкодити своїм очам і спині.

Охорона праці та здоров'я у сфері ІТ – це комплекс заходів, які спрямовані на забезпечення безпечних і здорових умов праці для працівників, які використовують інформаційні технології, а також на запобігання травматизму, професійним захворюванням і стресу.

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно-обчислювальних машин. 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

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

різних видів робіт, пов'язаних обробкою текстових та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом.

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207 [4], який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням [2]. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста включають наступні фактори:

– параметри повітряного середовища в приміщенні;

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

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

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

## 8.2 Пожежна безпека

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.

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

При обслуговуванні, ремонтних та профілактичних роботах використовуються різні легкозаймісті 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

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

приміщеннях. Звукобирне облицювання стін, стель приміщень треба виконувати з негорючих та важко горючих матеріалів.

Електроустановки (можливість їх застосування, монтаж, накладка експлуатація) повинні відповідати вимогам чинних правил улаштування електроустановок, правил технічної експлуатації, електроустановок та інших нормативних документів.

Ймовірність виникнення пожежі від електротехнічного та іншого одиничного виробу не повинна перевищувати  $10^{-6}$  на рік. При короткому замиканні в місцях з'єднання проводів опір практично дорівнює нулю, звідси величина струму досягає дуже великих значень.

Персональні комп'ютери після закінчення роботи повинні відключатися від мережі не рідше 1 разу на квартал, необхідно очищати від пилу агрегати та вузли, кабельні канали та простір між підлогами. Не дозволяється розміщувати комп'ютерні зали ЕОМ у підвалах; проводити ремонт вузлів (блоків) ЕОМ безпосередньо у залах, де знаходяться ПК (персональні комп'ютери), залишати без нагляду ввімкнену в мережу електронну апаратуру, яка використовується для контролю ЕОМ.

Електричний струм силою 0,1 А є небезпечним для людини. Для попередження травм усе електричне обладнання повинне бути заземлене. Приступаючи до роботи необхідно перевірити справність обладнання, ізоляцію проводів і надійність заземлення. Доторкання до оголених струмоведучих і незахищених частин в електроустаткуванні забороняється. В разі виявлення порушень ізоляції електропроводів, відкритих струмоведучих частин електроустаткування або порушення заземлення треба негайно повідомити про це свого начальника для вжиття заходів щодо усунення несправності. Проводити самому ремонт електроустаткування забороняється.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97



1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови Головного державного санітарного лікаря України [5], робота, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22,5-23	45-60	0,11
Тепла	23-25	50-70	0,1	23-25	52-70	0,1

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

В літню пору застосовується кондиціонер.

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

У приміщенні знаходяться наступні джерела шуму: принтер HP Laser 107a, електродвигуни вентиляторів ЕОМ.

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

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

## 8.4 Розробка заходів з умов поліпшення охорони праці

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

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

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

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

## 8.5 Розрахункова частина

Завдання: розрахувати *штучне освітлення робочого приміщення*.

*Початкові дані:* ширина робочого приміщення: 4,4 м.; довжина – 6 м.; висота – 3 м.

Розрахунок штучного освітлення проведемо за методом коефіцієнта використання світлового потоку.

					VKPM-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

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

$$F=ESKZ/n,$$

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

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

$S$  – площа освітлюваного приміщення (у нашому випадку  $S=4,4 \times 6 = 26,4$  м<sup>2</sup>);

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

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

$n$  – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ( $\rho_{стін.}$ ) і стелі ( $\rho_{стелі}$ ), значення коефіцієнтів дорівнюють  $\rho_{стін} = 50\%$  і  $\rho_{стелі} = 50\%$  [6].

Обчислимо індекс приміщення за формулою:

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

де:  $S$  – площа приміщення,  $S = 11,9$  м<sup>2</sup>;

$h$  – розрахункова висота підвісу,  $h = 3$  м;

$A$  – ширина приміщення,  $A = 4,4$  м;

$B$  – довжина приміщення,  $B = 6$  м.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$i=0,57.$$

Знаючи індекс приміщення, за знаходимо  $n = 0.29$  (з табличних даних коефіцієнтів використання світлового потоку ( $n$ ) світильників відповідного типу). Підставимо всі значення у формулу, визначемо світловий потік:  $F=28409$  Лм.

Для штучного освітлення приміщення використовуються LED панель MAXUS ASSISTANCE PRO 80W 5000K WHITE (M1052480531), світловий потік яких  $F_n = 8000$  Лм.

Число світильників визначається по формулі:

$$N=F/F_n$$

де:  $F$  – світловий потік,

$F_n$  – світловий потік одного світильника.

$$N= 28409/ 8000=3,55 \text{ шт.}$$

Приймаємо необхідну кількість світильників 4 шт.

### Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок захисного штучного освітлення. Розроблено заходи з охорони праці.

### Список використаних джерел інформації

1. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2.007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
3. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

					ВКРМ-123.23.0088.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

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

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

6. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення: 16.06.2023).

7. Методичні рекомендації до виконання розділу «Заходи з охорони праці та техніки безпеки» у магістерській дисертації / Л.Д. Третьякова; М-во освіти і науки України, Національний технічний університет України «Київський політехнічний інститут» – Київ, КПІ, 2014. – 26 с. – Режим доступу до ресурсу: <http://surl.li/dhulo> (дата звернення: 16.06.2023).

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>104</b>

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи ідентифікації професійних знань операторів автоматизованих систем управління.

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 ідентифікації професійних знань операторів автоматизованих систем управління.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем ідентифікації професійних знань операторів автоматизованих систем управління.
- Досліджена система ідентифікації професійних знань операторів автоматизованих систем управління.
- На основі отриманих результатів досліджень створена програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання ідентифікації професійних знань операторів автоматизованих систем управління.

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.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

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.

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. 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 CRYPTON.

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 13233 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,5 роки.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>106</b>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гордієнко Ю.Ю. Дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
3. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
4. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
5. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
6. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
7. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
8. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
9. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

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

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

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

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

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

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

16. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

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

					<b>БКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>108</b>

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

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

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

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

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

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

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

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

26. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

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

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

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

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

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

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

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

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

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

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

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

37. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Электронный Журнал]. Georgia. Tbilisi: SCSA – 2018.

38. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian

Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

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

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

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

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

43. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів .Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

44. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). - Полтава: ПолтНТУ. - 2016. - С. 98-103.

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

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

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

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

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

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

51. Смірнов О.А., Коваленко А.С., Коваленко О.В., Доренський О.П. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи. Системи озброєння і військова техніка. – Випуск 2(46) – Х.: ХУПС – 2016. – С. 103-107.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

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

53. Смірнов О.А., Євсєєв С.П., Король О.Г., Коваленко О.В., Коваленко А.С., Смірнов С.А. Архітектура мікропроцесорів та компонентів ЕОМ. Навчальний посібник – Кіровоград: Вид. Лисенко В.Ф., 2015. – 550 с.

54. Смірнов О.А., Коваленко О.В., Мелешко Є.В., Константинова Л.В., Кожанова А.С. Інженерія програмного забезпечення. Навчальний посібник. За ред. О.А. Смірнова. – Кіровоград: КНТУ 2013. – 409с.

55. Смірнов О.А., Осадчій С.І., Мелешко Є.В., Іванов С.Г., Павленко М.А., Усачов О.М. Основи технічної експлуатації АСУ. Навчальний посібник. – Кіровоград: КНТУ 2013. – 322с.

56. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лєвошко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.

57. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

					<b>ВКРМ-123.23.0088.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

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-123.23.0088.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Гордієнко Ю.Ю.				Дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління	Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22МЗ			
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи ідентифікації професійних знань операторів автоматизованих систем управління.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 36-13 від 04.08.2023 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи ідентифікації професійних знань операторів автоматизованих систем управління.

## 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-123.23.0088.00.00.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 ідентифікації професійних знань операторів автоматизованих систем управління;
- 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-123.23.0088.00.00.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.

					ВКРМ-123.23.0088.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

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

## 7 Економічні вимоги

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

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

## 8 Вимоги щодо охорони праці

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

					ВКРМ-123.23.0088.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

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

## 10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 21.12.2023 р.

					<b>ВКРМ-123.23.0088.00.00.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

\_\_\_\_\_ Смірнов О.А.

*Дослідження та програмна реалізація  
системи ідентифікації професійних знань операторів автоматизованих  
систем управління*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 42

Літера: РП

Кропивницький – 2023 року

## Файл Main.pas - основна програма

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, ExtCtrls, Menus, ToolWin, ComCtrls, IniFiles,
  SomeClasses,
  ImgList, Buttons, About;

type
  TForm1 = class(TForm)
    Frame21: TFrame2;
    Panel1: TPanel;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    Button1: TButton;
    Image1: TImage;
    Button2: TButton;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    Panel2: TPanel;
    Panel3: TPanel;
    ToolBar1: TToolBar;
    ToolButton5: TToolButton;
    Panel4: TPanel;
    StatusBar1: TStatusBar;
    Panel5: TPanel;
    N10: TMenuItem;
    Panel6: TPanel;
    ComboBox1: TComboBox;
    N9: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    N13: TMenuItem;
    N14: TMenuItem;
    Button4: TButton;
    Label1: TLabel;
    N15: TMenuItem;
    N16: TMenuItem;
    TreeView1: TTreeView;
    ImageList1: TImageList;
    Button3: TButton;
    Button5: TButton;
    N17: TMenuItem;
    N18: TMenuItem;
    PopupMenu1: TPopupMenu;
    N19: TMenuItem;
    N20: TMenuItem;
    N21: TMenuItem;
    N22: TMenuItem;
    ImageList2: TImageList;
    N23: TMenuItem;
    N24: TMenuItem;
    PopupMenu2: TPopupMenu;
    N25: TMenuItem;
    PopupMenu3: TPopupMenu;
    N26: TMenuItem;
    ToolButton2: TToolButton;
    ImageList3: TImageList;
```

```

ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ToolButton9: TToolButton;
ImageList4: TImageList;
N27: TMenuItem;
N28: TMenuItem;
Button6: TButton;
Button7: TButton;
Button8: TButton;
OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
N29: TMenuItem;
N30: TMenuItem;
N31: TMenuItem;
N32: TMenuItem;
N33: TMenuItem;
N34: TMenuItem;
N35: TMenuItem;
ToolButton1: TToolButton;
N36: TMenuItem;
N37: TMenuItem;
procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Image1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormCreate(Sender: TObject);
procedure FormMouseWheel(Sender: TObject; Shift: TShiftState;
  WheelDelta: Integer; MousePos: TPoint; var Handled: Boolean);
procedure FormActivate(Sender: TObject);
procedure Frame21Image1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure N3Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure Frame21Image1MouseDown(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Frame21Image1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
Procedure DrawSelectionRectangle (SP : TPoint);
procedure N8Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure Panel3Click(Sender: TObject);
procedure Panel4Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure DesktopSetup;
procedure InitCfg;
procedure SaveCfg;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ComboBox1Change(Sender: TObject);
procedure RepaintElements;
procedure N11Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure N14Click(Sender: TObject);
procedure Frame21Image1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure TreeView1Db1Click(Sender: TObject);
procedure TreeView1Click(Sender: TObject);
procedure AssociateTree;
procedure N19Click(Sender: TObject);
procedure N21Click(Sender: TObject);

```

```

procedure N24Click(Sender: TObject);
procedure Image1Db1Click(Sender: TObject);
procedure N25Click(Sender: TObject);
procedure DrawField;
procedure N26Click(Sender: TObject);
procedure ToolButton6Click(Sender: TObject);
procedure ToolButton7Click(Sender: TObject);
procedure ToolButton8Click(Sender: TObject);
procedure ToolButton9Click(Sender: TObject);
procedure N27Click(Sender: TObject);
procedure N28Click(Sender: TObject);
procedure N15Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure N34Click(Sender: TObject);
procedure N37Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
TCons = array of TPoint;
TConn = record
  Elem: integer;
  Leg: integer;
end;
TConns = array of TConn;

procedure Delay(n: Cardinal);

const
  Con_Color = clBlue;

var
  F_Stop: Boolean;
  F_Pause: Boolean;
  Form1: TForm1;
  Cons: array of TCons;
  Conns: array of TConns;
  Desktop_Color, Grid_Color : TColor;
  Selection_Metod : Boolean;
  IniFile : TIniFile;
  Koeff_Mashtab: integer;
  Old_Mashtab : integer;
  Elements: array [1..1000] of TElement;
  Chips_count : integer;
  Select_on_creation : boolean;
  Draw_connections : boolean;
  Select_connections : boolean;
  Legs_sel : Boolean;
  Legs_sel_Elem : TObject;
  Legs_sel_Leg : integer;
  Legs_sel2 : Boolean;
  Legs_sel_Elem2 : TObject;
  Legs_sel_Leg2 : integer;
  Selected_Connection : integer;
  Conn_Prov : integer;
  Conn_Prov_x : integer;
  Conn_Prov_y : integer;
  Last_Elem : integer;
  Test_Count : integer;
  Oscill : array [1..5] of integer;
  fio : file of TElemsave;
  fio2 : file of TConn;

```

implementation

```

    uses Parameters, Db_elements, Connector, Db_micro, Types, Tracer,
    Time_scale,
        Main6, Test;

    {$R *.dfm}

    var
        Mouse_Status, Selection_Status: Boolean;
        M_x, M_y: Integer;
    //   Cur_Elem: integer;
        Base_Selection_Point : TPoint;
        Second_Selection_Point : TPoint;
        Old_Second_Selection_Point : TPoint;
        Connect_Status : Boolean;
        Connect_Start : record
            Elem, Leg: Integer;
        end;

    procedure Delay(n:cardinal);
    var
        FirstTick: Cardinal;
    begin
        FirstTick := GetTickCount;
        repeat Application.ProcessMessages;
        until GetTickCount - FirstTick >= n;
    end;

    procedure TForm1.AssociateTree;
    var i, j : integer;
    begin
        for i := 0 to treeview1.Items.Count - 1 do
            if TreeView1.Items[i].ImageIndex = 1 then
                for j:= 1 to last_elem + 1 do
                    begin
                        if j = Last_Elem + 1 then begin TreeView1.Items[i].ImageIndex :=
2; TreeView1.Items[i].SelectedIndex := 2; break end;
                        if Elements[j].Caption = TreeView1.Items[i].Text then
                            begin TreeView1.Items[i].Data := @elements[j]; break end;
                    end;
                end;
        end;

    Procedure TForm1.RepaintElements;
    var i: integer;
    begin
        for i := 1 to Last_Elem do
            begin
                Elements[i].Image.Width:=Elements[i].Image.Picture.Bitmap.Width*Koeff_Mashtab
                div 10;

                Elements[i].Image.Height:=Elements[i].Image.Picture.Bitmap.Height*Koeff_Mashtab
                div 10;
                Elements[i].Image.Left:=Elements[i].image.left div Old_Mashtab *
                Koeff_Mashtab + 2;
                Elements[i].Image.Top:=Elements[i].image.Top div Old_Mashtab *
                Koeff_Mashtab + 2;
            end;
        end;

        //ініціалізація параметрів програми
        Procedure TForm1.InitCfg;
        begin
            IniFile := Tinifile.Create
            (ExtractFilePath(Application.ExeName)+'\config\main.ini'); //відкриття файлу з
            параметрами програми
            Desktop_Color := IniFile.ReadInteger('Colors','Desktop',clWhite);
            //колір фону
            Grid_Color := IniFile.ReadInteger('Colors','Grid',clBlack);
            //колір розмітки

```

```

Selection_Metod := IniFile.ReadBool('Selection','Full',False);
//параметри виділення професійних знань операторів автоматизованих
систем управління
Select_on_creation := IniFile.ReadBool('Creation','Selection',False);
Draw_connections := IniFile.ReadBool('Selection','Draw_conn',False);
Select_connections := IniFile.ReadBool('Selection','Sel_conn',True);
if IniFile.ReadBool('Main','LockBases',True) then
begin
  N11.Enabled:=false;
  N12.Enabled:=false;
  N13.Enabled:=false;
  N14.Enabled:=false;
end;
end;

//збереження параметрів програми
Procedure TForm1.SaveCfg;
begin
  IniFile.WriteInteger('Colors','Desktop',Desktop_Color);
  IniFile.WriteInteger('Colors','Grid',Grid_Color);
  IniFile.WriteBool('Selection','Full',Selection_Metod);
  IniFile.WriteBool('Creation','Selection',Select_on_creation);
  IniFile.WriteBool('Selection','Draw_conn',Draw_connections);
  IniFile.WriteBool('Selection','Sel_conn',Select_connections);
end;

Procedure TForm1.DesktopSetup;
var i, j : integer;
begin
  for i:=0 to 200 do for j:=0 to 200 do Field[i,j]:=0;
  Frame21.Imagel.Canvas.Brush.Style := bsSolid;
  Frame21.Imagel.Canvas.Pen.Mode := pmCopy;
  Frame21.Imagel.Canvas.Pen.Color := Desktop_Color;
  Frame21.Imagel.Canvas.Brush.Color := Desktop_Color;
  Frame21.Panell1.Width:=2000*Koeff_Mashtab div 10;
  Frame21.Panell1.Height:=2000*Koeff_Mashtab div 10;
  Frame21.Imagel.Canvas.Rectangle(0,0,2000*Koeff_Mashtab div
10,2000*Koeff_Mashtab div 10);
  for i:=0 to 200 do
    for j:=0 to 200 do

Frame21.Imagel.Canvas.Pixels[i*Koeff_Mashtab,j*Koeff_Mashtab]:=Grid_Color;
// Button4Click(nil);
end;

//видалення виділених професійних знань операторів автоматизованих систем
управління
procedure DeleteSelectedElements;
var i, j, k, l : integer;
begin
  i := 1;
  while i <= Last_Elem do
  begin
    if Elements[i].Selected then
    begin
      Elements[i].destroy;
      for j:=1 to 5 do
        if Oscill[i]=i then Oscill[i]:=0;
      if Form17.Visible then Form17.CheckListBox1ClickCheck(nil);
      for j:=0 to High(Conns) do
      begin
        k:=0;
        while k<=High(Conns[j]) do
        begin
          if Conns[j][k].Elem=i then
          begin
            for l:=k to High(Conns[j])-1 do
              Conns[j][l]:=Conns[j][l+1];
            k:=k-1;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        SetLength (Conns [j], High (Conns [j]));
        end;
        k:=k+1;
        end;
        end;
        j:=0;
        while j<=High (Conns) do
        begin
            if High (Conns [j])<1 then
            begin
                for l:=j to High (Conns)-1 do
                    Conns [l]:=Conns [l+1];
                    SetLength (Conns, High (Conns));
                end;
                j:=j+1;
            end;
            for j:=0 to High (Conns) do
                for k:=0 to High (Conns [j]) do
                    if Conns [j] [k].Elem>=i then
Conns [j] [k].Elem:=Conns [j] [k].Elem-1;
                    for j:= i to Last_Elem-1 do
                        begin
                            Elements [j] := Elements [j+1];
                            Elements [j].Image.Tag := Elements [j].Image.Tag -1;
                        end;
                        dec (Last_Elem);
                        dec (i);
                    end;
                    inc (i);
                end;
            end;

//видалення професійних знань операторів автоматизованих систем управління
procedure DeselectElements;
var i :integer;
begin
    for i:=1 to Last_Elem do
        Elements [i].Deselect;
    end;

procedure SelectAllElements;
var i : integer;
begin
    for i:=1 to Last_Elem do
        Elements [i].Select;
    end;

procedure InversElementSelection;
var i: integer;
begin
    for i:=1 to Last_Elem do
        Elements [i].Swich_Selection;
    end;

//промальовування виділення професійних знань операторів автоматизованих
систем управління мишкою
Procedure TForm1.DrawSelectionRectangle (SP : TPoint);
begin
    Frame21.Image1.Canvas.Brush.Style := bsClear;
    Frame21.Image1.Canvas.Pen.Color := clWhite;
    Frame21.Image1.Canvas.Pen.Style := psDash;
    Frame21.Image1.Canvas.Pen.Mode := pmXor;
    Frame21.Image1.Canvas.Pen.Width := 1;

Frame21.Image1.Canvas.Rectangle (Base_Selection_Point.X,Base_Selection_Point.Y,SP
.X,SP.Y);
    end;

procedure Draw_Leg (Sender:TObject; i:integer);

```

```

begin
  TImage(Sender).Canvas.Pen.Mode:=pmXor;
  TImage(Sender).Canvas.Pen.Width:=10;
  TImage(Sender).Canvas.Pen.Color:=clRed xor
Form1.Frame21.Image1.Canvas.Pixels[1,1];
  if Elements[TImage(Sender).Tag].Legs[i].x=0 then begin

TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Element
s[TImage(Sender).Tag].Legs[i].y*10);

TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Element
s[TImage(Sender).Tag].Legs[i].y*10);
  end else begin

TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);

TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);
  end;
end;

//з'єднання професійних знань операторів автоматизованих систем управління
function IsConnected(E,L: Integer): Boolean;
var i,j: Integer;
begin
  Result:=true;
  for i:=0 to High(Conns) do
    for j:=0 to High(Conns[i]) do
      if (Conns[i][j].Elem=E) and (Conns[i][j].Leg=L) then exit;
  Result:=false;
end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var selcon: Boolean;
    i: Integer;
begin
  if (Button <> mbLeft) and (Button <> mbRight) then exit;
  if Connect_Status then exit;
  selcon:= false;
  i:=1;
  if TImage(Sender).tag=0 then exit;
  if (x<Koeff_Mashtab) or (x>(Elements[TImage(Sender).tag].Size.x-
1)*Koeff_Mashtab) then
    for i:=0 to High(Elements[TImage(Sender).tag].Legs) do
      if (x>(Elements[TImage(Sender).tag].Legs[i].x-1)*Koeff_Mashtab)
        and (x<(Elements[TImage(Sender).tag].Legs[i].x+1)*Koeff_Mashtab)
          and (y>(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
            and
              (y<(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab+Koeff_Mashtab div 2)
then begin selcon:=true; break; end;
  if Button = mbRight then
    begin
      if selcon and IsConnected(TImage(Sender).tag,i) then
        begin
          TImage(Sender).PopupMenu:=PopupMenu3;
          Connect_Start.Elem:=TImage(Sender).tag;
          Connect_Start.Leg:=i;
        end else TImage(Sender).PopupMenu:=nil;
      exit;
    end;
  if selcon then begin
    Connect_Status:=true;
    Connect_Start.Elem:=TImage(Sender).tag;
    Connect_Start.Leg:=i;
    Legs_sel2:=false;
  end;
end;

```

```

end;

M_x:=X;
M_y:=Y;
if not (ssCtrl in Shift) then Mouse_Status:=True;
end;

//додавання з'єднання
procedure AddConnect(E1,L1,E2,L2: integer);
label 1,2;
var i,j,k:integer;
begin
  for i:=0 to High(Conns) do
    for j:= 0 to High(Conns[i]) do
      if (Conns[i][j].Elem=E1) and (Conns[i][j].Leg=L1) then goto 1;
1:
    for k:=0 to High(Conns) do
      for j:= 0 to High(Conns[k]) do
        if (Conns[k][j].Elem=E2) and (Conns[k][j].Leg=L2) then goto 2;
2:
    if ((i=High(Conns)+1) and (k=High(Conns)+1)) or (High(Conns)=-1) then
      begin
        SetLength(Conns,High(Conns)+2);
        SetLength(Conns[High(Conns)],2);
        j:=0;
        Conns[High(Conns)][j].Elem:=E1;
        Conns[High(Conns)][j].Leg:=L1;
        j:=1;
        Conns[High(Conns)][j].Elem:=E2;
        Conns[High(Conns)][j].Leg:=L2;
        InitTrace;
        Exit;
      end;
    if k=i then exit;
    if i=High(Conns)+1 then
      begin
        SetLength(Conns[k],High(Conns[k])+2);
        j:=High(Conns[k]);
        Conns[k][j].Elem:=E1;
        Conns[k][j].Leg:=L1;
        InitTrace;
        Exit;
      end;
    if k=High(Conns)+1 then
      begin
        SetLength(Conns[i],High(Conns[i])+2);
        j:=High(Conns[i]);
        Conns[i][j].Elem:=E2;
        Conns[i][j].Leg:=L2;
        InitTrace;
        Exit;
      end;
    SetLength(Conns[i],High(Conns[i])+2+High(Conns[k]));
    for j:=High(Conns[i])-High(Conns[k]) to High(Conns[i]) do
      begin
        Conns[i][j].Elem:=Conns[k][j-High(Conns[i])+High(Conns[k])].Elem;
        Conns[i][j].Leg:=Conns[k][j-High(Conns[i])+High(Conns[k])].Leg;
      end;
    SetLength(Conns[k],0);
    for j:=k to High(Conns)-1 do
      Conns[j]:=Conns[j+1];
    SetLength(Conns[j],0);
    SetLength(Conns,High(Conns));
    InitTrace;
  end;

procedure TForm1.Image1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var NX, NY, j, k, t3 : integer;
    selcon: Boolean;

```

```

begin
  if Button <> mbLeft then exit;
  if Button = mbRight then exit;
  t3:=Koeff_Mashtab div 5;
  Mouse_Status:= false;
  if TImage(Sender).tag=0 then exit;
  selcon:= false;
  j:=1;
  if Connect_Status then
    begin
      x:=TImage(Sender).Left+x;
      y:=TImage(Sender).Top+y;
      for k:=1 to Last_Elem do
        begin
          nx:=x-Elements[k].Image.Left;
          ny:=y-Elements[k].Image.Top;
          if (nx<Koeff_Mashtab) or (nx>(Elements[k].Size.x-
1)*Koeff_Mashtab) then
            for j:=0 to High(Elements[k].Legs) do
              if (nx>(Elements[k].Legs[j].x-1)*Koeff_Mashtab)
                and (nx<(Elements[k].Legs[j].x+1)*Koeff_Mashtab)
                  and (ny>(Elements[k].Legs[j].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
                    and
(ny<(Elements[k].Legs[j].y)*Koeff_Mashtab+Koeff_Mashtab div 2) then begin
selcon:=true; break; end;
              if selcon then break;
            end;
            TImage(Sender).Cursor:=crDefault;
            Connect_Status:=false;
            if (Conn_Prov<>0) then
              begin
                Conn_Prov:=0;
                Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

                Form1.Frame21.Image1.Canvas.Pen.Width:=t3;

AddConnect(Connect_Start.Elem,Connect_Start.Leg,Conns[MField[Conn_Prov_x,Conn_Pr
ov_y]][0].Elem,Conns[MField[Conn_Prov_x,Conn_Prov_y]][0].Leg);
                Button4Click(nil);
              end;
            if Legs_sel2 then
              begin
                Draw_Leg(Legs_sel_Elem2,Legs_sel_Leg2);
                Legs_sel2:=false;
              end;
            if selcon then
              begin
                if (Connect_Start.Elem=k) and (Connect_Start.Leg=j) then exit;
                AddConnect(Connect_Start.Elem,Connect_Start.Leg,k,j);
                Label1.Caption:=inttostr(High(Conns)+1);
                if Legs_sel2 then
                  begin
                    Draw_Leg(Legs_sel_Elem2,Legs_sel_Leg2);
                    Legs_sel2:=false;
                  end;
                Button4Click(nil);
              end;
            exit;
          end;
        if not (ssCtrl in Shift) then
          begin
            if not Elements[TImage(Sender).tag].Selected then
              begin

```

```

        DeselectElements;
        Elements[TImage(Sender).tag].Swich_Selection;
    end
end else Elements[TImage(Sender).tag].Swich_Selection;
Button4Click(nil);
end;

procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
var NX, NY, i, j, k, t3 : integer;
    selcon: Boolean;
begin
    t3:=Koeff_Mashtab div 5;
    if TImage(Sender).tag=0 then exit;
    selcon:= false;
    if Connect_Status then
        begin
            x:=TImage(Sender).Left+x;
            y:=TImage(Sender).Top+y;
            if Legs_sel2 then
                begin
                    Draw_Leg(Legs_sel_Elem2,Legs_sel_Leg2);
                    Legs_sel2:=false;
                end;
            for k:=1 to Last_Elem do
                begin
                    nx:=x-Elements[k].Image.Left;
                    ny:=y-Elements[k].Image.Top;
                    if (nx<Koeff_Mashtab) or (nx>(Elements[k].Size.x-
1)*Koeff_Mashtab) then
                        for j:=0 to High(Elements[k].Legs) do
                            if (nx>(Elements[k].Legs[j].x-1)*Koeff_Mashtab)
                                and (nx<(Elements[k].Legs[j].x+1)*Koeff_Mashtab)
                                and (ny>(Elements[k].Legs[j].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
                                    and
                                        (ny<(Elements[k].Legs[j].y)*Koeff_Mashtab+Koeff_Mashtab div 2) then begin
                                        selcon:=true; break; end;
                                    if selcon then break;
                                end;
                            if selcon then
                                begin
                                    Draw_Leg(Elements[k].Image,j);
                                    Elements[k].Image.Repaint;
                                    Legs_sel2:=true;
                                    Legs_sel_Elem2:=Elements[k].Image;
                                    Legs_sel_Leg2:=j;
                                    exit;
                                end;
                            if not selcon then
                                begin
                                    if MField[(x + Koeff_Mashtab div 2) div Koeff_Mashtab, (y +
Koeff_Mashtab div 2) div Koeff_Mashtab] <> -1 then
                                        begin
                                            if (Conn_Prov<>0) then
                                                begin
                                                    if (Conn_Prov_x = (x + Koeff_Mashtab div 2) div
Koeff_Mashtab) and (Conn_Prov_y = (y + Koeff_Mashtab div 2) div Koeff_Mashtab)
                                                        then
                                                            else begin
                                                                Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Conn_Prov:=Field[(x + Koeff_Mashtab div 2) div
Koeff_Mashtab, (y + Koeff_Mashtab div 2) div Koeff_Mashtab];

```

```

Conn_Prov_x := (x + Koeff_Mashtab div 2) div
Koeff_Mashtab;
Conn_Prov_y := (y + Koeff_Mashtab div 2) div
Koeff_Mashtab;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);
Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
end
end else begin
Conn_Prov_x := (x + Koeff_Mashtab div 2) div
Koeff_Mashtab;
Conn_Prov_y := (y + Koeff_Mashtab div 2) div
Koeff_Mashtab;
Conn_Prov:=Field[Conn_Prov_x,Conn_Prov_y];
Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);
Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
end;
end else begin
if (Conn_Prov<>0) then
begin
Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);
end;
Conn_Prov:=0;
end;
end;
exit;
end;
if Mouse_Status then
begin
for i:=1 to Last_Elem do
if (Elements[i].Selected) and
(Elements[TImage(Sender).tag].Selected) then
with Elements[i].Image do begin
NX:=Left+X-M_x;
NY:=Top+Y-M_y;
if (NX>0) and (NX<Frame21.Pane11.Width-Width) then Left:=NX
div Koeff_Mashtab * Koeff_Mashtab +2;
if (NY>0) and (NY<Frame21.Pane11.Height-Height) then Top:=NY
div Koeff_Mashtab * Koeff_Mashtab +2;
Elements[Tag].Coord.x:=Elements[Tag].Image.Left div
Koeff_Mashtab;
Elements[Tag].Coord.y:=Elements[Tag].Image.Top div
Koeff_Mashtab;
end;
if Draw_connections then Button4Click(nil);
exit;
end;
if Legs_sel then
begin
Legs_sel:=false;
Draw_Leg(Legs_sel_Elem,Legs_sel_Leg);
end;
end;

```

```

    if (x<Koeff_Mashtab) or (x>(Elements[TImage(Sender).tag].Size.x-
1)*Koeff_Mashtab) then
    for i:=0 to High(Elements[TImage(Sender).tag].Legs) do
        if (x>(Elements[TImage(Sender).tag].Legs[i].x-1)*Koeff_Mashtab)
            and (x<(Elements[TImage(Sender).tag].Legs[i].x+1)*Koeff_Mashtab)
            and (y>(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
                and
(y<(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab+Koeff_Mashtab div 2)
then begin selcon:=true; break; end;
    if selcon then
        begin
            Legs_sel:=true;
            Legs_sel_Elem:=Sender;
            Legs_sel_Leg:=i;
            Draw_Leg(Sender,i);
        end;
    if selcon then exit;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    Chips_count := 0;
    Last_Elem:=0;
    Koeff_Mashtab:=10;
    Old_Mashtab := 10;
    Selected_Connection:=-1;
    Connect;
    SetLength(Cons,0);
    SetLength(Conns,0);
    SaveDialog1.InitialDir:=ExtractFilePath(Application.ExeName);
    OpenFileDialog1.InitialDir:=ExtractFilePath(Application.ExeName);
end;

procedure TForm1.FormMouseWheel(Sender: TObject; Shift: TShiftState;
WheelDelta: Integer; MousePos: TPoint; var Handled: Boolean);
begin
    if Selection_Status then begin DrawSelectionRectangle
(Old_Second_Selection_Point); end;
    Selection_Status := false;
    if not (ssShift in Shift) then begin
Frame21.VertScrollBar.Position:=Frame21.VertScrollBar.Position-WheelDelta div 8;
Abort; end;
    if ssShift in Shift then begin
Frame21.HorzScrollBar.Position:=Frame21.HorzScrollBar.Position-WheelDelta div 8;
Abort; end;
    end;

procedure TForm1.FormActivate(Sender: TObject);
begin
    // InitCfg;
    // Form1.Align := alClient;
    Form1.Align := alNone;
    // DesktopSetup;
end;

procedure TForm1.Frame21Image1Click(Sender: TObject);
begin
    DeselectElements;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    DeleteSelectedElements;
end;

//виділити все
procedure TForm1.N3Click(Sender: TObject);
begin

```

```

    SelectAllElements;
end;

//зняти виділення
procedure TForm1.N4Click(Sender: TObject);
begin
    DeselectElements;
end;

//інвертувати виділення
procedure TForm1.N5Click(Sender: TObject);
begin
    InversElementSelection;
end;

procedure TForm1.Frame21Image1MouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
    // Connect_Status:=false;
    if Button = mbRight then
        begin
            if Selected_Connection<>-1 then Frame21.Image1.PopupMenu:=PopupMenu2
                else Frame21.Image1.PopupMenu:=nil;
            exit;
        end;
    if Button <> mbLeft then exit;
    Base_Selection_Point.X := X;
    Base_Selection_Point.Y := Y;
    Old_Second_Selection_Point := Base_Selection_Point;
    Selection_Status := true;
end;

procedure TForm1.Frame21Image1MouseUp(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var i : integer;
Left_Up, Right_Down : TPoint;
begin
    // Connect_Status:=false;
    if Button <> mbLeft then exit;
    if Button = mbRight then exit;
    Second_Selection_Point.X := X;
    Second_Selection_Point.Y := Y;
    Selection_Status := false;
    DrawSelectionRectangle (Old_Second_Selection_Point);
    if (Base_Selection_Point.X <> Second_Selection_Point.X) or
    (Base_Selection_Point.Y <> Second_Selection_Point.Y) then
        begin
            Left_Up.X := Min(Base_Selection_Point.X,Second_Selection_Point.X);
            Left_Up.Y := Min(Base_Selection_Point.Y,Second_Selection_Point.Y);
            Right_Down.X := Max(Base_Selection_Point.X,Second_Selection_Point.X);
            Right_Down.Y := Max(Base_Selection_Point.Y,Second_Selection_Point.Y);
            for i := 1 to Last_Elem do
                begin
                    if Selection_Metod then
                        if (Elements[i].Image.Left > Left_Up.X) and (Elements[i].Image.Top
> Left_Up.Y) and
                            (Elements[i].Image.Left + Elements[i].Image.Width <
Right_Down.X) and (Elements[i].Image.Top + Elements[i].Image.Height <
Right_Down.Y) then
                                Elements[i].Select;
                            if not Selection_Metod then
                                if (((Elements[i].Image.Left > Left_Up.X) and
(Elements[i].Image.Left < Right_Down.X)) and
                                    ((Elements[i].Image.Top > Left_Up.Y) and
(Elements[i].Image.Top < Right_Down.Y))) or
                                    (((Elements[i].Image.Left + Elements[i].Image.Width >
Left_Up.X) and (Elements[i].Image.Left + Elements[i].Image.Width <
Right_Down.X)) and

```

```

        ((Elements[i].Image.Top > Left_Up.Y) and
(Elements[i].Image.Top < Right_Down.Y)) or
        (((Elements[i].Image.Left + Elements[i].Image.Width >
Left_Up.X) and (Elements[i].Image.Left + Elements[i].Image.Width <
Right_Down.X)) and
        ((Elements[i].Image.Top + Elements[i].Image.Height > Left_Up.Y)
and (Elements[i].Image.Top + Elements[i].Image.Height < Right_Down.Y))) or
        (((Elements[i].Image.Left > Left_Up.X) and
(Elements[i].Image.Left < Right_Down.X)) and
        ((Elements[i].Image.Top + Elements[i].Image.Height > Left_Up.Y)
and (Elements[i].Image.Top + Elements[i].Image.Height < Right_Down.Y))) then
            Elements[i].Select;
        end;
    end else DeselectElements;
    Second_Selection_Point.X := 0;
    Second_Selection_Point.Y := 0;
    Base_Selection_Point.X := 0;
    Base_Selection_Point.Y := 0;
end;

//відкриття вікна додавання мікросхеми
procedure TForm1.N8Click(Sender: TObject);
begin
    form13.ShowModal;
end;

//видалити вибрані елементи
procedure TForm1.N7Click(Sender: TObject);
begin
    Button2Click(Sender);
    AssociateTree;
    Button4Click(Sender);
end;

procedure TForm1.Panel3Click(Sender: TObject);
var i : integer;
begin
    for i := 1 to 135 do
        begin
            panel2.width := panel2.width-1;
            end;
        Panel4.Visible := true;
    end;

procedure TForm1.Panel4Click(Sender: TObject);
var i: integer;
begin
    for i := 1 to 135 do
        begin
            panel2.width := panel2.width+1;
            end;
        panel4.Visible := false;
    end;

//відкриття вікна вибору параметрів програми
procedure TForm1.N10Click(Sender: TObject);
begin
    Form4.ShowModal;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    SaveCfg;
    F_Stop:=true;
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
    Old_Masshtab := Koeff_Mashtab;

```

```

    Koeff_Mashtab := (ComboBox1.ItemIndex + 3);
    DesktopSetup;
    RepaintElements;
end;

//відкриття бази даних зображень професійних знань операторів
автоматизованих систем управління
procedure TForm1.N11Click(Sender: TObject);
begin
    form5.ShowModal;
end;

//відкриття бази даних професійних знань операторів автоматизованих систем
управління
procedure TForm1.N12Click(Sender: TObject);
begin
    form7.ShowModal;
end;

procedure TForm1.DrawField;
var i,j: integer;
    t1,t2,t3:integer;
begin
    t1:=Koeff_Mashtab div 2;
    t2:=Koeff_Mashtab div 2 - 1;
    t3:=Koeff_Mashtab div 5;
    Form1.Frame21.Image1.Canvas.Pen.Mode:=pmXor;
    Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
    Form1.Frame21.Image1.Canvas.Pen.Color:=clWhite;
    for i:=0 to 200 do
        for j:=0 to 200 do begin
            if (Field[i,j]>=10000) and (Field[i,j]<=10010) then
                if MField[i,j] = Selected_Connection then
                    Form1.Frame21.Image1.Canvas.Pen.Color:=Con_Color xor
                    Form1.Frame21.Image1.Canvas.Pixels[1,1]
                    else Form1.Frame21.Image1.Canvas.Pen.Color:=clWhite;
                case Field[i,j] of
                    10000: begin
                        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
                        t1,j*Koeff_Mashtab);
                        Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);
                        end;
                    10001: begin
                        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);
                        Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
                        end;
                    10002: begin
                        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
                        t1,j*Koeff_Mashtab);
                        Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
                        end;
                    10003: begin
                        Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);
                        end;
                    10004: begin
                        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
                        t1,j*Koeff_Mashtab);
                        Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
                        end;
                    10005: begin
                        Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
                        end;
                end;
            end;
        end;
    end;
    Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);
end;

```

```

Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);
    end;
    10005: begin
Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);
Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
    end;
    10006: begin
        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
t1,j*Koeff_Mashtab);
Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);
Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);
Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
    end;
    10010: begin
        Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;
Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab,j*Koeff_Mashtab);
Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
        Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
    end;
    end;
end;

procedure TForm1.Button4Click(Sender: TObject);
var i,j:integer;
begin
    DrawField;
    SetLength(Cons,High(Conns)+1);
    for i:=0 to High(Conns) do
        begin
            SetLength(Cons[i],High(Conns[i])+1);
            for j:= 0 to High(Conns[i]) do
                with Cons[i][j] do
                    begin
X:=elements[Conns[i][j].Elem].Coord.x+Elements[Conns[i][j].Elem].Legs[Conns[i][j].Leg].x;
Y:=elements[Conns[i][j].Elem].Coord.y+Elements[Conns[i][j].Elem].Legs[Conns[i][j].Leg].y;
                    end;
                end;
            Connect;
            DrawField;
        end;

//відкриття бази даних питань
procedure TForm1.N14Click(Sender: TObject);
begin
    form10.ShowModal;
end;

procedure TForm1.Frame21Image1MouseMove(Sender: TObject;
    Shift: TShiftState; X, Y: Integer);
begin
    if Legs_sel then
        begin

```

```

        Legs_sel:=false;
        Draw_Leg(Legs_sel_Elem,Legs_sel_Leg);
    end;
    if Selection_Status then
    begin
        Second_Selection_Point.X := X;
        Second_Selection_Point.Y := Y;
        DrawSelectionRectangle (Second_Selection_Point);
        DrawSelectionRectangle (Old_Second_Selection_Point);
        Old_Second_Selection_Point := Second_Selection_Point;
    end;
    if MField[(x + Koeff_Mashtab div 2) div Koeff_Mashtab, (y +
Koeff_Mashtab div 2) div Koeff_Mashtab]<>Selected_Connection then
    begin
        if Select_connections then DrawField;
        Selected_Connection:=MField[(x + Koeff_Mashtab div 2) div
Koeff_Mashtab, (y + Koeff_Mashtab div 2) div Koeff_Mashtab];
        if Select_connections then DrawField;
    end;
    end;

    //додати елемент чи мікросхему
    procedure TForm1.Button3Click(Sender: TObject);
    begin
        PopupMenu1.Popup(Left+Button3.Left+Button3.Width div
2+Panel2.Left,Button3.Top+Button3.Height+Top+Panel2.Top+24);
    end;

    //видалити елемент чи мікросхему
    procedure TForm1.Button5Click(Sender: TObject);
    var i : integer;
    begin
        if TreeView1.Items.Count<1 then exit;
        if TreeView1.Selected.Parent = nil then
        begin
            DeselectElements;
            for i:=0 to TreeView1.Items.Count - 1 do
            begin
                if (TreeView1.Items[i].Parent = TreeView1.Selected) and
(TreeView1.items[i].imageindex = 1) then
                begin
                    telement(TreeView1.items[i].Data^).Select;
                end;
            end;
            DeleteSelectedElements;
            TreeView1.Selected.Free;
            AssociateTree;
        end
        else
            if TreeView1.Selected.ImageIndex = 1 then
            begin
                DeselectElements;
                telement(TreeView1.Selected.Data^).Select;
                DeleteSelectedElements;
                TreeView1.Selected.ImageIndex := 2;
                TreeView1.Selected.SelectedIndex := 2;
                AssociateTree;
            end;
        Button4Click(Sender);
    end;

    procedure TForm1.TreeView1Db1Click(Sender: TObject);
    var i, j : integer;
        legs1, legs2: String;
    begin
        if TreeView1.Items.Count = 0 then exit;
        if TreeView1.Selected.ImageIndex <= 1 then exit;
        Last_Elem:=Last_Elem+1;
        Elements[Last_Elem].create;
    end;

```

```

legs1:='';
legs2:='';
if TreeView1.Selected.Text[1]='D' then
begin
form10.Table1.first;
for i := 0 to form10.Table1.RecordCount - 1 do
begin
begin
if TreeView1.Selected.Parent.Text =
form10.Table1.fieldbyname('Mark').asstring then break;
form10.Table1.Next;
end;
Elements[Last_Elem].Image.Picture := form5.dbimager1.picture;
Elements[Last_Elem].Image.Canvas.Font.Size := 10;
Elements[Last_Elem].Image.Canvas.Font.Style := [fsBold];
Elements[Last_Elem].Image.Canvas.Brush.Style := bsSolid;
Elements[Last_Elem].Image.Canvas.Brush.Color:= $F0F0F0;

Elements[Last_Elem].Image.Canvas.TextOut(Elements[Last_Elem].Image.Picture.Bitmap
p.Width div 2 -
Elements[Last_Elem].Image.Canvas.TextWidth(TreeView1.Selected.Text) div 2,
Elements[Last_Elem].Image.Picture.Bitmap.Height - 20, TreeView1.Selected.Text);
legs1:=Form7.DBEdit3.Text;
legs2:=Form7.DBEdit4.Text;
Elements[Last_Elem].Reaction_LH:=strtoint(Form10.DBEdit4.Text);
Elements[Last_Elem].Reaction_HL:=strtoint(Form10.DBEdit6.Text);
end;

if TreeView1.Selected.Text[1]='U' then
begin
ImageList2.GetBitmap(0,Elements[Last_Elem].Image.Picture.Bitmap);
legs1:='1';
Elements[Last_Elem].Special_bool:=false;
end;
if TreeView1.Selected.Text[1]='Π' then
begin
ImageList2.GetBitmap(2,Elements[Last_Elem].Image.Picture.Bitmap);
legs2:='1';
Elements[Last_Elem].Image.OnDblClick:=Imager1.OnDblClick;
Elements[Last_Elem].Special_bool:=false;
end;
if TreeView1.Selected.Text[1]='Γ' then
begin
ImageList4.GetBitmap(0,Elements[Last_Elem].Image.Picture.Bitmap);
legs2:='010001';
Elements[Last_Elem].Image.OnDblClick:=Imager1.OnDblClick;
Elements[Last_Elem].Reaction_LH:=100;
Elements[Last_Elem].Reaction_HL:=100;
end;
if TreeView1.Selected.Text[1]='O' then
begin
ImageList4.GetBitmap(1,Elements[Last_Elem].Image.Picture.Bitmap);
legs1:='0001';
Elements[Last_Elem].Image.OnDblClick:=Imager1.OnDblClick;
Elements[Last_Elem].Special_bool:=false;
Oscill[strtoint(copy(TreeView1.Selected.Text,12,2))]:=Last_Elem;
if Form17.Visible then
begin
Form17.CheckListBox1.Checked[strtoint(copy(TreeView1.Selected.Text,12,2))-
1]:=true;
Form17.CheckListBox1ClickCheck(nil);
end;
end;

TreeView1.Selected.ImageIndex := 1;
TreeView1.Selected.SelectedIndex := 1;
TreeView1.Selected.Data := @elements[last_elem];

Elements[Last_Elem].Script.AddStrings(Form7.DBMemol.Lines);

```

```

Elements[Last_Elem].Caption := TreeView1.Selected.Text;
Elements[Last_Elem].Image.tag := Last_Elem;
Elements[Last_Elem].Image.Left:=2;
Elements[Last_Elem].Image.Top:=2;

Elements[Last_Elem].Image.Width:=Elements[Last_Elem].Image.Picture.Bitmap.Width*
Koeff_Mashtab div 10;

Elements[Last_Elem].Image.Height:=Elements[Last_Elem].Image.Picture.Bitmap.Heigh
t*Koeff_Mashtab div 10;
  Elements[Last_Elem].Image.Proportional:=true;
  Elements[Last_Elem].Image.Transparent:=true;
  Elements[Last_Elem].Image.Stretch:=true;
  Elements[Last_Elem].Image.Parent:=Frame21.Panell;
  Elements[Last_Elem].Image.OnMouseDown:=Image1.OnMouseDown;
  Elements[Last_Elem].Image.OnMouseMove:=Image1.OnMouseMove;
  Elements[Last_Elem].Image.OnMouseUp:=Image1.OnMouseUp;

Elements[Last_Elem].Size.x:=Elements[Last_Elem].Image.Picture.Bitmap.Width div
10;

Elements[Last_Elem].Size.y:=Elements[Last_Elem].Image.Picture.Bitmap.Height div
10;
  Elements[Last_Elem].Coord.x:=Elements[Last_Elem].Image.Left div
Koeff_Mashtab;
  Elements[Last_Elem].Coord.y:=Elements[Last_Elem].Image.Top div
Koeff_Mashtab;
  if Select_on_creation then Elements[Last_Elem].Select;

  j:=0;
  for i:=1 to Length(legs1) do
    if legs1[i]='1' then j:=j+1;
  for i:=1 to Length(legs2) do
    if legs2[i]='1' then j:=j+1;
  SetLength(Elements[Last_Elem].Legs, j);
  j:=0;
  for i:=1 to Length(legs1) do
    if legs1[i]='1' then
      with Elements[Last_Elem] do begin
        Legs[j].x:=0;
        Legs[j].y:=i;
        j:=j+1;
      end;
  for i:=1 to Length(legs2) do
    if legs2[i]='1' then
      with Elements[Last_Elem] do begin
        Legs[j].x:=Size.x;
        Legs[j].y:=i;
        j:=j+1;
      end;
  InitTrace;
end;

procedure TForm1.TreeView1Click(Sender: TObject);
var i : integer;
begin
  if TreeView1.Items.Count = 0 then exit;
  DeselectElements;
  if TreeView1.Selected.ImageIndex = 1 then
    telement(TreeView1.Selected.Data^).Select;
  if (TreeView1.Selected.ImageIndex = 0) and
(TreeView1.Selected.Text[1]<>'C') then
    begin
      for i := 0 to TreeView1.Items.Count -1 do
        if (TreeView1.Items[i].Parent = TreeView1.Selected) and
(TreeView1.Items[i].ImageIndex = 1) then
          telement(TreeView1.items[i].Data^).Select;
    end;
end;
end;

```

```

procedure TForm1.N19Click(Sender: TObject);
begin
  form13.ShowModal;
end;

//додати індикатор
procedure TForm1.N21Click(Sender: TObject);
var i,j,k:integer;
    t,l:TTreeNode;
begin
  t:=nil;
  for i:=0 to TreeView1.Items.Count-1 do
    if TreeView1.Items[i].Text='Спец. елементи' then
      begin
        t:=TreeView1.Items[i];
        break;
      end;
  if (t=nil) or (TreeView1.Items.Count=0) then
    t:=TreeView1.Items.AddChild(nil, 'Спец. елементи');
  j:=0;
  k:=0;
  for i:=1 to TreeView1.Items.Count-1 do
    if copy(TreeView1.Items[i].Text,1,9)='Індикатор' then
      begin
        if j<strtoint(copy(TreeView1.Items[i].Text,10,2)) then
          j:=strtoint(copy(TreeView1.Items[i].Text,10,2));
          k:=i;
        end;
        j:=j+1;
        if k=0 then l:=TreeView1.Items.AddChild(t, 'Індикатор1')
          else if k=TreeView1.Items.Count-1 then
            l:=TreeView1.Items.Add(TreeView1.Items[k], 'Індикатор'+inttostr(j))
          else
            l:=TreeView1.Items.Insert(TreeView1.Items[k+1], 'Індикатор'+inttostr(j));
            l.ImageIndex:=2;
            l.SelectedIndex:=2;
        end;
      end;

  procedure TForm1.N24Click(Sender: TObject);
  var i,j,k:integer;
      t,l:TTreeNode;
  begin
    t:=nil;
    for i:=0 to TreeView1.Items.Count-1 do
      if TreeView1.Items[i].Text='Спец. елементи' then
        begin
          t:=TreeView1.Items[i];
          break;
        end;
    if (t=nil) or (TreeView1.Items.Count=0) then
      t:=TreeView1.Items.AddChild(nil, 'Спец. елементи');
    j:=0;
    k:=0;
    for i:=1 to TreeView1.Items.Count-1 do
      if copy(TreeView1.Items[i].Text,1,13)='Перемикач' then
        begin
          if j<strtoint(copy(TreeView1.Items[i].Text,14,2)) then
            j:=strtoint(copy(TreeView1.Items[i].Text,14,2));
            k:=i;
          end;
          j:=j+1;
          if k=0 then l:=TreeView1.Items.AddChild(t, 'Перемикач1')
            else if k=TreeView1.Items.Count-1 then
              l:=TreeView1.Items.Add(TreeView1.Items[k], 'Перемикач'+inttostr(j))
            else
              l:=TreeView1.Items.Insert(TreeView1.Items[k+1], 'Перемикач'+inttostr(j));
              l.ImageIndex:=2;
              l.SelectedIndex:=2;
          end;
        end;
      end;

```

```

end;

procedure TForm1.Image1DbClick(Sender: TObject);
begin
  if Elements[TImage(Sender).Tag].Caption[1]='Π' then
  begin
    Elements[TImage(Sender).Tag].Special_bool := not
Elements[TImage(Sender).Tag].Special_bool;
    if Elements[TImage(Sender).Tag].Special_bool then

ImageList2.GetBitmap(3,Elements[TImage(Sender).Tag].Image.Picture.Bitmap)
    else
ImageList2.GetBitmap(2,Elements[TImage(Sender).Tag].Image.Picture.Bitmap);
    TImage(Sender).Repaint;
    Elements[TImage(Sender).Tag].Selected:=false;
  end;
  if Elements[TImage(Sender).Tag].Caption[1]='O' then
  begin
    UpdateOsc;
    Form17.Show;
  end;
  if Elements[TImage(Sender).Tag].Caption[1]='Γ' then
  begin
    F_Pause:=true;

Form16.Edit1.Text:=inttostr(Elements[TImage(Sender).Tag].Reaction_HL);
Form16.Edit2.Text:=inttostr(Elements[TImage(Sender).Tag].Reaction_LH);
    if Form16.ShowModal=mrOk then
    begin
      try

Elements[TImage(Sender).Tag].Reaction_HL:=strtoint(Form16.Edit1.Text);
Elements[TImage(Sender).Tag].Reaction_LH:=strtoint(Form16.Edit2.Text);
      except
        end;
      end;
      Elements[TImage(Sender).Tag].DeSelect;
      F_Pause:=false;
    end;
  end;

procedure TForm1.N25Click(Sender: TObject);
var j: Integer;
begin
  SetLength(Conns[Selected_Connection],0);
  for j:= Selected_Connection to High(Conns)-1 do
    Conns[j]:=Conns[j+1];
  SetLength(Conns,High(Conns));
  Label1.Caption:=inttostr(High(Conns)+1);
  Button4Click(Sender);
  InitTrace;
end;

procedure TForm1.N26Click(Sender: TObject);
var l,j,k : Integer;
begin
  for j:=0 to High(Conns) do
  begin
    k:=0;
    while k<=High(Conns[j]) do
    begin
      if (Conns[j][k].Elem=Connect_Start.Elem) and
(Conns[j][k].Leg=Connect_Start.Leg) then
      begin
        for l:=k to High(Conns[j])-1 do
          Conns[j][l]:=Conns[j][l+1];
        k:=k-1;

```

```

        SetLength (Conns [j], High (Conns [j]));
    end;
    k:=k+1;
end;
end;
j:=0;
while j<=High (Conns) do
begin
    if High (Conns [j])<1 then
    begin
        for l:=j to High (Conns)-1 do
            Conns [l]:=Conns [l+1];
        SetLength (Conns, High (Conns));
    end;
    j:=j+1;
end;
Button4Click (nil);
InitTrace;
end;

//запуск процесу емуляції роботи створеної цифрової схеми
procedure TForm1.ToolButton6Click (Sender: TObject);
begin
    F_Stop:=false;
    F_Pause:=false;
    ToolButton6.Enabled:=false;
    ToolButton7.Enabled:=true;
    ToolButton8.Enabled:=true;

    InitTrace;
    // TrTime:=0;
    repeat
        Delay (Form15.TrackBar1.Position);
        if not F_Pause then Trace;
        if not F_Pause then Trace;
        if not F_Pause then TrTime := TrTime + 1;
        if (Test_Count>0) and Form17.Visible {and
Form17.CheckListBox1.Checked [0]} then Form17.Button2Click (nil);
        Application.ProcessMessages;
    until F_Stop;
    ToolButton6.Enabled:=true;
    ToolButton7.Enabled:=false;
    ToolButton8.Enabled:=false;
end;

//стоп
procedure TForm1.ToolButton7Click (Sender: TObject);
begin
    F_Stop:=true;
end;

//пауза
procedure TForm1.ToolButton8Click (Sender: TObject);
begin
    F_Pause:=not F_Pause;
end;

//вибір масштабу часу
procedure TForm1.ToolButton9Click (Sender: TObject);
begin
    Form15.Visible:=ToolButton9.Down;
    SetFocus;
end;

procedure TForm1.N27Click (Sender: TObject);
var i, j, k: integer;
    t, l: TTreeNode;
begin
    t:=nil;

```

```

for i:=0 to TreeView1.Items.Count-1 do
  if TreeView1.Items[i].Text='Спец. элементы' then
    begin
      t:=TreeView1.Items[i];
      break;
    end;
  if (t=nil) or (TreeView1.Items.Count=0) then
    t:=TreeView1.Items.AddChild(nil,'Спец. элементы');
  j:=0;
  k:=0;
  for i:=1 to TreeView1.Items.Count-1 do
    if copy(TreeView1.Items[i].Text,1,9)='Генератор' then
      begin
        if j<strtoint(copy(TreeView1.Items[i].Text,10,2)) then
          j:=strtoint(copy(TreeView1.Items[i].Text,10,2));
          k:=i;
        end;
        j:=j+1;
        if k=0 then l:=TreeView1.Items.AddChild(t,'Генератор1')
        else if k=TreeView1.Items.Count-1 then
          l:=TreeView1.Items.Add(TreeView1.Items[k],'Генератор'+inttostr(j))
        else
          l:=TreeView1.Items.Insert(TreeView1.Items[k+1],'Генератор'+inttostr(j));
          l.ImageIndex:=2;
          l.SelectedIndex:=2;
        end;

procedure TForm1.N28Click(Sender: TObject);
var i,j,k:integer;
    t,l:TTreeNode;
begin
  t:=nil;
  for i:=0 to TreeView1.Items.Count-1 do
    if TreeView1.Items[i].Text='Спец. элементы' then
      begin
        t:=TreeView1.Items[i];
        break;
      end;
    if (t=nil) or (TreeView1.Items.Count=0) then
      t:=TreeView1.Items.AddChild(nil,'Спец. элементы');
    j:=0;
    k:=0;
    for i:=1 to TreeView1.Items.Count-1 do
      if copy(TreeView1.Items[i].Text,1,11)='Тест' then
        begin
          if j<strtoint(copy(TreeView1.Items[i].Text,12,2)) then
            j:=strtoint(copy(TreeView1.Items[i].Text,12,2));
            k:=i;
          end;
          if j=5 then exit;
          j:=j+1;
          if k=0 then l:=TreeView1.Items.AddChild(t,'Тест1')
          else if k=TreeView1.Items.Count-1 then
            l:=TreeView1.Items.Add(TreeView1.Items[k],'Тест'+inttostr(j))
          else
            l:=TreeView1.Items.Insert(TreeView1.Items[k+1],'Тест'+inttostr(j));
            if k=0 then Test_Count:=1 else Test_Count:=j;
            l.ImageIndex:=2;
            l.SelectedIndex:=2;
            if Form17.Visible then
              begin
                UpdateOsc;
                Form17.CheckListBox1ClickCheck(nil);
              end;
            end;
          end;

//тести
procedure TForm1.N15Click(Sender: TObject);
begin

```

```

    if Test_Count < 1 then Exit;
    UpdateOsc;
    Form17.Show;
end;

procedure TForm1.FormShow(Sender: TObject);
begin
    InitCfg;
    DesktopSetup;
end;

//збереження цифрової схеми у файл
procedure TForm1.Button6Click(Sender: TObject);
//var fEl : file of Telement;
var i, j : integer;
    fdata : TElemsave;
    CData : TConn;
    FN : string;
begin
    if not SaveDialog1.Execute then exit;
    FN := SaveDialog1.FileName;

TreeView1.SaveToFile (ExtractFilePath (FN)+copy (ExtractFileName (FN), 1, length (Extra
ctFileName (FN))-3)+'tsv');

assignfile (fio, ExtractFilePath (FN)+copy (ExtractFileName (FN), 1, length (ExtractFile
Name (FN))-3)+'esv');
rewrite (fio);
for i:= 1 to Last_Elem do
    begin
        fdata.coord.X := elements[i].Coord.X;
        fdata.coord.Y := elements[i].Coord.Y;
        fdata.Caption := elements[i].Caption;
        fdata.React_HL := elements[i].Reaction_HL;
        fdata.React_LH := elements[i].Reaction_LH;
        write (fio, fdata);
    end;
closefile (fio);

assignfile (fio2, ExtractFilePath (FN)+copy (ExtractFileName (FN), 1, length (ExtractFil
eName (FN))-3)+'csv');
rewrite (fio2);
for i:= 0 to High (Conns) do
    begin
        for j:= 0 to High (Conns[i]) do
            begin
                CData:=Conns[i][j];
                write (fio2, CData);
            end;
        CData.Elem:=0;
        write (fio2, CData);
    end;
closefile (fio2);
DesktopSetup;
Button4Click (nil);
end;

//завантаження цифрової схеми з файлу
procedure TForm1.Button7Click(Sender: TObject);
var i : integer;
    fdata : TElemsave;
    CData : TConn;
    FN : String;
begin
    if not OpenFileDialog1.Execute then exit;
    FN := OpenFileDialog1.FileName;
    Button8Click (nil);

```

```

TreeView1.LoadFromFile(ExtractFilePath(FN)+copy(ExtractFileName(FN),1,length(ExtractFileName(FN))-3)+'tsv');
  for i:= 0 to TreeView1.Items.Count -1 do
    if TreeView1.Items[i].Parent <> nil then
      begin
        TreeView1.Items[i].ImageIndex := 2;
        TreeView1.Items[i].SelectedIndex := 2;
      end;

assignfile(fio,ExtractFilePath(FN)+copy(ExtractFileName(FN),1,length(ExtractFileName(FN))-3)+'esv');
  reset(fio);
  while not EOF(fio) do
    begin
      read(fio, fdata);
      for i:= 0 to TreeView1.Items.Count - 1 do
        if TreeView1.Items[i].Text = fdata.Caption then
          begin
            TreeView1.Selected := TreeView1.Items[i];
            TreeView1Db1Click(nil);
            Elements[Last_Elem].Coord.x := fdata.coord.X;
            Elements[Last_Elem].Coord.Y := fdata.coord.Y;
            Elements[Last_Elem].Image.Left :=
fdata.coord.X*Koeff_Mashtab+2;
            Elements[Last_Elem].Image.top :=
fdata.coord.Y*Koeff_Mashtab+2;
            Elements[Last_Elem].Reaction_LH := fdata.React_LH;
            Elements[Last_Elem].Reaction_HL := fdata.React_HL;
            if Elements[Last_Elem].Caption[1] = 'O' then
Test_Count:=Test_Count+1;
          end;
        end;
      closefile (fio);

assignfile(fio2,ExtractFilePath(FN)+copy(ExtractFileName(FN),1,length(ExtractFileName(FN))-3)+'csv');
  reset(fio2);
  while not EOF(fio2) do
    begin
      SetLength(Conns,High(Conns)+2);
      i:=High(Conns);
      while not EOF(fio2) do
        begin
          read(fio2,cdata);
          if CData.Elem=0 then break;
          SetLength(Conns[i],High(Conns[i])+2);
          Conns[i][High(Conns[i])]:=CData;
        end;
      end;
    closefile (fio2);
    Button4Click(nil);
    UpdateOsc;
    DesktopSetup;
    Button4Click(nil);
  end;

//створення нового проекту
procedure TForm1.Button8Click(Sender: TObject);
var i: Integer;
begin
  TreeView1.Items.Clear;
  SelectAllElements;
  DeleteSelectedElements;
  Button4Click(nil);
  Test_Count:=0;
  for i:=1 to 5 do
    Oscill[i]:=0;

```

```
Form17.Hide;
end;

//вихід з програми
procedure TForm1.N34Click(Sender: TObject);
begin
  Close;
end;

//відкриття довідки
procedure TForm1.N37Click(Sender: TObject);
begin
  Form19.Show;
end;

end.
```

КБПЗ\_2023

## Файл Parameters.pas - вікно вибору параметрів програми

```

unit Parameters;

interface

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

type
  TForm4 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Panel1: TPanel;
    GroupBox1: TGroupBox;
    CheckBox1: TCheckBox;
    GroupBox2: TGroupBox;
    ColorBox1: TColorBox;
    ColorBox2: TColorBox;
    Label4: TLabel;
    Label5: TLabel;
    CheckBox3: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox4: TCheckBox;
    Image1: TImage;
    procedure FormActivate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure ColorBox1Change(Sender: TObject);
    procedure ColorBox2Change(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure CheckBox3Click(Sender: TObject);
    procedure CheckBox4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var Form4: TForm4;
implementation

uses Main;
{$R *.dfm}

procedure TForm4.FormActivate(Sender: TObject);
begin
  CheckBox1.Checked := Selection_Metod;
  ColorBox1.Selected := Desktop_Color;
  ColorBox2.Selected := Grid_Color;
  CheckBox2.Checked := Select_on_creation;
  CheckBox3.Checked := Draw_connections;
  CheckBox4.Checked := Select_connections;
end;

//процедура, що обробляє натиснення на кнопку "Відміна"
procedure TForm4.Button2Click(Sender: TObject);
begin
  close
end;

//процедура, що обробляє натиснення на кнопку "Застосувати"
procedure TForm4.Button3Click(Sender: TObject);
begin

```

```
Form1.SaveCfg;
end;

//процедура, що обробляє натиснення на кнопку "ОК"
procedure TForm4.Button1Click(Sender: TObject);
begin
    Form1.SaveCfg;
    Close
end;

procedure TForm4.ColorBox1Change(Sender: TObject);
begin
    Desktop_Color := ColorBox1.Selected; //вибір кольору фону
    Form1.DesktopSetup;
end;

procedure TForm4.ColorBox2Change(Sender: TObject);
begin
    Grid_Color := ColorBox2.Selected; //вибір кольору розмітки
    Form1.DesktopSetup;
end;

procedure TForm4.CheckBox1Click(Sender: TObject);
begin
    Selection_Metod := CheckBox1.Checked; //якщо вибрано, то виділяти тільки
повністю обведений елемент
end;

procedure TForm4.CheckBox2Click(Sender: TObject);
begin
    Select_on_creation := CheckBox2.Checked; //якщо вибрано, то виділяти
елементи при додаванні в схему
end;

procedure TForm4.CheckBox3Click(Sender: TObject);
begin
    Draw_connections := CheckBox3.Checked; //якщо вибрано, то промальовувати
з'єднання при пересуванні
end;

procedure TForm4.CheckBox4Click(Sender: TObject);
begin
    Select_connections:= CheckBox4.Checked; //якщо вибрано, то виділяти
з'єднання кольором
end; end.
```

## Файл Test.pas - вікно тестів

```

unit Test;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Main8, StdCtrls, CheckLst, Main, Tracer;

type
  TForm17 = class(TForm)
    GroupBox2: TGroupBox;
    Frame181: TFrame18;
    Button2: TButton;
    CheckListBox1: TCheckListBox;
    procedure CheckListBox1ClickCheck(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form17: TForm17;

procedure UpdateOsc;

implementation

uses Unit3;

{$R *.dfm}

procedure UpdateOsc;
var i:integer;
begin
  with Form17 do begin
    Frame181.Panel1.Visible:=false;
    Frame181.Panel2.Visible:=false;
    Frame181.Panel3.Visible:=false;
    Frame181.Panel4.Visible:=false;
    Frame181.Panel5.Visible:=false;
    CheckListBox1.Items.Clear;
    for i := 1 to Test_Count do
      begin
        CheckListBox1.Items.Add('Тест'+inttostr(i));
        CheckListBox1.Checked[i-1]:=true;
      end;
    CheckListBox1ClickCheck(nil);
  end;
end;

procedure TForm17.CheckListBox1ClickCheck(Sender: TObject);
var i:integer;
label 1;
begin
  for i:=1 to CheckListBox1.Items.Count do
    if CheckListBox1.Checked[i-1] and (Oscill[i]=0) then
      CheckListBox1.Checked[i-1]:=false;
  i:=0;
  if Test_Count<1 then goto 1;
  if CheckListBox1.Checked[0] then
    begin
      Frame181.Panel1.Visible:=true;
      Frame181.Panel1.Top:=i;
    end;
  goto 1;
end;

```

```

        i:=i+60;
    end else Frame181.Panel1.Visible:=false;
if Test_Count<2 then goto 1;
if CheckListBox1.Checked[1] then
begin
    Frame181.Panel2.Visible:=true;
    Frame181.Panel2.Top:=i;
    i:=i+60;
end else Frame181.Panel2.Visible:=false;
if Test_Count<3 then goto 1;
if CheckListBox1.Checked[2] then
begin
    Frame181.Panel3.Visible:=true;
    Frame181.Panel3.Top:=i;
    i:=i+60;
end else Frame181.Panel3.Visible:=false;
if Test_Count<4 then goto 1;
if CheckListBox1.Checked[3] then
begin
    Frame181.Panel4.Visible:=true;
    Frame181.Panel4.Top:=i;
    i:=i+60;
end else Frame181.Panel4.Visible:=false;
if Test_Count<5 then goto 1;
if CheckListBox1.Checked[4] then
begin
    Frame181.Panel5.Visible:=true;
    Frame181.Panel5.Top:=i;
    i:=i+60;
end else Frame181.Panel5.Visible:=false;
1:
    if i<>0 then Form17.Height:=65+i;
end;

procedure TForm17.Button2Click(Sender: TObject);
var i,j,k:Integer;
begin
    if Frame181.Panel1.Visible then with Elements[Oscill[1]] do
begin
    j:=30;
    k:=0;
//    Frame181.ScrollBar1.Max:=TrTime-Frame181.Image1.Width;
    Frame181.Image1.Picture:=nil;
    if TrTime<Frame181.Image1.Width then
Frame181.Image1.Canvas.MoveTo(Frame181.Image1.Width-TrTime-2,j)
    else Frame181.Image1.Canvas.MoveTo(-1,j);
    for i:=0 to TrTime do
begin
        if High(Test_Data)<>-1 then
            if Test_Data[k]=i then
begin
                if j=30 then j:=5 else j:=30;
                k:=k+1;
            end;
            if TrTime-i<=Frame181.Image1.Width then
Frame181.Image1.Canvas.LineTo(Frame181.Image1.Width-TrTime+i-1,j);
            end;
        end;
    if Frame181.Panel2.Visible then with Elements[Oscill[2]] do
begin
    j:=30;
    k:=0;
//    Frame181.ScrollBar1.Max:=TrTime-Frame181.Image1.Width;
    Frame181.Image2.Picture:=nil;
    if TrTime<Frame181.Image2.Width then
Frame181.Image2.Canvas.MoveTo(Frame181.Image2.Width-TrTime-2,j)
    else Frame181.Image2.Canvas.MoveTo(-1,j);

```

```

for i:=0 to TrTime do
begin
  if High(Test_Data)<>-1 then
    if Test_Data[k]=i then
      begin
        if j=30 then j:=5 else j:=30;
        k:=k+1;
      end;
    if TrTime-i<=Frame181.Image2.Width then
Frame181.Image2.Canvas.LineTo(Frame181.Image2.Width-TrTime+i-1,j);
    end;
  end;

  if Frame181.Panel3.Visible then with Elements[Oscill[3]] do
begin
  j:=30;
  k:=0;
//   Frame181.ScrollBar1.Max:=TrTime-Frame181.Image1.Width;
  Frame181.Image3.Picture:=nil;
  if TrTime<Frame181.Image3.Width then
Frame181.Image3.Canvas.MoveTo(Frame181.Image3.Width-TrTime-2,j)
  else Frame181.Image3.Canvas.MoveTo(-1,j);
  for i:=0 to TrTime do
begin
  if High(Test_Data)<>-1 then
    if Test_Data[k]=i then
      begin
        if j=30 then j:=5 else j:=30;
        k:=k+1;
      end;
    if TrTime-i<=Frame181.Image3.Width then
Frame181.Image3.Canvas.LineTo(Frame181.Image3.Width-TrTime+i-1,j);
    end;
  end;
end;

  if Frame181.Panel4.Visible then with Elements[Oscill[4]] do
begin
  j:=30;
  k:=0;
//   Frame181.ScrollBar1.Max:=TrTime-Frame181.Image1.Width;
  Frame181.Image4.Picture:=nil;
  if TrTime<Frame181.Image4.Width then
Frame181.Image4.Canvas.MoveTo(Frame181.Image4.Width-TrTime-2,j)
  else Frame181.Image4.Canvas.MoveTo(-1,j);
  for i:=0 to TrTime do
begin
  if High(Test_Data)<>-1 then
    if Test_Data[k]=i then
      begin
        if j=30 then j:=5 else j:=30;
        k:=k+1;
      end;
    if TrTime-i<=Frame181.Image4.Width then
Frame181.Image4.Canvas.LineTo(Frame181.Image4.Width-TrTime+i-1,j);
    end;
  end;
end;

  if Frame181.Panel5.Visible then with Elements[Oscill[5]] do
begin
  j:=30;
  k:=0;
//   Frame181.ScrollBar1.Max:=TrTime-Frame181.Image1.Width;
  Frame181.Image5.Picture:=nil;
  if TrTime<Frame181.Image5.Width then
Frame181.Image5.Canvas.MoveTo(Frame181.Image5.Width-TrTime-2,j)
  else Frame181.Image5.Canvas.MoveTo(-1,j);
  for i:=0 to TrTime do
begin
  if High(Test_Data)<>-1 then

```

```
        if Test_Data[k]=i then
            begin
                if j=30 then j:=5 else j:=30;
                k:=k+1;
            end;
        if TrTime-i<=Frame181.Image5.Width then
Frame181.Image5.Canvas.LineTo(Frame181.Image5.Width-TrTime+i-1,j);
        end;
    end;
end;

end.
```

К6П3\_2023

**Файл Db\_elements.pas - доступ до бази даних професійних знань операторів  
автоматизованих систем управління**

```

unit Db_elements;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, ExtCtrls, Grids, DBGrids, DBCtrls, StdCtrls, Mask,
  ComCtrls;

type
  TForm7 = class(TForm)
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    Table1: TTable;
    DataSource1: TDataSource;
    Panel2: TPanel;
    Panel3: TPanel;
    Button5: TButton;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    GroupBox1: TGroupBox;
    DBMemo1: TDBMemo;
    TabSheet2: TTabSheet;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    Label3: TLabel;
    Label4: TLabel;
    GroupBox5: TGroupBox;
    Label5: TLabel;
    Button4: TButton;
    DBEdit1: TDBEdit;
    Panel4: TPanel;
    Image1: TImage;
    Button6: TButton;
    Button7: TButton;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    Procedure FindPicture;
    procedure DBGrid1KeyPress(Sender: TObject; var Key: Char);
    procedure DBGrid1KeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure DBGrid1CellClick(Column: TColumn);
    procedure Button4Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure DrawLegs;
    procedure DBEdit3Change(Sender: TObject);
    procedure DBEdit4Change(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Table1AfterScroll(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form7: TForm7;

```

implementation

uses

```
    unit5, Unit8, Unit9;
    {$R *.dfm}
```

```
Procedure TForm7.DrawLegs;
```

```
var i : integer;
```

```
begin
```

```
    if form5 <> nil then
```

```
        form7.Imagel.Picture := form5.DBImagel.Picture;
```

```
    Imagel.Canvas.Pen.Color := clRed;
```

```
    for i := 1 to length (DBEdit3.text) do
```

```
        begin
```

```
            if DBEdit3.text[i] = '1' then Imagel.Canvas.Ellipse(0,i*10-3,5,i*10+3);
```

```
        end;
```

```
    for i := 1 to length (DBEdit4.text) do
```

```
        begin
```

```
            if DBEdit4.text[i] = '1' then
```

```
                Imagel.Canvas.Ellipse(Imagel.Picture.Bitmap.Width-6,i*10-3,Imagel.Picture.Bitmap.Width,i*10+3);
```

```
            end;
```

```
        end;
```

```
Procedure TForm7.FindPicture;
```

```
var i: integer;
```

```
begin
```

```
    if form5.Table1.FieldByName('Name').AsString =  
table1.FieldByName('Image_Name').AsString then exit;
```

```
    form5.Table1.First;
```

```
    for i:= 1 to form5.Table1.RecordCount -1 do
```

```
        begin
```

```
            if form5.Table1.FieldByName('Name').AsString =  
table1.FieldByName('Image_Name').AsString then break;
```

```
            form5.Table1.Next;
```

```
        end;
```

```
    end;
```

```
procedure TForm7.FormCreate(Sender: TObject);
```

```
begin
```

```
    Table1.DatabaseName := ExtractFilePath(Application.ExeName)+'bases\';
```

```
    Table1.TableName := 'elements.db';
```

```
    Table1.Active := true;
```

```
end;
```

```
procedure TForm7.DBGrid1KeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
// if key = #72 then FindPicture;
```

```
end;
```

```
procedure TForm7.DBGrid1KeyUp(Sender: TObject; var Key: Word;
```

```
    Shift: TShiftState);
```

```
begin
```

```
    if (key = VK_UP) or (key = VK_DOWN) then FindPicture;
```

```
end;
```

```
procedure TForm7.DBGrid1CellClick(Column: TColumn);
```

```
begin
```

```
    FindPicture;
```

```
end;
```

```
procedure TForm7.Button4Click(Sender: TObject);
```

```
begin
```

```
    form8.ShowModal;
```

```
end;
```

```
procedure TForm7.FormActivate(Sender: TObject);
```

```
begin
```

```
    FindPicture;    form7.Imagel.Picture := form5.DBImagel.Picture;
```

```
    DrawLegs;
end;

procedure TForm7.DBEdit3Change(Sender: TObject);
begin
    DrawLegs;
end;

procedure TForm7.DBEdit4Change(Sender: TObject);
begin
    DrawLegs;
end;

procedure TForm7.Button5Click(Sender: TObject);
begin
    close;
    Table1.Edit;
    Table1.Post;
end;

procedure TForm7.Button7Click(Sender: TObject);
begin
    if Table1.RecordCount > 0 then
    begin
        begin
            if Application.MessageBox('Ви дійсно хочете видалити цей елемент',
            'Увага', MB_YESNO) = IDYES then
                Table1.Delete;
            end;
        end;
    end;

procedure TForm7.Button6Click(Sender: TObject);
begin
    form9.ShowModal;
end;

procedure TForm7.Button1Click(Sender: TObject);
begin
    Table1.Edit;
    Table1.Post;
end;

procedure TForm7.Table1AfterScroll(DataSet: TDataSet);
begin
    if form5 <> nil then
    begin
        begin
            FindPicture;    form7.Image1.Picture := form5.DBImage1.Picture;
            DrawLegs;
        end
    end;
end;

end.
```

## Файл Db\_micro.pas - доступ до бази даних питань

```

unit Db_micro;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, Grids, DBGrids, DB, DBTables, ExtCtrls, StdCtrls,
  Mask, DBCtrls;

type
  TForm10 = class(TForm)
    Panel1: TPanel;
    Table1: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    GroupBox1: TGroupBox;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    Button1: TButton;
    DBEdit2: TDBEdit;
    Label2: TLabel;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    GroupBox4: TGroupBox;
    Panel2: TPanel;
    Image1: TImage;
    GroupBox5: TGroupBox;
    Label3: TLabel;
    Button2: TButton;
    DBEdit5: TDBEdit;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    DBEdit6: TDBEdit;
    Label4: TLabel;
    Label5: TLabel;
    Panel3: TPanel;
    TabSheet3: TTabSheet;
    procedure FormCreate(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure DBGrid1TitleClick(Column: TColumn);
    procedure Table1AfterScroll(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form10: TForm10;

implementation

uses Main1, Main2, Db_elements;

{$R *.dfm}

```

```

procedure TForm10.FormCreate(Sender: TObject);
begin
  Table1.DatabaseName := ExtractFilePath(Application.ExeName)+'bases\';
  Table1.TableName := 'microchips.db';
  Table1.Active := true;
end;

procedure TForm10.Button5Click(Sender: TObject);
begin
  Close;
  Table1.Edit;
  Table1.Post;
end;

procedure TForm10.Button4Click(Sender: TObject);
begin
  if Table1.RecordCount > 0 then
  begin
    begin
      if Application.MessageBox('Ви дійсно хочете видалити поточний елемент',
'Увага', MB_YESNO) = IDYES then
        Table1.Delete;
    end;
  end;
end;

procedure TForm10.Button3Click(Sender: TObject);
begin
  form11.ShowModal;
end;

procedure TForm10.Button1Click(Sender: TObject);
begin
  form12.ShowModal;
end;

procedure TForm10.DBGrid1TitleClick(Column: TColumn);
begin
  if column.Index = 0 then begin Table1.IndexFieldNames := 'Mark';
DBGrid1.Columns[0].Title.Caption := 'Маркировка*';
DBGrid1.Columns[1].Title.Caption := 'Тип'; end;;
  if column.Index = 1 then begin Table1.IndexFieldNames := 'Type';
DBGrid1.Columns[0].Title.Caption := 'Маркировка';
DBGrid1.Columns[1].Title.Caption := 'Тип*'; end;
end;

procedure TForm10.Table1AfterScroll(DataSet: TDataSet);
var i: integer;
begin
  Form7.Table1.First;
  for i := 0 to form7.Table1.RecordCount - 1 do
  begin
    if form7.Table1.fieldbyname('Name').AsString =
table1.FieldName('Elem_name').AsString then break;
    form7.Table1.Next;
  end;
end;
end.

```

**Файл Search.pas - вікно пошуку професійних знань операторів автоматизованих систем управління та питань**

```

unit Search;

interface

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

type
  TForm14 = class(TForm)
    LabeledEdit1: TLabeledEdit;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Image1: TImage;
    procedure FormActivate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form14: TForm14;

implementation

uses Db_micro;

{$R *.dfm}

procedure TForm14.FormActivate(Sender: TObject);
begin
  LabeledEdit1.SetFocus;
  LabeledEdit1.SelStart := 0;
  LabeledEdit1.SelLength := 255;
end;

procedure TForm14.Button1Click(Sender: TObject);
var i : integer;
begin
  form10.Table1.first;
  for i := 0 to form10.Table1.RecordCount - 1 do
    begin
      if pos (LabeledEdit1.Text, form10.Table1.fieldbyname('Mark').asString) <>
        0 then begin close; {Form10.DBGrid1.SetFocus;} exit end;
      form10.Table1.Next;
    end;
  form10.Table1.First;
  Application.MessageBox('Елемент не знайдено', 'Увага!', MB_OK);
  Close;
  Exit; end; end.

```

## Файл Time\_scale.pas - вікно вибору масштабу часу

```

unit Time_scale;

interface

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

type
  TForm15 = class(TForm)
    TrackBar1: TTrackBar;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormActivate(Sender: TObject);
    procedure TrackBar1Change(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form15: TForm15;

implementation

uses Main;

{$R *.dfm}

procedure TForm15.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Form1.ToolButton9.Down:=false;
end;

procedure TForm15.FormActivate(Sender: TObject);
begin
  Left:=Form1.Left+Form1.Frame21.Left+10;
  Top:=Form1.Top+Form1.Frame21.Top+45;
end;

procedure TForm15.TrackBar1Change(Sender: TObject);
begin
  Edit1.Text:=inttostr(TrackBar1.Position);
end;

procedure TForm15.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if Key=#13 then
    begin
      try
        if strtoint(Edit1.Text)>1000 then Edit1.Text:='1000';
        TrackBar1.Position:=strtoint(Edit1.Text);
      except
        Edit1.Text:=inttostr(TrackBar1.Position);
      end;
    end;
  exit;
end;
case Key of
  '0'..'9':;
  #8:;
end;

```

```
else
  Key:=chr(0);
end;
end;
end.
```

КБПЗ\_2023

## Файл About.pas - вікно довідки

```
unit About;

interface

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

type
  TForm19 = class(TForm)
    Image1: TImage;
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form19: TForm19;

implementation

{$R *.dfm}

procedure TForm19.Button1Click(Sender: TObject);
begin
  Form19.Close;
end;

end.
```