

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

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

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи вдосконалення**  
**технічних компонентів комп’ютерних систем на основі**  
**альтернативних форм представлення логічних функцій”**

Виконав здобувач вищої освіти  
II курсу, групи КІ-22М-2  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Смутко В.О.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Дреєв О.М.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Рецензент \_\_\_\_\_  
\_\_\_\_\_

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Смутку Віталію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи вдосконалення технічних компонентів комп’ютерних систем на основі альтернативних форм представлення логічних функцій

2. Керівник роботи Дресєв Олександр Миколайович, канд. техн. наук, доцент

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

затверджені наказом вищого навчального закладу № 35-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи вдосконалення технічних компонентів комп’ютерних систем на основі альтернативних форм представлення логічних функцій

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

## 6. Консультанти розділів роботи

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

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

## КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання  
« 6 » вересня 2023 р.

Підпис керівника

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

Підпис здобувача

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

**Смутко В.О. Дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій. 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 вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

Метою розробки є дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

Об'єктом дослідження є процес вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

Предметом дослідження є методи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

Методи дослідження базуються на методах комп'ютерної інженерії, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

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, логічні функції

## ABSTRACT

**Smutko V.O. Research and software implementation of the system of improvement of technical components of computer systems based on alternative forms of representation of logical functions. 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 improving the technical components of computer systems based on alternative forms of representation of logical functions.

The purpose of the development is research and software implementation of a system for improving the technical components of computer systems based on alternative forms of representation of logical functions.

The object of research is the process of improving the technical components of computer systems based on alternative forms of representation of logical functions.

The subject of research is methods of improving technical components of computer systems based on alternative forms of representation of logical functions.

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

The result of the work is the software implementation of the system for improving the technical components of computer systems based on alternative forms of representation of logical functions.

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, logical functions

## 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.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання .....	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	25
3.1 Опис функціонування системи .....	25
3.2 Розробка структурної схеми.....	36
3.3 Розробка функціональної схеми .....	38
3.4 Розробка діаграми процесів.....	41
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	43
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	43
4.2 Захист розробленого програмного забезпечення.....	49
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	53
6 НАУКОВА НОВИЗНА .....	59

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>			
<b>Вим.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	<i>Дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій</i>	<b>Літ.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<i>Розроб.</i>	<i>Смутко В.О.</i>			<i>М</i>		1	97	
<i>Перев.</i>	<i>Дресв О.М.</i>							
<i>Н.контр.</i>	<i>Коваленко А.С.</i>					<i>ЦНТУ КІ-22М-2</i>		
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	60
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	60
7.2 Розрахунок трудомісткості розробки програмної продукції.....	62
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	64
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	69
7.5 Визначення собівартості розробки та ціни програмної продукції.....	73
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	76
7.7 Визначення експлуатаційних витрат.....	76
7.8 Визначення економічної ефективності програмної продукції.....	78
7.9 Висновок.....	80
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	81
8.1 Вступ.....	81
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	82
8.3 Аналіз умов праці на робочому місці програміста.....	83
8.4 Розрахункова частина .....	86
8.5 Висновки до розділу.....	89
9 ОСНОВНІ ВИСНОВКИ.....	90
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	92

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АФП	–	алгебраїчна форма представлення
КФП	–	класична форма представлення
ЛФ	–	логічні функції
ОФП	–	оптимальна форма представлення
ПЛМ	–	програмовані логічні матриці
РМФП	–	Ріда-мюллерівська форма представлення
ФП	–	форми представлення

КБГПЗ-2023

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Широке впровадження обчислювальної техніки та систем управління в усі сфери людської діяльності, а також ускладнення задач, які розв'язуються цими системами, обумовлюють необхідність постійного пошуку шляхів вдосконалення технічних компонентів комп'ютерних систем. Покращення загальної якості відповідних об'єктів, наслідком яких є мініатюризація, зменшення енергоспоживання, збільшення швидкодії можливе на основі прогресу як в області технологій, так і шляхом оптимізації структур інтегральних мікросхем, як основної елементної бази технічних компонентів комп'ютерних систем. Оскільки, на думку більшості фахівців, можливості сучасної мікроелектроніки наближаються до своєї фізичної межі, гостро постає проблема пошуку потенційних шляхів вдосконалення технічних компонентів комп'ютерних систем саме за рахунок структурних змін мікросхем.

Структурна реалізація суттєво залежить від аналітичного представлення логічних функцій, як основних математичних моделей пристроїв сучасної техніки. Сукупність варіантів такого представлення логічної функції обумовлена, в першу чергу, формою представлення логічних функцій.

Внаслідок цього виникає можливість проведення структурної оптимізації технічних компонентів на основі застосування альтернативних форм представлення логічних функцій. За останнє десятиріччя дослідження з теорії логічних функцій, що пов'язані з представленням функцій різними способами, достатньо інтенсивно розвиваються.

Науковцями обрано шлях нетрадиційних підходів до представлення та перетворення логічних функцій, до яких, зокрема, належать: концепція багатозначного алфавіту, кінечних предикатів, логічних шкал, інтерполяційних поліномів, спектрального представлення, концепція оптимальної форми представлення. Зазначені методи продемонстрували позитивний ефект в задачах аналізу та синтезу дискретних пристроїв та довели достатньо потужний потенціал

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

нових підходів до представлення логічних функцій в аспекті зменшення їх параметрів складності структурної реалізації.

Проте залишається невирішеним досить широкий спектр важливих задач, що пов'язані із застосуванням альтернативних представлень логічних функцій в аспекті покращення якості систем обробки, зберігання та перетворення інформації.

Отже, тема магістерського дослідження є достатньо важливою та актуальною.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

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

– Огляд існуючих систем вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

– Дослідження системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

– Програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

*Об'єктом дослідження* є процес вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

*Предметом дослідження* є методи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

*Методи дослідження* базуються на методах комп'ютерної інженерії, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

– Розроблено вітчизняний продукт вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

**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 вищеперахованого, дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Система призначена для вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій. Під час проведення досліджень були використані наступні різні форми представлення логічних функцій:

- АФП – алгебраїчна форма представлення.
- КФП – класична форма представлення.
- ЛФ – логічні функції.
- МОФ – мінімізація логічних функцій в ортогональній формі представлення з урахуванням однакових фрагментів.
- ОРФП – ортогональна форма представлення.
- ОФП – оптимальна форма представлення.
- ПЛМ – програмовані логічні матриці.
- РМФП – Ріда-мюллерівська форма представлення.
- ФП – форми представлення.

## 1.2 Область застосування

Областю застосування є вдосконалення технічних компонентів комп'ютерних систем. Комп'ютерна система – це інформаційно-технічний комплекс, до складу якого входить обчислювальна техніка (комп'ютери, сервери тощо) із програмним забезпеченням, призначена для обробки, зберігання та введення–виводу інформації.

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Обмін інформацією в комп'ютерній системі відбувається через комп'ютерну мережу за допомогою системи передачі даних (локальної та/або глобальної).

Перевагою комп'ютерних систем є можливість розпаралелювати обчислення, підвищувати продуктивність і відмовостійкість, що у свою чергу дозволяє вирішувати прикладні завдання в предметних галузях діяльності (автоматизація процесів, технологічна підготовка, облік, керування тощо).

Уніфікація й різноманіття комп'ютерного устаткування дає можливість інтегрувати комп'ютерні системи з іншими інженерними технологіями, тим самим розширити можливості й створити єдине середовище для керування.

Операції з комп'ютерними системами, як правило включають наступне:

- Консалтинг і аудит комп'ютерних систем (обстеження, консультації).
- Проектування комп'ютерних систем (розробка проекту, узгодження проекту).
- Будівництво комп'ютерних систем (монтаж пасивного устаткування, установлення активного устаткування).
- Пусконаладження комп'ютерних систем (настроювання устаткування, запуск в експлуатацію).
- Сертифікація комп'ютерних систем (оцінка якості, тестування).
- Відновлення комп'ютерних систем (діагностика, ремонт).
- Комплексне обслуговування комп'ютерних систем.
- Модернізація й оптимізація комп'ютерних систем.
- Перенесення комп'ютерних систем і компонентів.
- Автоматизація комп'ютерних систем.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>8</b>

## 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

#### Proteus VSM

Наймогутніша система автоматизованого проектування, що дозволяє віртуально змоделювати роботу величезної кількості аналогових і цифрових пристроїв.

Програмний пакет Proteus VSM дозволяє зібрати схему будь-якого електронного пристрою й симулювати його роботу, виявляючи помилки, допущені на стадії проектування й трасування. Програма складається із двох модулів. ISIS – редактор електронних схем з наступною імітацією їхньої роботи. ARES – редактор друкованих плат, оснащений автотрасувальником Electra, убудованим редактором бібліотек і автоматичною системою розміщення компонентів на платі. Крім цього ARES може створити тривимірну модель друкованої плати.

Proteus VSM містить у собі більше 6000 електронних компонентів з усіма довідковими даними, а також демонстраційні ознайомлювальні проекти. Програма має інструменти USBCONN і COMPIM, які дозволяють підключити віртуальний пристрій до портів USB і COM комп'ютера. При приєднанні до цих портів будь-якого зовнішнього приладу віртуальна схема буде працювати з ним, як якби вона існувала в реальності. Proteus VSM підтримують наступні компілятори: CodeVisionAVR і WinAVR (AVR), ICC (AVR, ARM7, Motorola), HiTECH (8051, PIC Microchip) і Keil (8051, ARM). Існує можливість експорту моделей електронних компонентів із програми PSpice.

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

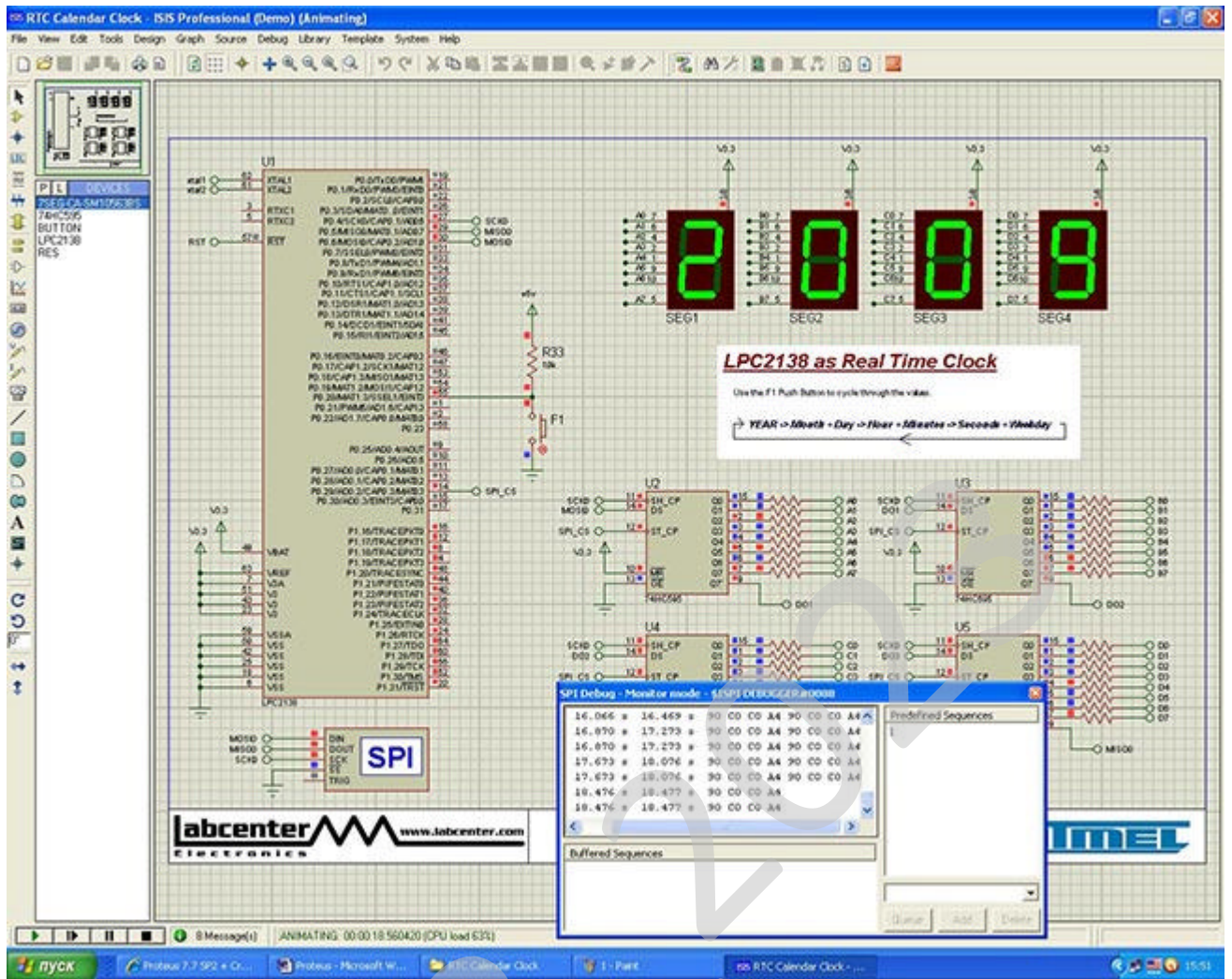


Рисунок 2.1 – Інтерфейс користувача Proteus VSM

Незважаючи на те, що програма працює із пристроями, що складаються з декількох мікроконтролерів і навіть із чипами від різних виробників в одному пристрої, необхідно чітко розуміти, що симуляція повторює роботу реальної схеми не абсолютно точно! Щоб уникнути помилок, потрібно ясно представляти кінцевий результат.

Proteus VSM є комерційним продуктом. Є безкоштовна демонстраційна версія. Вона має всі функції й можливостями платного пакета, але не дозволяє зберегти або роздрукувати результат роботи, створити свій власний мікроконтролер.

Меню програми англomовне. Повного українзатора для Proteus VSM немає. Установлювати програму необхідно в папку без кириличних символів у назві.

Операційні системи, у яких працює дана САПР – це Windows 2000 / XP / Vista / 7. Зверніть увагу, що Proteus VSM працює в Windows 7, тільки починаючи з версії 7.8.

### **Micro-Cap**

Професійна програма аналогового, цифрового й змішаного моделювання й аналізу ланцюгів електронних пристроїв середнього ступеня складності.

Інтуїтивно зрозумілий інтерфейс, невимогливість до обчислювальних ресурсів персонального комп'ютера й великий спектр можливостей послужили основою популярності Micro-Cap серед радіоаматорів, студентів і викладачів мікроелектроніки. Алгоритм роботи містить у собі створення електричного кола в графічному редакторі, завдання параметрів аналізу й вивчення отриманих даних. Програма самостійно становить рівняння ланцюга й проводить моментальний розрахунок. Будь-яка зміна схеми або параметрів елементів приводить до автоматичного відновлення результатів. Графічний редактор опирається на бібліотеки електронних компонентів, які можна поповнювати на основі експериментальних або довідкових даних за допомогою убудованого модуля Shape Editor. Всі номінали й параметри елементів можуть бути як незмінними, так і залежними від температури, часу, частоти, стану схеми, параметрів інших компонентів. Анімовані деталі (світлодіоди, реле, семисегментні індикатори й деякі інші елементи) змінюють стан відповідно до вступників на них сигналами. Моделювання містить у собі цілий набір різних аналізів: перехідних процесів, передатних характеристик по постійному струмі, малосигнальних частотних характеристик, чутливостей по постійному струмі, нелінійних перекручувань, методу Монте-Карло й багатьох інших. Досвідчені користувачі можуть створювати свої макромоделі, які полегшують імітаційне моделювання без втрат інформації. Допускається одночасно використовувати різні стандарти елементів

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

схеми. Повна підтримка SPICE-моделей дозволяє застосовувати проекти з інших програм (DesignLab, OrCAD, P-CAD). З недоліків можна відзначити лише необхідність установки додаткових елементів, тому що обсяг бібліотек Micro-Cap (навіть у повній версії) явно недостатній.

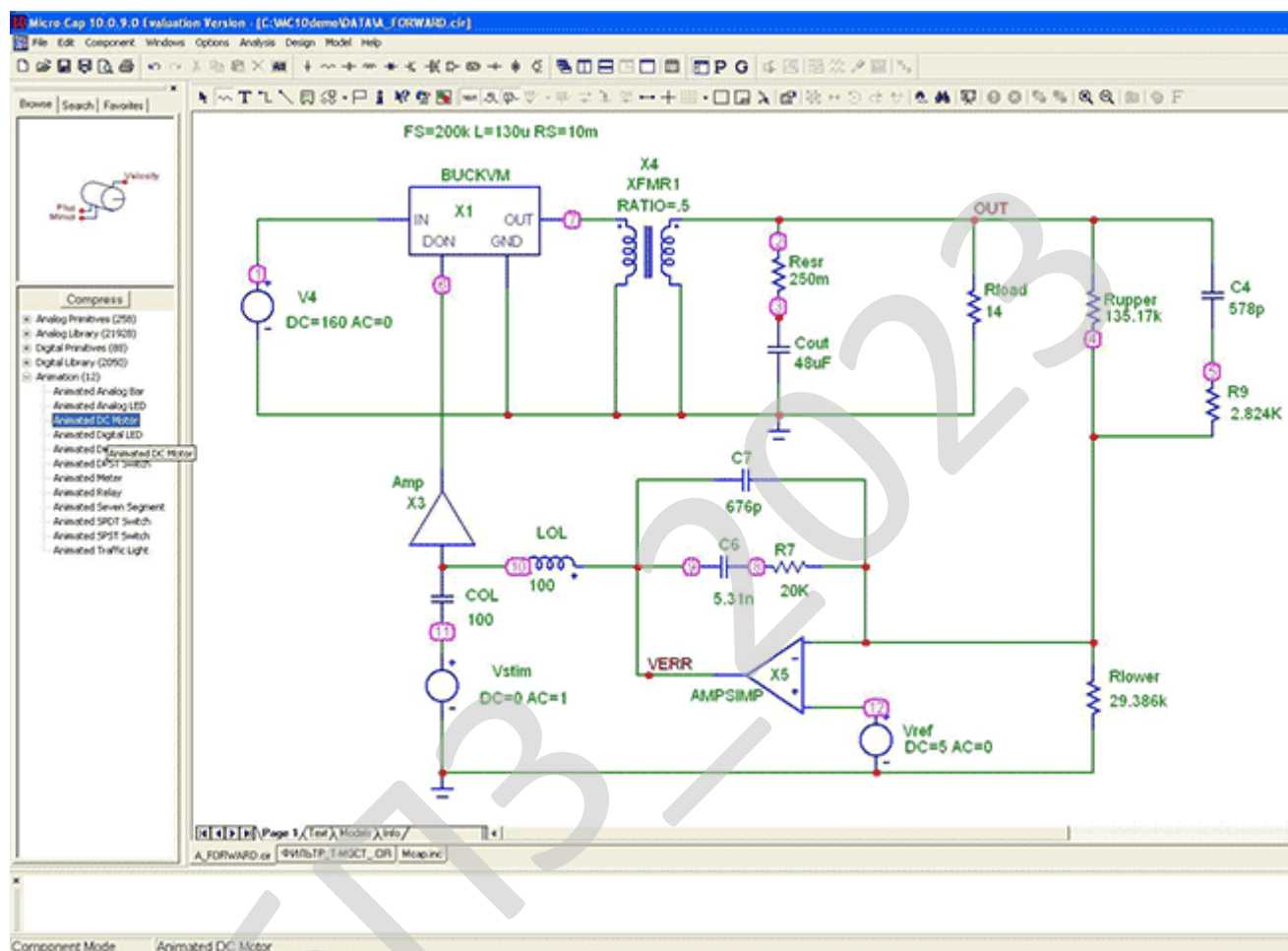


Рисунок 2.2 – Інтерфейс користувача Micro-Cap

Micro-Cap – англомовний пакет, але бажаючі можуть легко відшукати в мережі українізатор програми від відомих фахівців з роботи з ним – Сергія й Марини Амеліних. У деяких користувачів після українізації в середовищі моделювання виникають проблеми з відображенням кирилических шрифтів.

Програма була написана в 1982 році фірмою Spectrum Software, з тих пор вона постійно розширюється й удосконалюється. Фірма, у свою чергу, була

заснована Енді Томпсоном у лютому 1980 року, споконвічно позиціонуючись на написанні програм для Apple. Вона розташована в одному з основних міст Силіконової долини – Саннивейлі (штат Каліфорнія, США).

Вартість Micro-Cap становить кілька тисяч доларів, однак на сайті розроблювача можна скачати вільно розповсюджену Evaluation Version, що має багато можливостей повнофункціональної. Основні відмінності – це не більше 50 елементів у схемі, урізана бібліотека компонентів, обмеження на побудову ряду графіків і повільна швидкість роботи.

Micro-Cap працездатний у всіх операційних системах сімейства Windows. Проблем при роботі в 10/11 виявлено не було.

### **NI Multisim**

Популярний програмний пакет, що дозволяє моделювати електронні схеми й розводити друковані плати.

Головна особливість NI Multisim – простий наочний інтерфейс, потужні засоби графічного аналізу результатів моделювання, наявність віртуальних вимірювальних приладів, що копіюють реальні аналоги. Бібліотека елементів містить більше 2000 SPICE-моделей компонентів National Semiconductor, Analog Devices, Phillips, NXP і інших виробників. Присутні електромеханічні моделі, імпульсні джерела живлення, перетворювачі потужності. Інструмент Convergence Assistant автоматично виправляє параметри SPICE, коректуючи помилки моделювання. NI Multisim випускається у двох варіантах – Professional і Education.

Версія Multisim Education призначена для навчальних закладів і містить у собі навчальні курси, підготовлені апаратні рішення й робочі підручники. Основне завдання – закріпити теоретичний матеріал, наочно продемонструвавши роботу тих або інших законів і процесів у реальних проектах. Для цього крім інтерактивних компонентів програма здатна взаємодіяти з апаратними платформами NI myDAQ (бібліотека контрольно-вимірювального встаткування) і NI ELVIS (віртуальний інструментарій для навчальної майстерні), що

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

УМОЖЛИВЛЮЄ створення цілих віртуальні лабораторії систем керування, енергетики, мехатроніки й силової техніки.

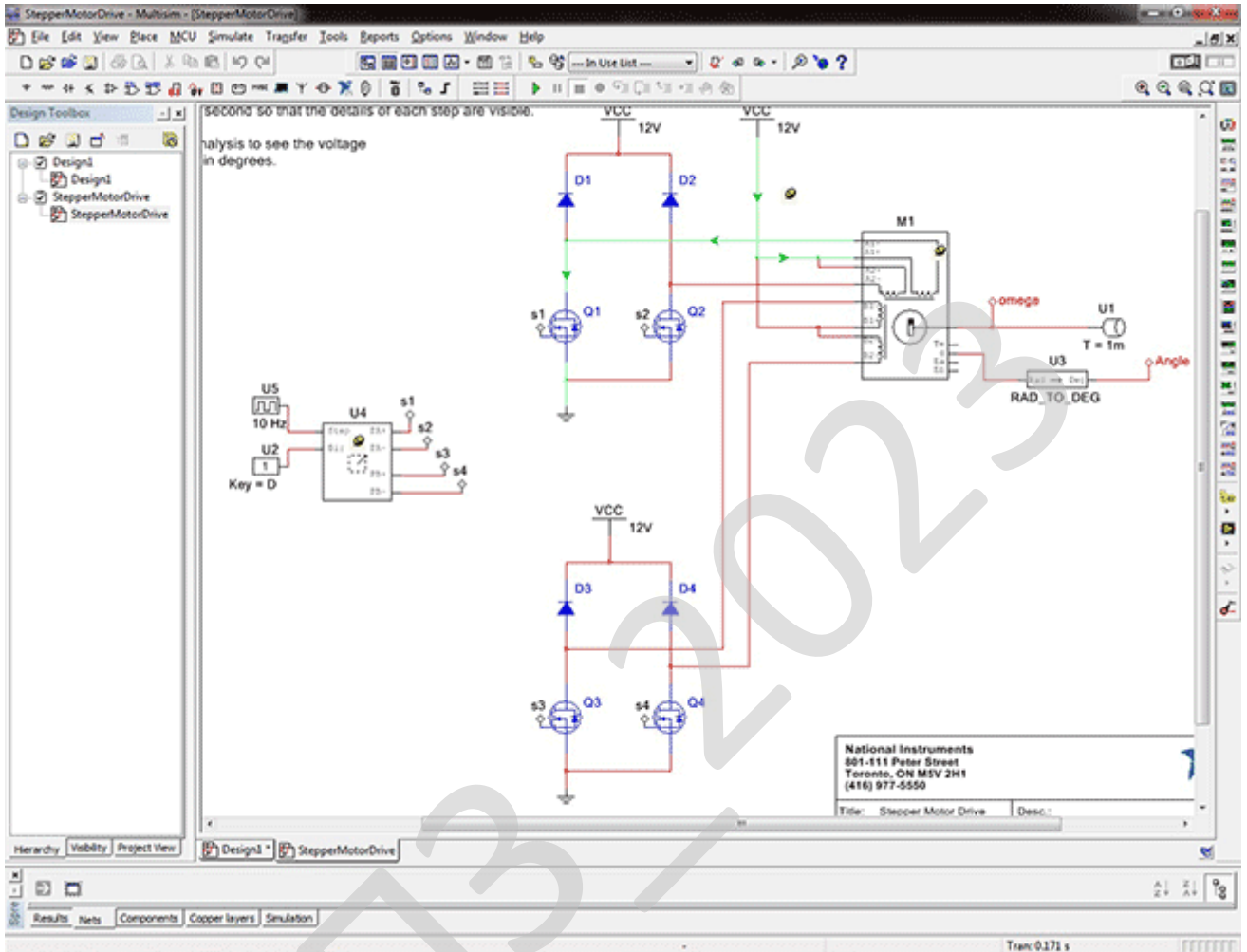


Рисунок 2.3 – Інтерфейс користувача NI Multisim

Версія Multisim Professional спеціально створена для швидкого прототипування й рішення завдань оптимізації з'єднань. Пропонується розширений користувальницький інтерфейс, нестандартні методи аналізу, засновані на фірмовій системі NI LabVIEW, і звичайні алгоритми імітаційного моделювання схем по стандарті SPICE.

Останні версії програми мають поліпшену функціональність, новими інструментами для моделювання, розширеною базою елементів, завдяки чому розробка й створення проектів електричних схем може виконуватися набагато

більш точно й швидко. NI Multisim може взаємодіяти із середовищем розробки систем виміру LabVIEW, що дозволяє зіставляти теоретичні дані з реальними, прямо в ході створення схем друкованих плат. Це зменшує кількість проектних помилок і прискорює реалізацію проектів. Зворотною стороною цього стали завищені системні вимоги, пропоновані до встаткування. Навантаження на процесор і пам'ять при роботі з більшими схемами й при трасуванні дуже велике.

Перші версії програми мали назва Electronics Workbench і розроблялися однойменною фірмою. У цей час Electronics Workbench є дочірньою компанією, що належить National Instruments. Штаб-квартира NI розташован у місті Остин (Техас, США).

Для полегшення процесу створення друкованих плат компанія дає можливість кожному розроблювачеві вступити в онлайн-співтовариство NI Circuit Design Community для того, щоб обмінюватися своїми роботами, прототипами, шаблонами, обговорювати нюанси розробок і одержувати нові знання від колег і однодумців, що живуть по усьому світі.

Мова інтерфейсу тільки англійська, але існують саморобні варіанти українізації пакета.

### **LabVIEW**

Графічне середовище для створення програм у системах збору, аналізу, виміру, візуалізації й обробки даних, а також для керування й автоматизації технічних об'єктів і технологічних процесів.

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) має унікальний графічний інтерфейс і програмування, що істотно відрізняється від роботи на Java або C. Створення додатків являє собою процес утворення блок-діаграми із графічних образів (іконок), що дозволяє сконцентрувати всю свою увагу тільки на роботі з потоком даних. Будь-яка програма є віртуальним приладом, що має «лицьову панель» (всі засоби введення-виводу для керування приладом: перемикачі, кнопки, світлодіоди, інформаційні табло, лампочки, графіки, текстові поля та інше) і «блок-схему» (логіка роботи програми). Всі

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

частини програми з'єднані між собою нитками, по яких відбувається передача даних. Кожний віртуальний прилад може містити в собі інші віртуальні прилади. Система, створена в LabVIEW, набагато перевершує будь-який реально існуючий лабораторний інструмент, дозволяючи самостійно визначати потрібні функції створюваного апарата. При необхідності, зміни можна внести всього за пару хвилин.

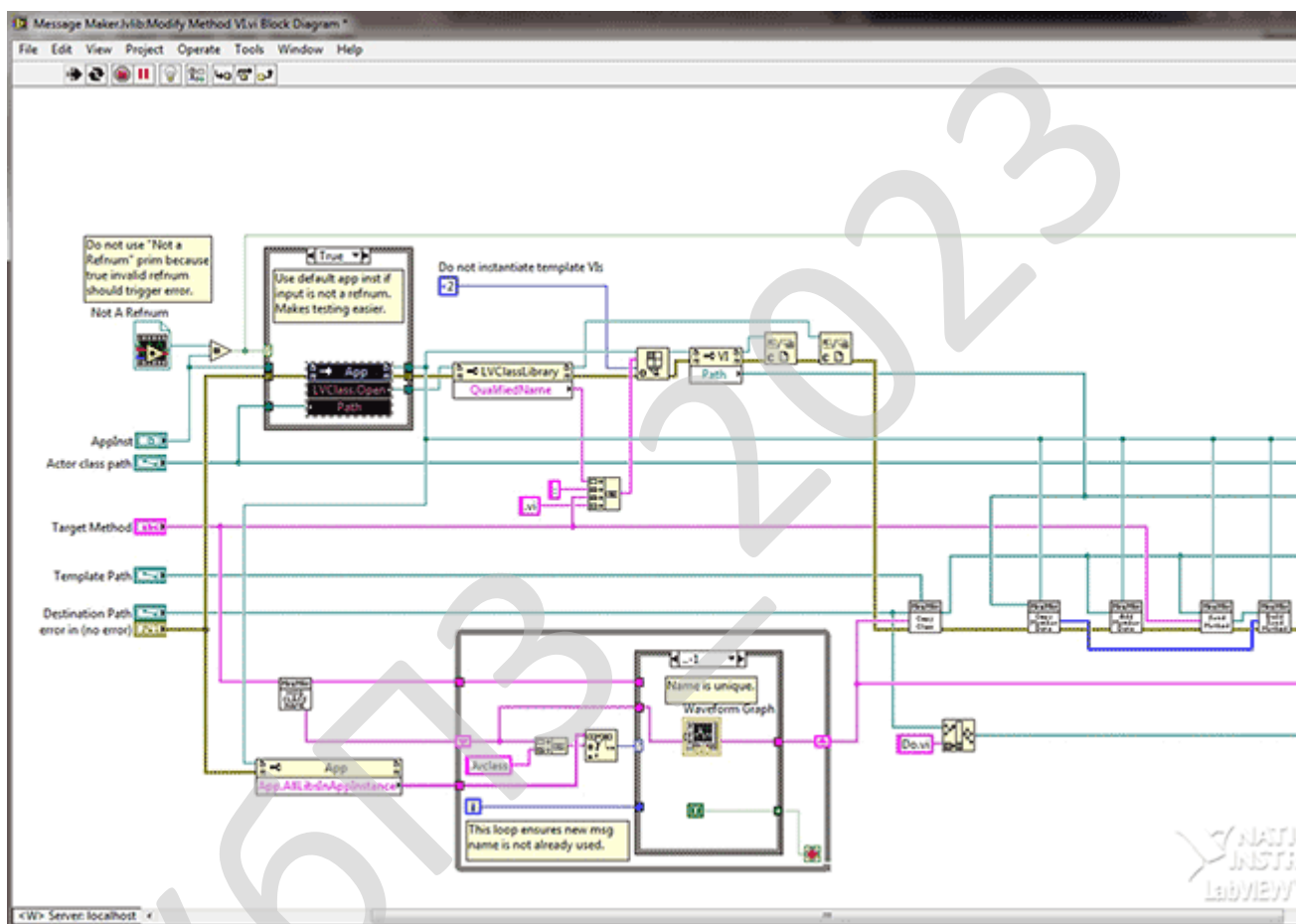


Рисунок 2.4 – Інтерфейс користувача LabVIEW

Програмний пакет підтримує величезна кількість устаткування від різних виробників і містить у собі численні бібліотеки компонентів, а також розвинені засоби зв'язку для вилученої взаємодії з об'єктом. LabVIEW має власну математичну підтримку й може інтегрувати програми, створені в середовищі MATLAB. У сполученні з апаратними засобами можливе створення систем

практично будь-якої складності для рішення абсолютно різних завдань. LabVIEW застосовується для керування (у тому числі й вилученого) різним устаткуванням (пристроями збору даних, датчиками, пристроями спостереження, руховими пристроями, роботами), збору даних, тестувань і вимірів, візуалізації результатів, моделювання процесів, зберігання інформації й генерації звітів. Оскільки LabVIEW ще й справжній 32-бітний компілятор, то він здатний створювати бібліотеки функцій і незалежні файли, які виконуються.

LabVIEW була створена для зменшення часу програмування будь-яких вимірювальних приладів. Але надалі сфера застосовності програми охопила електротехніку, механіку, фізику, біологію, психологію, хімію, утворення й безліч інших галузей науки. LabVIEW використовують передові наукові центри CERN (Європа), Livermore, Batelle, Lawrence (США), найбільші військові й промислові об'єкти.

Інтерфейс LabVIEW, як і убудована інтерактивна навчальна система, контекстнозалежна допомога й безліч прикладів по використанню прийомів програмування, виконані на англійському, китайському, німецькому, але не на російських мовах.

Перша версія LabVIEW була написана в 1986 році компанією National Instruments (Техас, США, [www.ni.com](http://www.ni.com)), продукцію якої купують більше 25 тисяч організації, розташованих в 90 країнах миру. Останні десять років National Instruments входить до числа 100 кращих роботодавців США.

LabVIEW є пропріетарним програмним забезпеченням, має закритий вихідний код і вимагає активації. В ознайомлювальних цілях вільний доступ до програми надається лише на кілька днів.

Споконвічно LabVIEW випускалася для Apple Macintosh, але сьогодні існують версії для MacOS, HP-UX, Linux, Solaris і, звичайно, Microsoft Windows. Додатка, написані в LabVIEW, легко портуються на інші платформи.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17



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

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

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

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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

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

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

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

### **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 звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

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

### **Єдине керування пам'яттю**

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

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

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

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

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

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

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

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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

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

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

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

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

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

## Поліпшена кроссплатформеність

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

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.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

## 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

З метою підвищення ефективності концепції оптимальної форми представлення (ОФП) запропонуємо математичну модель статистичної оцінки кількості втрат від неоптимальності форми представлення (ФП) логічної функції (ЛФ) на основі аналізу інтегральних оцінок найбільш суттєвих для розробників інтегральних мікросхем показників структурної складності реалізації ЛФ для повних множин логічних функцій  $L(n)$  при кількості аргументів ЛФ  $n \in (3,5)$ . В результаті проведених обчислювальних експериментів отримано важливий практичний висновок про те, що виключне використання класичної форми представлення (КФП) приводить до технічно необґрунтованих суттєвих втрат в площі мікросхеми і в абсолютних, і у відносних розмірах. Зокрема, за параметром структурної складності реалізації габаритної площі  $S_S$  в середньому вони становлять близько 40%.

Запропонуємо метод дослідження структур повних множин  $L(n)$  ЛФ великих розмірностей, необхідність якого обумовлена складністю аналізу окремих елементів множини  $L(n)$  внаслідок експоненціального зростання потужності  $L(n)$  при збільшенні кількості аргументів  $n$  ЛФ, на основі теореми Муавра-Лапласа.

На основі розробленого методу для дослідження структур  $L(n)$  аналізуються спеціально сформовані статистично достовірні вибірки, які забезпечують достатньо прийнятні рівні похибок при визначенні статистичних показників  $L(n)$  при досить великих значеннях ймовірності. Пропонується, зокрема, аналізувати структуру множини  $L(5)$  на основі вибірки для статистично обґрунтованої кількості її представників в обсязі 65536 логічних функцій, яка забезпечує достовірність висновків на рівні не менше 95%. Отримана

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

інструментальна база даних *MINFORM\_5* відіграє велике значення у систематизації інформації по  $L(5)$  та дозволяє дослідити подальшу диференціацію логічних функцій на підмножини пріоритетів, виявити нові та підтвердити вже виявлені тенденції.

В рамках розширення концепції ОФП запропонуємо варіант бінарних форм представлення на основі паралельного застосування двох базових форм для реалізації логічних функцій:

– класичної форми представлення (КФП) та алгебраїчної форми представлення (АФП),

або:

– класичної форми представлення (КФП) та Ріда-мюллерівської форми представлення (РМФП).

Об'єднання зазначених ФП обумовлюється урахуванням схемотехнічної складності та мінімальністю необхідних термів для представлення ЛФ і сприяє спрощенню структури програмованих логічних матриць в частині додавання кон'юнкцій.

Найбільш перспективними з точки зору структурної оптимізації мікросхем вважаються модифікації бінарних форм, що полягають у застосуванні КФП ЛФ без інвертування вхідних аргументів. Це дозволяє ліквідувати основний недолік КФП – необхідність подвоєння вхідних шин, оскільки аргументи в КФП можуть подаватися як в прямому, так і в інверсному станах.

Розроблено два нових методи мінімізації ЛФ в АФП на основі задачі лінійного програмування (ЗЛП). В першому методі в систему обмежень ЗЛП вводяться додаткові спеціальні обмеження для невідомих системи. Праві частини системи являють собою значення ЛФ на відповідних наборах вхідних аргументів, в якості невідомих виступають коефіцієнти ЛФ в АФП, а сама система обмежень являє собою систему нерівностей. Оцінка ефективності пропонованого методу визначається як інтегральний показник кількості доданків ЛФ множини  $L(n)$

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26



розроблена ортогональна форма представлення ЛФ (ОРФП). Для даної ФП характерною є декомпозиція вхідних аргументів ЛФ на інформативну  $X_Q$  та базисну  $X_\Phi$  підмножини з потужностями  $k$  та  $n-k$  відповідно. Аргументи  $X_\Phi$  утворюють ортогональні базисні функції, серед яких лише одна приймає значення, відмінне від нульового на будь-якому наборі вхідних аргументів. Аналітична форма представлення логічної функції в ОРФП має вигляд

$$f(x_1, x_2, \dots, x_n) = \bigvee_{i=0}^{2^{n-k}-1} Q_i \Phi_i, \quad (3.2)$$

де  $k$  – кількість інформативних аргументів підмножини  $X_Q$ ;  $Q_i$  – інформативні функції, утворені аргументами  $X_Q$ ;  $\Phi_i$  – базисні функції, утворені аргументами  $X_\Phi$  потужністю  $n-k$ .

Інформативні функції виступають в ролі вагових коефіцієнтів для базисних функцій, на які накладається вимога взаємної ортогональності. Відзначимо позитивні риси ОРФП:

– По-перше, ОРФП здатна повністю ліквідувати існуюче ускладнення щодо практичного впровадження концепції ОФП, оскільки в частині додавання кон'юнкцій програмованої логічної матриці (ПЛМ) достатньо простого логічного елемента типу *OR*.

– По-друге, ОРФП є багатоваріантною формою, що визначається кількістю інформативних аргументів та розподілом вхідних змінних на підмножини  $X_Q$  та  $X_\Phi$ . Частковий випадок ОРФП при  $k=0$  відповідає загальновідомій класичній формі представлення.

Структуру множин  $L(n)$  з урахуванням появи нової ОРФП ЛФ демонструє діаграма Венна на рисунку 3.1. Межі підмножин пріоритетів  $L(n)$  обумовлені обраними критеріями складності реалізації ЛФ.

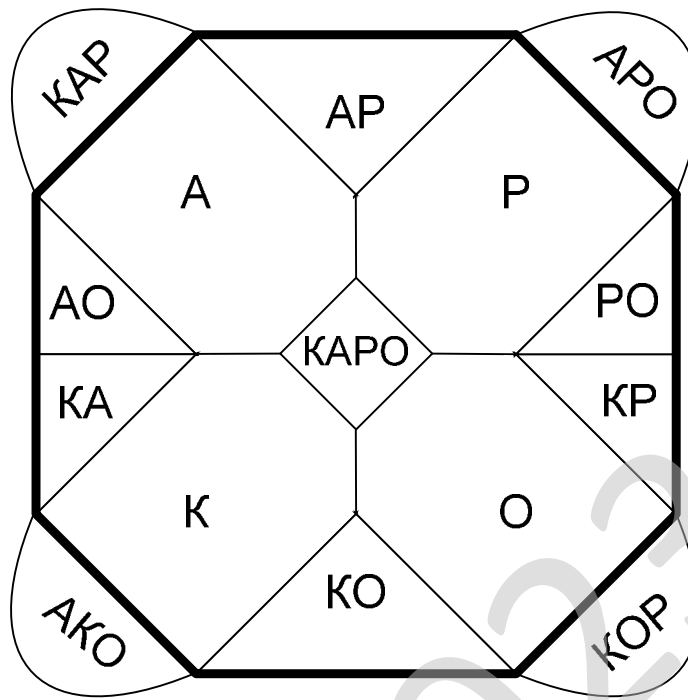


Рисунок 3.1 – Структура  $L(n)$  за підмножинами пріоритетів

На рисунку 3.1:

- $K, A, P, O$  – так звані «чисті» підмножини пріоритетів ЛФ, для реалізації яких найбільш доцільна одна з ФП: КФП, АФП, РМФП або ОРФП відповідно;
- $KA, KP, KO, AP, AO, PO, KAP, KAO, KPO, APO, KAPO$  – так звані «проміжні» підмножини пріоритетів ЛФ, для реалізації яких однаково доцільні декілька ФП.

Експериментально визначимо питомий склад підмножин пріоритетів  $L(n)$  при  $n = \overline{3,5}$  для нової диференціації повних множин  $L(n)$  з урахуванням ОРФП ЛФ за найбільш суттєвим показником структурної складності реалізації  $S_S$ . Результати відповідних досліджень представлені у табл. 3.1-3.2. Вищезгаданий показник для ОРФП ЛФ визначається, виходячи з наявності двох конструктивних частин ПЛМ – БФК-1 та БФК-2 (рисунок 3.2), що формують відповідно інформативні та базисні функції:

$$S_s(OP\Phi\Pi) = S_s(Q) + S_s(\Phi), \quad ($$

3.3)

де  $S_s(Q)$  – площа ПЛМ для реалізації інформативних функцій  $Q_i$  (частина БФК-1);  $S_s(\Phi)$  – площа ПЛМ для реалізації базисних функцій  $\Phi_i$  (частина БФК-2).

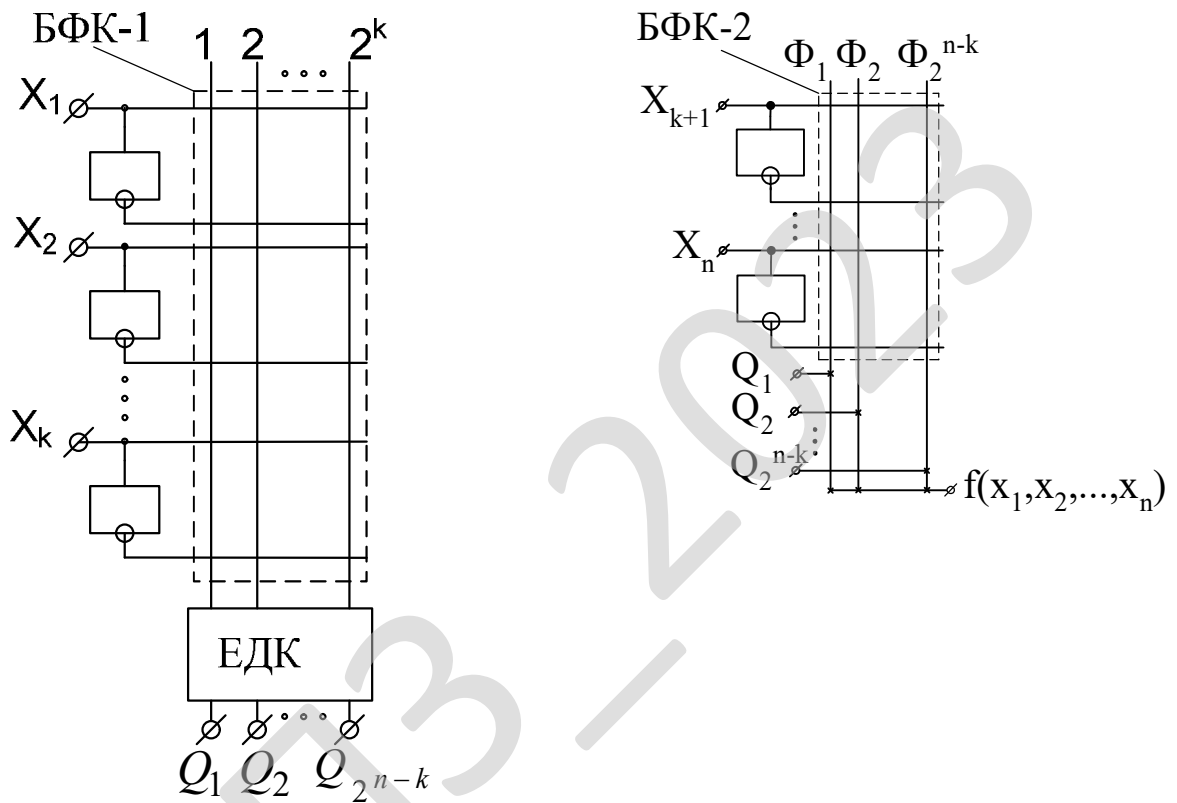


Рисунок 3.2 – Типова схема реалізації ЛФ у вигляді ОРФП

Таблиця 3.1 – Склад  $L(n)$  для «чистих» підмножин пріоритетів ЛФ за показником  $S_s$

$n$	Відносний вміст чистих підмножин пріоритетів, %				
	К	А	Р	О	Всього
$L(3)$	–	18,7	25	2,3	46
$L(4)$	–	31	37,4	2	70,4
$L(5)$	0,78	8,4	68	11,5	88,7

Таблиця 3.2 – Склад  $L(n)$  для «проміжних» підмножин пріоритетів ЛФ за показником  $S_s$

	Відносний вміст проміжних підмножин пріоритетів, %											
	КА	КР	КО	АР	АО	РО	КАР	КАО	КРО	АРО	КАРО	Всього
$L(3)$	–	–	–	51,6	–	–	0,4	–	–	–	2	54
$L(4)$	–	–	–	24,8	0,73	0,83	–	0,11	–	2,81	0,46	29,7
$L(5)$	0,14	0,61	0,6	4,8	0,42	3	0,13	0,13	0,76	0,48	0,17	11,3

Визначимо сумарні потужності підмножин, пов'язаних з відповідними ФП – *CLASSIC (C)*, *ALGEBRAIC (A)*, *RID-MULLER (R)* та *ORTOGONAL (O)*, а також встановлено, що в загальній диференціації  $L(n)$  сумарна потужність *ORTOGONAL=OUKOUAOUPOUKAOUKPOUAPOUKARO* з ростом кількості аргументів  $n$  ЛФ зростає, як наведено в табл.3.3 та на рисунку 3.3-3.5.

Таблиця 3.3 – Сумарні потужності підмножин в структурі  $L(n)$

Підмножини	$L(3)$	$L(4)$	$L(5)$
<i>CLASSIC (C)</i>	2,4	0,6	3,34
<i>ALGEBRAIC (A)</i>	73,1	60,1	14,84
<i>RID-MULLER (R)</i>	79,4	66,5	78,2
<i>ORTOGONAL (O)</i>	4,3	6,9	17,1

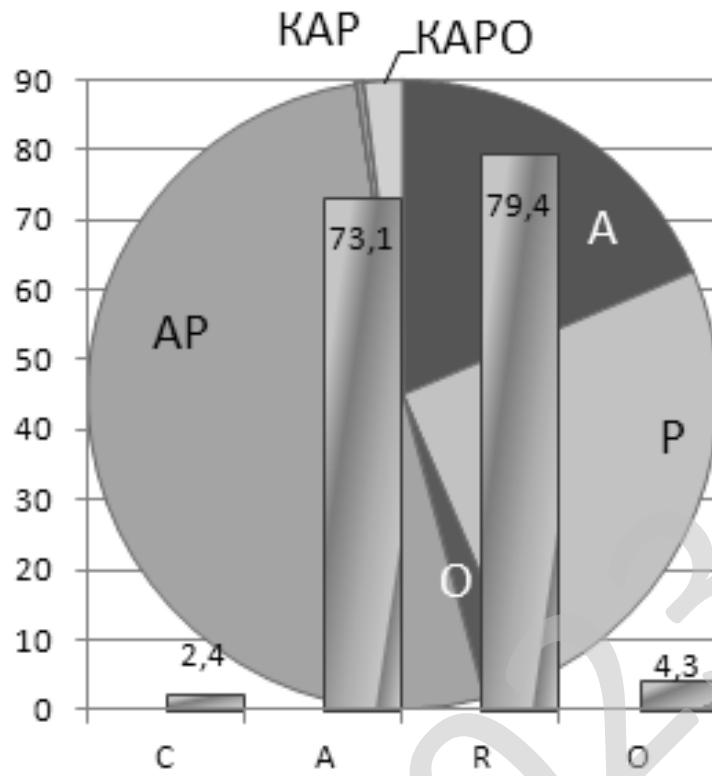


Рисунок 3.3 – Вміст сумарних підмножин  $L(3)$

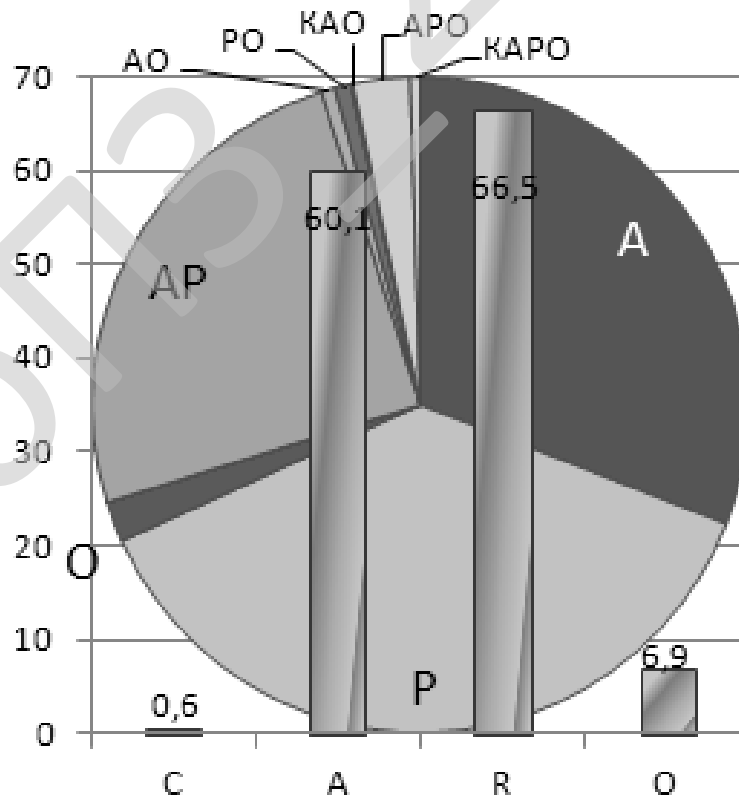


Рисунок 3.4 – Вміст сумарних підмножин  $L(4)$

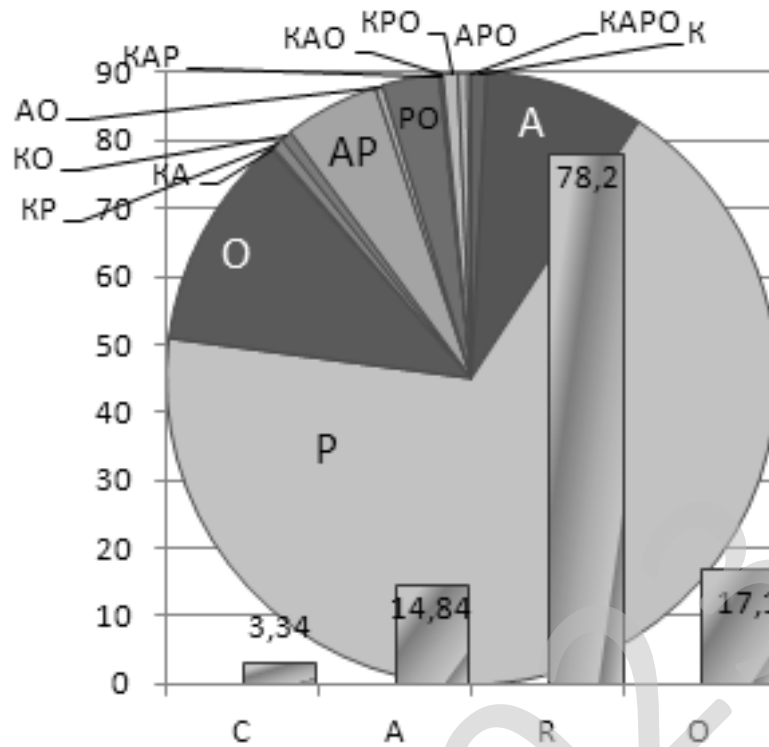


Рисунок 3.5 – Вміст сумарних підмножин  $L(5)$

В табл. 3.4 наведено результати обчислювальних експериментів для кількісного дослідження ефективності ОРФП у порівнянні з класичною формою представлення.

Таблиця 3.4 – Потужності підмножин пріоритетів для КФП, ОРФП та їх перетину

$L(n)$	Абсолютний та відносний вміст підмножин		
	КФП	ОРФП	КОРФП
$L(3)$	5 (2%)	172 (67%)	79 (31%)
$L(4)$	1368 (2,1%)	62963 (96%)	1204 (1,9%)
$L(5)$	6258 (9,5%)	55313 (84,3%)	3966 (6,0%)

Проведемо порівняння ефективності ОРФП з усіма базовими ФП на основі відносного показника ефективності (ВПЕ) ФП (табл. 3.5), що відображає

залежність кількості ЛФ для конкретної ФП від необхідної величини деякого критерія структурної складності реалізації, зокрема,  $S_S$  та визначається як:

$$\eta_i(n) = \frac{\int_0^{S_{\max}(n)} N_j(S) ds}{N_{\max}(n) \cdot S_{\max}(n)}, \quad (3.4)$$

де  $N_j(S)$  – кількість ЛФ при заданому значенні відповідного критерію;  
 $N_{\max}(n)$  – повна кількість ЛФ заданого числа аргументів  $n$ ;  
 $S_{\max}(n)$  – максимальне значення обраного критерію для всіх ФП, що забезпечує реалізацію всіх ЛФ.

Відзначимо суттєве покращення показників структурної складності реалізації ОРФП ЛФ у випадку застосування альтернативних форм представлення в частині формування інформативних функцій – БФК1. Це продемонстровано на прикладі логічної функції з номером 23312. Зокрема, при застосуванні альтернативних ФП значення  $S_S$  становить 20, без застосування – 32.

Таблиця 3.5 – Порівняння ВПЕ для всіх ФП за критерієм  $S_S$

	КФП	ОРФП	АФП	РМФП
L(3)	0,53	0,62	0,63	0,76
L(4)	0,51	0,61	0,67	0,68
L(5)	0,54	0,59	0,61	0,64

Дослідження ефективності ОРФП дозволяють оцінити її як перспективну форму, а також окреслити ряд специфічних питань. В рамках багатоваріантності ортогональної ФП визначимо вплив на її ефективність кількості аргументів підмножини  $X_\phi$ , згідно якому відбувається зниження ефективності ОРФП при збільшенні кількості базисних аргументів, що підтверджується зростанням значень інтегральних показників структурної складності  $S_S$  для  $L(3)$ ,  $L(4)$  та  $L(5)$ , встановлених в результаті обчислювальних експериментів.



	$\Phi_0$	$\Phi_1$	...	$\Phi_{2^{n-k}-1}$
$X^{(0)}_{Q_i}$	$f_{Q_0}^{(0)}$	$f_{Q_1}^{(0)}$	...	$f_{Q_{2^{n-k}-1}}^{(0)}$
$X^{(1)}_{Q_i}$	$f_{Q_0}^{(1)}$	$f_{Q_1}^{(1)}$	...	$f_{Q_{2^{n-k}-1}}^{(1)}$
...	...	...	...	...
$X^{(2^k-1)}_{Q_i}$	$f_{Q_0}^{(2^k-1)}$	$f_{Q_1}^{(2^k-1)}$	...	$f_{Q_{2^{n-k}-1}}^{(2^k-1)}$

Рисунок 3.6 – Інформативна Q-матриця

### 3.2 Розробка структурної схеми

Проведемо аналіз сучасних шляхів удосконалення технічних компонентів комп'ютерних систем, виділено основний напрямок вирішення цієї задачі на основі структурних змін мікросхем, як основної елементної бази технічних компонентів комп'ютерних систем.

Структурна реалізація цифрових блоків в значній мірі обумовлена аналітичними представленнями логічних функцій (ЛФ), як основних математичних моделей дискретних пристроїв. В свою чергу аналітичні представлення ЛФ залежать від форми представлення (ФП) ЛФ. В результаті для вдосконалення технічних компонентів комп'ютерних систем обрано структурну оптимізацію схем на основі використання альтернативних ФП ЛФ. Альтернативними вважаються такі способи представлення логічних функцій, що є відмінними від традиційного класичного.

Покажемо, що класична форма (КФП) не є єдиною можливою формою представлення ЛФ, та наведено особливості реалізації логічних функцій в відомих альтернативних формах представлення – алгебраїчній (АФП) та Ріда-мюллерівській (РМФП). Розглядається поширений варіант реалізації ЛФ в КФП –

програмовані логічні матриці (ПЛМ). Окреслено параметри структурної складності реалізації ЛФ.

Продемонструємо ефективність альтернативних форм представлення на основі експериментально виявленої в роботах попередників диференціації повної структури логічних функцій на так звані підмножини пріоритетів, елементами яких є ЛФ, для яких найбільш доцільною з точки зору параметрів структурної складності реалізації є та чи інша ФП або, можливо, їх комбінація.

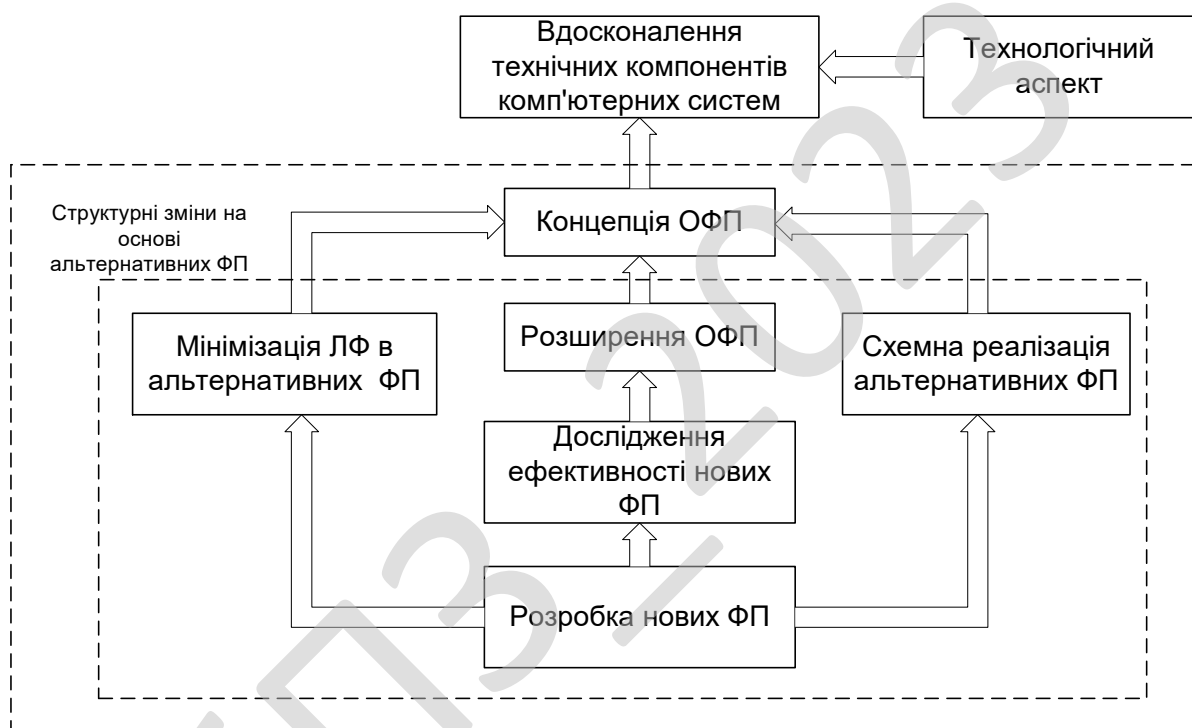


Рисунок 3.7 – Структурна схема системи

Визначимо вирішальну роль в проблемі вдосконалення технічних компонентів комп'ютерних систем концепції оптимальної форми представлення (ОФП) ЛФ, яка полягає в реалізації ЛФ в найбільш доцільній для неї ФП з точки зору забезпечення мінімальності параметрів складності реалізації ЛФ. Покажемо, що потенціал ОФП є значно вищим, ніж окремо взятої форми представлення ЛФ. Окреслено фактори, що стримують на даний час застосування концепції ОФП в широкій інженерній практиці та комплекс задач, які виникають у зв'язку із впровадженням ОФП ЛФ.

### 3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.8. Для кількісного оцінювання якості мінімізації за методом МОФ ЛФ обчислено середні значення показників  $S_S$  ЛФ множин  $L(3)$ ,  $L(4)$  та  $L(5)$  в ОРФП з метою порівняння з аналогічними показниками для класичної форми представлення. Отриманий метод забезпечує спрощення схемної реалізації, що виражається зменшенням середнього значення показника  $S_S$  в ОРФП для  $L(n)$  при  $n = \overline{3,5}$  на 23,8% порівняно з аналогічним показником в КФП.

Наведемо схемотехнічні рішення по застосуванню бінарних безінверторних форм представлення – БІ-К&АФП (паралельно застосовуються класична та алгебраїчна ФП) та БІ-К&РМФП (паралельно застосовуються класична та Ріда-мюллерівська ФП).

На відміну від класичних ПЛМ, для яких необхідне подвоєння вхідних шин для забезпечення реалізації будь-яких ЛФ на  $n$  входів і на  $S_{ad}$  членів логічної суми в традиційній класичній ФП, сутність запропонованого рішення полягає в доповненні існуючих ПЛМ, в яких ЛФ реалізуються виключно в класичній ФП, елементами додавання кон'юнкцій (ЕДК), що відповідають одній з альтернативних ФП в доповнення до класичного елемента додавання типу  $OR$ .

В формі представлення БІ-К&РМФП ЛФ реалізуються у вигляді кон'юнкцій, які додаються на елементах  $OR$  або  $XOR$  в залежності від характеру конкретної ЛФ. В формі БІ-К&АФП кон'юнкції додаються класично або алгебраїчно з ваговими коефіцієнтами на компараторах.

Вибір типу суматора (логічного, алгебраїчного або за модулем 2) здійснюється на етапі логічного проектування ПЛМ з урахуванням зменшення схемотехнічної складності ЕДК та забезпечення мінімальності необхідних доданків в аналітичному представленні ЛФ.

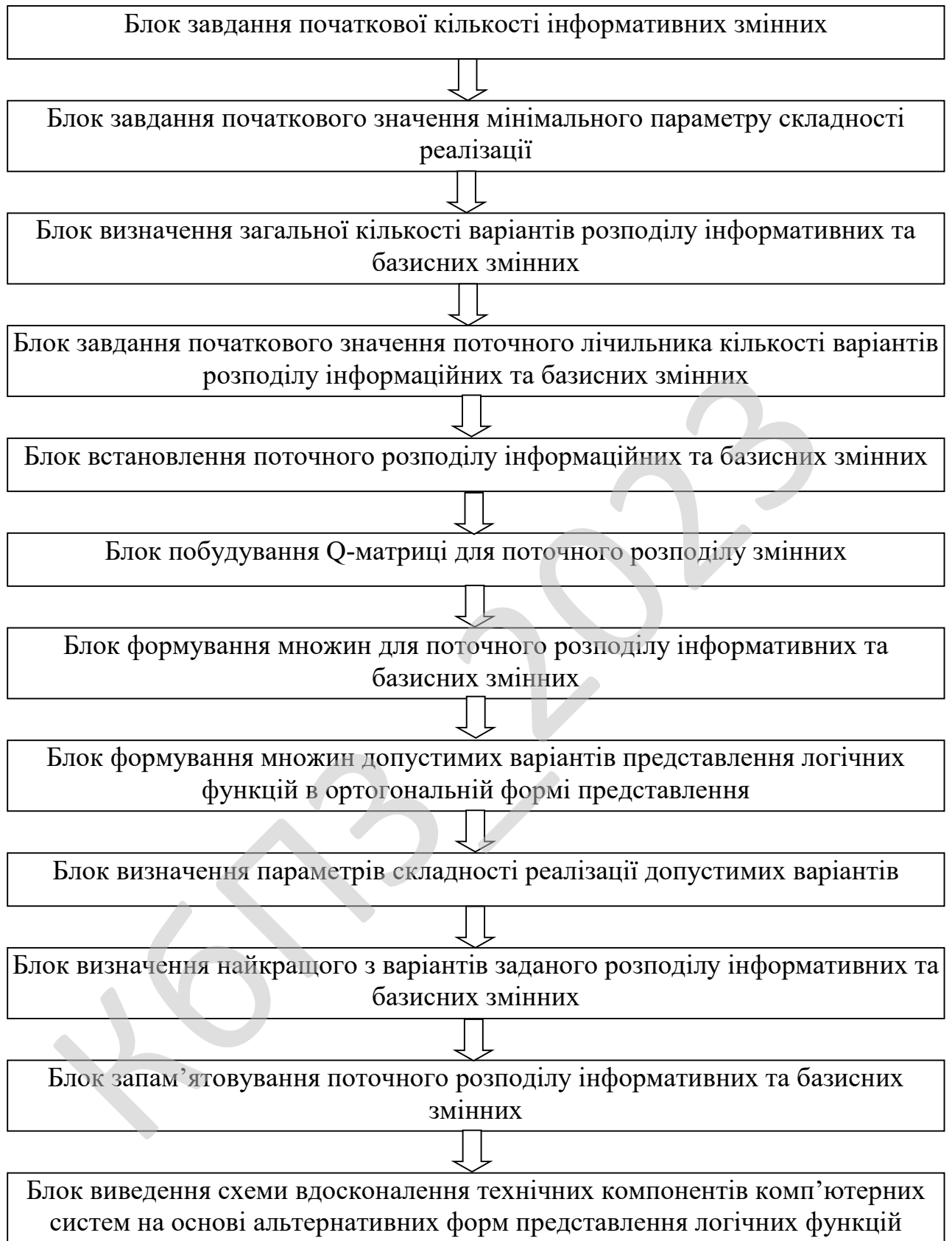


Рисунок 3.8 – Функціональна схема системи

Застосування бінарних форм дозволяє ліквідувати основний недолік КФП – дублювання входних шин ПЛМ та у багатьох випадках досягти зменшення потрібної кількості доданків в аналітичному рівнянні ЛФ порівняно з класичною реалізацією. Внаслідок зазначених схмотехнічних рішень зменшується площа частини формування кон'юнкцій та одночасно розширюються функціональні можливості, що обумовлено збільшенням кількості ЛФ, які оптимально можуть бути реалізованими на заданій площі.

Наведена типова схемна реалізація ортогональної форми представлення у вигляді окремих конструктивних частин ПЛМ – БФК-1 та БФК-2. На прикладі ЛФ з номером 32617 продемонстровано результат роботи розробленого методу мінімізації в ОРФП та схемну реалізацію зазначеної ЛФ до і після проведення мінімізації запропонованим методом. Виходячи з рисунку 3.9, застосування методу МОФ зменшує кількість елементів, необхідних для реалізації інформативних функцій ЛФ в ОРФП.

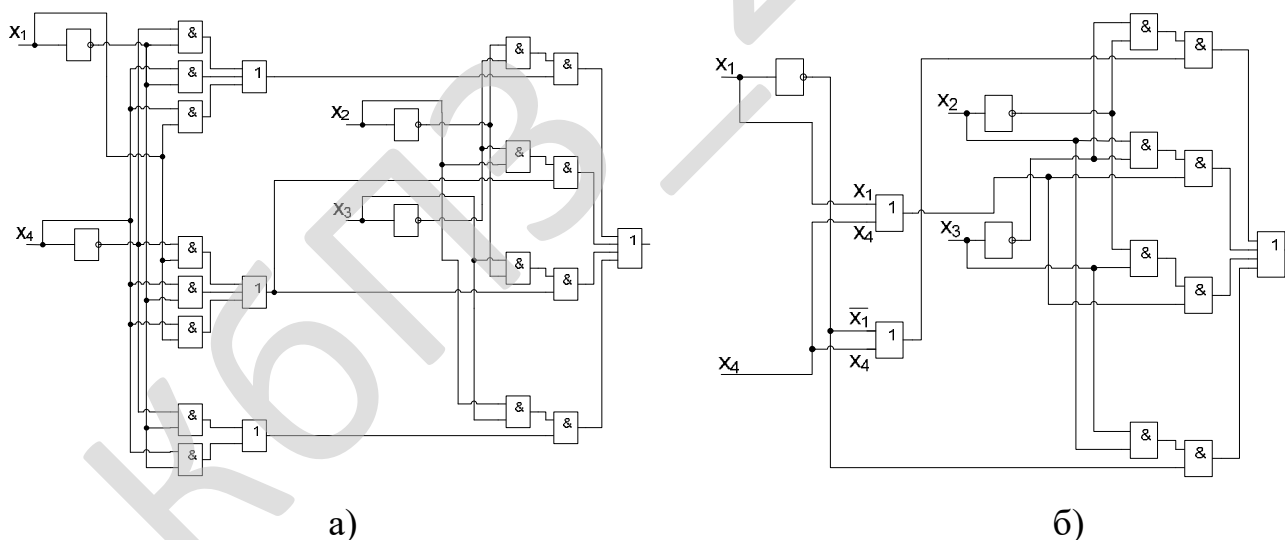


Рисунок 3.9 – Схема реалізації ЛФ з номером 32617 до (а) і після (б) обробки за методом мінімізації МОФ в ОРФП

Запропоновано схмотехнічну реалізацію нейронів в нейрочипах на основі використання АФП для реалізації ЛФ. Зазначене технічне рішення досягається

внаслідок введення в структуру нейрону на вході формувача спеціальних S-функцій, що дозволило розширити функціональність нейрочипів. Це обумовлюється зменшенням кількості нейронів в нейрочипі у зв'язку з тим, що достатньо одного нейрону запропонованої структури для реалізації будь-якої ЛФ, а не тільки порогової.

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.10. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

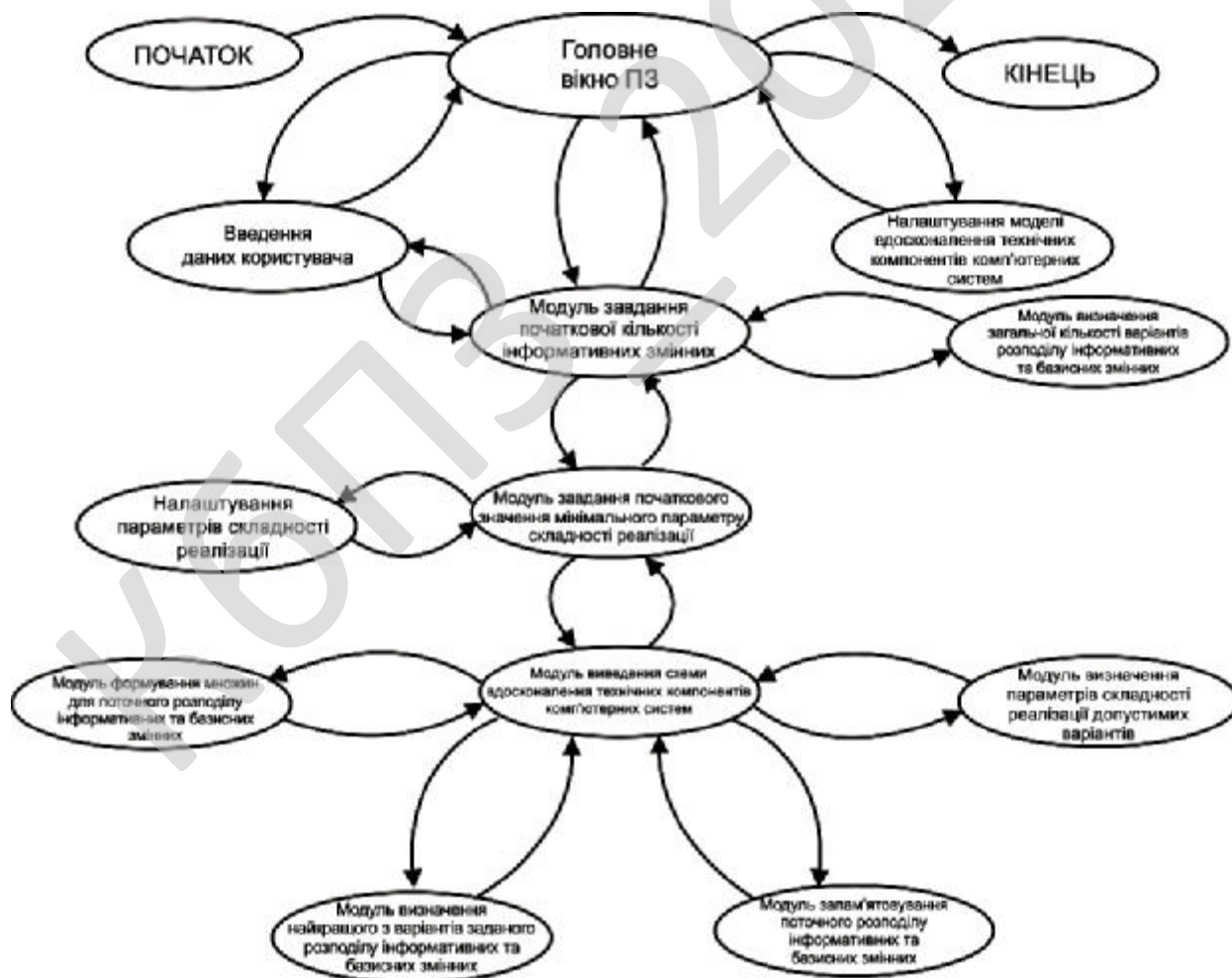


Рисунок 3.10 – Діаграма взаємодії процесів

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). 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.

					VKPM-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		42

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

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

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 вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

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.

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.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

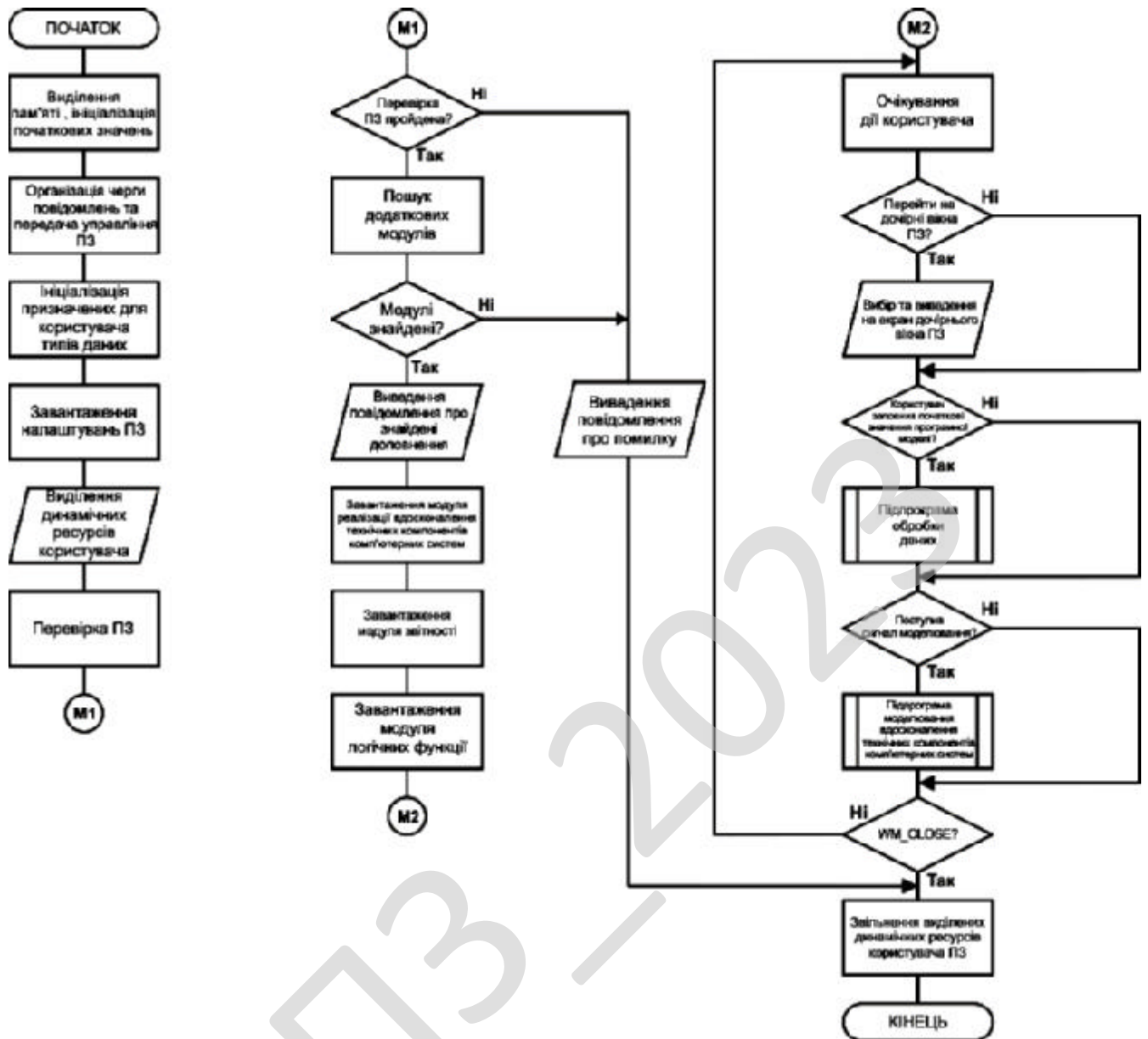


Рисунок 4.1 – Блок-схема основної програми

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

Це фундаментальна одиниця визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних

сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії.

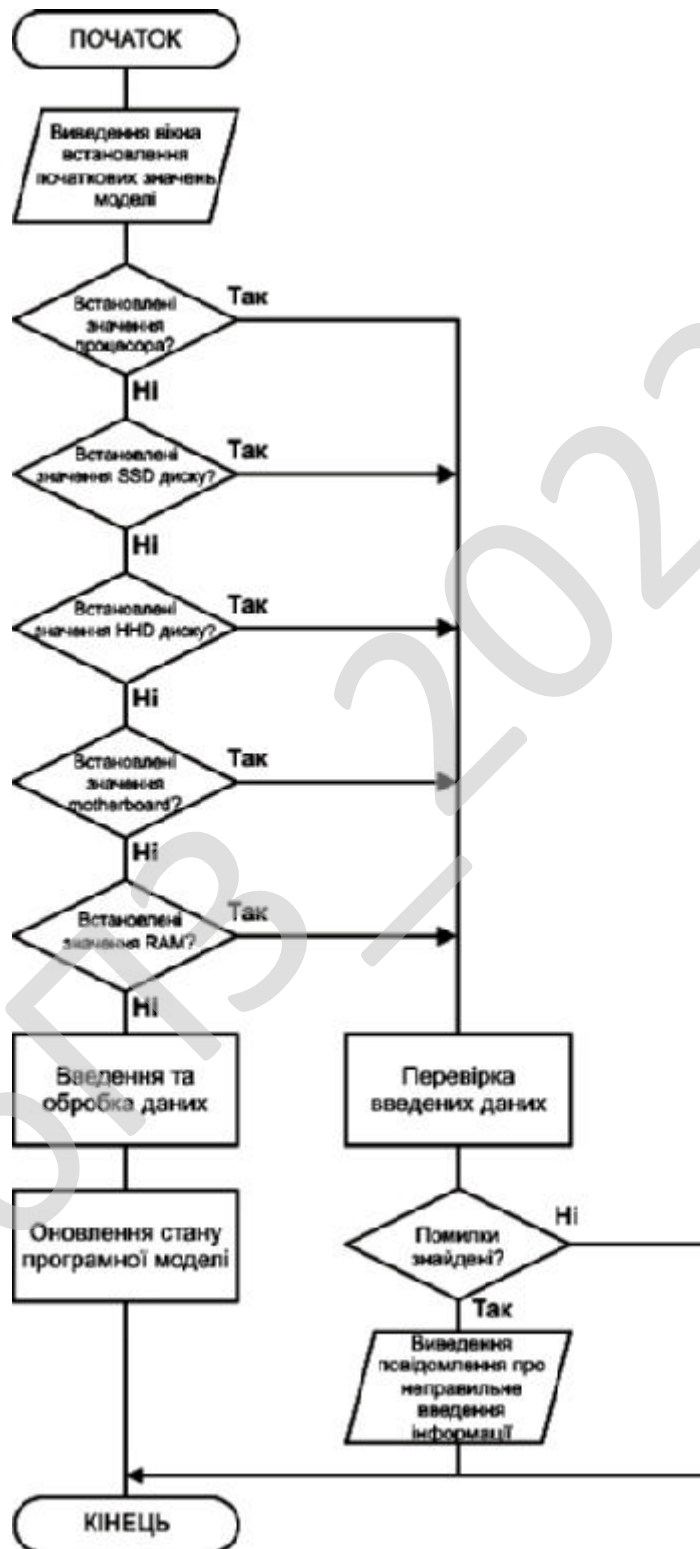


Рисунок 4.2 – Блок-схема роботи підпрограми



Рисунок 4.3 – Блок-схема роботи підпрограми

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.23.0048.00.00.ПЗ

Арк.

46

Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

При написанні програми використовувався модульно-реверсивний алгоритм, суть якого полягає в тому, що взаємодія всіх модулів проходить через блок керування модулями, цей алгоритм використовується при написанні програмних моделей та справ очних систем. Спрощена схема взаємодії приводе до оптимізації ПЗ.

Для цього 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

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

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.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

- Керівникам проектів, які керують розподілом завдань і контролем за проектом.
- Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.
- Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.
- Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

## 4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7564:2014 («Купина»). В Україні на основі консервативного підходу із залученням відомих і добре досліджених конструкцій була розроблена геш-функція, що базується на новому блоковому шифрі „Калина” (ДСТУ 7624:2014).

Національний стандарт ДСТУ 7564:2014 визначає криптографічну функцію гешування „Купина”, додатковий режим її застосування для формування коду автентифікації повідомлення (імітовставки), а також значення для перевірки реалізацій.

Для скорочення обсягу тексту національного стандарту була застосована математична нотація, що дозволяє отримати точний і компактний запис.

Водночас, такий підхід може ускладнювати сприйняття сутності алгоритму для фахівців, що не мають фундаментальної криптологічної освіти.

У роботі наводиться розгорнутий альтернативний опис функції гешування „Купина” із позначеннями, традиційними для галузі комп’ютерних наук.

### Термінологія та позначення

Вектор ініціалізації – бітова послідовність фіксованої довжини (512 або 1024 біта), що використовується як початкове значення при обчисленні геш-значення.

Внутрішній стан – бітова послідовність фіксованої довжини (512 або 1024 біта), що є проміжним значенням на кожній ітерації перетворення функції гешування, а також вхідним та вихідним значеннями перетворень  $P$  і  $Q$ ; для цих перетворень внутрішній стан подається як матриця розміром  $8 \times s$  байт.

Геш-значення (геш-вектор) – бітова послідовність фіксованої довжини ( $n = 8 \cdot s, s \in \{1, 2, \dots, 64\}$ ), що є результатом роботи функції гешування.

Доповнення – вставка додаткових біт у кінець повідомлення для отримання кратності довжини бітової послідовності довжині внутрішнього стану функції гешування.

Повідомлення – бітова послідовність довжини від 0 біт (порожній рядок) до  $2^{96}-1$  біт.

Функція стиснення – ітеративне перетворення, що відображає  $l$ -бітний блок повідомлення та  $l$ -бітне значення, отримане функцією стиснення на попередньому кроці, у нове  $l$ -бітне значення.

Далі використовуються наступні позначення:

- $\oplus$  – додавання за модулем 2 (XOR);
- $0x$  – префікс числа, що записане у шістнадцятковій системі числення;
- $a \bmod b$  – ціле невід’ємне число, що дорівнює залишку від ділення цілого числа  $a$  на натуральне число  $b$ ;
- $B_i$  –  $i$ -й байт вхідної послідовності;
- $C^i$  – константа перетворення XORRoundKey або Add64RoundKey для  $i$ -го циклу;
- $c$  – кількість стовпців внутрішнього стану в матричному поданні;
- $\varphi$  – функція стиснення;
- $H$  – визначена у стандарті функція гешування;
- $H(M)$  – результат обчислення функції гешування для повідомлення  $M$  (гешзначення);
- $IV$  – вектор ініціалізації;
- $l$  – розмір внутрішнього стану функції гешування (у бітах),  $l \in \{512, 1024\}$ ;
- $M$  – повідомлення;
- $m_i$  –  $i$ -й блок повідомлення  $M$ ;
- $n$  – довжина обчисленого геш-значення;
- $N$  – довжина повідомлення  $M$  без доповнення;
- $P, Q$  – складові перетворення функції стиснення;
- $P_{512}$  – перетворення  $P$  для 512-бітного внутрішнього стану;
- $P_{1024}$  – перетворення  $P$  для 1024-бітного внутрішнього стану;
- $Q_{512}$  – перетворення  $Q$  для 512-бітного внутрішнього стану;

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

- $Q_{1024}$  – перетворення  $Q$  для 1024-бітного внутрішнього стану;
- $r$  – кількість ітерацій у перетвореннях  $P$  і  $Q$  ( $r = t$  в ДСТУ 7564:2014);
- $S$  – внутрішній стан геш-функції;
- $t$  – кількість блоків  $m$ , з яких складається повідомлення  $M$ , включаючи доповнення;
- $v_i$  –  $i$ -й біт вхідної послідовності;
- $\Omega$  – завершальне перетворення;
- Купина- $n$  – режим використання функції гешування з усіченням обчисленого гешзначення до розміру  $n$  біт.

### Загальні положення

Під функцією гешування  $H$  розуміється залежне від вектора ініціалізації IV відображення послідовності біт  $M$  у геш-значення  $H(M)$  фіксованої довжини  $n$ .

ДСТУ 7564:2014 визначає функцію гешування, яка виконує перетворення «Купина-256» або «Купина-512», що забезпечують обчислення геш-значення з довжинами 256 або 512 біт відповідно.

Геш-значення довжиною 256 бітів додатково може бути усічено до бітової послідовності довжиною від 8 до 248 біт з кроком у 8 біт, 512 бітів може бути усічене до бітової послідовності довжиною від 264 до 504 біт з кроком у 8 біт.

Режим роботи для формування геш-значення довжиною  $n$  біт позначається як «Купина- $n$ ».

Основними режимами роботи функції гешування, що рекомендуються до застосування, є «Купина-256», «Купина-384» і «Купина-512».

### Структура перетворення

Функція гешування, визначена в ДСТУ 7564:2014, формує геш-значення для повідомлення, що складається з бітової послідовності довжини від 0 біт (порожній рядок) до  $2^{96}-1$  біт.

При формуванні геш-значення повідомлення доповнюється, далі поділяється на  $l$ -бітні блоки  $m_0, \dots, m_t$ , після чого виконується обробка кожного блоку шляхом ітеративного виконання функції стиснення  $\varphi$ .

При цьому формуються значення  $h_i = \varphi(h_{i-1}, m_i)$  де  $i = 1, \dots, t$ , а початкове значення  $h_0 = IV$ .

Після обробки останнього блоку повідомлення результуюче геш-значення обчислюється як  $H(M) = \Omega(h_t)$ , де  $\Omega$  – завершальне перетворення, що повертає  $n$  - бітне значення, кратне 8 ( $0 < n \leq l/2$ ).

КБГПЗ - 2023

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52



Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1.

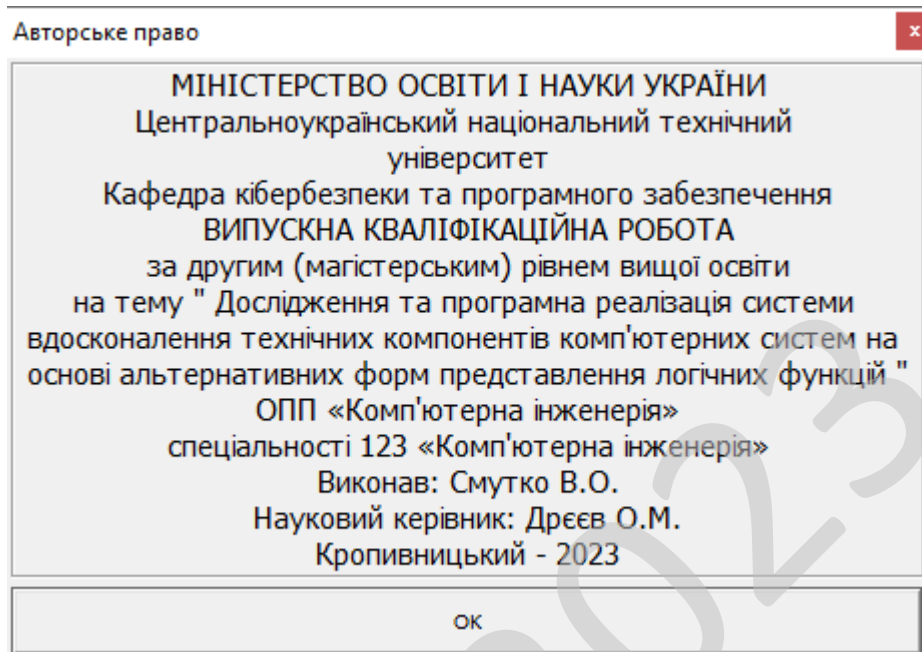


Рисунок 5.2 – Авторське право

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;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки та чорної скриньки. Тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки» пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– 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.

Проводилось тестування чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

– Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс;
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій.
- Помилки інтерфейсу.
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних.
- Помилки характеристик (необхідна ємність пам'яті і т.д.).
- Помилки ініціалізації та завершення.

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 (zareestruvatisya), заплативши авторів певну суму.

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

КБПЗ - 2023

					VKPM-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

*Метою розробки є дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.*

*Об'єктом дослідження є процес вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.*

*Предметом дослідження є методи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.*

*Методи дослідження базуються на методах комп'ютерної інженерії, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.
- Розроблено вітчизняний продукт вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій, який має більш широкі можливості, на відміну від існуючих аналогів.

## 7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці). В магістерській роботі було проведене дослідження та виконана програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій. 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	280
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

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.	–	28000
33. Норматив додаткової зарплати, % :	Н <sub>д</sub>	10
34. Норматив відрахувань у соціальні фонди, %	Н <sub>с</sub>	22
35. Норматив загальногосподарських витрат, %	Н <sub>г</sub>	15
36. Норматив витрат на освоєння нових мов програмування, %	Н <sub>п</sub>	15
37. Рівень рентабельності програмної продукції, %	Р <sub>е</sub>	50
38. Ставка податку на додану вартість, %	Н <sub>дв</sub>	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-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

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.:

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

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

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

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)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 48 = 81 \text{ люд/день.}$$

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

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63



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	7	630	10,5
Монітор	60	7	420	7
Клавіатура	30	7	210	3,5
Маніпулятор «мишка»	30	7	210	3,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	1	30	0,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	250	625	10,42
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 <sub>ч</sub>	41,08

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:

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

$$\Phi_{op}^c = \frac{41 \cdot 3}{1,2} = 102,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 102,5 / (60 \cdot 8) = 0,2 \text{ ставки.}$$

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	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	



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	17646	52938
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	2,25	14000	94500
Інженер-електронщик	0,2	12000	7200
Інженер-системотехнік	0,25	12000	9000
Адміністратор мережі	0,5	12000	18000
Системний програміст	0,25	12000	9000
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	12000	9000
Бухгалтер-економіст	0,5	12000	18000
Всього за період розробки	$R_{cn} = 5,7$	-	$\Phi_{роб} = 235638$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{co} = \frac{235638}{5,7 \cdot 60} = 689 \text{ грн.}$$

#### 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-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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

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

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

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

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно 8  $m^2$ . З урахуванням цього:

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

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:

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

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

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

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 фірми Компбест за 03.11.23 – джерело <https://compbest.com.ua/>.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	AMD Ryzen 5 2400G (YD2400C5M4MFB) AM4, 4 ядра, 8 потоків, 3.6 GHz, 3.9 GHz, TDP - 65 Вт, 14nm	-
Системна плата	ASRock A520M-HVS сокет - AM4, DDR4, LAN - 1 Гбит/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Відеокарта	Radeon R7 350 2Gb Afox (AFR7350- 2048D5H4-V3) PCI-Express 3.0, 2 ГБ, GDDR5, 128 Bit	-
Жорсткий диск	SSD 2.5" 256GB Mibrand (MI2.5SSD/CA256GBST) 256 GB, 3D TLC NAND, 2.5", SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 8GB 3200 MHz Dato (DT8G4DLND32)	-
DVD-привод	DVDRW Pioneer DVR-TD10RS SATA Slim Black Bulk (DVR-TD10RS)	-
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA-489, PSU 350W(FSP Brand: ATX- 350PNR, 12cm), black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	–	–
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB3.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D ( 5ms, 300/3000:1, 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

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	199177

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 7.8 – 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
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	0	20	0,0
6. Господарський інвентар	28000	25	7000
Всього по групі	33190	-	8297,5
7. Нематеріальні активи	120000	10	12000
Разом	$K_p = 1760367$		$A_p = 190286$



Згідно прийнятих норм на підприємстві  $n_{\text{вум}}$  приймаємо 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає  $Ц_n=210$  грн., визначаємо вартість паперу за період розробки:

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

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

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 150):

$$З_{M2} = \sum Ц_d, \quad (7.17)$$

де:  $Ц_d$  – вартість дисків CD/DVD: CDR box – 22,4 грн./шт., DVD-R box – 35 грн./шт.

$$З_{M2} = 150 \cdot 22,4 = 3360 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$З_{M3} = \sum Ц_z, \quad (7.18)$$

де:  $Ц_z$  – 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.

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

$$З_M = (105 + 3360 + 1702) / 280 = 18 \text{ грн.}$$

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

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

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

$$O_n = 300 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 280$  прим.):

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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

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

$$A_m = 190286 \cdot 3 / (280 \cdot 12) = 170 \text{ грн.}$$

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

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

$$C_n = 300 + 30 + 73 + 45 + 18 + 45 + 170 = 681 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1. Основна зарплата виконавців	$Z_o$	300
2. Додаткова зарплата виконавців	$Z_d$	30
3. Відрахування на соціальні потреби	$C_{oc}$	73
4. Загальногосподарські витрати	$\Gamma_{ocn}$	45
5. Витрати на матеріали	$Z_m$	18
6. Освоєння нових операційних систем, мов програмування	$O_n$	45
7. Амортизація основних фондів	$A_m$	170
8. Повна собівартість програмного забезпечення	$C_n$	681
9. Плановий прибуток	$P_p$	341
10. Ціна підприємства $C_n = C_n + P_p$	$C_n$	1022
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{ов} \cdot C_n$	$ПДВ$	204,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	$C$	1226,4

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, %.

$$P_p = 0,01 \cdot 50 \cdot 681 = 341 \text{ грн.}$$

## 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	–	1226
Всього капітальних витрат	–	1226

## 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.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	$Z_p$	31000	18500
2. Витрати на електроенергію	$Z_{ел}$	2267	1570
3. Витрати на амортизацію	$Z_{ам}$	0	307
Всього витрат за рік	$I$	33267	20377

Витрати на профілактичні роботи:

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

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

$Z_2$  – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення витрати на обслуговування системи зменшились з 31000 грн до 18500 грн на рік.

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	–	1226	–	306,5
Всього відрахувань	-	–	1226	–	306,5

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

$$Z_{ел} = П_{ел} \cdot T_p \cdot Ц_{ел} \quad (7.24)$$

$$Z_{ел баз} = 0,545 \cdot 1300 \cdot 3,2 = 2267 \text{ грн.}$$

$$Z_{ел нов} = 0,545 \cdot 900 \cdot 3,2 = 1570 \text{ грн.}$$

## 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:

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

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

$$E_e = (1022 - 681) \cdot 280 - (0,05 \cdot 1408000 + 0,5 \cdot 199177 + 0,25 \cdot 33190 + 0,1 \cdot 28000) \cdot 3/12 = 50208 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

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

$$T_e = \frac{1760367}{(1022 - 681) \cdot 280 \cdot 12 / 3} = 4,5 \text{ роки}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

$$E_{cn} = (33267 - 20377) - 0,25 \cdot 1226 = 12584 \text{ грн.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	280
2. Повна собівартість розробленої програми	Грн.	681
3. Ціна розробленої програми	Грн.	1022
4. Плановий прибуток від реалізації розробленої програми	Грн.	341
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1760367
7. Загальний прибуток від реалізації програмної продукції	Грн.	95480
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	50208
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Роки	4,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1226
11. Величина економічного ефекту у користувача програмної продукції	Грн.	12584
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

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

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

$$T_{cn} = \frac{1226}{33267 - 20377} = 0,1 \text{ року} .$$

## 7.9 Висновки

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

КБПЗ-2023

					ВКРМ-123.23.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80



- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці.

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

Електронно-обчислювальні машини (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98 [2].

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

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

### 8.3 Аналіз умов праці на робочому місці програміста

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 4,8 м×7,2 м×2,8 м. Одна з її більших стін має шість двостулкових вікон, розмірами 2,1 м×1,9 м, які виходять на північний схід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита леноліумом, всі стіни пофарбовані світло оранжевого кольору до висоти 2,8м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1245 мм×840 мм. Висота столів 750 мм. Висота стільців від рівня підлоги становить 425 мм.

Згідно НПАОП 0.00 – 1.28 – 10 «Правила охорони праці під час електронно-обчислювальних машин» площа повинна задовольняти умові – не менш 6 м<sup>2</sup> на одне робоче місце. Кратність повітрообміну в приміщенні вузла також

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83









і коефіцієнтів відбиття поверхонь. Чим ближче світильники до стелі і стін, тим менше світла доходить до робочої поверхні. Чим ближче робоча поверхня до підлоги, тим більше світла вона отримує. Чим більші коефіцієнти відбиття стелі, стін і підлоги, тим більше світла повертається до робочої поверхні.

– Коефіцієнт запасу  $Z$  – Коефіцієнт запасу залежить від типу і частоти обслуговування світильників, ламп і поверхонь, а також від режиму роботи освітленн. Зазвичай коефіцієнт запасу приймається в межах 1,1-1,5. Приймаємо 1,1.

Для розрахунку кількості світильників в приміщенні, вам потрібно врахувати рівень освітлення, світловий потік кожного світильника, коефіцієнт використання і площу приміщення.

Загальний алгоритм розрахунку:

– **Визначте бажаний рівень освітлення (E):** Це рівень освітлення, який вам потрібно в приміщенні. Він вимірюється в люксах (лк). Для офісних приміщень зазвичай рекомендується рівень освітлення близько 300-500 лк, залежно від конкретних потреб.

– **З'ясуйте світловий потік кожного світильника (F):** Це кількість світла, яке виділяє кожен світильник і вимірюється в люменах (лм). Інформацію про світловий потік зазвичай надає виробник світильника.

– **Визначте коефіцієнт використання ( $K_c$ ):** Цей коефіцієнт вказує на частину світла, яка досягає поверхні, де працюють співробітники. Виробники світильників також надають інформацію про цей параметр.

– **Визначте виправлення світлового потоку (K):** Виправлення світлового потоку враховує втрати світла через пил, забруднення та відбиття від поверхонь. Для офісних приміщень, де чистота зазвичай добра, цей коефіцієнт може бути близько 0,9.

– **Виміряйте площу приміщення (S):** Виміряйте площу приміщення, в якому планується розташування світильників, в квадратних метрах ( $m^2$ ).

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88



## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

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 вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

– Досліджена система вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

– На основі отриманих результатів досліджень створена програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

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

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 ДСТУ 7564:2014.

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 12584 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Смутко В.О. Дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.

2. Панаско О.М. Мінімізація логічних функцій у вигляді кусочно-постійних рядів / О.М. Панаско // Шістнадцята наукова сесія осередку наукового товариства ім. Т.Шевченка у Черкасах: матеріали доп. – 2005. – С.151-152.

3. Кочкар'юв Ю.О. Оптимізація структури комбінаційних схем шляхом використання оптимальної форми представлення логічних функцій / Ю.О. Кочкар'юв, І.В. Синько, О.М. Панаско // Інформаційні і моделюючі технології: II міжнародна науково-технічна конференція [«ІМТ-2009»] : матеріали наук.-техн. конф. – Черкаси. – 2009. – С. 38-39.

4. Kochkarev Y.A. Ortogonal forms of presentation of boolean functions in device blocks / Y.A. Kochkarev, I.I. Osipenkova, E.N. Panasko // Датчики, приборы и системы ДПС – 2009 : международная научно-техническая конференция [«ДПС 2009»] : сб. научн. труд. – Ялта. – 2009. – С.39-42.

5. Пат. 10201 Україна, МПК G06F7 / 544. Програмована логічна матриця / Ю.О. Кочкар'юв, О.М. Панаско, С.О. Шакун. – № u200502059; заявл. 05.03.05; опубл. 15.11.05. Бюл. №11.

6. Пат. 42063 Україна, МПК G06F 7/48. Нейрочип / Ю.О. Кочкар'юв, Н.С. Кучерова, О.М. Панаско. – № u200814795; заявл. 22.12.08; опубл. 25.06.09. Бюл. №12.

7. Aggarwal C.C. Neural Networks and Deep Learning: A Textbook 1st ed. 2018 Edition. – Springer, 2018. – 520 p

8. Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. – Pearson, 2001. – 620 c.

					<b>ВКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

9. Brink H., Richards J., Fetherolf M. Real-World Machine Learning. – Manning, 2016. – 474 p.
10. 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.
11. Fenner M. Machine Learning with Python for Everyone (Addison-Wesley Data & Analytics Series) 1st Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 586 p.
12. Foreman J.W. Data Smart: Using Data Science to Transform Information into Insight 1st Edition. – Wiley, 2013. – 432 p.
13. Hurbans R. Grokking Artificial Intelligence Algorithms. – Manning, 2020. – 631 p.
14. Gusfield D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology 1st Edition. – Cambridge University Press, 2008. – 556 p.
15. Kotu V., Deshpande B. Data Science: Concepts and Practice. – Elsevier Science, 2018. – 953 p.
16. Knowledge Base A Complete Guide – 2021 Edition // The Art of Service – Knowledge Base Publishing, 2020. – 306 p.
17. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
18. Mattmann C. Machine Learning with TensorFlow, Second Edition. – Manning, 2020. – 1124 p.
19. Mueller J.P., Massaron L. Machine Learning For Dummies. – Wiley, 2016. – 714 p.
20. Teofili T. Deep Learning for Search. – Manning, 2019. – 695 p.
21. Rungta K. TensorFlow in 1 Day: Make your own Neural Network. – Publishdrive, 2019. – 587 p.
22. Weidman S. Deep Learning from Scratch: Building with Python from First Principles. – O’Reilly. 2019. – 252 p.

					<b>БКРМ-123.23.0048.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>93</b>

23. Rajasekaran S., Vijayalakshmi Pai G.A. Neural networks, fuzzy logic, and genetic algorithms: synthesis and applications (with cd-rom) Kindle Edition. – PHI, 2013. – 628 p.

24. 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.

25. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12.

26. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

27. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskiy, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34.

28. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems. Volume 152, 2021, Pages 85-98.

29. 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>.

30. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207.

31. 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.

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

33. 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.

34. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

35. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

36. 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.

37. 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.

38. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019,



Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

46. Smirnov, S., Bulekbaeva, G., Kikvidze, O.G., Lakhno, V., Brzhanov, R., Tabylov, A. «Computer simulation in the MathCAD package of plastic deformation of the deposited layer on the flat surface of the part». Journal of Theoretical and Applied Information Technology Volume 97, Issue 20, 2019, Pages 2467-2484. (Scopus).

47. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

48. 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.

49. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95

50. 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.

51. 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.

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-123.23.0048.00.00.ТЗ</b>			
<i>Вим.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>	<i>Смутко В.О.</i>				<i>Дослідження та програмна реалізація системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>	<i>Дресв О.М.</i>					<i>М</i>	<i>1</i>	<i>6</i>
<i>Н. Контр.</i>	<i>Коваленко А.С.</i>					<i>ЦНТУ КІ-22М-2</i>		
<i>Затв.</i>	<i>Смірнов О.А.</i>							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 35-13 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 вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій.

## 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;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

					<b>ВКРМ-123.23.0048.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи вдосконалення технічних компонентів комп'ютерних систем на основі альтернативних форм представлення логічних функцій;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

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

## 5.4 Вимоги до архітектури

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

## 5.5 Вимоги до надійності

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

					ВКРМ-123.23.0048.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Delphi 10.

					ВКРМ-123.23.0048.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

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

## 7 Економічні вимоги

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

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

## 8 Вимоги щодо охорони праці

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

					<b>ВКРМ-123.23.0048.00.00.ТЗ</b>	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 – 97 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 13.12.2023 р.

					<b>ВКРМ-123.23.0048.00.00.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  
\_\_\_\_\_ Дреєв О.М.

***Дослідження та програмна реалізація  
системи вдосконалення технічних компонентів комп'ютерних систем на  
основі альтернативних форм представлення логічних функцій***

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 38

Літера: РП

Кропивницький – 2023 року

**Основний файл проекту розробленого програмного забезпечення  
(Program\_Model\_DevelopComputingSystem.dpr)**

```
program ProgramModelDevelopComputingSystem;

uses
  Forms,
  Messages,Graphics,
  Windows,Unit1 in 'Unit1.pas' {Form1},
  Unit2 in 'Report.pas' {Form2},
  Unit3 in 'fset.pas' {Form3},
  Unit4 in 'data.pas' {Form4},
  Unit5 in 'DevelopComputingSystem.pas' {Form5},
  Unit6 in 'U6.pas' {Form6},
  U_splash in 'U7.pas' {U_Form_Splash}.

const
  WM_CONST1=WM_USER+322; WM_CONST2=WM_USER+105; WM_CONST3=WM_USER+99;

  {$R *.res}

  ///Розробив Смутко Віталій Олександрович, ЦНТУ, 2023

begin
  Application.HintPause:=400; Application.HintHidePause:=1000;
  Application.Title:= 'ProgramModelDevelopComputingSystem';
  Application.Initialize;
  try
    U_Form_Splash:=TU_Form_Splash.Create(Application);
    U_Form_Splash.Show;
    U_Form_Splash.Update;
    Sendmessage(U_Form_Splash.Handle,WM_MY,0,'Start');
    Application.CreateForm(TForm1, Form1);
    Application.CreateForm(TForm2, Form2);
    Application.CreateForm(TForm3, Form3);
    Application.CreateForm(TForm4, Form4);
    Sendmessage(U_Form_Splash.Handle,WM_MY,0,'End');
    Application.CreateForm(TForm5, Form5);
  finally U_Form_Splash.free; end;
  Application.Run;
end.
```

**Файл форми взаємодії та відображення звіту роботи  
(Report.pas)**

```

unit Report;

interface

uses
Windows, glReport, glCapt, glBevel, glPage, Printers,
glLabel, glRuler, Mask, glLBox, dsgnintf, Spin, geRPEdit,
Menus, ExtCtrls, StdCtrls, Buttons, ComCtrls, Controls, Dialogs,
Forms, Classes, Sysutils;
///Розробив Смутко Віталій Олександрович, ЦНТУ, 2023

type

tmyRep_Property = class( TPropertyEditor )
function GetAttributes: TPropertyAttributes; override;
function GetValue: string; override;
procedure Edit; override;
end;

tmyRep_Editor = class(TComponentEditor)
procedure ExecuteVerb(Index: Integer); override;
function GetVerb(Index: Integer): string; override;
function GetVerbCount: Integer; override;
end;

tmyReportEditor = class(TComponent)
FReport: tmyReport;
protected
procedure Notification(AComponent: TComponent; Operation: TOperation); override;
public
procedure Preview;
procedure Edit;
published
property Report: tmyReport read FReport write FReport;
end;

tmyRepEditor = class(TForm) //основний клас взаємодії
OpenDialog1: TOpenDialog; SaveDialog1: TSaveDialog;
PM_Control: TPopupMenu; N_Linktofile: TMenuItem;
N1: TMenuItem; N2: TMenuItem;
PC: tmyPageControl; Panel1: TPanel;
Bevel4: TBevel; P_Sides: TPanel;
Panel2: TPanel; glBevel1: tmyBevel;
Bevel2: TBevel; Bevel1: TBevel;
B_Label: TSpeedButton; sb_Open: TSpeedButton;
sb_Save: TSpeedButton; sb_Preview: TSpeedButton;
sb_Book: TSpeedButton; sb_Album: TSpeedButton;
Bevel3: TBevel; sb_OLE: TSpeedButton;
sb_SnapToGrid: TSpeedButton; b_Bevel: TSpeedButton;
sb_Print: TSpeedButton; TabSheet1: TTabSheet;
TabSheet2: TTabSheet; Memo1: TMemo;
P_Font: TPanel; ColorDialog1: TColorDialog;
N3: TMenuItem; N_DeleteObject: TMenuItem;
P_HRuler: TPanel; P_Main: TPanel;
ScrollBar_: TScrollBar; ShapeSize: TShape;
P_VRuler: TPanel; TabSheet3: TTabSheet;
ImageList1: TImageList; HRuler: tmyRuler;
VRuler: tmyRuler; glBevel4: tmyBevel;
sb_FixAllMoving: TSpeedButton; sb_FixMoving: TSpeedButton;
glLabel3: TLabel; N_OLESize: TMenuItem;
N_Clip: TMenuItem; N_Center: TMenuItem;
N_Scale: TMenuItem; N_Stretch: TMenuItem;
N_AutoSize: TMenuItem; P_SBar: TPanel;
Panel5: TPanel; glLabel4: TLabel;
glLabel5: TLabel; glLabel7: TLabel;

```

```

se_Width: TSpinEdit; se_Top: TSpinEdit;
se_Left: TSpinEdit; glLabel6: TLabel;
se_Height: TSpinEdit; cb_Components: TComboBox;
glLabel8: TLabel; N4: TMenuItem;
Bevel5: TBevel; SB_Left: TSpeedButton;
SB_Bottom: TSpeedButton; SB_Right: TSpeedButton;
SB_Top: TSpeedButton; sb_AlignL: TSpeedButton;
sb_AlignC: TSpeedButton; sb_AlignR: TSpeedButton;
sb_AlignW: TSpeedButton; RxSpeedButton8: TSpeedButton;
RxSpeedButton9: TSpeedButton; sb_BevelBold: TSpeedButton;
glBevel2: tmyBevel; Panel3: TPanel;
RxSpinEdit1: TSpinEdit; Panel6: TPanel;
sbFontColor: TSpeedButton; glBevel3: tmyBevel;
Panel7: TPanel; Edit1: TMemo;
FE_OLE: TEdit; SpeedButton1: TSpeedButton;
OpenOLEFile: TOpenDialog; sb_FontUnderline: TSpeedButton;
sb_FontItalic: TSpeedButton; sb_FontBold: TSpeedButton;
TabSheet4: TTabSheet; glLabel1: TLabel;
lb_Params: tmyListBox; Panel4: TPanel;
SpeedButton2: TSpeedButton; sbBackColor: TSpeedButton;
SpeedButton3: TSpeedButton; CheckBox1: TCheckBox;
FontComboBox1: TComboBox; ColorComboBox1: TComboBox;
procedure OpenClick(Sender: TObject);
procedure SaveClick(Sender: TObject);
procedure ScrollBox_MouseDown(Sender: TObject; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
procedure FormCreate(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure SB_LeftClick(Sender: TObject);
procedure FontComboBox1Change(Sender: TObject);
procedure RxSpinEdit1Change(Sender: TObject);
procedure ColorComboBox1Change(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure sb_BookClick(Sender: TObject);
procedure N1Click(Sender: TObject);
procedure sb_FontBoldClick(Sender: TObject);
procedure sb_AlignLClick(Sender: TObject);
procedure Memo1Change(Sender: TObject);
procedure sbFontColorClick(Sender: TObject);
procedure N_DeleteObjectClick(Sender: TObject);
procedure ScrollBox_MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure ScrollBox_MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure FormDestroy(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure sb_FixMovingClick(Sender: TObject);
procedure N_AutoSizeClick(Sender: TObject);
procedure sb_SnapToGridClick(Sender: TObject);
procedure se_SizeChange(Sender: TObject);
procedure cb_ComponentsChange(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure sb_BevelBoldClick(Sender: TObject);
procedure se_TopChange(Sender: TObject);
procedure se_WidthChange(Sender: TObject);
procedure se_HeightChange(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FE_OLEChange(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure sb_PrintClick(Sender: TObject);
procedure sb_PreviewClick(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure cb_ComponentsKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure CheckBox1Click(Sender: TObject);
private
ScrollBox: TlaScrollBox;
SelectedControl: tmyReportItem;

```

```

fSelection: boolean;
SelectionRect: TRect;
procedure RemakeComponentsList;
procedure read( FileName: string; ParentWnd: TWinControl );
procedure Save( FileName: string );
procedure OnMouseDown_(Sender: TObject; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
procedure OnMouseUp_(Sender: TObject; Button: TMouseButton; Shift: TShiftState;
X, Y: Integer);
procedure OnMouseMove_(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure OnResize_(Sender: TObject);
procedure ShowComponentPos(Control: TControl);
procedure AssignEventsToAllComponents;
procedure UpdateToolBar( Control: tmyReportItem);
public
Component: tmyReport;
procedure Preview(glReport: tmyReport);
procedure Edit(glReport: tmyReport);
end;

TPublicControl = class(TControl)
public
property Caption;
end;

TPublicControlClass = class of TPublicControl;

const
IGNORE_VALUE = 65536;
var
glRepEditor: tmyRepEditor;
Form2: TComponent;

implementation
uses glTypes, glUtils;
{$R *.DFM}

procedure tmyRepEditor.OnMouseMove_(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
var
DC: HDC;
i: integer;
begin
if fSelection then ScrollBox_MouseMove(Sender, Shift, X, Y);
if sb_FixAllMoving.Down then exit;
if TControl(Sender).Tag = 0 then exit;
if not fMouseDown then exit;
with ScrollBox do
for i:=0 to ControlCount-1 do
if (Controls[i] is tmyReportItem) then with tmyReportItem(Controls[i]) do
if Selected and not bool(Fixed) then
begin
Left:= ((Left + X - ControlPos.x)div Step.X)*Step.X;
Top:= ((Top + Y - ControlPos.y)div Step.Y)*Step.Y;
end;
fSkipSizeUpdate:= true;
ShowComponentPos(SelectedControl);
TControl(Sender).Left:= TControl(Sender).Left + X - ControlPos.x;
TControl(Sender).Top:= TControl(Sender).Top + Y - ControlPos.y;
{
TControl(Sender).Tag:= 2;//...on moving
DC:= GetDC( TControl(Sender).Parent.Handle );
DrawFocusRect( DC, FocusRect );
FocusRect:= Bounds( TControl(Sender).Left+X-ControlPos.x,
TControl(Sender).Top+Y-ControlPos.y, SelectedControl.Width,
SelectedControl.Height );
DrawFocusRect( DC, FocusRect );
ReleaseDC( TControl(Sender).Parent.Handle, DC );
}
}

```

```

end;

procedure tmyRepEditor.OnResize_(Sender: TObject);
begin
fSkipSizeUpdate:= true;
if Sender = SelectedControl then ShowComponentPos(TControl(Sender));
end;

procedure tmyRepEditor.read( FileName: string; ParentWnd: TWinControl );
begin
ScrollBar.HorzScrollBar.Position:= 0;
ScrollBar.VertScrollBar.Position:= 0;
SelectedControl:= nil;
UpdateToolBar( nil );
Component.LoadFromFile( FileName );
Component.CreateReport( ParentWnd, true );
AssignEventsToAllComponents;
RemakeComponentsList;
end;

procedure tmyRepEditor.Save( FileName: string );
begin
ScrollBar.HorzScrollBar.Position:= 0;
ScrollBar.VertScrollBar.Position:= 0;
Component.SaveToFile( FileName );
end;

procedure tmyRepEditor.OpenClick(Sender: TObject);
begin
OpenDialog1.InitialDir:= ExtractFilePath(ParamStr(0));
if OpenDialog1.Execute then
Read( OpenDialog1.FileName, ScrollBox );
end;

procedure tmyRepEditor.Save1Click(Sender: TObject);
begin
SaveDialog1.InitialDir:= ExtractFilePath(ParamStr(0));
if SaveDialog1.Execute then Save( SaveDialog1.FileName );
end;

procedure tmyRepEditor.ScrollBox_MouseDown(Sender: TObject; Button:
TMouseButton;
Shift: TShiftState; X, Y: Integer);
var
l, Compon: tmyReportItem;
R: TRect;
pt: TPoint;
begin
if ssCtrl in Shift then
begin
SelectionRect:= Rect( 0,0,0,0 );
SelPt.X:= X - ScrollBox.HorzScrollBar.Position;
SelPt.Y:= Y - ScrollBox.VertScrollBar.Position;
SelPt:= ScrollBox.ClientToScreen(SelPt);
fSelection:= true;
end;
if (B_Label.Down) or (B_Bevel.Down) or (sb_OLE.Down) then
begin
Compon:= Component.AddComponent;
with Compon do
begin
Left:= X - ScrollBox.HorzScrollBar.Position;
Top:= Y - ScrollBox.VertScrollBar.Position;
if B_Label.Down then
begin
SideLeft:= 0;
SideTop := 0;
SideRight:= 0;
SideBottom:= 0;

```

```

end;
OnMouseDown:= OnMouseDown_;
OnMouseUp:= OnMouseUp_;
OnMouseMove:= OnMouseMove_;
OnResize:= OnResize_;
ContainOLE:= sb_OLE.Down;
B_Label.Down:= false;
B_Bevel.Down:= false;
sb_OLE.Down:= false;
RemakeComponentsList;
end;
end else
begin
R:= ScrollBox.ClientRect;
pt.x:= 0; pt.y:= 0; pt:= ScrollBox.ClientToScreen(pt);
OffsetRect( R, pt.x, pt.y );
ClipCursor( @R );
end;
end;

procedure tmyRepEditor.ScrollBox_MouseMove(Sender: TObject; Shift: TShiftState;
X, Y: Integer);
var
DC: HDC;
pt: TPoint;
begin

if not fSelection then exit;
DC:= GetDC( 0 );
DrawFocusRect( DC, SelectionRect );
Pt.X:= X - ScrollBox.HorzScrollBar.Position;
Pt.Y:= Y - ScrollBox.VertScrollBar.Position;
pt:= ScrollBox.ClientToScreen(pt);
SelectionRect:= Bounds( min( SelPt.X, pt.X ), min( SelPt.Y, pt.Y ), abs(
SelPt.X-pt.X ), abs( SelPt.Y-pt.Y ) );
DrawFocusRect( DC, SelectionRect );
ReleaseDC( 0, DC );
end;

procedure tmyRepEditor.sb_BookClick(Sender: TObject);
begin
if TControl(Sender).Tag=1 then
f_PrintReport.CReport1.Orientation:=f_PrintReport.PrintWin1.Orientation else
f_PrintReport.CReport1.Orientation:= f_PrintReport.PrintWin2.Orientation;
if TControl(Sender).Tag=1 then Printer.Orientation:=poPortrait
else Printer.Orientation:= poLandscape;
UpdatePageSize;
end;

procedure tmyRepEditor.N1Click(Sender: TObject);
begin
ScrollBox.RemoveControl( SelectedControl );
ScrollBox.InsertControl( SelectedControl );
end;

procedure tmyRepEditor.sb_FontBoldClick(Sender: TObject);
begin
if not Assigned(SelectedControl) then exit;
with SelectedControl do
case TControl(Sender).Tag of
1: FStyle:= FStyle xor 1;
2: FStyle:= FStyle xor 2;
3: FStyle:= FStyle xor 4;
end;
end;

procedure tmyRepEditor.sb_AlignLClick(Sender: TObject);
begin
if not Assigned(SelectedControl) then exit;

```

```

SelectedControl.Alignment:= TControl(Sender).Tag;
end;

procedure tmyRepEditor.sbFontColorClick(Sender: TObject);
var i: integer;
begin
if not Assigned(SelectedControl) then exit;
with ColorDialog1 do
begin
case TControl(Sender).Tag of
0: Color:= SelectedControl.FColor;
1: Color:= SelectedControl.BkColor;
else Color:= SelectedControl.BvColor;
end;

if Execute then
for i:=0 to ScrollBox.ControlCount-1 do
if ScrollBox.Controls[i] is tmyReportItem then with
tmyReportItem(ScrollBox.Controls[i]) do
if tmyReportItem(ScrollBox.Controls[i]).Selected then
case TControl(Sender).Tag of
0: tmyReportItem(ScrollBox.Controls[i]).FColor:= Color;
1: tmyReportItem(ScrollBox.Controls[i]).BkColor:= Color;
else tmyReportItem(ScrollBox.Controls[i]).BvColor:= Color;
end;
end;
end;

procedure tmyRepEditor.N_DeleteObjectClick(Sender: TObject);
begin
if Assigned(SelectedControl) then
begin
if Windows.MessageBox(0, 'Delete object?', 'Confirm', MB_OKCANCEL ) <> IDOK then
exit;

if SelectedControl.ContainOLE then
ScrollBox.RemoveControl( SelectedControl.OLEContainer );
ScrollBox.RemoveControl( SelectedControl );
SelectedControl.Free;
SelectedControl:= nil;
RemakeComponentsList;
end;
end;

procedure tmyRepEditor.OnDrawScrollBox(Sender: TObject);
begin
VRuler.Top:= ShapeSize.Top;
HRuler.Left:= ShapeSize.Left + P_VRuler.Width;
end;

procedure tmyRepEditor.RemakeComponentsList;
var i: integer;
begin
cb_Components.Items.Clear;
for i:= 0 to ScrollBox.ControlCount-1 do
if ScrollBox.Controls[i] is tmyReportItem then
cb_Components.Items.Add( tmyReportItem(ScrollBox.Controls[i]).CompName );
cb_Components.Text:= '';
lb_Params.Items.Clear;
for i:=0 to Component.ParamNames.Count-1 do
lb_Params.Items.Add( Component.ParamNames[i] );
end;

procedure tmyRepEditor.UpdatePageSize;
const
Sizes:array[boolean,1..2] of integer = ((21,29),(29,21));
begin
ShapeSize.Width:=round(Sizes[Printer.Orientation=
poLandscape][1]*GetDeviceCaps(Canvas.Handle,LOGPIXELSX)* 1.541*2.54/10);

```

```

ShapeSize.Height:=round( Sizes[Printer.Orientation=poLandscape][2]
*GetDeviceCaps(Canvas.Handle,LOGPIXELSY)*1.541*2.54/10);
HRuler.Width:=ShapeSize.Width+10;
VRuler.Height:=ShapeSize.Height+10;
end;

procedure tmyRepEditor.FormKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
var
l, t, w, h: integer;
begin
w:= 0; h:= 0;
if (Shift = [ssCtrl])and(chr(Key)='Z')and fCanUndo then
begin
fCanUndo:=false;
ResizeReportControls(UndoPosShift.X, UndoPosShift.Y,0,0,true);
end;

if Assigned(ActiveControl) then exit;
l:=0; t:=0;
case Key of
VK_UP: if Shift = [ssShift] then h:= -1 else if Shift = [ssCtrl] then t:= -1;
VK_DOWN: if Shift = [ssShift] then h:= 1 else if Shift = [ssCtrl] then t:= 1;
VK_LEFT: if Shift = [ssShift] then w:= -1 else if Shift = [ssCtrl] then l:= -1;
VK_RIGHT: if Shift = [ssShift] then w:= 1 else if Shift = [ssCtrl] then l:= 1;
else exit;
end;
ResizeReportControls( l, t, w, h, true );

end;

procedure tmyRepEditor.sb_FixMovingClick(Sender: TObject);
var i: integer;
begin
with ScrollBox do
for i:=0 to ControlCount-1 do
if (Controls[i] is tmyReportItem)and tmyReportItem(Controls[i]).Selected then
tmyReportItem(Controls[i]).Fixed:= integer(sb_FixMoving.Down);
end;

procedure tmyRepEditor.N_AutoSizeClick(Sender: TObject);
begin
SelectedControl.OLESizeMode:= TMenuItem(Sender).Tag;
end;

procedure tmyRepEditor.sb_SnapToGridClick(Sender: TObject);
begin
if sb_SnapToGrid.Down then
begin Step.X:= Grid.X; Step.Y:= Grid.Y; end else
begin Step.X:= 1; Step.Y:= 1; end;
end;

procedure tmyRepEditor.se_SizeChange(Sender: TObject);
begin
if (not fMouseDown)and not fSkipSizeUpdate then
ResizeReportControls( se_Left.Value, IGNORE_VALUE,
IGNORE_VALUE, IGNORE_VALUE,
false{fUseParamsAsShifts} );
ShowComponentPos(SelectedControl);
fSkipSizeUpdate:= false;
end;

procedure tmyRepEditor.se_TopChange(Sender: TObject);
begin
if (not fMouseDown)and not fSkipSizeUpdate then
ResizeReportControls( IGNORE_VALUE, se_Top.Value,
IGNORE_VALUE, IGNORE_VALUE,
false{fUseParamsAsShifts} );
ShowComponentPos(SelectedControl);

```

```

fSkipSizeUpdate:= false;
end;

procedure tmyRepEditor.se_WidthChange(Sender: TObject);
begin
if (not fMouseDown)and not fSkipSizeUpdate then
ResizeReportControls( IGNORE_VALUE, IGNORE_VALUE,
  se_Width.Value, IGNORE_VALUE,
  false{fUseParamsAsShifts} );
ShowComponentPos(SelectedControl);
fSkipSizeUpdate:= false;
end;

procedure tmyRepEditor.se_HeightChange(Sender: TObject);
begin
if (not fMouseDown)and not fSkipSizeUpdate then
ResizeReportControls( IGNORE_VALUE, IGNORE_VALUE,
  IGNORE_VALUE, se_Height.Value,
  false{fUseParamsAsShifts} );
ShowComponentPos(SelectedControl);
fSkipSizeUpdate:= false;
end;

procedure tmyRepEditor.ResizeReportControls( l, t, w, h: integer;
fUseParamsAsShifts: boolean );
var i: integer;
begin
with ScrollBox do
for i:=0 to ControlCount-1 do
if (Controls[i] is tmyReportItem) then with tmyReportItem(Controls[i]) do
if Selected and not bool(Fixed) then
if fUseParamsAsShifts then
begin
if l < IGNORE_VALUE then Left:= Left + l; if t < IGNORE_VALUE then Top:= Top +
t;
if w < IGNORE_VALUE then Width:= Width + w; if h < IGNORE_VALUE then Height:=
Height + h;
end else
begin
if l < IGNORE_VALUE then Left:= l; if t < IGNORE_VALUE then Top:= t;
if w < IGNORE_VALUE then Width:= w; if h < IGNORE_VALUE then Height:= h;
end;
end;
end;

procedure tmyRepEditor.cb_ComponentsChange(Sender: TObject);
var i: integer;
begin
if Assigned(SelectedControl) then
if SelectedControl.CompName = cb_Components.Text then exit;
with ScrollBox do
for i:=0 to ControlCount-1 do
if (Controls[i] is tmyReportItem) then
if tmyReportItem(Controls[i]).CompName = cb_Components.Text then
begin
OnMouseDown_( Controls[i], mbLeft, [], 0, 0 );
OnMouseUp_( Controls[i], mbLeft, [], 0, 0 );
exit;
end;
end;

procedure tmyRepEditor.ShowComponentPos(Control: TControl);
begin
if Component = nil then exit;
se_Left.Value:= Control.Left;
se_Top.Value:= Control.Top;
se_Width.Value:= Control.Width;
se_Height.Value:= Control.Height;
end;

```

```

procedure tmyRepEditor.AssignEventsToAllComponents;
var i: integer;
begin
with Component.ParentWnd do
for i:=0 to ControlCount-1 do
if (Controls[i] is tmyReportItem) then with tmyReportItem(Controls[i]) do
begin
OnMouseDown:= OnMouseDown_;
OnMouseUp:= OnMouseUp_;
OnMouseMove:= OnMouseMove_;
OnResize:= OnResize_;
end;
end;

function CanAlignControl(Control: TControl): boolean;
begin
Result:= (Control is tmyReportItem) and (bool(tmyReportItem(Control).Selected))
and(not bool(tmyReportItem(Control).Fixed));
end;

procedure tmyRepEditor.N4Click(Sender: TObject);
begin
if not Assigned(AlignForm) then AlignForm:= TAlignForm.Create(nil);
if AlignForm.ShowModal = mrOK then
AlignControlsInWindow( Component.ParentWnd, CanAlignControl, AlignForm.Horz,
AlignForm.Vert );
end;

procedure tmyRepEditor.sb_BevelBoldClick(Sender: TObject);
var i: integer;
begin
with Component.ParentWnd do
for i:=0 to ControlCount-1 do
if (Controls[i] is tmyReportItem) and bool(tmyReportItem(Controls[i]).Selected)
then
tmyReportItem(Controls[i]).PenWidth:= 1+integer(sb_BevelBold.Down);
end;

procedure tmyRepEditor.UpdateToolBar( Control: tmyReportItem);
begin
with Control do
begin
{
se_Left.Enabled:= Assigned(Control);
se_Top.Enabled:= Assigned(Control);
se_Width.Enabled:= Assigned(Control);
se_Height.Enabled:= Assigned(Control);}
P_Sides.Enabled:= Assigned(Control);
P_Font.Enabled:= Assigned(Control);
P_SBar.Enabled:= Assigned(Control);
if not Assigned(Control) then exit;
Edit1.Text:= Text;
Memo1.Text:= Text;
FontComboBox1.Text:= FName;
RxSpinEdit1.Value:= FSize;
ColorComboBox1.Color:= FColor;

SB_Left.Down := SideLeft = 1;
SB_Top.Down := SideTop = 1;
SB_Right.Down:= SideRight = 1;
SB_Bottom.Down:= SideBottom = 1;
case Alignment of
1: sb_AlignL.Down:= true;
2: sb_AlignR.Down:= true;
3: sb_AlignC.Down:= true;
4: sb_AlignW.Down:= true;
end;
sb_FixMoving.Down:= bool(Fixed);

```

```

sb_FontBold.Down:= boolean(FStyle and 1);
sb_FontItalic.Down:= boolean(FStyle and 2);
sb_FontUnderline.Down:= boolean(FStyle and 4);
FE_OLE.Text:= OLELinkToFile;
FE_OLE.Enabled:= ContainOLE;
sb_BevelBold.Down:= bool(PenWidth-1);
fSkipSizeUpdate:= true;
ShowComponentPos( Control );

cb_Components.Text:= CompName;

end;
end;

procedure tmyRepEditor.FormClose(Sender: TObject; var Action: TCloseAction);
var msS, msT: TMemoryStream;
begin
if Assigned(AlignForm) then AlignForm.Free;
if Assigned(ReportParamEditor) then ReportParamEditor.Free;
ScrollBar.HorzScrollBar.Position:= 0;
ScrollBar.VertScrollBar.Position:= 0;
Component.Save;
end;

procedure tmyRepEditor.FE_OLEChange(Sender: TObject);
var
str: string;
begin
if (not Assigned(SelectedControl))or (not FileExists(FE_OLE.Text)) then exit;
str:= FE_OLE.Text;
if ExtractFilePath(Name) = ExtractFilePath(ParamStr(0)) then
str:= ExtractFileName(Name);
if SelectedControl.OLELinkToFile <> str then
SelectedControl.OLELinkToFile:= str;
end;

procedure tmyRepEditor.SpeedButton1Click(Sender: TObject);
begin
if OpenOLEFile.Execute then FE_OLE.Text:= OpenOLEFile.FileName;
end;

procedure tmyRepEditor.sb_PrintClick(Sender: TObject);
begin
if Assigned(Component)and (Component.ComponentList.Count > 0) then
Component.Print;
Component.OwnerWnd:= self;
Component.ParentWnd:= ScrollBox;
end;

procedure tmyRepEditor.sb_PreviewClick(Sender: TObject);
var
Form: TForm;
Image: TImage;
bmp: TBitmap;
R: TRect;
i, W, H: integer;
begin
if not Assigned(Component) then exit;
Form:= TForm.Create(nil);
Form.Caption:= 'Page Preview';
Image:= TImage.Create(Form);
bmp:= TBitmap.Create;
Image.Parent:= Form;
H:= SantimsToPixels( Form.Canvas.Handle, 29, true );
W:= SantimsToPixels( Form.Canvas.Handle, 21, false );
// Image.Width:= W+8;
Image.Left:= 0; Image.Top:= 0;
bmp.Width:= W+7;
bmp.Height:= H+7;

```

```
try
```

```
with Component do
for i:=0 to ComponentList.Count-1 do with tmyReportItem(ComponentList[i]) do
begin
PaintTo bmp.Canvas);
if ContainOle then OLEContainer.PaintTo( bmp.Canvas.Handle, Left, Top );
end;
```

```
bmp.Canvas.Brush.Color:= clBtnFace;
R:= Bounds(bmp.Width-7, 0, 7, bmp.Height-7);
bmp.Canvas.FillRect( R );
R:= Bounds(0, bmp.Height-7, bmp.Width-7, 7);
bmp.Canvas.FillRect( R );
bmp.Canvas.Brush.Color:= 0;
R:= Bounds(bmp.Width-7, 7, 7, bmp.Height-7);
bmp.Canvas.FillRect( R );
R:= Bounds(7, bmp.Height-7, bmp.Width-7, 7);
bmp.Canvas.FillRect( R );
```

```
Image.Picture.bitmap:= bmp;
Image.Stretch:= true;
Image.Width:= W div 2;
Image.Height:= H div 2;
Form.ClientWidth:= Image.Width;
Form.ClientHeight:= Image.Height;
```

```
bmp.Free; bmp:= nil;
Form.ShowModal;
finally
if Assigned bmp then bmp.Free;
Form.Free;
end;
end;
```

```
procedure tmyRepEditor.SpeedButton2Click(Sender: TObject);
begin
if not Assigned(ReportParamEditor) then
ReportParamEditor:= TReportParamEditor.Create(nil);
ReportParamEditor.ShowModal;
end;
```

```
procedure tmyRepEditor.cb_ComponentsKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if (Key=VK_RETURN) then
begin
SelectedControl.CompName:= trim(cb_Components.Text);
RemakeComponentsList;
cb_Components.Text:= SelectedControl.CompName;
end;
if (Key=VK_ESCAPE) then cb_Components.Text:= SelectedControl.CompName;
end;
```

```
procedure tmyRepEditor.CheckBox1Click(Sender: TObject);
begin
if Assigned(SelectedControl) then SelectedControl.Transparent:=
integer(TCheckBox(Sender).Checked);
end;
```

```
end.
```

**Файл форми виведення графічного представлення  
(fset.pas)**

```

unit SET;

interface

uses Windows, Messages, Classes, Forms, dialogs;

///Розробив Смутко Віталій Олександрович, ЦНТУ, 2023

type

TEMUL_DEVELOPCOMPUTINGSYSTEM = class ( TComponent )
private
FResult: boolean;
FFileName: string;
FOnTerminated: TNotifyEvent;
si: TStartupInfo;
public
pi: TProcessInformation;
function Run: boolean;
function Kill: boolean;
destructor Destroy; override;
published
property FileName: string read FFileName write FFileName;
property Result: boolean read FResult stored false;
property OnTerminated: TNotifyEvent read FOnTerminated write FOnTerminated;
end;

procedure Register;

implementation

procedure Register;
begin
RegisterComponents('Proba', [TEMUL_DEVELOPCOMPUTINGSYSTEM]);
end;

destructor TEMUL_DEVELOPCOMPUTINGSYSTEM.Destroy;
begin
Kill;
inherited;
end;

function TEMUL_DEVELOPCOMPUTINGSYSTEM.Run: boolean;
begin
GetStartupInfo(si);
si.wShowWindow:= SW_NORMAL;
FResult:= CreateProcess( PChar(FFileName), nil, nil, nil, false,
NORMAL_PRIORITY_CLASS, nil, nil, si, pi);
Run:= FResult;
if Result then
begin
while WaitForSingleObjectDevelopComputingSystem(pi.hProcess, 100) = WAIT_TIMEOUT do
Application.ProcessMessages;
if Assigned(OnTerminated) then OnTerminated(self);
end;
end;

function TEMUL_DEVELOPCOMPUTINGSYSTEM.Kill: boolean;
begin
if FResult {and(WaitForSingleObject(pi.hProcess, 100) <> WAIT_TIMEOUT)}
then Kill:= TerminateProcess( pi.hProcess, 0 ) else Kill:= false;
end;

end.

```

**Файл форми даних розробленої програми  
(data.pas)**

```

unit DATA;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  dsgnintf,
  StdCtrls, glPropCn, ComCtrls, ExtCtrls, TypInfo, Buttons;

type

  Ttdata_DevelopComputingSystem1 = class( TPropertyEditor )
    function GetAttributes: TPropertyAttributes; override;
    function GetValue: string; override;
    procedure Edit; override;
  end;

  Ttdata_DevelopComputingSystem2 = class(TComponentEditor)
    procedure ExecuteVerb(Index: Integer); override;
    function GetVerb(Index: Integer): string; override;
    function GetVerbCount: Integer; override;
  end;

  Ttdata_DevelopComputingSystem3 = class(TForm)
    lvAll: TListView;
    Label1: TLabel;
    Bevel1: TBevel;
    Bevel2: TBevel;
    Label2: TLabel;
    Bevel3: TBevel;
    Bevel4: TBevel;
    lvSel: TListView;
    pbAdd: TBitBtn;
    pbRemove: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    ImageList1: TImageList;
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure lvAllDb1Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure lvSelDb1Click(Sender: TObject);
    procedure lvSelChange(Sender: TObject; Item: TListItem;
      Change: TItemChange);
    procedure lvAllChange(Sender: TObject; Item: TListItem;
      Change: TItemChange);
  private
    ComponentList: TStringList;
  end;

type
  TDesigner = IDesigner;
  TFormDesigner = IFormDesigner;

var
  dataCompListEditor: Ttdata_DevelopComputingSystem3;

implementation
  {$R *.DFM}

procedure ShowCompListEditor(Designer: TDesigner; dataPropertyCenter:
  TdataPropertyCenter);
var

```

```

    Dialog:Ttdata_DevelopComputingSystem3;
    I:Integer;
begin
    Dialog:=Ttdata_DevelopComputingSystem3.Create( Application );
    Dialog.Component:=dataPropertyCenter;
    Dialog.ShowModal;
    Dialog.free;
end;

function Ttdata_DevelopComputingSystem1.GetAttributes: TPropertyAttributes;
begin
    Result:=[paDialog];
end;

function Ttdata_DevelopComputingSystem1.GetValue: string;
begin
    Result:= Format( '%s', [ GetPropType^.Name ] );
end;

procedure Ttdata_DevelopComputingSystem1.Edit;
begin
    ShowCompListEditor(Designer, dataPropertyCenter(GetComponent(0)));
    GetComponent(0).Owner.Name
end;

procedure Ttdata_DevelopComputingSystem2.ExecuteVerb(Index: Integer);
begin
    case Index of
        0: ShowCompListEditor(Designer, dataPropertyCenter(Component));
    end;
end;

function Ttdata_DevelopComputingSystem2.GetVerbCount: Integer;
begin
    Result:= 1;
end;

procedure Ttdata_DevelopComputingSystem3.FormCreate(Sender: TObject);
begin
    ControlsList:= TList.create;
end;

procedure Ttdata_DevelopComputingSystem3.FormDestroy(Sender: TObject);
begin
    ControlsList.Free;
end;

procedure Ttdata_DevelopComputingSystem3.FormShow(Sender: TObject);
var
    i, j: integer;
    ListItem: TListItem;
    Comp: TComponent;
    ColorPropInfo, FontPropInfo: PPropInfo;
const
    aSigns: array [boolean] of string = ('-', '+');
begin
    lvAll.Items.Clear;
    lvSel.Items.Clear;

    for i:= 0 to Component.ComponentList.Count-1 do
        begin
            ListItem:= lvSel.Items.Add;
            ListItem.Caption:= Component.ComponentList[i];
            Comp:= Component.Owner.FindComponent( Component.ComponentList[i] );
            if Comp = nil then continue;
            ColorPropInfo:= GetPropInfo( Comp.ClassInfo, 'Color');
            FontPropInfo:= GetPropInfo( Comp.ClassInfo, 'Font');
            ListItem.SubItems.Add(aSigns[Assigned(ColorPropInfo)]);
            ListItem.SubItems.Add(aSigns[Assigned(FontPropInfo)]);
        end;
    end;
end;

```

```

    ListItem.ImageIndex:= 1;
end;

with Component.Owner do
for i:=0 to ComponentCount-1 do
begin
    ColorPropInfo:= GetPropInfo( Components[i].ClassInfo, 'Color');
    FontPropInfo:= GetPropInfo( Components[i].ClassInfo, 'Font');
    if (ColorPropInfo <> nil)or(FontPropInfo <> nil) then
        begin
            ListItem:= lvAll.Items.Add;
            ListItem.Caption:= Components[i].Name;
            ListItem.SubItems.Add(aSigns[Assigned(ColorPropInfo)]);
            ListItem.SubItems.Add(aSigns[Assigned(FontPropInfo)]);

            for j:=0 to lvSel.Items.Count - 1 do
                if lvSel.Items[j].Caption = ListItem.Caption then
                    begin ListItem.ImageIndex:= 1; break; end;

            end;
            SetOrdProp( FormX.Components[i], PropInfo, clGreen );
        end;
end;

procedure Ttdata_DevelopComputingSystem3.lvAllDbClick(Sender: TObject);
var ListItem: TListItem;
begin
    if (lvAll.Selected = nil)or(lvAll.Selected.ImageIndex = 1) then exit;

    ListItem:= lvSel.Items.Add;
    ListItem.Caption:= lvAll.Selected.Caption;
    ListItem.SubItems.Add(lvAll.Selected.SubItems[0]);
    ListItem.SubItems.Add(lvAll.Selected.SubItems[1]);
    lvAll.Selected.ImageIndex:= 1;
    ListItem.ImageIndex:= 1;
end;

procedure Ttdata_DevelopComputingSystem3.BitBtn4Click(Sender: TObject);
begin close; end;

procedure Ttdata_DevelopComputingSystem3.BitBtn3Click(Sender: TObject);
var
    i:integer;
    Comp:TComponent;
begin
    Component.ComponentList.Clear;
    Component.CompList.Clear;
    for i:=0 to lvSel.Items.Count-1 do
        begin
            Comp:=Component.Owner.FindComponent(lvSel.Items[i].Caption);
            if Comp = nil then continue;
            Component.CompList.Add(Comp);
            Component.ComponentList.Add(lvSel.Items[i].Caption);
        end;
    close;
end;

procedure Ttdata_DevelopComputingSystem3.lvSelDbClick(Sender: TObject);
var i: integer;
begin
    if not Assigned(lvSel.Selected) then exit;
    for i:=0 to lvAll.Items.Count - 1 do
        if lvAll.Items[i].Caption = lvSel.Selected.Caption then break;
        lvAll.Items[i].ImageIndex:= 0;
        lvSel.Items.Delete(lvSel.Selected.Index);
    end;

procedure Ttdata_DevelopComputingSystem3.lvSelChange(Sender:
TObject;Item:TListItem;Change:TItemChange);

```

```
begin
  pbRemove.Enabled:=Assigned(lvSel.Selected);
end;

procedure Ttdata_DevelopComputingSystem3.lvAllChange(Sender: TObject; Item:
TListItem; Change: TItemChange);
begin
  pbAdd.Enabled:=Assigned(lvAll.Selected) and (lvAll.Selected.ImageIndex=0);
end;

end.
```

К6П3-2023

**Файл форми роботи програмної моделі  
(DevelopComputingSystem.pas)**

```

unit DevelopComputingSystem;
///Розробив Смутко Віталій Олександрович, ЦНТУ, 2023
interface
uses Windows,Graphics,Controls,Classes,ExtCtrls,glTypes,SysUtils;

type
TTwainColors = class; Tem_DevelopComputingSystem_CustomLabelColors = class;
Tem_DevelopComputingSystem_LabelColors = class; TCustomGradient = class;
TGradient = class; TThreeDGradient = class;
T2DAlign = class; Tem_DevelopComputingSystem_CustomBoxStyle= class;
Tem_DevelopComputingSystem_CustomTextBoxStyle= class;
Tem_DevelopComputingSystem_TextBoxStyle = class;
TPointClass = class; Tem_DevelopComputingSystem_Bevel = class;
Tem_DevelopComputingSystem_ExtBevel = class; TILLUMINATION = class;
Tem_DevelopComputingSystem_LabelTextStyles = class;
Tem_DevelopComputingSystem_CustomTextColors = class;
Tem_DevelopComputingSystem_SimleLabelColors = class;
Tem_DevelopComputingSystem_BevelLines = class; TTwainColors = class(TPersistent)
Tem_DevelopComputingSystem_GrBoxColors = class;
Tem_DevelopComputingSystem_ListBoxItemStyle = class;
Tem_DevelopComputingSystem_AskListBoxItemStyle= class;
private
  FFromColor:TColor; FToColor:TColor;
  procedure SeTem_DevelopComputingSystem_omColor( Value:TColor );
  procedure SetToColor( Value:TColor );
public
  FRGBFromColor:Longint;
  FRGBToColor:Longint;
  OnChanged:TNotifyEvent;
  constructor Create; virtual;
published
  property FromColor:TColor read FFromColor write
SeTem_DevelopComputingSystem_omColor default $00808080;
  property ToColor:TColor read FToColor write SetToColor
default 0;
end;

TCustomGradient = class(TTwainColors)
private
  FBufferedDraw: boolean;
  FSteps: integer;
  FPercentFilling: Tem_DevelopComputingSystem_Percent;
  FBrushStyle: TBrushStyle;
procedure SetActive( Value:boolean );
procedure SetOrientation( Value:Tem_DevelopComputingSystem_GradientDir );
procedure SetSteps( Value: integer );
procedure SetPercentFilling( Value: Tem_DevelopComputingSystem_Percent );
procedure SetBrushStyle( Value: TBrushStyle );
public
FOrientation: Tem_DevelopComputingSystem_GradientDir;
FActive: boolean;
fReverse: boolean;
procedure TextOut(DC:HDC;Str:string; TextR:TRect; x, y:integer);
function GetColorFromGradientLine( GradientLineWidth, Position: word ):
COLORREF;
constructor Create; override;
destructor Destroy;override;
protected
property Active:boolean read FActive write SetActive;
property BufferedDraw:boolean read FBufferedDraw write FBufferedDraw
default false;
property Orientation:Tem_DevelopComputingSystem_GradientDir read FOrientation
write SetOrientation;
property Steps: integer read FSteps write SetSteps default 255;
property PercentFilling: Tem_DevelopComputingSystem_Percent read FPercentFilling
write SetPercentFilling

```

```

default 100;
property BrushStyle: TBrushStyle read FBrushStyle write SetBrushStyle
default bsSolid;
end;
TGradient = class(TCustomGradient)
private
public
procedure Draw(DC: HDC; r: TRect; PenStyle, PenWidth: integer);
published
property Active;
property BufferedDraw;
property Orientation;
property Steps;
property PercentFilling;
property BrushStyle;
end;
TThreeDGradient = class(TCustomGradient)
private
FDepth:word;
FGType:TThreeDGradientType;
procedure SetDepth(Value: word);
procedure SetGType(Value: TThreeDGradientType);
public
constructor Create; override;
published
property Depth: word
read FDepth write SetDepth default 16;
property GType:TThreeDGradientType read FGType write SetGType default fgtFlat;
end;
T2DAlign = class(TPersistent)
private
FHorizontal: Tem_DevelopComputingSystem_HorAlign;
FVertical: Tem_DevelopComputingSystem_VertAlign;
procedure SetHorizontal(Value: Tem_DevelopComputingSystem_HorAlign);
procedure SetVertical(Value: Tem_DevelopComputingSystem_VertAlign);
public
OnChange:TNotifyEvent;
constructor Create;
published
property Horizontal:Tem_DevelopComputingSystem_HorAlign read FHorizontal write
SetHorizontal
default fhaLeft;
property Vertical:Tem_DevelopComputingSystem_VertAlign read FVertical write
SetVertical
default fvaTop;
end;
TPointClass = class(TPersistent)
private
FX: integer;
FY: integer;
procedure SetX(Value: integer);
procedure SetY(Value: integer);
public
OnChange:TNotifyEvent;
published
property X:integer read FX write SetX;
property Y:integer read FY write SetY;
end;
Tem_DevelopComputingSystem_Bevel = class(TPersistent)
private
FInner: TPanelBevel;
FOuter: TPanelBevel;
FSides: Tem_DevelopComputingSystem_Sides;
FBold: boolean;
procedure SetInner(Value: TPanelBevel);
procedure SetOuter(Value: TPanelBevel);
procedure SetSides(Value: Tem_DevelopComputingSystem_Sides);
procedure SetBold(Value: boolean);
public

```

```

OnChanged:TNotifyEvent;
constructor Create; virtual;
function BordersHeight: integer;
function BordersWidth: integer;
published
property Inner: TPanelBevel read FInner write SetInner stored true;// default
bvLowered;
property Outer: TPanelBevel read FOuter write SetOuter stored true;// default
bvNone;
property Sides: Tem_DevelopComputingSystem_Sides read FSides write SetSides
stored true default ALLGLSIDES;
property Bold: boolean read FBold write SetBold stored true;// default false;
end;
Tem_DevelopComputingSystem_ExtBevel = class(Tem_DevelopComputingSystem_Bevel)
private
FActive: boolean;
FBevelPenStyle: TPenStyle;
FBevelPenWidth: word;
FInteriorOffset: word;
procedure SetActive(Value: boolean);
procedure SetBevelPenStyle(Value: TPenStyle);
procedure SetBevelPenWidth(Value: word);
procedure SetInteriorOffset(Value: word);
public
constructor Create; override;
published
property Active: boolean read FActive write SetActive
default true;
property BevelPenStyle: TPenStyle read FBevelPenStyle write SetBevelPenStyle
default psSolid;
property BevelPenWidth: word read FBevelPenWidth write SetBevelPenWidth
default 1;
property InteriorOffset: word read FInteriorOffset write SetInteriorOffset
default 0;
end;
TILLUMINATION = class(T2DAlign)
private
procedure SetShadowDepth(Value: integer);
public
FShadowDepth: integer;
OnChanged:TNotifyEvent;
constructor Create;
published
property ShadowDepth: integer
read FShadowDepth write SetShadowDepth default 2;
end;
Tem_DevelopComputingSystem_LabelTextStyles = class(TPersistent)
private
FPassive: Tem_DevelopComputingSystem_TextStyle;
FActive: Tem_DevelopComputingSystem_TextStyle;
FDisabled: Tem_DevelopComputingSystem_TextStyle;
procedure SetPassive(Value: Tem_DevelopComputingSystem_TextStyle);
procedure SetActive(Value: Tem_DevelopComputingSystem_TextStyle);
procedure SetDisabled(Value: Tem_DevelopComputingSystem_TextStyle);
public
OnChanged:TNotifyEvent;
constructor Create;
published
property Passive: Tem_DevelopComputingSystem_TextStyle read FPassive write
SetPassive
default fstRaised;
property Active: Tem_DevelopComputingSystem_TextStyle read FActive write
SetActive
default fstRaised;
property Disabled: Tem_DevelopComputingSystem_TextStyle read FDisabled write
SetDisabled
default fstPushed;
end;
Tem_DevelopComputingSystem_CustomTextColors = class(TPersistent)

```

```

private
FText: TColor;
FTextDisabled: TColor;
FDelineate: TColor;
FBackground: TColor;
public
FHighlight: TColor;
FShadow: TColor;
private
procedure SetText(Value: TColor);
procedure SetTextDisabled(Value: TColor);
procedure SetDelineate(Value: TColor);
procedure SetBackground(Value: TColor);
procedure SetHighlight(Value: TColor);
procedure SetShadow(Value: TColor);
public
OnChange:TNotifyEvent;
constructor Create; virtual;
protected
property Text: TColor read FText write SetText
default clBlack;
property TextDisabled: TColor read FTextDisabled write SetTextDisabled
default clGray;
property Delineate: TColor read FDelineate write SetDelineate
default clWhite;
property Shadow:TColor read FShadow write SetShadow default clBtnShadow;
property Highlight:TColor read FHighlight write SetHighlight default
clBtnHighlight;
property Background:TColor read FBackground write SetBackground default
clBtnFace;
end;
Tem_DevelopComputingSystem_SimleLabelColors =
class(Tem_DevelopComputingSystem_CustomTextColors)
published
property Text stored true;
property Delineate stored true;
property Shadow stored true;
property Highlight;
property Background stored true;
end;
Tem_DevelopComputingSystem_CustomLabelColors =
class(Tem_DevelopComputingSystem_CustomTextColors)
private
FTextActive: TColor;
FDelineateActive: TColor;
FAutoHighlight: boolean;
FAutoShadow: boolean;
FBackgroundActive: TColor;
public
FColorHighlightShift: integer;
FColorShadowShift: integer;
private
procedure SetTextActive(Value: TColor);
procedure SetDelineateActive(Value: TColor);
procedure SetBackgroundActive(Value: TColor);
procedure SetAutoHighlight(Value: boolean);
procedure SetAutoShadow(Value: boolean);
procedure SetColorHighlightShift(Value: integer);
procedure SetColorShadowShift(Value: integer);
public
OnChange:TNotifyEvent;
constructor Create; override;
protected
property TextActive: TColor read FTextActive write SetTextActive
default clBlack;
property DelineateActive: TColor read FDelineateActive write SetDelineateActive
default clWhite;
property AutoHighlight: boolean
read FAutoHighlight write SetAutoHighlight default false;

```

```

property AutoShadow: boolean
read FAutoShadow write SetAutoShadow default false;
property ColorHighlightShift: integer
read FColorHighlightShift write SetColorHighlightShift default 40;
property ColorShadowShift: integer
read FColorShadowShift write SetColorShadowShift default 60;
property BackgroundActive: TColor
read FBackgroundActive write SetBackgroundActive default clBtnFace;
end;
Tem_DevelopComputingSystem_LabelColors =
class(Tem_DevelopComputingSystem_CustomLabelColors)
published
property Text;
property TextDisabled;
property Delineate;
property Shadow;
property Highlight;
property Background;
property TextActive;
property DelineateActive;
property AutoHighlight;
property AutoShadow;
property ColorHighlightShift;
property ColorShadowShift;
property BackgroundActive;
end;
Tem_DevelopComputingSystem_GrBoxColors =
class(Tem_DevelopComputingSystem_CustomLabelColors)
private
FCaption: TColor;
FCaptionActive: TColor;
FClient: TColor;
FClientActive: TColor;
procedure SetCaption(Value: TColor);
procedure SetCaptionActive(Value: TColor);
procedure SetClient(Value: TColor);
procedure SetClientActive(Value: TColor);
public
constructor Create; override;
published
property Text;
property Delineate;
property Shadow;
property Highlight;
property Background;
property TextActive;
property DelineateActive;
property AutoHighlight;
property AutoShadow;
property ColorHighlightShift;
property ColorShadowShift;
property BackgroundActive;
property Caption: TColor read FCaption write SetCaption;
property CaptionActive: TColor read FCaptionActive write SetCaptionActive;
property Client: TColor read FClient write SetClient;
property ClientActive: TColor read FClientActive write SetClientActive;
end;
Tem_DevelopComputingSystem_CustomListBoxItemStyle = class(TPersistent)
private
FColor: TColor;
FDelineateColor: TColor;
FFont: TFont;
FBevel: Tem_DevelopComputingSystem_Bevel;
FTextStyle: Tem_DevelopComputingSystem_TextStyle;
FOnChanged: TNotifyEvent;
procedure SetColor(Value: TColor);
procedure SetDelineateColor(Value: TColor);
procedure SetFont(Value: TFont);
procedure SetTextStyle(Value: Tem_DevelopComputingSystem_TextStyle);

```

```

protected
procedure SetOnChanged(Value: TNotifyEvent); virtual;
public
property OnChanged: TNotifyEvent read FOnChanged write SetOnChanged;
constructor Create; virtual;
destructor Destroy; override;
function HighlightColor: TColor;
function ShadowColor: TColor;
published
property Color: TColor read FColor write SetColor;
property DelineateColor: TColor read FDelineateColor write SetDelineateColor;
property Font: TFont read FFont write SetFont;
property Bevel: Tem_DevelopComputingSystem_Bevel read FBevel write FBevel;
property TextStyle: Tem_DevelopComputingSystem_TextStyle read FTextStyle write
SetTextStyle;
end;
Tem_DevelopComputingSystem_ListBoxItemStyle=class(Tem_DevelopComputingSystem_Cus
tomListBoxItemStyle)
private
FGradient: TGradient;
FTextGradient: TGradient;
protected
property TextGradient: TGradient read FTextGradient write FTextGradient;
procedure SetOnChanged(Value: TNotifyEvent); override;
public
constructor Create; override;
destructor Destroy; override;
published
property Gradient: TGradient read FGradient write FGradient;
end;

Tem_DevelopComputingSystem_HintStyle=class(Tem_DevelopComputingSystem_ListBoxIte
mStyle)
end;

Tem_DevelopComputingSystem_SpeedButtonStyle=class(Tem_DevelopComputingSystem_Lis
tBoxItemStyle)
published
property TextGradient;
end;
Tem_DevelopComputingSystem_AskListBoxItemStyle=
class(Tem_DevelopComputingSystem_CustomListBoxItemStyle)
private
FBtnColor: TColor;
FBtnFont: TFont;
FBtnTextStyle: Tem_DevelopComputingSystem_TextStyle;
procedure SetBtnColor(Value: TColor);
procedure SetBtnFont(Value: TFont);
procedure SetBtnTextStyle(Value: Tem_DevelopComputingSystem_TextStyle);
public
constructor Create; override;
destructor Destroy; override;
published
property BtnColor: TColor read FBtnColor write SetBtnColor;
property BtnFont: TFont read FBtnFont write SetBtnFont;
property BtnTextStyle: Tem_DevelopComputingSystem_TextStyle read FBtnTextStyle
write SetBtnTextStyle;
end;
Tem_DevelopComputingSystem_CustomBoxStyle=
class(Tem_DevelopComputingSystem_Bevel)
private
FPenStyle: TPenStyle;
FHighlightColor: TColor;
FShadowColor: TColor;
procedure SetPenStyle(Value: TPenStyle);
procedure SetHighlightColor(Value: TColor);
procedure SetShadowColor(Value: TColor);
public
OnChanged: TNotifyEvent;

```

```

constructor Create; override;
protected
property PenStyle: TPenStyle read FPenStyle write SetPenStyle
default psSolid;
property HighlightColor: TColor read FHighlightColor write SetHighlightColor
default clBtnHighlight;
property ShadowColor: TColor read FShadowColor write SetShadowColor
default clBtnShadow;
end;
Tem_DevelopComputingSystem_CustomTextBoxStyle=
class(Tem_DevelopComputingSystem_CustomBoxStyle)
private
FTextColor: TColor;
FBackgroundColor: TColor;
procedure SetTextColor(Value: TColor);
procedure SetBackgroundColor(Value: TColor);
public
constructor Create; override;
protected
property TextColor: TColor read FTextColor write SetTextColor
default clBlack;
property BackgroundColor: TColor read FBackgroundColor write SetBackgroundColor
default clWindow;
end;
Tem_DevelopComputingSystem_TextBoxStyle =
class(Tem_DevelopComputingSystem_CustomTextBoxStyle)
published
property Inner;
property Outer;
property Sides;
property Bold;
property PenStyle;
property TextColor;
property BackgroundColor;
property HighlightColor;
property ShadowColor;
end;
Tem_DevelopComputingSystem_BevelLines = class(TPersistent)
private
FCount: cardinal;
FStep: cardinal;
FOrigin: Tem_DevelopComputingSystem_Origin;
FStyle: TPanelBevel;
FBold: boolean;
FThickness: byte;
FIgnoreBorder: boolean;

procedure SetCount(Value: cardinal);
procedure SetStep(Value: cardinal);
procedure SetOrigin(Value: Tem_DevelopComputingSystem_Origin);
procedure SetStyle(Value: TPanelBevel);
procedure SetBold(Value: boolean);
procedure SetThickness(Value: byte);
procedure SetIgnoreBorder(Value: boolean);
public
OnChange: TNotifyEvent;
constructor Create;
published
property Count: cardinal read FCount write SetCount
default 0;
property Step: cardinal read FStep write SetStep
default 0;
property Origin: Tem_DevelopComputingSystem_Origin read FOrigin write SetOrigin
default forLeftTop;
property Style: TPanelBevel read FStyle write SetStyle
default bvLowered;
property Bold: boolean read FBold write SetBold
default false;
property Thickness: byte read FThickness write SetThickness

```

```

default 1;
property IgnoreBorder: boolean read FIgnoreBorder write SetIgnoreBorder
default false;
end;
implementation

{$I Debug.inc}

constructor TTWainColors.Create;
begin
inherited Create;
FFromColor:= $00808080; FRGBFromColor:= ColorToRGB( FFromColor );
FToColor:= 0;FRGBToColor:= ColorToRGB( FToColor );
end;

procedure TTWainColors.SeTem_DevelopComputingSystem_omColor( Value:TColor );
begin
if FFromColor = Value then exit;
FFromColor:= Value; FRGBFromColor:= ColorToRGB( Value );
if Assigned(OnChanged) then OnChanged(self);
end;

procedure TTWainColors.SetToColor( Value:TColor );
begin
if FToColor = Value then exit;
FToColor:= Value; FRGBToColor:= ColorToRGB( Value );
if Assigned(OnChanged) then OnChanged(self);
end;
constructor TCustomGradient.Create;
begin
inherited Create;
FActive:= false;
FBufferedDraw:= false;
FOrientation:= fgdHorizontal;
FSteps:= 255;
FPercentFilling:= 100;
FBrushStyle:= bsSolid;
end;

destructor TCustomGradient.Destroy;
begin inherited Destroy; end;

procedure TCustomGradient.Free;
begin if self<>nil then Destroy; end;

procedure TCustomGradient.SetActive( Value:boolean );
begin
if FActive = Value then exit;
FActive:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure TCustomGradient.SetOrientation(
Value:Tem_DevelopComputingSystem_GradientDir );
begin
if FOrientation = Value then exit;
FOrientation:= Value;
if FActive and Assigned(OnChanged) then OnChanged(self);
end;

procedure TCustomGradient.SetSteps( Value: integer );
begin
if Value > 255 then Value:= 255
else if Value < 1 then Value:= 1;
if FSteps = Value then exit;
FSteps:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

```

```

procedure TCustomGradient.SetPercentFilling( Value:
Tem_DevelopComputingSystem_Percent );
begin
if FPercentFilling = Value then exit;
FPercentFilling:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure TCustomGradient.SetBrushStyle( Value: TBrushStyle );
begin
FBrushStyle:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

function TCustomGradient.GetColorFromGradientLine
( GradientLineWidth, Position: word ): COLORREF;
var
c1F,c2F,c3F:byte;
c1T,c2T,c3T:byte;
Step1,Step2,Step3:SinEm_DevelopComputingSystem;
begin
c1F:= Byte( self.FRGBFromColor );
c2F:= Byte( WORD(self.FRGBFromColor) shr 8);
c3F:= Byte( self.FRGBFromColor shr 16 );
c1T:= Byte( self.FRGBTToColor );
c2T:= Byte( WORD(self.FRGBTToColor) shr 8);
c3T:= Byte( self.FRGBTToColor shr 16 );

Step1:=(c1T-c1F)/GradientLineWidth;
Step2:=(c2T-c2F)/GradientLineWidth;
Step3:=(c3T-c3F)/GradientLineWidth;

Result:=RGB( trunc(c1F+Step1*Position),
trunc(c2F+Step2*Position),
trunc(c3F+Step3*Position) );
end;

procedure TCustomGradient.TextOut( DC: HDC; Str:string; TextR:TRect; x,
y:integer );
var
i,Steps:integer;
r:TRect;
c1F,c2F,c3F:byte;
c1T,c2T,c3T:byte;
c1,c2,c3:SinEm_DevelopComputingSystem;
OldTextColor:TCOLORREF;
begin
if (not Active)or(GetDeviceCaps( DC, BITSPIXEL )<16) then
begin Windows.TextOut( DC, x,y, PChar(str), Length(str)); exit; end;
r:= TextR;
c1F:= Byte( FRGBFromColor );
c2F:= Byte( WORD(FRGBFromColor) shr 8);
c3F:= Byte( FRGBFromColor shr 16 );
c1T:= Byte( FRGBTToColor );
c2T:= Byte( WORD(FRGBTToColor) shr 8);
c3T:= Byte( FRGBTToColor shr 16 );

c1:=c1F; c2:=c2F; c3:=c3F;
if FOrientation = fgdVertical then Steps:=r.right-r.left
else Steps:=r.bottom-r.top;
Step1:=(c1T-c1F)/Steps; Step2:=(c2T-c2F)/Steps; Step3:=(c3T-c3F)/Steps;

OldTextColor:= SetTextColor( DC, 0 );
Steps:= MulDiv( Steps, PercentFilling, 100 );
for i:=0 to Steps do
begin
SetTextColor( DC, RGB( trunc(c1),trunc(c2),trunc(c3)) );

```

```

if FOrientation = fgdVertical then
begin r.left:=i; r.right:=r.left+1; end
else
begin r.top:=i; r.bottom:=r.top+1; end;

Windows.ExtTextOut( DC, x, y, ETO_CLIPPED, @r,
PChar(str), Length(str), nil);
c1:= c1+ Step1; c2:= c2+ Step2; c3:= c3+ Step3;
end;
SetTextColor( DC, OldTextColor );
end;
constructor TThreeDGradient.Create;
begin
inherited Create;
Depth:=16;
FGType:=fgtFlat;
FActive:=true;
end;

procedure TThreeDGradient.SetGType(Value: TThreeDGradientType);
begin
if FGType = Value then exit;
FGType:= Value; if FActive and Assigned(OnChanged) then OnChanged(self);
end;

procedure TThreeDGradient.SetDepth(Value: word);
begin
if FDepth = Value then exit;
FDepth:= Value; if FActive and Assigned(OnChanged) then OnChanged(self);
end;

constructor T2DAlign.Create;
begin
inherited Create;
FHorizontal:= fhaLeft;
FVertical:= fvaTop;
end;

procedure T2DAlign.SetHorizontal(Value: Tem_DevelopComputingSystem_HorAlign);
begin
if FHorizontal = Value then exit; FHorizontal:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure T2DAlign.SetVertical(Value: Tem_DevelopComputingSystem_VertAlign);
begin
if FVertical = Value then exit; FVertical:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure TPointClass.SetX(Value: integer);
begin
if FX = Value then exit; FX:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure TPointClass.SetY(Value: integer);
begin
if FY = Value then exit; FY:= Value;
if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_Bevel.Create;
begin
inherited;
FSides:= ALLGLSIDES;
end;

procedure Tem_DevelopComputingSystem_Bevel.SetOuter(Value: TPanelBevel);

```

```

begin
if FOuter=Value then exit; FOuter:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_Bevel.SetInner(Value: TPanelBevel);
begin
if FInner=Value then exit; FInner:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_Bevel.SetSides(Value:
Tem_DevelopComputingSystem_Sides);
begin
if FSides=Value then exit; FSides:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_Bevel.SetBold(Value: boolean);
begin
if FBold=Value then exit; FBold:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tillumination.Create;
begin
inherited;
FShadowDepth:= 2;
end;

procedure Tillumination.SetShadowDepth(Value: integer);
begin
if Value < 0 then Value:=0;
if FShadowDepth=Value then exit; FShadowDepth:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_LabelTextStyles.Create;
begin
inherited;
FActive:= fstRaised;
FPassive:= fstRaised;
FDisabled:= fstPushed;
end;

procedure Tem_DevelopComputingSystem_LabelTextStyles.SetPassive(Value:
Tem_DevelopComputingSystem_TextStyle);
begin
if FPassive=Value then exit; FPassive:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_LabelTextStyles.SetActive(Value:
Tem_DevelopComputingSystem_TextStyle);
begin
if FActive=Value then exit; FActive:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_LabelTextStyles.SetDisabled(Value:
Tem_DevelopComputingSystem_TextStyle);
begin
if FDisabled=Value then exit; FDisabled:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_CustomTextColors.Create;
begin
inherited;

```

```

FText:= clBlack;
FTextDisabled:= clGray;
FDelineate:= clWhite;
FHighlight:= clBtnHighlight;
FShadow:= clBtnShadow;
FBackground:= clBtnFace;
end;

procedure Tem_DevelopComputingSystem_CustomTextColors.SetText(Value: TColor);
begin
if FText=Value then exit; FText:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomTextColors.SetTextDisabled(Value:
TColor);
begin
if FTextDisabled=Value then exit; FTextDisabled:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomTextColors.SetDelineate(Value:
TColor);
begin
if FDelineate=Value then exit; FDelineate:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomTextColors.SetHighlight(Value:
TColor);
begin
if FHighlight=Value then exit; FHighlight:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomTextColors.SetShadow(Value: TColor);
begin
if FShadow=Value then exit; FShadow:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomTextColors.SetBackground(Value:
TColor);
begin
if FBackground=Value then exit; FBackground:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_CustomLabelColors.Create;
begin
inherited;
FTextActive:= clBlack;
FDelineateActive:= clWhite;
FAutoHighlight:= false;
FAutoShadow:= false;
FColorHighlightShift:= 40;
FColorShadowShift:= 60;
FBackgroundActive:= clBtnFace;
end;

procedure Tem_DevelopComputingSystem_CustomLabelColors.SetTextActive(Value:
TColor);
begin
if FTextActive=Value then exit; FTextActive:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomLabelColors.SetDelineateActive(Value:
TColor);

```

```

begin
if FDelineateActive=Value then exit; FDelineateActive:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomLabelColors.SetAutoHighlight (Value:
boolean);
begin
if FAutoHighlight=Value then exit; FAutoHighlight:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomLabelColors.SetAutoShadow (Value:
boolean);
begin
if FAutoShadow=Value then exit; FAutoShadow:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure
Tem_DevelopComputingSystem_CustomLabelColors.SetColorHighlightShift (Value:
integer);
begin
if FColorHighlightShift=Value then exit; FColorHighlightShift:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure
Tem_DevelopComputingSystem_CustomLabelColors.SetColorShadowShift (Value:
integer);
begin
if FColorShadowShift=Value then exit; FColorShadowShift:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

procedure
Tem_DevelopComputingSystem_CustomLabelColors.SetBackgroundActive (Value: TColor);
begin
if FBackgroundActive=Value then exit; FBackgroundActive:=Value;
if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_GrBoxColors.Create;
begin
inherited;
FCaption:= clBtnFace;
FCaptionActive:= clBtnFace;
FClient:= clBtnFace;
FClientActive:= clBtnFace;
end;

procedure Tem_DevelopComputingSystem_GrBoxColors.SetCaption (Value: TColor);
begin FCaption:= Value; if Assigned(OnChanged) then OnChanged(self); end;

procedure Tem_DevelopComputingSystem_GrBoxColors.SetCaptionActive (Value:
TColor);
begin FCaptionActive:= Value; if Assigned(OnChanged) then OnChanged(self); end;

procedure Tem_DevelopComputingSystem_GrBoxColors.SetClient (Value: TColor);
begin FClient:= Value; if Assigned(OnChanged) then OnChanged(self); end;

procedure Tem_DevelopComputingSystem_GrBoxColors.SetClientActive (Value: TColor);
begin FClientActive:= Value; if Assigned(OnChanged) then OnChanged(self); end;

constructor Tem_DevelopComputingSystem_ExtBevel.Create;
begin
inherited;
FActive:= true;
FBevelPenStyle:= psSolid;

```

```

FBevelPenWidth:= 1;
end;

procedure Tem_DevelopComputingSystem_ExtBevel.SetActive(Value: boolean);
begin
if FActive = Value then exit;
FActive:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_ExtBevel.SetBevelPenStyle(Value:
TPenStyle);
begin
if FBevelPenStyle = Value then exit;
FBevelPenStyle:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_ExtBevel.SetBevelPenWidth(Value: word);
begin
if FBevelPenWidth = Value then exit;
FBevelPenWidth:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_ExtBevel.SetInteriorOffset(Value: word);
begin
if FInteriorOffset = Value then exit;
FInteriorOffset:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_CustomListBoxItemStyle.Create;
begin
inherited Create;
FBevel:= Tem_DevelopComputingSystem_Bevel.Create;
FFont:= TFont.Create;
end;

destructor Tem_DevelopComputingSystem_CustomListBoxItemStyle.Destroy;
begin
FFont.Free; FBevel.Free; inherited;
end;

procedure Tem_DevelopComputingSystem_CustomListBoxItemStyle.SetOnChanged(Value:
TNotifyEvent);
begin
FOnChanged:= Value; FBevel.OnChanged:= Value;
end;

procedure Tem_DevelopComputingSystem_CustomListBoxItemStyle.SetColor(Value:
TColor);
begin
if FColor = Value then exit;
FColor:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

procedure
Tem_DevelopComputingSystem_CustomListBoxItemStyle.SetDelineateColor(Value:
TColor);
begin
if FDelineateColor = Value then exit;
FDelineateColor:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomListBoxItemStyle.SetFont(Value:
TFont);
begin
FFont.Assign(Value); if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomListBoxItemStyle.SetTextStyle(Value:
Tem_DevelopComputingSystem_TextStyle);

```

```

begin
FTextStyle:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_ListBoxItemStyle.Create;
begin
inherited Create;
FGradient:= TGradient.Create;
FTextGradient:= TGradient.Create;
end;

destructor Tem_DevelopComputingSystem_ListBoxItemStyle.Destroy;
begin
FGradient.Free;
FTextGradient.Free;
inherited;
end;

procedure Tem_DevelopComputingSystem_ListBoxItemStyle.SetOnChanged(Value:
TNotifyEvent);
begin
inherited SetOnChanged(Value);
FGradient.OnChanged:= Value;
end;

constructor Tem_DevelopComputingSystem_AskListBoxItemStyle.Create;
begin
inherited Create;
FBtnFont:= TFont.Create;
end;

destructor Tem_DevelopComputingSystem_AskListBoxItemStyle.Destroy;
begin
FBtnFont.Free; inherited;
end;

procedure Tem_DevelopComputingSystem_AskListBoxItemStyle.SetBtnColor(Value:
TColor);
begin
if FBtnColor = Value then exit;
FBtnColor:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_AskListBoxItemStyle.SetBtnFont(Value:
TFont);
begin
FBtnFont.Assign(Value); if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_AskListBoxItemStyle.SetBtnTextStyle(Value:
Tem_DevelopComputingSystem_TextStyle);
begin
FBtnTextStyle:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_CustomBoxStyle.Create;
begin
inherited;
FPenStyle:= psSolid;
FHighlightColor:= clBtnHighlight;
FShadowColor:= clBtnShadow;
end;

procedure Tem_DevelopComputingSystem_CustomBoxStyle.SetPenStyle(Value:
TPenStyle);
begin FPenStyle:= Value; if Assigned(OnChanged) then OnChanged(self); end;

procedure Tem_DevelopComputingSystem_CustomBoxStyle.SetHighlightColor(Value:
TColor);

```

```

begin FHighlightColor:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

procedure Tem_DevelopComputingSystem_CustomBoxStyle.SetShadowColor(Value:
TColor);
begin FShadowColor:= Value; if Assigned(OnChanged) then OnChanged(self); end;

constructor Tem_DevelopComputingSystem_CustomTextBoxStyle.Create;
begin
inherited;
FTextColor:= clBlack;
FBackgroundColor:= clWindow;
end;

procedure Tem_DevelopComputingSystem_CustomTextBoxStyle.SetTextColor(Value:
TColor);
begin FTextColor:= Value; if Assigned(OnChanged) then OnChanged(self); end;

procedure
Tem_DevelopComputingSystem_CustomTextBoxStyle.SetBackgroundColor(Value: TColor);
begin FBackgroundColor:= Value; if Assigned(OnChanged) then OnChanged(self);
end;

constructor Tem_DevelopComputingSystem_BevelLines.Create;
begin
inherited;
FStyle:= bvLowered;
FThickness:= 1;
end;

procedure Tem_DevelopComputingSystem_BevelLines.SetCount(Value: cardinal);
begin FCount:= Value; if Assigned(OnChanged) then OnChanged(self); end;
procedure Tem_DevelopComputingSystem_BevelLines.SetStep(Value: cardinal);
begin FStep:= Value; if Assigned(OnChanged) then OnChanged(self); end;
procedure Tem_DevelopComputingSystem_BevelLines.SetOrigin(Value:
Tem_DevelopComputingSystem_Origin);
begin FOrigin:= Value; if Assigned(OnChanged) then OnChanged(self); end;
procedure Tem_DevelopComputingSystem_BevelLines.SetStyle(Value: TPanelBevel);
begin FStyle:= Value; if Assigned(OnChanged) then OnChanged(self); end;
procedure Tem_DevelopComputingSystem_BevelLines.SetBold(Value: boolean);
begin FBold:= Value; if Assigned(OnChanged) then OnChanged(self); end;
procedure Tem_DevelopComputingSystem_BevelLines.SetThickness(Value: byte);
begin FThickness:= Value; if Assigned(OnChanged) then OnChanged(self); end;
procedure Tem_DevelopComputingSystem_BevelLines.SetIgnoreBorder(Value: boolean);
begin FIgnoreBorder:= Value; if Assigned(OnChanged) then OnChanged(self); end;

{ TGradient }

procedure TGradient.Draw(DC: HDC; r: TRect; PenStyle, PenWidth: integer);
var
i, j, x, y, x2, y2, h, w, NumberOfColors: integer;
c1F, c2F, c3F: byte;
c1T, c2T, c3T: byte;
c1D, c2D, c3D: integer;
_R, _G, _B: byte;
Pen, OldPen: HPen;
FillBrush: HBrush;
BufferBmp, OldBMP: HBITMAP;
BufferDC, TargetDC: HDC;
ColorR: TRect;
LOGBRUSH: TLOGBRUSH;

procedure SwapColors;
var TempColor: Longint;
begin
TempColor:= FRGBFromColor; FRGBFromColor:= FRGBToColor; FRGBToColor:= TempColor;
end;
begin
if (Steps = 1) or (GetDeviceCaps( DC, BITSPIXEL ) < 16) then

```

```

begin exit;
FillBrush:= CreateSolidBrush( ColorToRGB( FromColor ) );
FillRect( DC, r, FillBrush );
DeleteObject( FillBrush );
exit;
end;
x:=r.left; y:=r.top; h:=r.bottom-r.top; w:=r.right-r.left;
x2:= 0; y2:= 0; pen:= 0; oldpen:= 0; BufferDC:= 0;

if Orientation = fgdHorzConvergent then
begin
FOrientation:= fgdHorizontal;
Draw(DC, Rect(R.Left, R.Top, R.Right, R.Bottom- h div 2), PenStyle, PenWidth);
SwapColors;
Draw(DC, Rect(R.Left, R.Top+ h div 2, R.Right, R.Bottom), PenStyle, PenWidth);
SwapColors;
FOrientation:= fgdHorzConvergent;
exit;
end;
if Orientation = fgdVertConvergent then
begin
FOrientation:= fgdVertical;
Draw(DC, Rect(R.Left, R.Top, R.Right- w div 2, R.Bottom), PenStyle, PenWidth);
SwapColors;
Draw(DC, Rect(R.Left+ w div 2, R.Top, R.Right, R.Bottom), PenStyle, PenWidth);
SwapColors;
FOrientation:= fgdVertConvergent;
exit;
end;

c1F:=Byte(FRGBFromColor );
c2F:=Byte(WORD(FRGBFromColor) shr 8);
c3F:=Byte(FRGBFromColor shr 16 );
c1T:=Byte(FRGBToColor );
c2T:=Byte(WORD(FRGBToColor) shr 8);
c3T:=Byte(FRGBToColor shr 16 );
c1D:=c1T- c1F;
c2D:=c2T- c2F;
c3D:=c3T- c3F;

if BufferedDraw then
begin
BufferDC:= CreateCompatibleDC(DC);
BufferBmp:= CreateBitmap( w, h, GetDeviceCaps( DC, PLANES ), GetDeviceCaps( DC,
BITSPIXEL ), nil );
OldBMP:= SelectObject( BufferDC, BufferBmp );
SetMapMode( BufferDC, GetMapMode(DC) );
TargetDC:= BufferDC;
end else TargetDC:= DC;

case Orientation of
fgdHorizontal:
begin
NumberOfColors:= min( Steps, h );
ColorR.Left:= r.left; ColorR.Right:= r.right;
end;
fgdVertical:
begin
NumberOfColors:= min( Steps, w );
ColorR.Top:= r.top; ColorR.Bottom:= r.bottom;
end;
fgdLeftBias, fgdRightBias:
begin
NumberOfColors:= min( Steps, w+h );
if PenStyle=0 then PenStyle:=PS_SOLID; if PenWidth=0 then PenWidth:=1;
y2:= y;
if Orientation = fgdLeftBias then x2:= x else
begin x:= r.right; x2:= r.right; end;
end;

```

```

else{fgdRectanEm_DevelopComputingSystem}
begin
h:=h div 2; w:=w div 2;
NumberOfColors:= min( Steps, min(w,h) );
end;
end;
LOGBRUSH.lbStyle:= BS_HATCHED;
LOGBRUSH.lbHatch:= Ord(BrushStyle)- Ord(bsHorizontal);
for i:=0 to NumberOfColors-1 do
begin
_R:= c1F+ MulDiv(i, c1D, NumberOfColors- 1);
_G:= c2F+ MulDiv(i, c2D, NumberOfColors- 1);
_B:= c3F+ MulDiv(i, c3D, NumberOfColors- 1);

case Orientation of
fgdHorizontal, fgdVertical, fgdRectanEm_DevelopComputingSystem:
begin
if BrushStyle = bsSOLID then
FillBrush:= CreateSolidBrush( RGB( _R, _G, _B ) )
else
begin
LOGBRUSH.lbColor:= RGB( _R, _G, _B );
FillBrush:= CreateBrushIndirect(LOGBRUSH);
end;

case Orientation of
fgdHorizontal:
begin
if fReverse then begin
ColorR.Top:= r.bottom- MulDiv( i, h, NumberOfColors);
ColorR.Bottom:= r.bottom- MulDiv( i+ 1, h, NumberOfColors);
end else begin
ColorR.Top:= r.top+ MulDiv( i, h, NumberOfColors);
ColorR.Bottom:= r.top+ MulDiv( i+ 1, h, NumberOfColors);
end;
end;
fgdVertical:
begin
if fReverse then begin
ColorR.Left:= r.right- MulDiv( i,w, NumberOfColors);
ColorR.Right:= r.right- MulDiv( i+ 1, w, NumberOfColors);
end else begin
ColorR.Left:= r.left+ MulDiv( i,w, NumberOfColors);
ColorR.Right:= r.left+ MulDiv( i+ 1, w, NumberOfColors);
end;
end;
fgdRectanEm_DevelopComputingSystem:
begin
ColorR.Top:= r.top+ MulDiv( i, h, NumberOfColors);
ColorR.Bottom:= r.bottom- MulDiv( i, h, NumberOfColors);
ColorR.Left:= r.left+ MulDiv( i, w, NumberOfColors);
ColorR.Right:= r.right - MulDiv( i, w, NumberOfColors);
end;
end;
FillRect( TargetDC, ColorR, FillBrush );
DeleteObject( FillBrush );
end;
else {fgdLeftBias, fgdRightBias:}
begin
if Pen <> 0 then
DeleteObject(SelectObject( TargetDC, OldPen ));
Pen:= CreatePen( PenStyle, PenWidth, RGB( _R, _G, _B ) );
OldPen:= SelectObject( TargetDC, Pen );
for j:= 1 to MulDiv( i+ 1, h+w, NumberOfColors)- MulDiv( i, h+w, NumberOfColors)
do
begin
case Orientation of
fgdLeftBias:
begin

```

```

if y >= r.bottom then inc( x, PenWidth ) else y:= y+ PenWidth;
if x2 >= r.right then inc( y2, PenWidth ) else x2:= x2+ PenWidth;
MoveToEx( TargetDC, x, y, nil );
LineTo( TargetDC, x2, y2 );
end;
else{fgdRightBias:}
begin
if x <= r.left then inc( y, PenWidth ) else x:= x- PenWidth;
if y2 >= r.bottom then dec( x2, PenWidth ) else y2:= y2+ PenWidth;
MoveToEx( TargetDC, x, y, nil );
LineTo( TargetDC, x2, y2 );
end;
end;
end;
DeleteObject( SelectObject( TargetDC, OldPen ) );
end;
end;
if NumberOfColors=0 then exit;
if i/NumberOfColors*100 > PercentFilling then break;
end;

if BufferedDraw then
begin
BitBlt( DC, 0, 0, r.right-r.left, r.bottom-r.top, BufferDC, 0, 0, SRCCOPY );
DeleteObject( SelectObject( BufferDC, OldBMP ) );
DeleteDC( BufferDC );
end;

end;

function Tem_DevelopComputingSystem_Bevel.BordersHeight: integer;
begin
Result:= 0;
if Inner <> bvNone then
begin
if fsdTop in Sides then inc(Result);
if fsdBottom in Sides then
if Bold then inc(Result, 1) else inc(Result);
end;
if Outer <> bvNone then
begin
if fsdTop in Sides then inc(Result);
if fsdBottom in Sides then
if Bold then inc(Result, 1) else inc(Result);
end;
end;

function Tem_DevelopComputingSystem_Bevel.BordersWidth: integer;
begin
Result:= 0;
if Inner <> bvNone then
begin
if fsdLeft in Sides then inc(Result);
if fsdRight in Sides then
if Bold then inc(Result, 1) else inc(Result);
end;
if Outer <> bvNone then
begin
if fsdLeft in Sides then inc(Result);
if fsdRight in Sides then
if Bold then inc(Result, 1) else inc(Result);
end;
end;

function Tem_DevelopComputingSystem_CustomListBoxItemStyle.HighlightColor:
TColor;
begin
Result:= incColor(Color, 60);
end;

```

```
function Tem_DevelopComputingSystem_CustomListBoxItemStyle.ShadowColor: TColor;  
begin  
Result:= decColor(Color, 60);  
end;  
  
end.
```

К6П3 - 2023