

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

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
« \_\_\_\_ » \_\_\_\_\_ 2022 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи big data**  
**наукових досліджень”**

Виконав здобувач вищої освіти  
II курсу, групи КН-21М-1,4  
ОПП «Комп’ютерні науки»  
спеціальності 122 «Комп’ютерні науки»  
\_\_\_\_\_ Пилипенко А.А.  
« \_\_\_\_ » \_\_\_\_\_ 2022 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Коваленко А.С.  
« \_\_\_\_ » \_\_\_\_\_ 2022 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет *Механіко-технологічний*  
Кафедра *Кібербезпеки та програмного забезпечення*  
Рівень вищої освіти *магістр*  
Галузь знань *12* "Інформаційні технології"  
Спеціальність *122* "Комп'ютерні науки"  
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2022 року

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

*Пилипенку Андрію Андрійовичу*

(прізвище, ім'я, по батькові)

- |  |   |
|--|---|
| 1. Тема роботи   | <i>Дослідження та програмна реалізація системи big data наукових досліджень</i>   |
| 2. Керівник роботи   | <i>Коваленко Анна Степанівна, канд. техн. наук, доцент</i><br>(прізвище, ім'я, по батькові, науковий ступінь, вчене звання) |
| затверджені наказом вищого навчального закладу № 18-13 від 17.08.2022 року           |   |
| 3. Строк подання студентом роботи до захисту   | <i>10.12.2022 р.</i>  |
| 4. Мета та завдання випускної кваліфікаційної роботи:                                | <i>Метою розробки є дослідження та програмна реалізація системи big data наукових досліджень</i>                            |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) |   |
| <i>1. Призначення та область використання.</i>                                       | <i>6. Наукова новизна.</i>  |
| <i>2. Перегляд аналогічних існуючих систем.</i>                                      | <i>7. Економічна ефективність розробленої програми.</i>   |
| <i>3. Опис і обґрунтування проектних рішень.</i>                                     | <i>8. Заходи з охорони праці та техніки безпеки.</i>  |
| <i>4. Етапи програмування системи.</i>   | <i>9. Висновки.</i>   |
| <i>5. Впровадження системи в промислову експлуатацію</i>                             |   |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)         |   |
| <i>Наукова новизна</i>   | <i>1 аркуш</i>  |
| <i>Структурна схема системи</i>  | <i>1 аркуш</i>  |
| <i>Функціональна схема системи</i>   | <i>1 аркуш</i>  |
| <i>Діаграма процесів</i>   | <i>1 аркуш</i>  |
| <i>Блок-схема алгоритму роботи додатку</i>   | <i>2 аркуша</i>   |
| <i>Показники економічної ефективності</i>  | <i>1 аркуш</i>  |

## 6. Консультанти розділів роботи

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

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

## КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання  
« 6 » вересня 2022 р.

Підпис керівника

Коваленко А.С.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 6 » вересня 2022 р.

Підпис здобувача

Пилипенко А.А.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Пилипенко А.А. Дослідження та програмна реалізація системи big data наукових досліджень. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2022.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи big data наукових досліджень.

Метою розробки є дослідження та програмна реалізація системи big data наукових досліджень.

Об'єктом дослідження є процес big data наукових досліджень.

Предметом дослідження є методи big data наукових досліджень.

Методи дослідження базуються на методах big data, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи big data наукових досліджень.

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

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

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

Програму розроблено в середовищі Delphi 10.4 Sydney.

**Ключові слова:** комп'ютерні науки, big data, наукові дослідження

## ABSTRACT

**Pylypenko A.A. Research and software implementation of the big data system of scientific research. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.**

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the big data system of scientific research.

The purpose of the development is research and software implementation of the big data system of scientific research.

The object of research is the big data process of scientific research.

The subject of research is big data methods of scientific research.

Research methods are based on big data methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the big data system of scientific research.

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

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

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

The program was developed in the Delphi 10.4 Sydney environment.

**Keywords:** computer science, big data, scientific research

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання .....	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	25
3.1 Опис функціонування системи .....	25
3.2 Розробка структурної схеми.....	38
3.3 Розробка функціональної схеми .....	44
3.4 Розробка діаграми процесів.....	46
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	48
4.2 Захист розробленого програмного забезпечення.....	54
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	61
6 НАУКОВА НОВИЗНА .....	65

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>			
<b>Вим.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	Дослідження та програмна реалізація системи big data наукових досліджень	<b>Лім.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<i>Розроб.</i>	Пилипенко А.А.					<b>М</b>	1	106
<i>Перев.</i>	Коваленко А.С.							
<b>Н.контр.</b>	Гермак В.С.					ЦНТУ КН-21М-1,4		
<b>Затв.</b>	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	66
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	66
7.2 Розрахунок трудомісткості розробки програмної продукції.....	68
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	70
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	75
7.5 Визначення собівартості розробки та ціни програмної продукції.....	79
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	82
7.7 Визначення експлуатаційних витрат.....	83
7.8 Визначення економічної ефективності програмної продукції.....	84
7.9 Висновок.....	86
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	87
8.1 Вступ.....	87
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	88
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	89
8.4 Розробка заходів з умов поліпшення охорони праці.....	92
8.5 Розрахункова частина .....	93
8.6 Висновки до розділу.....	95
9 ОСНОВНІ ВИСНОВКИ.....	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	98

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АС	–	автоматизована система
АСДС	–	автоматизована система відповідних дослідницьких стендів
АСЕД	–	АС управління експериментальними дослідженнями
АСМ	–	автоматизована система моделювання гіпотетичних систем
АСНД	–	автоматизовані системи наукових досліджень
КМ	–	константи моделей
СВ	–	сигнали виміру
СМ	–	структури моделей
СУ	–	сигнали управління

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** При дослідженні технічних систем можуть використовуватися теоретичні й емпіричні методи пізнання. Кожний із цих напрямків має відносну самостійність, має свої переваги й недоліки. У загальному випадку, теоретичні методи у вигляді математичних моделей дозволяють описувати й пояснювати взаємозв'язки елементів досліджуваної системи або об'єкта у відносно широких діапазонах зміни змінних величин. Однак при побудові теоретичних моделей неминуче введення яких-небудь обмежень, допущень, гіпотез і т.п. Тому виникає задача оцінки вірогідності (адекватності) отриманої моделі реальному процесу або об'єкту. Для цього проводиться експериментальна перевірка розроблених теоретичних моделей. Практика є вирішальною основою наукового пізнання. У ряді випадків саме результати експериментальних досліджень дають поштовх до теоретичного узагальнення досліджуваного явища. Експериментальне дослідження дає більш точну відповідність між досліджуваними параметрами. Але не треба й перебільшувати результати експериментальних досліджень, які справедливі тільки в межах умов проведеного експерименту.

Таким чином, теоретичні й експериментальні дослідження доповнюють один одного і є складеними елементами процесу пізнання навколишнього нас миру.

Як правило, результати експериментальних досліджень мають потребу в певній математичній обробці. У цей час процедура обробки експериментальних даних досить добре формалізована й дослідникові необхідно тільки її правильно використовувати. Коло питань, розв'язуваних при обробці результатів експерименту, не так вуж великий. Це питання підбора емпіричних формул і оцінка їхніх параметрів, питання оцінки дійсних значень вимірюваних величин і точності вимірів, питання дослідження кореляційних залежностей і деякі інші. Усі ці задачі дуже зручно виконувати за допомогою обчислювальної техніки та з залученням комп'ютерних мереж.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи big data наукових досліджень.

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

- Огляд існуючих систем big data наукових досліджень.
- Дослідження системи big data наукових досліджень.
- Програмна реалізація системи big data наукових досліджень.

*Об'єктом дослідження є процес big data наукових досліджень.*

*Предметом дослідження є методи big data наукових досліджень.*

*Методи дослідження базуються на методах big data, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод big data наукових досліджень.
- Розроблено вітчизняний продукт big data наукових досліджень, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі big data наукових досліджень.

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

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи big data наукових досліджень, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Основою всіх наукових досліджень є спостереження й експеримент.

Спостереження – це систематичне, цілеспрямоване сприйняття того або іншого об'єкта або явища без впливу на досліджуваний об'єкт або явище. Спостереження дозволяє одержати первісну інформацію з досліджуваного об'єкта або явища.

Експеримент – метод вивчення об'єкта, коли дослідник активно й цілеспрямовано впливає на нього шляхом створення штучних умов або використовує природні умови, необхідні для виявлення відповідних властивостей. Перевагами експерименту в порівнянні зі спостереженням реального явища або об'єкта є:

1. Можливість вивчення в «чистому виді», без впливу побічних факторів, що затемнюють основний процес.
2. В експериментальних умовах можна одержати результат більш швидко й точно.
3. При експерименті можна проводити випробування стільки разів, скільки це необхідно.

Результат експерименту або виміру завжди містить деяку погрішність. Якщо погрішність мала, то нею можна зневажити. Однак при цьому неминуче виникають два питання: по-перше, що розуміти під малою погрішністю, і, по-друге, як оцінити величину погрішності. Тобто, і результати експерименту мають потребу в певнім теоретичному осмисленні.

Метою будь-якого експерименту є визначення якісного й кількісного зв'язку між досліджуваними параметрами, або оцінка чисельного значення якого-небудь параметра.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

У деяких випадках вид залежності між змінними величинами відомі за результатами теоретичних досліджень. Як правило, формули, що виражають ці залежності, містять деякі постійні, значення яких і необхідно визначити з досвіду.

Іншим типом задачі є визначення невідомого функціонального зв'язку між змінними величинами на основі даних експерименту. Такі залежності називають емпіричними.

Однозначно визначити невідому функціональну залежність між змінними неможливо навіть у тому випадку, якби результати експерименту не мали помилок. Тим більше не треба цього очікувати, маючи результати експерименту, що містять різні помилки виміру.

Тому варто чітко розуміти, що метою математичної обробки результатів експерименту є не знаходження істинного характеру залежності між змінними або абсолютною величиною якої-небудь константи, а подання результатів спостережень у вигляді найбільш простої формули з оцінкою можливої погрішності її використання.

## 1.2 Область застосування

Областю застосування розробляємої системи є обробка результатів наукових досліджень, що зберігаються розподілено в мережі, як локальній так і глобальній. Це дозволяє підвищувати швидкість та точність обчислень, що у свою чергу дозволяє більш широко застосовувати обчислювальну техніку для розвитку науки. Розвиток науки спрямований на полегшення життя усіх громадян, тому програма, розроблена у результаті виконання магістерського проектування, має дуже важливе значення.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи big data наукових досліджень, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

## 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

Задачі обробки даних зустрічаються практично у всіх областях наукових досліджень (зокрема, в економіці, фізиці, хімії, біології, медицині). До таких задач відносяться як описові статистики й регресійний аналіз, так і різні перетворення даних (гістограми, частотний аналіз і аналіз часових рядів), а також задачі добування ознак (feature extraction). Майже завжди потрібне відображення даних і результатів їхньої обробки в графічному виді.

Одним з обов'язкових етапів будь-якого наукового дослідження є статистичний аналіз даних. Тривалий час аналіз наукових даних був долею фахівців, тому що це вимагало серйозної попередньої підготовки. З появою й удосконалюванням сучасних програм обробки даних статистична обробка піднялася на новий рівень. Тепер дослідник-учений може й не мати математичної підготовки. Досить оперувати статистичними поняттями й, саме головне, правильно вибрати метод аналізу. Усе здійснено завдяки комп'ютеру й новітнім програмам.

Всі програми статистичної обробки даних можна розділити на професійні, напівпрофесійні (популярні) і спеціалізовані. Статистичні програми відносяться до наукомісткого програмного забезпечення, ціна їх часто недоступна індивідуальному користувачеві. Професійні пакети мають велика кількість методів аналізу, популярні пакети – кількість функцій, достатнє для універсального застосування. Спеціалізовані ж пакети орієнтовані на яку-небудь вузьку область аналізу даних. Творці програмних статистичних пакетів

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8





додатковими убудованими формулами. Також в MS Excel неможливо побудувати якісні наукові графіки. Безумовно, MS Excel добре підходить для нагромадження даних, проміжного перетворення, попередніх статистичних прикидок, для побудови деяких видів діаграм. Однак остаточний статистичний аналіз необхідно робити в програмах, які спеціально створені для цих цілей. Існує макрос-доповнення XLSTAT-Pro <http://www.xlstat.com> для MS Excel який, містить у собі більше 50 статистичних функцій, включаючи аналіз виживаності, яких в основних випадках досить для звичайного застосування. Пробну версію макросу можна взяти на сайті виробника.

### **STADIA**

Програма вітчизняної розробки з 16-і літньою історією. Містить у собі всі необхідні статистичні функції. Вона прекрасно справляється зі своєю задачею – статистичним аналізом. Але, програма зовні фактично не змінюється з 1996 року. Графіки й діаграми, побудовані за допомогою STADIA, виглядають у сучасних презентаціях архаїчно. Колірна гама програми (червоний шрифт на зеленому) дуже стомлює в роботі. До позитивних якостей програми можна віднести україномовний інтерфейс і наявність книг, які описують роботу. Зі сторінки <http://www.protein.bio.msu.su/~akula/index.htm> можна взяти демо-версію STADIA.

### **SPSS (Statistical Package for Social Science)**

Самий часто використовуваний пакет статистичної обробки даних з більш ніж 30-і літньою історією <http://www.spss.com>. Відрізняється гнучкістю, потужністю застосовуємо для всіх видів статистичних розрахунків застосовуваних у біомедицині. Недавно вийшла 13-я англomовна версія. Існує україномовне представництво компанії <http://www.spss.ru> яке пропонує повністю українізовану версію SPSS 12.0.2 для Windows. З'явився підручник українською мовою, що дозволяє крок за кроком освоїти можливості SPSS, репетитор по статистиці українською мовою, що допомагає у виборі потрібної статистичної або графічної процедури для конкретних даних і задач, а також довідка по SPSS Base і SPSS Tables. Український офіс SPSS регулярно проводить навчальні курси по аналізі

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

даних за допомогою програмного забезпечення SPSS. На українську мову перекладена книга по SPSS, що вийшла у світло в 2002 році в Київському видавництві «Діасофт» за назвою «SPSS 10: мистецтво обробки інформації. Аналіз статистичних даних і відновлення схованих закономірностей».

### STATA

Професійний статистичний програмний пакет з data-management system, що може застосовуватися для наукових цілей. Один із самих популярних в освітніх і наукових установах США поряд з SPSS. Офіційний сайт <http://www.stata.com> Програма добре документована, видається спеціальний журнал для користувачів системи. Однак можливості попереднього ознайомлення з демо-версією немає.

### STATISTICA

Виробником програми є фірма StatSoft Inc. (США) <http://www.statsoft.com> яка випускає статистичні додатки, починаючи з 1985 року. STATISTICA включає велика кількість методів статистичного аналізу (більше 250 убудованих функцій) об'єднаних наступними спеціалізованими статистичними модулями:

- Основні статистики й таблиці.
- Непараметрична статистика.
- Дисперсійний аналіз.
- Множинна регресія.
- Нелінійне оцінювання.
- Аналіз часових рядів і прогнозування.
- Кластерний аналіз, Факторний аналіз.
- Дискримінантний функціональний аналіз.
- Аналіз довжини життя.
- Канонічна кореляція.
- Багатомірне шкалювання.
- Моделювання структурними рівняннями й ін.

Нескладний в освоєнні цей статистичний пакет може бути рекомендований для наукових досліджень будь-якої складності. У цей час

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

випущена версія 7. Українське представництво компанії ([http://www.statsoft. ua/](http://www.statsoft.ua/)) пропонує повністю українізовану 6-у версію програми. Сайт компанії містить багато інформації зі статистичної обробки медичних даних, підручник по статистиці українською мовою.

### **JMR**

Один зі світових лідерів в аналізі даних. Розвиває цей статистичний пакет SAS Institute <http://www.jmp.com> який викупив наприкінці 2002 року відому статистичну програму StatView. Однак особливих переваг для статистики цей програмний продукт не має.

### **SYSTAT**

Статистична система для персональних комп'ютерів <http://systat.com> Остання 11 версія має непоганим інтуїтивно зрозумілий інтерфейс. Компанія Systat Software також розробляє популярні у вітчизняних дослідників SigmaStat і SigmaPlot, які є відповідно, програмою статистичної обробки й програмою побудови діаграм. При спільній роботі стають єдиним пакетом для статистичної обробки й візуалізації даних.

### **NCSS**

Програма розвивається з 1981 року й розрахована на непрофесіоналів в області статистичної обробки. Інтерфейс системи багатооконий і як наслідок цього явища – небагато незвичний у використанні. Всі дії користувача супроводжуються підказками. Зараз доступна версія 2004 р. Із сайту <http://www.ncss.com> можна переписати повнофункціональну пробну версію працюючих 30 днів.

### **MINITAB 14**

Статистичний пакет MINITAB у цей час випускається у версії 14. Із сайту виробника <http://www.minitab.com> можна взяти повнофункціональний пробний варіант програми, що працює 30 днів. Це досить зручний у роботі програмний пакет, що має гарний інтерфейс користувача, гарні можливості по візуалізації результатів роботи. Має докладну довідку.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13



– прогнозування методами ковзного середнього й експонентного згладжування;

– побудова гістограм;

– фур'є-аналіз;

– обчислення F-, T- і Z-тестів.

Gnumeric також є крос-платформним додатком, інформацію про наявність складання під конкретну операційну систему й версіях пакета можна одержати на сайті проекту <http://www.gnome.org/projects/gnumeric/>. Є невелика стаття в Wikipedia. Gnumeric також не відрізняється багатством і розмаїтістю варіантів графіків.

### **Grace (xmgrace)**

Пакет двовимірної наукової графіки. Призначений для обробки даних і підготовки результатів до публікації. На «вході» пакета, як правило, повинен бути текстовий файл зі стовпцями даних, а на «виході» виходить графічний файл заданого типу з високоякісним графіком. Основні можливості пакета в області обробки даних наступні:

– математичні операції (арифметика й застосування функцій) з рядами даних;

– побудова гістограм;

– пряме й зворотне перетворення Фур'є з можливістю використання цифрової фільтрації;

– інтерполяція методом ковзного середнього й сплайн-інтерполяція;

– чисельне диференціювання й інтегрування;

– аналіз часових рядів шляхом виключення періодичного компонента (seasonal differences);

– лінійна й нелінійна регресія, нелінійне припасування;

– обчислення кореляції, коваріації й згортки;

– добування ознак.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Пакет Grace поширюється у вихідних текстах (сайт проекту [http:// plasma-gate.weizmann.ac.il/Grace](http://plasma-gate.weizmann.ac.il/Grace)), але існують готові складання в деяких дистрибутивах Linux (ALT, Debian, SUSE ...). Складання для Windows не може знайти навіть Google, однак можливий запуск пакета в Cygwin-Оточенні. Про цей пакет також є невелика стаття в Wikipedia.

### **Maxima**

Пакет символної математики, що дозволяє проводити символні перетворення: спрощення виражень, розкриття дужок, піднесення в ступінь. Крім того, можна знаходити корінь квадратних (і не тільки) рівнянь, працювати із тригонометричними функціями й комплексними числами, будувати двовимірні й тривимірні графіки. Велика кількість убудованих функцій, можливість створювати користувальницькі функції й писати користувальницькі програми роблять пакет серйозним інструментом для професійного застосування. Сам по собі пакет працює в текстовому режимі, однак для Maxima існує кілька інтерфейсів (оболонок), серед яких найцікавіший TeXmacs – WYSIWYG-Редактор для документів TeX. Сесія Maxima в TeXmacs являє собою документ із розрахунками й формулами типографської якості, що може бути перетворений як в PDF, так і в HTML і в Te, а наявність в TeXmacs презентаційного режиму дозволяє працювати в програмі прямо в ході лекції або семінару.

### **SciLab**

Пакет числової обробки даних. Основний режим роботи – командний, однак є система візуального моделювання технічних систем (Scicos). До основних функцій пакета можна віднести роботу з матрицями (у тому числі розрідженими), рішення звичайних диференціальних рівнянь, чисельне диференціювання й інтегрування, побудова двовимірних і тривимірних графіків по формулах і за результатами розрахунків, рішення задач лінійного програмування, а також можливість створення користувальницьких програм. Пакет є крос-платформним, на сайті проекту ([www.scilab.org](http://www.scilab.org)) можна знайти самі останні складання для різних операційних систем.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

## **GRETL (GNU Regression, Econometrics and Time-series Library)**

Професійний кроссплатформений пакет для рішення задач економетрики й аналізу часових рядів (що й видно за назвою). На сайті проекту ([gretl.sourceforge.net](http://gretl.sourceforge.net)) можна знайти складання для різних платформ, а також додаткові бібліотеки й файли прикладів (англійською мовою). Українські Інтернет-ресурси, присвячені пакету, знайти досить важко, однак на початку 2007 року у видавництві «Гаряча лінія-телеком» вийшла книга Т.Куфеля «Економетрика. Рішення задач із застосуванням пакета програм GRETL» (ISBN N 5-93517-307).

Описані вище кілька програм, звичайно, не вичерпують усього розмаїтості вільних програмних засобів для наукових розрахунків і обробки даних, однак вони можуть скласти якийсь базовий набір, за допомогою якого можна забезпечити істотну економію засобів при рішенні наукових і дослідницьких задач. Із всіх перерахованих програм тільки Grace не є крос-платформної, але працює в Windows через знов-таки безкоштовний емулятор Cygwin. Таким чином, можна практично даром організувати потужну станцію для наукових розрахунків з використанням улюбленої операційної системи. А спеціально для людей, що займаються науковими розрахунками, але які не знають іноземних мов, з'являється усе більше літератури українською мовою.

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

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

### **Delphi 10.4 Sydney**

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

Основні можливості Delphi 10.4.1:

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

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

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

– Тип даних Delphi «record» тепер підтримуватиме 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.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18





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, libSMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

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

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

### **Підвищення якості й швидкодії інструментів**

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

– Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

### **Змінені стилі VCL для High DPI**

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

### **Поліпшена кроссплатформеність**

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

						<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			22

- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

### **Оновлений менеджер пакетів Getit**

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

### **Універсальний інсталятор для установки Online і Offline**

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

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

## **2.3 Розгорнута постановка завдання**

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи big data наукових досліджень.

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 методика побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

Розробити функціональну та структурну схеми системи;

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>23</b>

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

**Види помилок і порядок обробки результатів вимірів при проведенні наукових досліджень**

#### **Види вимірів і причини помилок**

Під виміром розуміють порівняння вимірюваної величини з іншою величиною, прийнятою за одиницю виміру.

Розрізняють два типи вимірів: прямі й непрямі. При прямому вимірі вимірювана величина рівняється безпосередньо зі своєю одиницею міри. Наприклад, вимір мікрометром лінійного розміру, проміжку часу за допомогою годинних механізмів, температури – термометром, сили струму – амперметром і т.п. Значення вимірюваної величини відлічується при цьому по відповідній шкалі приладу.

При непрямому вимірі вимірювана величина визначається (обчислюється) за результатами вимірів інших величин, які пов'язані з вимірюваною величиною певною функціональною залежністю. Наприклад, вимір швидкості по пройденому шляху й витраченому часу, вимір щільності тіла по вимірі маси й об'єму, температури при різанні по електрорушійній силі, величини сили – по пружних деформаціях і т.п.

При вимірі будь-якої фізичної величини роблять перевірку й установку відповідного приладу, спостереження їхніх показань і відлік. При цьому ніколи істинного значення вимірюваної величини не одержати. Це пояснюється тим, що вимірювальні засоби засновані на певному методі виміру, точність якого кінцева. При виготовленні приладу задається клас точності. Його погрішність визначається точністю ділень шкали приладу. Якщо шкала лінійки нанесена через 1 мм, то точність відліку  $\pm 0,5$  мм не змінити якщо застосуємо лупу для

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

розглядання шкали. Аналогічно відбувається вимір і при використанні інших вимірювальних засобів.

Крім приладової погрішності на результат виміру впливає ще ряд об'єктивних і суб'єктивних причин, що спричиняються появу помилки виміру – різниці між результатом виміру й істинним значенням вимірюваної величини. Помилка виміру звичайно невідома, як невідомо й істинне значення вимірюваної величини. Виключення становлять виміри відомих величин при визначенні точності вимірювальних приладів або їх тарировці. Тому однією з найважливіших задач математичної обробки результатів експерименту і є оцінка істинного значення вимірюваної величини за даними експерименту з можливо меншою помилкою.

### **Типи помилок виміру**

Крім приладової погрішності виміру (обумовленої методом виміру) існують і інші, які можна розділити на три типи:

1. Систематичні погрішності обумовлюють постійно діючими факторами. Наприклад, зсув початкової крапки відліку, вплив нагрівання тіл на їхнє подовження, зношування ріжучого леза й т.п. Систематичні помилки виявляють при відповідній тарировці приладів і тому вони можуть бути враховані при обробці результатів вимірів.

2. Випадкові помилки містять у своїй основі багато різних причин, кожна з яких не проявляє себе чітко. Випадкову помилку можна розглядати як сумарний ефект дії багатьох факторів. Тому випадкові помилки при багаторазових вимірах виходять різними як по величині, так і за знаком. Їх неможливо врахувати як систематичні, але можна врахувати їхній вплив на оцінку істинного значення вимірюваної величини. Аналіз випадкових помилок є найважливішим розділом математичної обробки експериментальних даних.

3. Грубі помилки (промахи) з'являються внаслідок неправильного відліку по шкалі, неправильному запису, невірної установки умов експерименту й т.п. Вони легко виявляються при повторному проведенні досвідів.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Надалі будемо вважати, що систематичні й грубі помилки з результатів експерименту виключені.

### Властивості випадкових помилок

Випадкові помилки бувають як позитивні, так і негативні різної величини, не переважаючи певної межі. Якщо позначити через  $X$  істинне значення вимірюваної величини,  $a$  результат першого виміру –  $a_1$ , то різниця  $X - a_1 = x_1$  або  $a_1 - X = x_1$  називають дійсною абсолютною помилкою одного виміру. Одночасно вона є випадковою (при виключенні систематичних і грубих помилок).

Якщо виміру провести багаторазово в тих самих умовах, то результати окремих вимірів однаково надійні. Таку сукупність вимірів  $a_1, a_2, \dots, a_n$  називають рівноточними вимірами. Якщо проаналізувати досить більшу серію рівноточних вимірів і відповідних випадкових помилок вимірів, то можна виділити 4 властивості випадкових помилок:

1. Число позитивних помилок майже дорівнює числу негативних;
2. Дрібні помилки зустрічаються частіше, ніж великі;
3. Величина найбільш великих помилок не перевершує деякого певної межі, що залежить від точності виміру. Саму більшу помилку в ряді рівноточних вимірів називають граничною помилкою;
4. Частки від ділення алгебраїчної суми всіх випадкових помилок на їх загальне близько до нуля, тобто:

$$\frac{x_1 + x_2 + \dots + x_n}{n} \approx 0. \quad (3.1)$$

На основі зазначених властивостей при обліку деяких допущень математично досить строго виводиться закон розподілу помилок, описуваний наступною функцією:

$$y = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \quad (3.2)$$

- де
- $\sigma$  – дисперсія вимірів (див. нижче);
  - $e$  – підстава натуральних логарифмів;

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27



$$X = \frac{\sum_{i=1}^n a_i}{n} + \frac{\sum_{i=1}^n \Delta x_i}{n}. \quad (3.5)$$

Величина  $\frac{1}{n} \sum_{i=1}^n a_i$  є середньоарифметичним величини  $X$ . Якщо число  $n$  досить велике (при  $n \rightarrow \infty$ ), те відповідно до четвертої властивості випадкових помилок:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \Delta x_i = 0. \quad (3.6)$$

Це ж видно й по кривій Гаусса, де всякої позитивної погрішності відповідає рівна їй негативна.

З викладеного треба, що  $X = a$  при  $n \rightarrow \infty$ , тобто при нескінченному числі вимірів істинне значення вимірюваної величини дорівнює середньоарифметичному значенню результатів всіх вимірів. При обмеженому числі вимірів істинне значення буде відрізнятися від середньоарифметичного й необхідно оцінити величину цієї розбіжності:  $X = a \pm \Delta x$ .

Треба ще раз підкреслити, що середньоарифметичне значення, прийняте за істинне значення вимірюваної величини, є найбільш імовірним значенням. Серед значень  $a_i$  можуть виявитися значення, які в дійсності ближче до істинного значення.

Відхилення  $\Delta x$  ймовірнішого значення  $a$  від його істинного значення  $X$  називають істинною абсолютною помилкою.

### Оцінка точності вимірів

Для ряду рівноточних вимірів  $a_1, a_2, \dots, a_n$  визначимо його середньоарифметичне значення  $a$  й складемо різниці  $(a - a_1), (a - a_2), \dots, (a - a_n)$ .

Кожну із цих різниць називають ймовірнішою помилкою окремого виміру ( $V_i$ ). Ймовірніші помилки, як і істинні помилки  $\Delta x_i = (X - a_i)$ , бувають позитивні й негативні, нульові. Розглянемо:

$$\sum_{i=1}^n V_i = V_1 + V_2 + \dots + V_n = na - (a_1 + a_2 + \dots + a_n) = 0, \quad (3.7)$$

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29









Якщо відомо точне значення  $\sigma$ , то ймовірність появи значення, що відрізняється від середньоарифметичного  $a$  більш ніж на  $3\sigma \leq 0,003$  і всі виміри, що відрізняються від  $a$  на  $3\sigma$  ( і більше ) можуть бути відкинуті, як малоімовірні.

Варто мати на увазі, що для сукупності вимірів імовірність появи виміру  $\geq 3\sigma$  від  $a$  завжди більше 0,003. Дійсно, імовірність того, що результат кожного виміру не буде відрізнятися від істинного більш ніж  $3\sigma$  становить  $1 - 0,003 = 0,997$ . Імовірність того, що всі  $n$  вимірів не будуть відрізнятися від середнього більш ніж на  $3\sigma$  за правилом множення ймовірностей складе  $(1 - 0,003)^n$ . Для не занадто великого  $n$ :  $(1 - 0,003)^n \approx 1 - 0,003 \cdot n$ .

Це значить, що ймовірність того, що з 10 вимірів хоча б одне буде випадково відрізнятися від середнього більш ніж на  $3\sigma$  ( буде вже не 0,003, а 0,03 або 3%. А при 100 вимірах імовірність такої події складе вже близько 30%.

Звичайне число вимірів не дуже велико. При цьому точне значення ( не відомо, отже, відкидати виміри, що відрізняються від середнього більш ніж на  $3\sigma$ , не можна.

Для оцінки ймовірності  $\beta$  випадкової появи "випадаючих" значень у ряді  $n$  вимірів складені відповідні таблиці.

Для застосування таблиці обчислюється середнє арифметичне  $a$  й середня квадратична погрішність  $s_n$  із всіх вимірів, включаючи й підозрюване значення  $a_k$ . Потім обчислюється відхилення підозрюваного значення  $a_k$  від середнього арифметичного в частках середньоквадратичної помилки:

$$V_{\text{макс}} = \left| \frac{a - a_k}{S_n} \right|. \quad (3.19)$$

По таблиці визначається якій ймовірності  $\beta$  відповідає отримане значення  $V_{\text{макс}}$ .

Якщо ймовірність появи даного виміру в ряді лежить у діапазоні  $0,1 > \beta > 0,01$ , то представляється однаково правильним – залишити цей вимір або відкинути. У випадку ж, коли  $\beta$  виходить за зазначені межі, питання про

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34



$$\frac{\sum_{i=1}^n \Delta y^2}{n} = \left( \frac{\partial f}{\partial x_1} \right)^2 \frac{\sum_{i=1}^n (\Delta x_1)^2}{n} + \left( \frac{\partial f}{\partial x_2} \right)^2 \frac{\sum_{i=1}^n (\Delta x_2)^2}{n} + \dots + \left( \frac{\partial f}{\partial x_m} \right)^2 \frac{\sum_{i=1}^n (\Delta x_m)^2}{n}. \quad (3.22)$$

Тут суми подвоєних добутків типу:

$$\frac{2 \sum_{i=1}^n \Delta x_k \sum_{i=1}^n \Delta x_{k+1}}{n} = 0. \quad (3.23)$$

Відповідно до четвертої властивості випадкових помилок ( $\Delta x_i = 0$ ).

Тоді в лівій і правій частинах маємо середньоквадратичні погрішності функції й аргументів:

$$S_y^2 = \left( \frac{\partial f}{\partial x_1} \right)^2 S_{x_1}^2 + \left( \frac{\partial f}{\partial x_2} \right)^2 S_{x_2}^2 + \dots + \left( \frac{\partial f}{\partial x_m} \right)^2 S_{x_m}^2. \quad (3.24)$$

### Правила округлення чисел

Величина погрішності результату вимірів фізичної величини дає подання про те, які цифри в числовому значенні вимірюваної величини сумнівні. Тому результати вимірів варто округляти перед тим, як робити з ними подальші обчислення.

Округляти числове значення результату вимірів треба відповідно до числового розряду значущої цифри погрішності. При цьому виконують загальні правила округлення.

Зайві цифри в цілих числах замінюються нулями, а в десяткових дробах відкидаються (як і зайві нулі). Наприклад, якщо погрішність виміру  $< 0,001$  мм, то результат 1,07005 округляється до 1,070.

Якщо перша зі змінюваних нулями й цифр, що відкидаються, менше 5, що залишаються цифри не змінюються. Наприклад, число 148935, точність виміру  $< 50$ , округлення = 148900.

Якщо перша із замінних нулями або цифр, що відкидаються, дорівнює 5, а за нею не треба ніяких цифр або йдуть нулі, то округлення виробляється до найближчого парного числа. Наприклад, число 123,50 округляється до 124.



11. Записати остаточний результат:

$$X = a \pm \Delta x. \quad (3.29)$$

12. Оцінити відносну погрішність результату серії вимірів:

$$e = \frac{\Delta x}{a} 100\%. \quad (3.30)$$

### 3.2 Розробка структурної схеми

Для побудови структурної схеми ми розглянемо автоматизовані системи наукових досліджень. Відповідно функціональна схема системи обробки результатів наукових досліджень, у даній роботі, буде розглянута як частина структурної схеми автоматизовані системи наукових досліджень.

Автоматизовані системи наукових досліджень (АСНД з використанням технологій big data) – системи, у яких для підвищення ефективності наукових досліджень ряд процедур одержання, аналізу, передачі й нагромадження інформації, пов'язаних з використанням у ході вишукувань методу математичного моделювання, формалізований і виконується автоматизовано. Задачі, які можна вирішувати за допомогою АСНД з використанням технологій big data:

- скоротити строки досліджень;
- підвищити точність моделей і одержати якісно нову інформацію;
- підвищити ефективність використання встаткування;
- скоротити допоміжний персонал дослідницьких груп.

Автоматизація досліджень вимагає насамперед розчленовування досліджуваної складної системи на складові частини (рівні будови) з метою встановлення закономірностей процесів на кожному рівні. На основі знання властивостей кожного рівня передвіщаються характеристики різних варіантів ще не існуючих (гіпотетичних) систем, а глибоке розуміння механізмів явищ дозволяє цілеспрямовано шукати найбільш ефективні елементи й структури

розроблювальних систем. У світовій практиці створені й експлуатуються сотні АСНД з використанням технологій big data для вивчення на різних рівнях складних систем.

Узагальнена структура найважливіших підсистем при автоматизації досліджень наведена на рисунку 3.1.

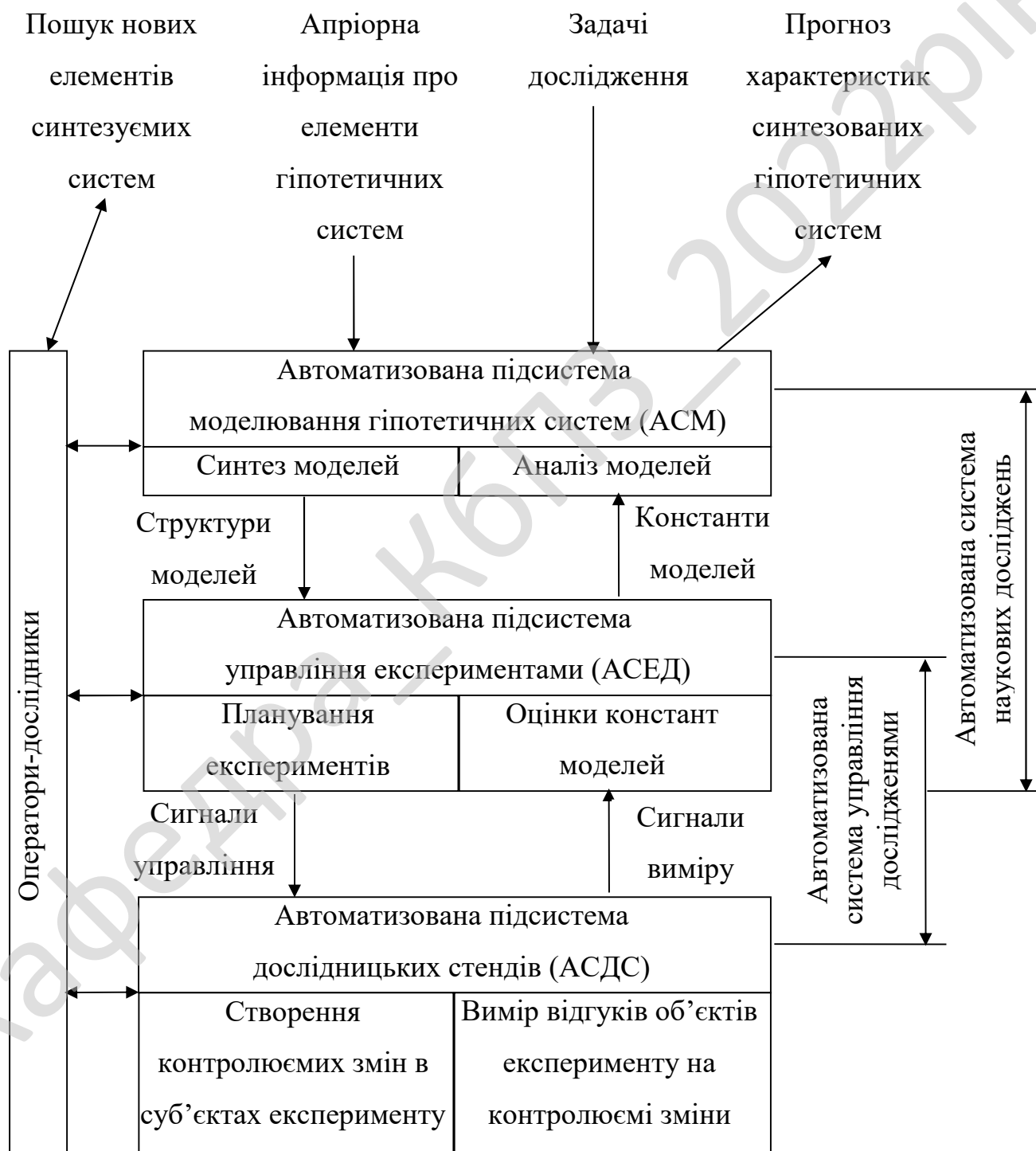


Рисунок 3.1 – Структурна схема системи



– визначення статистичних оцінок констант моделей порівнянням обчислених по моделі значень відгуку об'єкта на контрольовані збурювання з обмірюваними значеннями за заданими критеріями оцінки (зворотні задачі моделювання).

3. У підсистемі моделювання гіпотетичних систем (АСМ) автоматизують:

– синтез варіантів математичних моделей гіпотетичних систем і розрахунки відгуку моделей (прямі задачі моделювання) на основі апріорної інформації про елементи синтезованої системи на перших етапах досліджень і скоректованих моделей за експериментальним даними;

– оптимізація характеристик синтезованих гіпотетичних систем і порівняння їх із заданими цілями вишукувань;

– аналіз оцінок гіпотетичних систем для уточнення пізнавальних задач, розв'язуваних у підсистемі експериментальних досліджень (АСЕД), утвореної сполученням підсистем АСДС і АСЕД;

– аналіз чутливості оцінок гіпотетичних систем до параметрів елементів моделей для визначення напрямку пошуку більше ефективних елементів.

При об'єднанні підсистем АСЕД й АСМ утвориться АСНД з використанням технологій big data.

Головний принцип створення технічних і програмних засобів АСНД з використанням технологій big data – модульна побудова систем із забезпеченням сполучення користувачем окремих модулів у систему без спеціальних додаткових розробок (стандартизація інтерфейсів, створення уніфікованих магістралей для підключення цифрових приладів у систему). Найважливіша умова ефективного функціонування АСНД з використанням технологій big data – забезпечення можливості для дослідника активно контролювати всі виконувані АСНД з використанням технологій big data процедури й управляти ними. Для цього створюються системи програм, що забезпечують дослідникам зв'язок АСНД з використанням технологій big data через дисплеї в режимі діалогу мовою в термінах експериментаторів без знань спеціальних питань програмування. У

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

результаті комплексної автоматизації ряду процедур в АСНД з використанням технологій big data строки досліджень скорочуються в 3-5 разів, а точність даних підвищується в 2-3 рази. Широке використання фізичних методів вимірів дозволяє на основі вивчення динаміки процесів одержувати в АСНД з використанням технологій big data якісно нову інформацію для більше глибокого розуміння різних механізмів процесу й вибору оптимальних інженерних рішень.

Для рішення задач на рівні окремих видів устаткування створені АСНД з використанням технологій big data, у яких елементи системи представлені у вигляді математичних моделей елементів того або іншого апарата. У таких АСНД з використанням технологій big data загальний об'єм експериментів значно скорочується за рахунок пошуку оптимальних рішень на моделях гіпотетичних апаратів, а вартість досвідчених установок – у результаті зменшення масштабів досліджуваних фрагментів апаратури.

Одночасне застосування в АСНД з використанням технологій big data на рівні схем математичних моделей апаратів, отриманих в АСНД з використанням технологій big data інших рівнів, дає можливість скорочувати час на досвіди за рахунок вивчення й вибору на моделях оптимальних режимів і експериментів до початку досвідчених робіт і оперативної корекції ходу досліджень.

Подальший розвиток АСНД з використанням технологій big data пов'язане з організацією окремих систем у єдину ієрархічну галузеву систему, що дозволила б фахівцям різного профілю оперативно обмінюватися інформацією (колективний інтелект) для максимального скорочення витрат засобів і часу на реалізацію в промисловості результатів наукових досліджень.

На рисунку 3.2 зображена структурна схема локальної мережі лабораторно-стендового комплексу.

На цій схемі введені наступні позначення, та визначенні наступні поняття.

**Інтегроване робоче місце** – на комп'ютері встановлені всі модулі програмного забезпечення: інтегроване середовище, ПЗ експерименту, локальна база даних і підсистема тарировки.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42



Можливі наступні функції:

- створення, зміна, пошук і зберігання сценаріїв експериментів;
- післясеансовий аналіз результатів.

“Сліпий вузол” – на комп'ютері встановлене програмне забезпечення реєстрації експерименту реального часу.

Даний програмно-технічний комплекс призначений для реєстрації процесів, що відбуваються при проведенні експерименту й передачі зібраної інформації під час або після сеансу в базу даних результатів, розташовану на сервері.

### **Підсистема тарировки й інформаційного супроводу вимірювальних каналів**

Підсистема тарировки й інформаційного супроводу вимірювальних каналів призначена для конфігурування вимірювальних каналів автоматизованих систем із заданими метрологічними характеристиками або визначення метрологічних характеристик для вже існуючих каналів.

### **3.3 Розробка функціональної схеми**

На рисунку 3.3 зображена функціональна схема розробленої системи. Вона складається з наступних функціональних блоків, які взаємодіють між собою, та призначених для виконання наступних дій.

- Комплекс автоматизації експериментальних установок призначений для автоматизації робіт на дослідницьких, технологічних і контрольно-діагностичних комплексах і експериментальних установках різного призначення.
- Комплекс може функціонувати на РС сумісному комп'ютері промислового, мобільного або офісного виконання, оснащеного засобами збору даних.
- Можливості комплексу по кількості, составу й характеристикам вимірювальних каналів залежать від використаних пристроїв збору даних і потужності застосовуваного комп'ютера.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

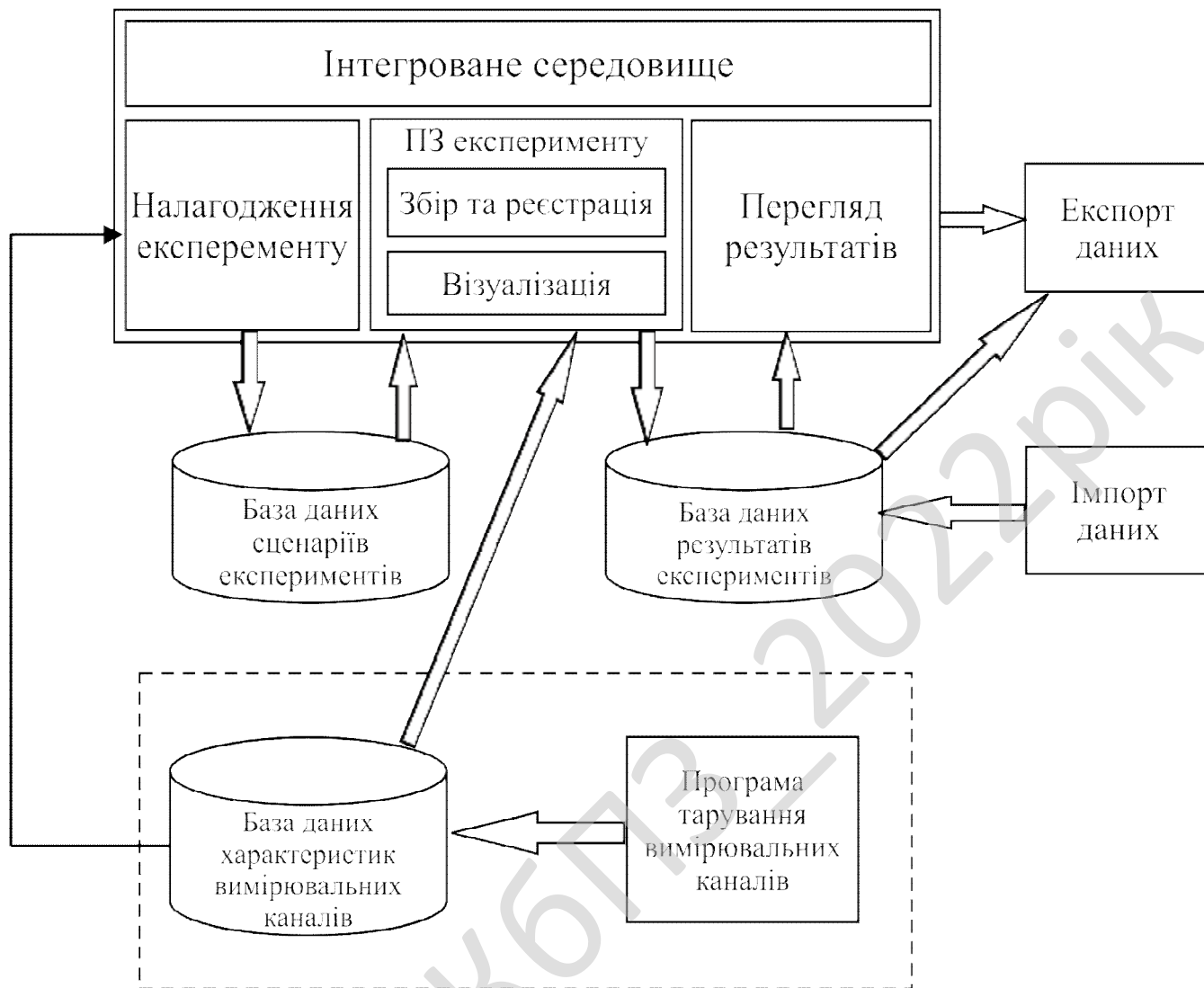


Рисунок 3.3 – Функціональна схема системи

Оснoву комплексу становить інтегроване середовище, що дозволяє:

- проводити налаштування експерименту;
- здійснювати пошук потрібного сценарію в базі даних;
- робити запуск програмного забезпечення експерименту реального часу;
- здійснювати перегляд і аналіз результатів.

Програмне забезпечення експерименту реального часу складається із двох незалежних частин здатних працювати як єдине ціле на одному комп'ютері або поодиноці на окремих комп'ютерах.

Підсистема тарировки й інформаційного супроводу вимірювальних каналів передає інформацію іншим підсистемам комплексу про склад і характеристики наявних вимірювальних каналів і їхніх елементів, дозволяє формувати вимірювальні канали й визначати їхні метрологічні характеристики.

Комплекс може функціонувати як на одиночному комп'ютері, так і з використанням клієнт-серверних технологій у рамках розподіленої системи збору й обробки даних.

### 3.4 Розробка діаграми процесів

На рисунку 3.4 зображена діаграма взаємодії процесів, які відбуваються у системі, розробленій у результаті виконання магістерського проектування.

З цієї діаграми видно, що спершу запускається процес початку/кінця роботи ПЗ. Він взаємодіє з наступними двома процесами:

- процес налагодження експерименту;
- процес збереження результатів експерименту у базі даних.

Процес налагодження експерименту взаємодіє з наступними процесами:

- процесом налагодження характеристик експерименту;
- процесом налагодження вимірювальних каналів;
- процес налагодження параметрів візуалізації;
- процес створення сценаріїв експерименту.

Останій процес у свою чергу взаємодіє з процесом створення сценаріїв експерименту. Цей процес взаємодіє з процесом вибору сценарію, що взаємодіє з процесом введення апріорної інформації.

Процесом введення апріорної інформації взаємодіє з ключовим процесом запуску та проведення експерименту. Він взаємодіє з наступними процесами:

- процесом проведення вимірювань;
- процесом математичної обробки;
- процесом реєстрації даних у відповідності зі сценарієм;

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

- процесом виведення вікна візуалізації;
- процесом обробки та аналізу отриманих даних.

Процес обробки та аналізу отриманих даних взаємодіє з процесом виведення результатів, який у свою чергу взаємодіє з наступними процесами:

- процесом створення файлу звіту;
- процесом післясеансової обробки даних.

Процес післясеансової обробки даних взаємодіє з заключним процесом збереження результатів експерименту у базі даних.



Рисунок 3.4 – Діаграма процесів системи

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

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

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

На рисунку 4.1 зображена блок-схема основної програми. З нього видно, що програма починається з запуску функції виведення основного вікна програми.

Після цього вибирається меню налагодження. Якщо налагодження потрібно, то відбувається виконання наступних етапів роботи програми:

- створення нового сценарію;
- налаштування загальних параметрів сценарію;
- налаштування конфігурації вимірювального каналу та параметрів збору інформації;
- створення та налагодження розрахункових каналів;
- тарування каналів;
- налагодження параметрів візуалізації;
- збереження створеного сценарію.

Після вибору закладки налагодження відбувається вибір закладки проведення експерименту.

Якщо цю закладку є потреба вибрати, то відбуваються наступні дії:

- відкривається файл з потрібним сценарієм;
- створюється новий експеримент на основі відкритого сценарію;
- запускається підпрограма проведення вимірювань;
- візуалізуються результати експерименту;
- виводяться результати;
- зберігаються результати у базі даних.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>48</b>

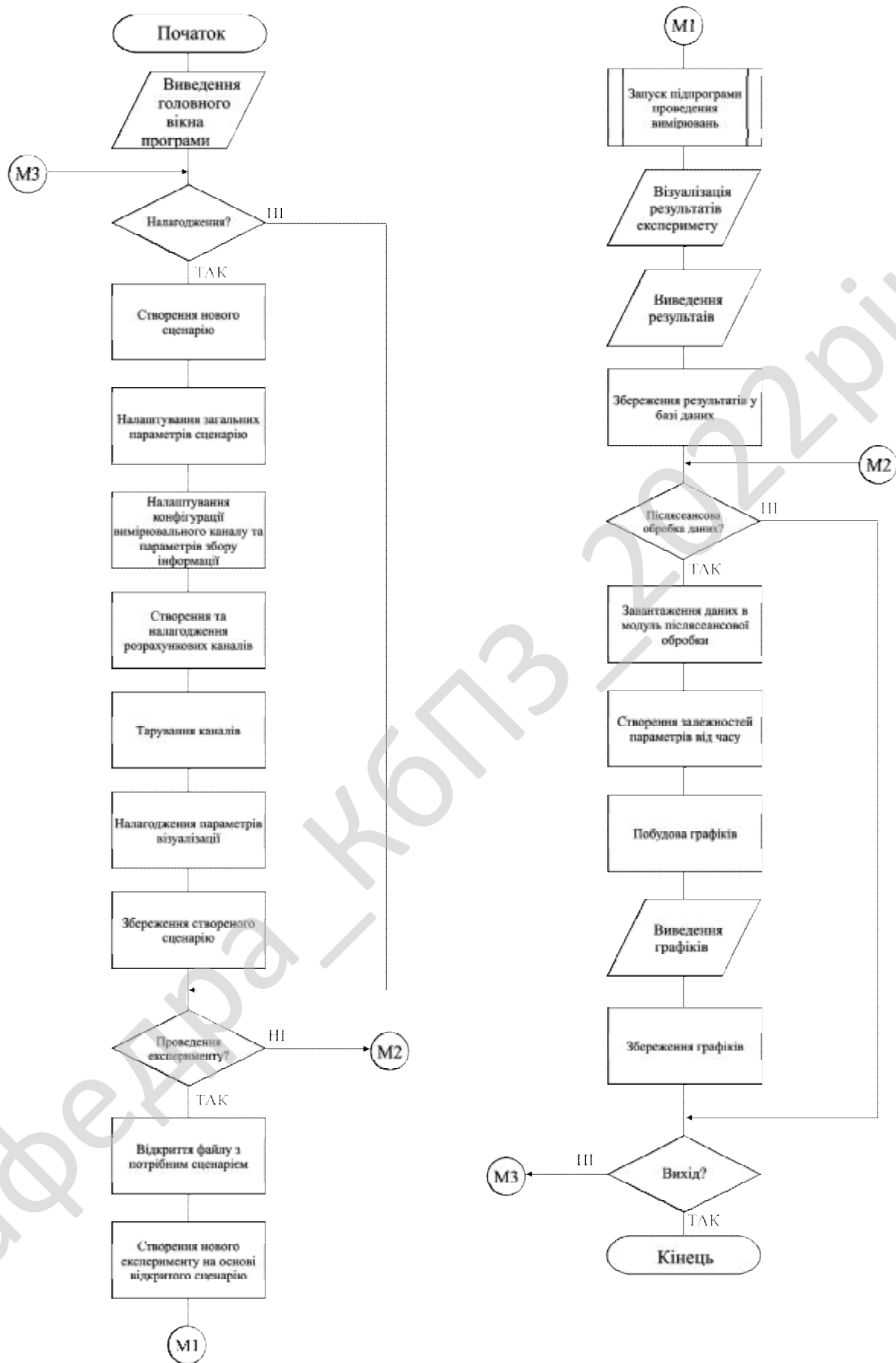


Рисунок 4.1 – Блок-схема роботи основної програми

Після проведення експерименту, відбувається післясеансова обробка даних. Для цього виконуються наступні дії:

- завантажуються дані в модуль післясеансової обробки;
- створюється залежність параметрів від часу;
- будуються графіки;
- виводяться графіки;
- зберігаються графіки.

На рисунку 4.2 зображена блок-схема підпрограми проведення вимірювань. З цієї блок-схеми бачимо, що відбувається активізація пристроїв зняття даних.

Після цього відбувається зняття показів з датчиків.

За цим виводиться вікно візуалізації.

Якщо пуск не відбувається то програма чекає команди пуск.

Якщо ж він відбувається то виконуються наступні дії:

- проводяться вимірювання;
- відбувається математична обробка результатів вимірювань за вказаним сценарієм;
- візуалізуються показання приладів;
- реєструються дані за вказаним сценарієм.

Якщо відбувається стоп, то виконуються наступні дії:

- вводиться післясеансова інформація;
- присвоюється експерименту статус «Завершений».

На цьому ми закінчуємо опис блок-схем алгоритму програмного продукту.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>50</b>



Рисунок 4.2 – Блок-схема роботи підпрограми проведення вимірювань



```

begin
    ValueMemo.Lines.Clear;
    VaxChart.Title.Text.Clear;
    VaxChart.Title.Text.Add('Графік вихідний ВАХ транзистора');
    SeriesNom:=0;
end;
//Установлення зв'язку із приладом
procedure TVAX_Tran.btnConnectClick(Sender: TObject);
begin
    FormRef:=InitLib(); //Підключення бібліотеки
    comm_setParam; //Установка параметрів СОМ порту
    if FormRef=0 then showMessage('Помилка установки зв'язку із приладом')
    else showMessage('Зв'язок із приладом установлений');
    btnConnect.Enabled:=False;
end;
procedure TVAX_Tran.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    if FormRef>0 then CloseLib; //Закриття СОМ порту при виході із програми
end;
procedure TVAX_Tran.btnStartClick(Sender: TObject);
type pBwr=^byte;
    p=^byte;
var tokMin,tokMax,lenWr,Len,n,i,tokBase: integer;
    BufWr,BufRd,temp:array [0..511] of byte; //Буфери запису/читання й часовий
        буфер
    Bwr: pBwr;
    B: p;
    ResBuf: array [0..255] of real; //буфер результатів
begin
    ValueMemo.Lines.Clear;
    tokBase:=strtoint(baseTokEdit.Text); //значення базового струму
    tokMin:=strtoint(tokEditMin.Text); //початкове значення струму уведене
        користувачем
    tokMax:=strtoint(tokEditMax.Text); //кінцеве
    //перевірка уведених значень
    if((tokMin<0) or (tokMin=tokMax)) or ((tokMax<tokMin) or (tokMax>20)) then
    begin
        showMessage('Невірний діапазон уведених значень');
        exit;
    end;
    if (tokBase<0) or (tokBase>20000) then
    begin

```

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>53</b>



## Основні відомості

Алгоритм був створений в 1996 році Карлайлом Адамсом (Carlisle Adams) і Стаффордом Таваресом (Stafford Tavares) використовуючи метод побудови шифрів CAST, який використовується також і іншим їхнім алгоритмом CAST-256 (алгоритм-кандидат AES).

CAST-128 складається з 12 або 16 раундів мережі Фейстеля з розміром блоку 64 біта й довжиною ключа від 40 до 128 біт (але тільки з інкрементацією по 8 біт). 16 раундів використовуються коли розміри ключа перевищують 80 біт. В алгоритмі використовуються 8x16 S- блоки, засновані на бент-функції, операції XOR і модулярної арифметиці (модулярне додавання й вирахування). Є три різні типи функцій раундів, але вони схожі за структурою й різняться тільки у виборі виконуваної операції (додавання, вирахування або XOR) у різних місцях.

Хоча CAST-128 захищений патентом Entrust, його можна використовувати в усьому світі для комерційних або некомерційних цілей безкоштовно.

## Опис

CAST – це популярний 64-бітовий шифр, що допускає розміри ключа аж до 128 біт

Алгоритм CAST використовує 64-бітовий блок і 64-бітовий ключ. CAST стійкий до диференціального й лінійного криптоаналізу. Сила алгоритму CAST укладена в його S-блоках. В CAST немає фіксованих S-блоків і для кожного додатка вони конструюються заново. Створений для конкретної реалізації CAST S-блок уже більше ніколи не міняється. Інакше кажучи, S-блоки залежать від реалізації, а не від ключа. Northern Telecom використовує CAST у своєму пакеті програм Entrust для комп'ютерів Macintosh, PC і робочих станцій UNIX. Обрані ними S-блоки не опубліковані, що втім не дивно.

CAST-128 належить компанії Entrust Technologies, але є безкоштовним як для комерційного, так і для некомерційного використання. CAST-256 – безкоштовне доступне розширення CAST-128, яке ухвалює розмір ключа до 256

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55





$$\begin{aligned}
x_4x_5x_6x_7 &= z_0z_1z_2z_3 \wedge S_5[x_0] \wedge S_6[x_2] \wedge S_7[x_1] \wedge S_8[x_3] \wedge S_8[z_2] \\
x_8x_9xAxB &= z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1] \\
xCxDxEzF &= zCzDzEzF \wedge S_5[xA] \wedge S_6[x_9] \wedge S_7[xB] \wedge S_8[x_8] \wedge S_6[z_3] \\
K_5 &= S_5[x_3] \wedge S_6[x_2] \wedge S_7[xC] \wedge S_8[xD] \wedge S_5[x_8] \\
K_6 &= S_5[x_1] \wedge S_6[x_0] \wedge S_7[xE] \wedge S_8[xF] \wedge S_6[xD] \\
K_7 &= S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_8] \wedge S_8[x_9] \wedge S_7[x_3] \\
K_8 &= S_5[x_5] \wedge S_6[x_4] \wedge S_7[xA] \wedge S_8[xB] \wedge S_8[x_7] \\
z_0z_1z_2z_3 &= x_0x_1x_2x_3 \wedge S_5[xD] \wedge S_6[xF] \wedge S_7[xC] \wedge S_8[xE] \wedge S_7[x_8] \\
z_4z_5z_6z_7 &= x_8x_9xAxB \wedge S_5[z_0] \wedge S_6[z_2] \wedge S_7[z_1] \wedge S_8[z_3] \wedge S_8[xA] \\
z_8z_9AzB &= xCxDxEzF \wedge S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_5[x_9] \\
zCzDzEzF &= x_4x_5x_6x_7 \wedge S_5[zA] \wedge S_6[z_9] \wedge S_7[zB] \wedge S_8[z_8] \wedge S_6[xB] \\
K_9 &= S_5[z_3] \wedge S_6[z_2] \wedge S_7[zC] \wedge S_8[zD] \wedge S_5[z_9] \\
K_{10} &= S_5[z_1] \wedge S_6[z_0] \wedge S_7[zE] \wedge S_8[zF] \wedge S_6[zC] \\
K_{11} &= S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_8] \wedge S_8[z_9] \wedge S_7[z_2] \\
K_{12} &= S_5[z_5] \wedge S_6[z_4] \wedge S_7[zA] \wedge S_8[zB] \wedge S_8[z_6] \\
x_0x_1x_2x_3 &= z_8z_9AzB \wedge S_5[z_5] \wedge S_6[z_7] \wedge S_7[z_4] \wedge S_8[z_6] \wedge S_7[z_0] \\
x_4x_5x_6x_7 &= z_0z_1z_2z_3 \wedge S_5[x_0] \wedge S_6[x_2] \wedge S_7[x_1] \wedge S_8[x_3] \wedge S_8[z_2] \\
x_8x_9xAxB &= z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1] \\
xCxDxEzF &= zCzDzEzF \wedge S_5[xA] \wedge S_6[x_9] \wedge S_7[xB] \wedge S_8[x_8] \wedge S_6[z_3] \\
K_{13} &= S_5[x_8] \wedge S_6[x_9] \wedge S_7[x_7] \wedge S_8[x_6] \wedge S_5[x_3] \\
K_{14} &= S_5[xA] \wedge S_6[xB] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_6[x_7] \\
K_{15} &= S_5[xC] \wedge S_6[xD] \wedge S_7[x_3] \wedge S_8[x_2] \wedge S_7[x_8] \\
K_{16} &= S_5[xE] \wedge S_6[xF] \wedge S_7[x_1] \wedge S_8[x_0] \wedge S_8[xD]
\end{aligned}$$

половина, що залишається, ідентична тому, що дане вище, продовження від останнього створило  $x_0..x_f$ , щоб генерувати ключі  $K_{17} - K_{32}$ .

$$\begin{aligned}
z_0z_1z_2z_3 &= x_0x_1x_2x_3 \wedge S_5[xD] \wedge S_6[xF] \wedge S_7[xC] \wedge S_8[xE] \wedge S_7[x_8] \\
z_4z_5z_6z_7 &= x_8x_9xAxB \wedge S_5[z_0] \wedge S_6[z_2] \wedge S_7[z_1] \wedge S_8[z_3] \wedge S_8[xA] \\
z_8z_9AzB &= xCxDxEzF \wedge S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_5[x_9] \\
zCzDzEzF &= x_4x_5x_6x_7 \wedge S_5[zA] \wedge S_6[z_9] \wedge S_7[zB] \wedge S_8[z_8] \wedge S_6[xB]
\end{aligned}$$

$K17 = S5[z8] \wedge S6[z9] \wedge S7[z7] \wedge S8[z6] \wedge S5[z2]$   
 $K18 = S5[zA] \wedge S6[zB] \wedge S7[z5] \wedge S8[z4] \wedge S6[z6]$   
 $K19 = S5[zC] \wedge S6[zD] \wedge S7[z3] \wedge S8[z2] \wedge S7[z9]$   
 $K20 = S5[zE] \wedge S6[zF] \wedge S7[z1] \wedge S8[z0] \wedge S8[zC]$   
 $x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$   
 $x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$   
 $x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$   
 $xCxDxEzF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$   
 $K21 = S5[x3] \wedge S6[x2] \wedge S7[xC] \wedge S8[xD] \wedge S5[x8]$   
 $K22 = S5[x1] \wedge S6[x0] \wedge S7[xE] \wedge S8[xF] \wedge S6[xD]$   
 $K23 = S5[x7] \wedge S6[x6] \wedge S7[x8] \wedge S8[x9] \wedge S7[x3]$   
 $K24 = S5[x5] \wedge S6[x4] \wedge S7[xA] \wedge S8[xB] \wedge S8[x7]$   
 $z0z1z2z3 = x0x1x2x3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$   
 $z4z5z6z7 = x8x9xAxB \wedge S5[z0] \wedge S6[z2] \wedge S7[z1] \wedge S8[z3] \wedge S8[xA]$   
 $z8z9zAzB = xCxDxEzF \wedge S5[z7] \wedge S6[z6] \wedge S7[z5] \wedge S8[z4] \wedge S5[x9]$   
 $zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$   
 $K25 = S5[z3] \wedge S6[z2] \wedge S7[zC] \wedge S8[zD] \wedge S5[z9]$   
 $K26 = S5[z1] \wedge S6[z0] \wedge S7[zE] \wedge S8[zF] \wedge S6[zC]$   
 $K27 = S5[z7] \wedge S6[z6] \wedge S7[z8] \wedge S8[z9] \wedge S7[z2]$   
 $K28 = S5[z5] \wedge S6[z4] \wedge S7[zA] \wedge S8[zB] \wedge S8[z6]$   
 $x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$   
 $x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$   
 $x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$   
 $xCxDxEzF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$   
 $K29 = S5[x8] \wedge S6[x9] \wedge S7[x7] \wedge S8[x6] \wedge S5[x3]$   
 $K30 = S5[xA] \wedge S6[xB] \wedge S7[x5] \wedge S8[x4] \wedge S6[x7]$   
 $K31 = S5[xC] \wedge S6[xD] \wedge S7[x3] \wedge S8[x2] \wedge S7[x8]$   
 $K32 = S5[xE] \wedge S6[xF] \wedge S7[x1] \wedge S8[x0] \wedge S8[xD]$

### Маскування й перестановка підключів

$K_{m_1}, \dots, K_{m_{16}}$  32-розрядні підключи маскування (один на раунд).

$K_{r_1}, \dots, K_{r_{16}}$  32-розрядні перестановки підключів (один на раунд); тільки молодші 5 бітів використовуються в кожному раунді.

```
for (i=1; i<=16; i++) {  $K_{m_i} = K_i$ ;  $K_{r_i} = K_{16+i}$ ; }
```

### Змінний розмір ключа

CAST-128 Алгоритм шифрування був розроблений, щоб розмір ключа міг варіюватися від 40 до 128 біт, в 8-бітному кроці (тобто припустимі розміри ключа рівняються 40, 48, 56, 64..., 112, 120, і 128 бітам). Для змінної роботи розміру ключа специфікація наступні:

- 1) Для розмірів ключа до й включаючи 80 бітів (тобто, 40, 48, 56, 64, 72, і 80 бітів) алгоритм точно такої ж, але використовує 12 раундів замість 16;
- 2) Для розмірів ключа більше, чим 80 бітів, алгоритм використовує повні 16 раундів;
- 3) Для розмірів ключа менше, чим 128 бітів ключ доповнений нульовими байтами (у самих правих, або молодших, позиціях) до 128 біток (тому що розклад ключа CAST 128 ухвалює вхідний ключ 128 бітів).

### Розшифрування

Розшифрування для CAST-128 відносно проста. Розшифрування працює в тому ж алгоритмічному напрямку, що й шифрування, починаючи із зашифрованого тексту як вхідних даних. При цьому підключ використовуються у зворотному напрямку.

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення призначене для автоматизації обробки даних наукових досліджень та дозволяє автоматизувати роботу на дослідницьких, випробовувальних, технологічних та контрольно-діагностичних установках. Програма здатна функціонувати на PC-сумісних комп'ютерах, оснащених засобами збору даних. Можливості програми по кількості, змісту й характеристикам вимірювальних каналів залежать від використовуваних пристроїв збору даних (ПЗД) і продуктивності комп'ютера.

Програма має зручний та інтуїтивно зрозумілий інтерфейс. Головне вікно програми зображене на рисунку 5.1.

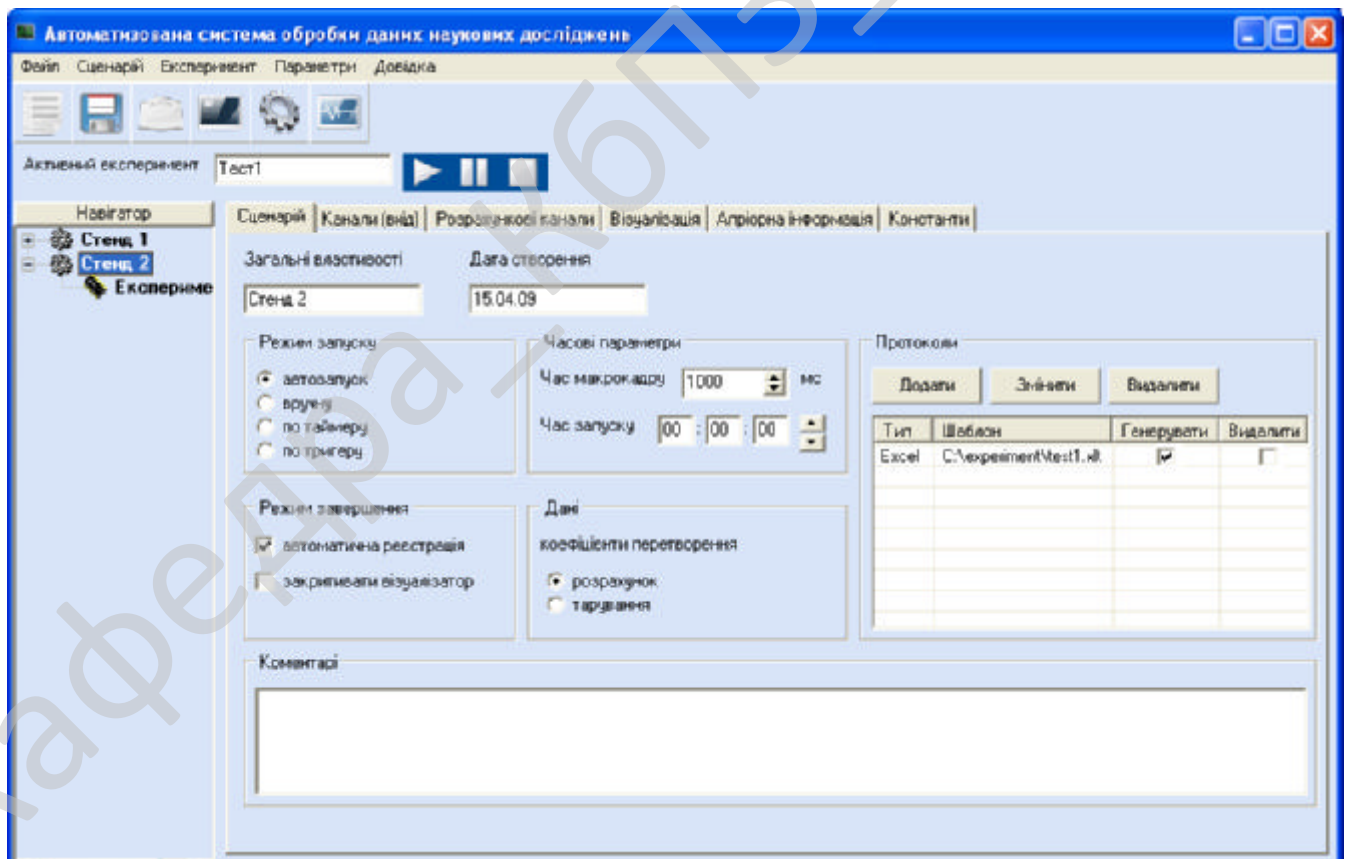


Рисунок 5.1 – Головне вікно програми



запуск» установлений, необхідно натиснути кнопку «Запустити експеримент» у верхній панелі керування.

4. Одночасно із проведенням вимірів буде проводитися математична обробка, візуалізація й реєстрація даних у відповідності зі сценарієм експерименту. Вимірювання завершаються автоматично, якщо це передбачено сценарієм, або вручну – натисканням кнопки «Стоп» у панелі керування збором даних.

5. Для реєстрації даних експерименту в базі даних необхідно натиснути на кнопку «ОК» у вікні введення післясеансової інформації. При цьому експерименту привласнюється статус «завершений». Післясеансова текстова інформація може вноситися або не вноситися. Якщо в налагодженнях сценарію експерименту не зазначено «Закривати візуалізатор після завершення», вікно візуалізації реального часу автоматично не закривається, і дані, що утримуються в ньому, можна продивитися по закінченню вимірювань. Для закриття вікна в ручному режимі необхідно зробити його активним і натиснути комбінацію клавіш Alt+F4, або закрити за допомогою контекстового меню панелі завдань.

Обробка й аналіз отриманих даних проводяться за допомогою модуля післясеансової обробки даних. Для обробки результатів вимірювань і створення звіту необхідно виконати наступні дії:

1. Створити монітор для перегляду даних (залежність параметрів від часу або параметрична залежність).
2. Вибрати дані вимірювань з активного експерименту, з бази даних результатів або з файлу.
3. Провести аналіз, маркерні вимірювання або необхідну математичну обробку.
4. Результати вивести на принтер, графічні залежності – записати у файл звіту.

### **Механізм збору даних**

Збір даних здійснюється протягом усього експерименту. Але на диску

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

зберігаються тільки дані за останній проміжок часу тривалістю, зазначеною в загальних параметрах сценарію в полі тривалість експерименту. Тривалість експерименту повинна перевищувати час макрокадру.

Макрокадр – це час збору безперервного блоку даних зі всіх пристроїв, рівний часу заповнення половини буфера. Механізм збору даних організований за принципом подвійної буферизації. Буфер, що представляє собою область, виділену в оперативній пам'яті, ділиться на дві половини. Спочатку дані записуються в першу частину буфера. Як тільки перша половина буфера заповнена, дані з неї відправляються на обробку й зберігаються на диску, а наступні дані починають заноситися в другу половину буферу. При заповненні другої половини починає знову заповнюватися перша порожня провина, а дані із другої – ідуть на обробку. Таким чином, процес заповнення половин буфера йде по колу. Обробка однієї половини буфера й запис в іншу відбуваються одночасно. Рекомендований час макрокадру лежить у діапазоні від 200 мс до 2 с (час 0,2 с визначається часом реакції Windows, час 2 с – об'ємом буферу плати збору).

Коротку довідку про розроблену програму та її автора можна переглянути вибравши пункт «Про програму...» з меню «Довідка», після чого з'явиться вікно зображене на рисунку 5.2.

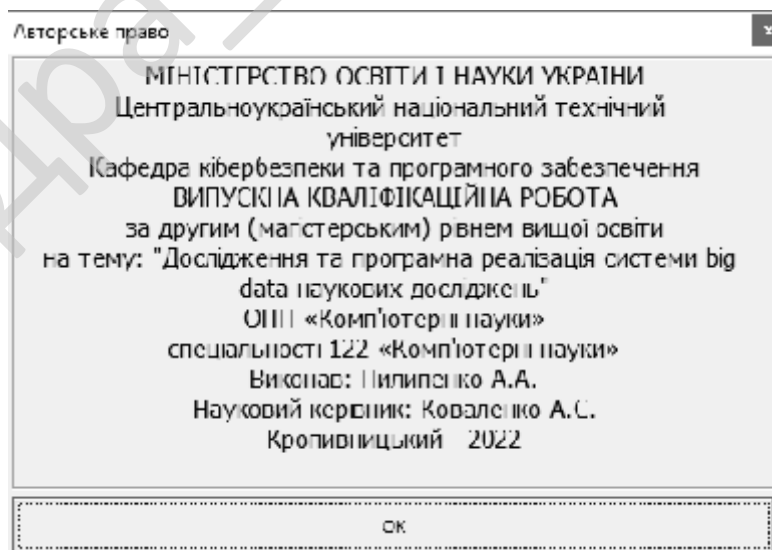


Рисунок 5.2 – Довідка про програму

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи big data наукових досліджень.

*Метою розробки є дослідження та програмна реалізація системи big data наукових досліджень.*

*Об'єктом дослідження є процес big data наукових досліджень.*

*Предметом дослідження є методи big data наукових досліджень.*

*Методи дослідження базуються на методах big data, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод big data наукових досліджень.
- Розроблено вітчизняний продукт big data наукових досліджень, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

## 7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

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

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи big data наукових досліджень.

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	70
3. Запланований термін розробки, днів	Fpq	24 (1 місяць)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Г
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

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

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	70000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	37
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	30
38. Ставка податку на додану вартість, %	Ндв	20

## 7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

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





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	14	1260	21
Монітор	60	14	840	14
Клавіатура	30	14	420	7
Маніпулятор «мишка»	10	14	140	2,33
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	2	40	0,67
Концентратор-маршрутизатор	30	3	90	1,5
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>	64,25

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{\text{ор}}^c = \frac{64 \cdot 1}{1,2} = 53 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 53/(24 \cdot 8) = 0,3 \text{ ставки.}$$

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

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 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 (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	2	0,5
	Створення графічних і стилістичних елементів сайту	1	
	Оформлення банерів і промо-сторінок	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		4	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	14653	14653
Продакт-менеджер	0,5	10000	5000
Інженер-програміст	6,8	11200	76160
Інженер-електронщик	0,3	9000	2700
Інженер-системотехнік	0,25	9000	2250
Адміністратор мережі	0,5	9000	4500
Системний програміст	0,25	9000	2250
Дизайнер WEB	0,5	9000	4500
Інженер-верстальник	0,25	9000	2250
Бухгалтер-економіст	0,5	10000	5000
Всього за період розробки	$R_{cn} = 10,85$	-	$\Phi_{роб} = 119263$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{119263}{10,85 \cdot 24} = 458 \text{ грн.}$$

#### 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-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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

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

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

$\Pi_{nl}$  – 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 170...1600 у.о./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 6560 грн./ $m^2$ . На кожне робоче місце у середньому потрібно 8  $m^2$ . З урахуванням цього:

$$B_{y\partial} = 13 \cdot 8 \cdot 6560 = 682240 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 68224 грн.

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом Інтернет магазину Компбест за 28.10.2022 – джерело <https://compbest.com.ua>.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок	VINGA ADVANCED B0166 (R3M8INT.B0166)	7347
Процесор	AMD Ryzen 3, 2100GE, 4 ядра, 4 потоки Частота процесора, 3.2 ГГц, Частота в Boost 3.6 ГГц	-
Системна плата	MB AM4, Чіпсет AMD A320, 1 x Headphone 4 x USB 3.0, 3 x Audio, 1 x Microphone, 4 USB 2.0, 2 x PS/2, 1 x HDMI, 1 x VGA, 1 RJ45, Realtek ALC887, 10/100/1000 Мбіт/с	-
Відеокарта	Вбудована AMD Radeon Vega 8	-
Жорсткий диск	SSD 120 GB	-
Оперативна пам'ять	8 ГБ DIMM, DDR4-2666 MHz, PC4-21300	-
DVD-привод	Не комплектується	-
Корпус	ATX Vinga CS108B, PSU 350W(FSP Brand ATX-400PNR, 12cm), black, (front bezel black+light silver; body material – 0.6mm) 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermal Advantaged Chassis	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: A Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	PHILIPS 223V5LSB2/10, 21.5", TN, 1920 1080, 16:9, WLED, матове покриття, 5мс	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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

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

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

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	682240	-	-
2. Передавальні пристрої	68224	-	-
Всього по групі	750464	5	37523,2
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	70000	50	35000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	45500	25	11375
Всього по групі	197531	-	42232,75
Разом	$K_p = 1217172$		$A_p = 214344,45$

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



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

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

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

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

$$З_{M2} = \sum Ц_d, \quad (7.17)$$

де:  $Ц_d$  – вартість дисків CD/DVD: CDR 52x Cake box – 30 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 33 грн./шт.

$$З_{M2} = 30 \cdot 2 + 33 = 93 \text{ грн.}$$

Згідно виданих викладачем 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: картридж для CANON LBP-3010 Black Canon 712 – 574 грн.; картридж для EPSON STYLUS PHOTO R390 – 558 грн.; картридж для CANON IR-1022A – LJ Q2612A Cart. HP LJ 1010/1012/1015/3015/3020/3030 (2500 стр.) – 570 грн.

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

$$З_M = (40 + 93 + 1702) / 70 = 26 \text{ грн.}$$

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, %.

$$O_n = 936 \cdot 15 \cdot 0,01 = 140 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 70$  прим.):

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>80</b>

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

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

$$A_m = 214344 \cdot 1 / (70 \cdot 12) = 255 \text{ грн.}$$

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

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

$$C_n = 936 + 93,6 + 381 + 140 + 26 + 140 + 255 = 1971,6 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 30%.

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

де:  $P_n$  – рівень рентабельності, %.

$$P_p = 0,01 \cdot 30 \cdot 1971,6 = 591 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	$Z_o$	936
2. Додаткова зарплата виконавців	$Z_d$	93,6
3. Відрахування на соціальні потреби	$C_{oc}$	381
4. Загальногосподарські витрати	$\Gamma_{ocn}$	140
5. Витрати на матеріали	$Z_m$	26
6. Освоєння нових операційних систем, мов програмування	$O_n$	140

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	$A_m$	255
8. Повна собівартість програмного забезпечення	$C_n$	1971,6
9. Плановий прибуток	$P_p$	591
10. Ціна підприємства $C_n = C_n + P_p$	$C_n$	2562,6
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	512,5
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	$C$	3075,1

**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	–	3075
Всього капітальних витрат	–	3075

## 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	$Z_p$	25122	16748
2. Витрати на електроенергію	$Z_{ел}$	421	263
3. Витрати на амортизацію	$Z_{ам}$	0	1538
Всього витрат за рік	$I$	25543	18549

Витрати на обслуговування системи:

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

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

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 160 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 78 \cdot 1,1 \cdot 1,22 = 25122 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 160 \cdot 78 \cdot 1,1 \cdot 1,22 = 16748 \text{ грн}.$$

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

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

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>83</b>

$$Z_{ел\ баз} = 7 \cdot 0,15 \cdot 240 \cdot 1,67 = 421 \text{ грн.}$$

$$Z_{ел\ нов} = 7 \cdot 0,15 \cdot 150 \cdot 1,67 = 263 \text{ грн.}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	3075	–	1537,5
Всього відрахувань	-	–	3075	–	1537,5

### 7.8 Визначення економічної ефективності програмної продукції

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

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

$$E_e = (C_n - C_n) \cdot N_e - E_n \cdot K_p, \quad (7.25)$$

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

$$E_e = (2562,6 - 1971,6) \cdot 70 - 0,15 \cdot 1217172 \cdot 1/12 = 26155,35 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

$$T_e = \frac{1217172}{(2562,6 - 1971,6) \cdot 70 \cdot 12/1} = 2,5 \text{ року.}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

$$E_{cn} = (25543 - 18549) - 0,5 \cdot 3075 = 5456 \text{ грн.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	70
2. Повна собівартість розробленої програми	Грн.	1971,6
3. Ціна розробленої програми	Грн.	2562,6
4. Плановий прибуток від реалізації розробленої програми	Грн.	591
5. Рентабельність програмної продукції	%	30
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1217172
7. Загальний прибуток від реалізації програмної продукції	Грн.	41370
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	26155
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	2,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3075
11. Величина економічного ефекту у користувача програмної продукції	Грн.	5456
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,44

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

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

$$T_{cn} = \frac{3075}{25543 - 18549} = 0,44 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

## 7.9 Висновки

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

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Умови праці програміста включають наступні фактори:

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

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

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

Електронно-обчислювальна машин (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.

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

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

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат;

### 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	3
Довжина	4,6
Висота	3

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого\*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м <sup>2</sup>	не менше 6.0	6,9
Об'єм, V	м <sup>3</sup>	не менше 20.0	20,7

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

У зазначеному приміщенні працюють двоє людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення

у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»). Таким чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Ia. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

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

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Ia, так і розглянутого приміщення. У

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90



приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

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

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

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

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

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

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92



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

Обчислимо індекс приміщення за формулою:

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

де:

$S$  – площа приміщення,  $S = 13,8 \text{ м}^2$ ;

$h$  – розрахункова висота підвісу,  $h = 3 \text{ м}$  (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

$A$  – ширина приміщення,  $A = 3 \text{ м}$ ;

$B$  – довжина приміщення,  $B = 4,6 \text{ м}$ .

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$i = 0,43.$$

Знаючи індекс приміщення, за знаходимо  $n = 0,23$  (з табличних даних коефіцієнтів використання світлового потоку ( $n$ ) світильників з відповідним типом лампам). Підставимо всі значення у формулу, визначимо світловий потік:  $F = 29700 \text{ Лм}$ .

Для розрахунку дудемо використовувати світлодіодні панелі *LED панель 42Вт 6000К SUNLED 000000127*, світловий потік яких  $F_{\text{л}} = 3990 \text{ Лм}$ .

Число ламп визначається по формулі:

$$N = F / F_{\text{л}}$$

де:

$F$  – світловий потік,

$F_{\text{л}}$  – світловий потік однієї лампи.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$N = 29700 / 3990 = 7,4 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 8 шт.

### 8.6 Висновки до розділу

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

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи big data наукових досліджень.

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 big data наукових досліджень.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем big data наукових досліджень.
- Досліджена система big data наукових досліджень.
- На основі отриманих результатів досліджень створена програмна реалізація системи big data наукових досліджень.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання big data наукових досліджень.

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

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

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

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

Програма реалізована на мові високого рівня Delphi 10.4 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм CAST-128.

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 5456 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,44 роки.

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пилипенко А.А. Дослідження та програмна реалізація системи big data наукових досліджень // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.
2. Данько В.М. Теорія експерименту. Текст лекцій. Алчевськ: «Ладо», 2005-с.117
3. Статистические методы в инженерных исследованиях. /лаб. практикум под ред. Круга Г.К.– М.: Высшая школа, 1983.-216с.
4. Э. Фёрстер, Б. Рёнц. Методы корреляционного и регрессионного анализа.– М.: Финансы и статистика, 1983.-302с.
5. Шенк Х. Теория инженерного эксперимента. М.: Мир, 1972.– 380с.
6. Грановский В.А., Сирая Т.Н. Методы обработки экспериментальных данных при измерениях. Л.: Энергоатомиздат, 1990.– 288с.
7. Египко В. М., Организация и проектирование систем автоматизации научно-технических экспериментов, К., 1978.
8. 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. (Scopus).*
9. 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. (Scopus).*
10. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskiy, V., Khorolska, K., Bebesko, 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.*

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

*Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. **Springer**, Singapore. pp. 21-34. **(Scopus)**.

11. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. **Springer**, Cham. 2022, pp. 2463-2477. **(Scopus)**.

12. 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> **(Scopus)**.

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

14. 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. **(Scopus)**.

15. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages 44-58. **(Scopus)**.

16. 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. **(Scopus)**.

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

18. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and

					<b>БКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>99</b>

cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346. **(Scopus)**.

19. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131. **(Scopus)**.

20. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14. **(Scopus)**.

21. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. **Springer**, Cham. 2021, pp 66-84. **(Scopus)**.

22. 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. **(Scopus)**.

23. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136. **(Scopus)**.

24. 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. **(Scopus)**.

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

26. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated

					BKPM-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645. **(Scopus)**.

27. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 646-660., **(Scopus)**.

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

29. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019. **(Scopus)**.

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

31. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings* Volume 2353, *CEUR Workshop Proceedings* 2019, Pages 618-629. **(Scopus)**.

32. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», *CEUR Workshop Proceedings* Volume 2353, *CEUR Workshop Proceedings* 2019, Pages 873-884. **(Scopus)**.

33. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». *ISCI'2020: Information Security in Critical Infrastructures. Collective monograph*. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).

					<b>BKPM-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>101</b>

34. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. **Collective monograph**. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

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

36. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

37. Смірнов О.А., Дреєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у *Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка*. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.

38. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования WEB-приложений. *Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка*. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.

39. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. *Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка*. – Х.: Видавець Рожко С.Г., 2017. – 447 с.

40. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в

інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98. 2022. (Фахове видання. Категорія «Б»)

41. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

42. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

43. 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

44. Смирнов А., Кузнецов А., Кузнецова Т. «Шумоподобные дискретные сигналы для асинхронных систем кодового разделения радиоканалов». *Радиотехника*, № 2(205), 175–183. 2021.

45. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». *CEUR Workshop Proceedings Volume 2732*, 2020, Pages 214-227.

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

47. Смирнов А.А, Кузнецов А.А., Киян А.С., Кузнецова Е.А. «Соккрытие данных на основе адресации шумоподобных сигналов». *Всеукраїнський*

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

міжвідомчий науково-технічний збірник "Радіотехніка" – Харків: ХНУРЕ. – 2020. – Вип. 203. – С. 38-49.

48. Смирнов А.А., Дудан А.В., Смирнова Т.В. «Формализация структуры технологического процесса электродугового напыления». *Сборник научных трудов «Актуальные вопросы машиноведения»*. Объединенный институт машиностроения Национальной Академии Наук Беларуси. №9. С. 308-312, 2020.

49. 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.

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

51. А.А. Смирнов, Т.В. Смирнова, А.Н. Дреев, А.В. Дудан. «Оптимизация технологического процесса восстановления и упрочнения поверхностей с заданными характеристиками в виде облачного сервиса». *Вестник Полоцкого государственного университета. Серия В, Промышленность. Прикладные науки*. Республика Беларусь – 2020. – № 3. – С. 50-61.

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

53. О.А. Смірнов, Т.В. Смірнова, О.М. Дреєв, Є.К. Солових, «Методи оптимізації технологічних процесів відновлення сталевих покриттів», *Shipbuilding & marine infrastructure / Суднобудування і морська інфраструктура* № 1 (11). с. 48-57, 2019.

54. Смірнов О.А., Дреєва Г.М., Дреєв О.М., «Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та

зміцнення поверхонь деталей». Центральнуукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

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

56. Смірнов О.А., Смірнова Т.В., Дреєв О.М., «Експертна система оптимізації процесу відновлення та зміцнення поверхонь деталей типу «вал» електродуговим напиленням», Системи управління, навігації та зв'язку, № 2 (54). с. 149-154, 2019.

57. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

58. Смирнов А.А., Лысенко И.А., Информационная технология проектирования тестовых наборов на основе требований к программному обеспечению, Системи управління, навігації та зв'язку. – Випуск 4 (44). – Полтава: ПолтНТУ. – 2017. – С. 112-115.

59. Смірнов О.А., Мелешко Є.В., Хох В.Д., Дослідження методів аудиту систем управління інформаційною безпекою, Системи управління, навігації та зв'язку. – Випуск 1 (41). – Полтава: ПолтНТУ. – 2017. – С. 38-42.

60. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

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

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

					ВКРМ-122.22.0009.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

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

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

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

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

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

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

					<b>ВКРМ-122.22.0009.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

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

					<b>ВКРМ-122.22.0009.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Пилипенко А.А.				Дослідження та програмна реалізація системи big data наукових досліджень	Літ.	Аркуш	Аркушів
Перевірів	Коваленко А.С.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КН-21М-1,4			
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи big data наукових досліджень.

## 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. № 18-13 від 17.08.2022 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи big data наукових досліджень.

## 4 Джерела розробки

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

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-122.22.0009.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи big data наукових досліджень;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

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

## 5.4 Вимоги до архітектури

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

## 5.5 Вимоги до надійності

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

					<b>ВКРМ-122.22.0009.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

					ВКРМ-122.22.0009.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

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

## 7 Економічні вимоги

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

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

## 8 Вимоги щодо охорони праці

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

					ВКРМ-122.22.0009.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

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

## 10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2022 р.

					<b>ВКРМ-122.22.0009.00.00.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

\_\_\_\_\_ Коваленко А.С.

*Дослідження та програмна реалізація  
системи big data наукових досліджень*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 38

Літера: РП

Кропивницький – 2022 року

## Файл Main.pas основної програми

```

unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, TeeProcs, TeEngine, Chart, DbChart, Series,
  Math_exp, About;

procedure ConvertToHex(var tmin:integer;var tmax:integer;var Buffer:array of
byte;var Length: integer);stdcall;
procedure ConvertToDec(var Buffer1:array of byte;var Buffer2:array of real;var
Length1:integer;var Length2: integer);stdcall;

type
  TVAX_Tran = class(TForm)
    btnConnect: TButton;
    VaxChart: TDBChart;
    tokEditMin: TEdit;
    tokEditMax: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    btnStart: TButton;
    btnReset: TButton;
    btnSave: TButton;
    btnGraf: TButton;
    ValueMemo: TMemo;
    btnExit: TButton;
    Label3: TLabel;
    Label4: TLabel;
    baseTokEdit: TEdit;
    Series1: TPointSeries;
    Series2: TPointSeries;
    Series3: TPointSeries;
    Series4: TPointSeries;
    procedure btnExitClick(Sender: TObject);
    procedure btnResetClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnConnectClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure btnStartClick(Sender: TObject);
    procedure btnGrafClick(Sender: TObject);
    procedure btnSaveClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  VAX_Tran: TVAX_Tran;
  FormRef: longint; //змінна ініціалізації бібліотеки
  SeriesNom: integer; //змінна для вибору графіка

implementation

{$R *.dfm}
//Функції бібліотеки для зв'язку із приладом по протоколі Modbus
function InitLib(): longint; external 'PIC18_Modbus.dll';
procedure comm_setParam; external 'PIC18_Modbus.dll';
procedure CloseLib; external 'PIC18_Modbus.dll';
function read_RAM(CID:byte;var bufData:array of
byte;adrRAMH:word;adrRAML:word;len:byte): word; external 'PIC18_Modbus.dll';
function write_RAM(CID:byte;bufData:pointer;adrRAMH:word;adrRAML:word;len:byte):
word; external 'PIC18_Modbus.dll';

```

```

//Вихід із програми
procedure TVAX_Tran.btnExitClick(Sender: TObject);
begin
    Close;
end;

//Скидання значень
procedure TVAX_Tran.btnResetClick(Sender: TObject);
begin
    tokEditMin.Text:='0';
    tokEditMax.Text:='20';
    baseTokEdit.Text:='0';
    ValueMemo.Lines.Clear;
    VaxChart.Series[0].Clear;
    VaxChart.Series[1].Clear;
    VaxChart.Series[2].Clear;
    VaxChart.Series[3].Clear;
    VaxChart.Series[0].Title:='-----';
    VaxChart.Series[1].Title:='-----';
    VaxChart.Series[2].Title:='-----';
    VaxChart.Series[3].Title:='-----';
    SeriesNom:=0;
end;

procedure TVAX_Tran.FormCreate(Sender: TObject);
begin
    ValueMemo.Lines.Clear;
    VaxChart.Title.Text.Clear;
    VaxChart.Title.Text.Add('Графік вихідний ВАХ транзистора');
    SeriesNom:=0;
end;

//Установлення зв'язку із приладом
procedure TVAX_Tran.btnConnectClick(Sender: TObject);
begin
    FormRef:=InitLib(); //Підключення бібліотеки
    comm_setParam; //Установка параметрів COM порту
    if FormRef=0 then showMessage('Помилка установки зв'язку із приладом')
    else showMessage('Зв'язок із приладом установлений');
    btnConnect.Enabled:=False;
end;

procedure TVAX_Tran.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    if FormRef>0 then CloseLib; //Закриття COM порту при виході із програми
end;

procedure TVAX_Tran.btnStartClick(Sender: TObject);
type pBwr=^byte;
    p=^byte;
var tokMin,tokMax,lenWr,Len,n,i,tokBase: integer;
    BufWr,BufRd,temp:array [0..511] of byte; //Буфери запису/читання й
    тимчасовий буфер
    Bwr: pBwr;
    B: p;
    ResBuf: array [0..255] of real; //буфер результатів
begin
    ValueMemo.Lines.Clear;
    tokBase:=strtoint(baseTokEdit.Text); //значення базового струму
    tokMin:=strtoint(tokEditMin.Text); //початкове значення струму уведене
    користувачем
    tokMax:=strtoint(tokEditMax.Text); //кінцеве
    //перевірка уведених значень
    if((tokMin<0) or (tokMin=tokMax)) or ((tokMax<tokMin) or (tokMax>20)) then
    begin
        showMessage('Невірний діапазон уведених значень');
        exit;
    end;
end;

```

```

if (tokBase<0) or (tokBase>20000) then
begin
  showMessage('Невірний діапазон уведених значень');
  exit;
end;
//посилка на прилад значення базового струму
n:=65535*tokBase div 20000;
BufWr[0]:=lo(n);
BufWr[1]:=hi(n);
B:=@(BufWr[0]);
write_RAM($02,B,$00,$02,$02);
sleep(50);
ConvertToHex(tokMin,tokMax,BufWr,LenWr); //одержання буфера запису
n:=0;
Bwr:=@(BufWr[0]);
while n<LenWr+1 do
begin
  write_RAM($02,Bwr,$00,$00,$02); //запис даних на прилад
  sleep(50); //затримка перед читанням даних із приладу
  read_RAM($02,BufRd,$00,$04,$02); //читання даних із приладу
  for i:=0 to 1 do temp[i+n]:=bufrd[i]; //збереження отриманих
значень у масив
  //for i:=0 to 3 do ValueMemo.Lines.Add(inttohex(BufRd[i],4));
  //ValueMemo.Lines.Add('=====');
  inc(Bwr,2);
  n:=n+2;
end;
ConvertToDec(Temp,ResBuf,LenWr,Len); //одержання остаточних результатів
for n:=0 to Len do ValueMemo.Lines.Add(FloatToStr(ResBuf[n],ffFixed,3,3));
//вивід значень
showMessage('Виміри кінчені');
end;

{*****
Процедура перетворення значень уведених користувачем (tokMin і tokMax)
у масив шіснадцяткових чисел(Buffer) для запису в RAM,
Length - довжина масиву. Buffer і Length - вихідні значення.
*****}
procedure ConvertToHex(var tmin:integer; var tmax:integer;var Buffer:array of
byte;var Length: integer);
var i:integer;
    min,max,n:word;
begin
  min:=65535*tmin div 20;
  max:=65535*tmax div 20;
  n:=min;
  i:=0;
  while n<=max+1 do
  begin
    Buffer[i]:=lo(n);
    Buffer[i+1]:=hi(n);
    n:=n+256;
    i:=i+2;
    if i>=511 then break;
  end;
  Length:= i-1;
end;

{*****
Процедура перетворення масиву даних зчитаних із приладу(Buffer1)
у масив десяткових чисел(Buffer2) для подальшої побудови графіка.
Length1 - довжина масиву Buffer1, Length2 - довжина масиву Buffer2.
Buffer2 і Length2 - вихідні значення.
*****}
procedure ConvertToDec(var Buffer1:array of byte;var Buffer2:array of real;var
Length1:integer;var Length2: integer);
var i,n:integer;
begin
  n:=0;

```

```

    i:=0;
    while n<=Length1 do
    Begin
        Buffer2[i]:=((5*Buffer1[n])/$FF30)+((5*256*Buffer1[n+1])/$FF30);
        n:=n+2;
        i:=i+1;
    end;
    Length2:= i-1;
end;

//Вивід графіка вихідний ВАХ
procedure TVAX_Tran.btnGrafClick(Sender: TObject);
var n:integer;
    i:real;
    //B_data:TStringList;
begin
    //B_data:=TStringList.Create;
    //B_Data.LoadFromFile('Results.txt');
    if SeriesNom>3 then SeriesNom:=0;
    i:=0;
    for n:=0 to ValueMemo.Lines.Count-1 do
    Begin
        VaxChart.Series[SeriesNom].Title:=baseTokEdit.Text+' мкА';

VaxChart.Series[SeriesNom].AddXY(strtfloat(ValueMemo.Lines[n]),i,'',{,Color});
        //Вивід графіка по значанням з таблиці
        i:=i+0.078;
    end;
    SeriesNom:=SeriesNom+1;
    {if (B_data.Count>0) then //перевірка наявності даних у файлі
    begin
        for n:=0 to B_Data.Count-1 do
            VaxChart.Series[0].AddXY(strtfloat(B_data.Strings[n]),n);
        //Вивід графіка за значеннями з файлу
        end;}
end;

//Збереження отриманих результатів у файл
procedure TVAX_Tran.btnSaveClick(Sender: TObject);
var B_data:TStringList;
    n:integer;
begin
    B_data:=TStringList.Create;
    if SeriesNom>3 then SeriesNom:=0;
    B_data.Add('Вихідна напруга для базового струму в '+baseTokEdit.Text+'
мкА');
    for n:=0 to ValueMemo.Lines.Count-1 do
        B_data.Add(ValueMemo.Lines[n]);
    B_data.SaveToFile('Results'+inttostr(SeriesNom)+'.txt');
    //ValueMemo.Lines.SaveToFile('Results'+inttostr(SeriesNom)+'.txt');
end;
end.

```

**Файл Math\_exp.pas - обробка отриманих у ході експерименту даних та математичні розрахунки**

```

unit Math_exp;

interface

uses
    Windows, SysUtils, Classes, Dialogs, Math, Forms, Controls, StdCtrls;

const _PI = 3.141592653589793238462643;

```

```

const  PI_2: extended = _PI/2;
const  _LDSF = 3;
const  _LDSC = 4;
const  _FUNC = 1;
const  _COP  = 2;

type
TAddress = Cardinal;

type TFloatType = double;

type PFloatType = ^TFloatType;

type
TArray4 = array [1..4] of byte;
type TCode = array of Byte;
type TArray = array of TFloatType;
type PArray = ^TArray;
type TAddFunc = function(N: Integer; SF: TArray): TFloatType;

type TArray = array of Integer;
type TArray = array of Cardinal;

type TIntType = array of Cardinal;

type TIntFunc  = array of
    record
        Type:  Cardinal;           //повертається тип ф-ії
        Addr: Cardinal;           //адреса ф-ії
        N:     Integer;            //N=-1 Infinite
        TV:   array of Cardinal;   //масив типів змінних (Size=N)
    end;

type TAlgFunction = array of
    record
        Name: String;             //ім'я ф-ії
        Func: TIntFunc;           //характеристики ф-ій з даним ім'ям
    end;

type TAlgOperation = array of
    record
        PR: Cardinal;             //пріоритет операції (крім '0'!)
        Name: String;             //ім'я символічної операції
        NFunc: Cardinal;          //номер масиву відповідної ф-ії
        GS: Integer;              //для одноарг. операцій:
        розташування
        //((-1)спереду; (-2)позаду)
    end;
    //для багатоарг. операцій: >= 0 !!!

    //пріоритет кожної одноарг. операції вище кожної
багатоарг. операції !!!

```

```

type TAlgObject = array of
    record
        Name: String;
        Type: Cardinal;           //тип змінної
        Addr: Cardinal;
    end;

type TAlgLoadFunction = array of
    record
        Type: Cardinal;         //тип змінної для завантаження
        Addr: Cardinal;
    end;

// Bri = 1 якщо ф-ія встановлена й NFi - номер масиву
type
TAlgBracket = record
    Br1: Integer;             // (,) не використовується !!!
    NF1: Cardinal;
    Br2: Integer;           // [,]
    NF2: Cardinal;
    Br3: Integer;           // {,}
    NF3: Cardinal;
    Br4: Integer;           // <, >
    NF4: Cardinal;
end;

type
PFunc = ^TFunc;

TFunc = record
    NL: Cardinal;           //порядковий номер команди на рівні
    S: Cardinal;           //Stack
    N: Cardinal;           //Node
    Type: Cardinal;        //повертається тип
    COP: Cardinal;
    NFunc: Cardinal;       //номер масиву ф-ій
    NF: Cardinal;          //номер ф-ії в масиві
    NObj: Cardinal;        //номер масиву змінних
    FD: Double;            //константа
    Arg: Cardinal;         //число змінних (для infinite)
    Id: Cardinal;
    Id: Cardinal;
end;

type
TArrayOfPFunc = array of PFunc;

type
TFunction = record
    Interp: TArrayOfPFunc;
    Comp: TCode;
    ICode: TAddress;
end;

type PFunction = ^TFunction;

type TNStack = record
    S: Cardinal; //поточний номер стека
    N: Cardinal; //число вузлів, що впливають
end;

type

```

```

PNode = ^TNode;
TNode = record
    Stack: TNStack;
    Id: Cardinal;
    Id: Cardinal;
    NL: Cardinal; //порядковий номер команди (складова
рівня);
    NCOP: Cardinal; //операція (номер масиву) (складова вузла);
    NFunc: Cardinal; // ф-ія (номер масиву ф-ій)
    NObj: Cardinal; //номер масиву змінної
    COP: Integer; //COP: = 1( ф-ія) в Func номер масиву ф-ій;
//2(операція) в NCOP номер масиву;
//3(завантаження змінної) в NObj - номер
масиву;
//4(завантаження константи) в FD -
константа;
    FD: Double; //константа
    Expression: String;
    Ch: PNode;
    Nb: PNode;
    Pr: PNode;
end;

type
TDataNode = record
    N: Cardinal;
    Id: Cardinal;
    NCOP: Cardinal;
    NFunc: Cardinal;
    NObj: Cardinal;
    COP: Integer;
    FD: Double;
    FX: Extended;
    Expression: String;
end;

type
TLevel = record
    NL: Cardinal; //порядковий номер команди (складова
рівня);
    Id: Cardinal;
    Expression: String;
end;

type PLevel = array of TLevel;

type
TArray = array of String;

type
TParamList = record
    Name: String;
    Value: TFloatType;
end;

{type
TArray = array of Integer;
}

type
TString1 = String[1];
type
TAlgebra = record
    Name: String;
    Types: TArray;
    Obj: TAlgObject;
    Opr: TAlgOperation;
    ExFunc: TAlgFunction;
    LdFunc: TAlgLoadFunction;

```

```

Bracket: TAlgBracket;
end;

type TExternalException = procedure (TypeError: Integer); {of object;} stdcall;

type

  TForeval = class

  private

    Alg: TAlgebra;
    CurrentExpression: String; //поточне вираження
    LFunc : Cardinal; //довжина скомпільованого вираження
    StackSize: Cardinal; //розмір стека
    Arg : TArray; //список аргументів у випадку > 2 арг.(відлік з
'1')
    Func: TFunction; //виконується код, що, скомпільованого
вираження
    Tree: PNode; //дерево розбору вираження
    //F_SyntaxExtension: Boolean; //вмик./вимк.: імена ф-ій > 1; '{' ;
'['; '!'; '^'; .
    BeginTree: PNode; //показчик початку дерева (не обов'язково)
    F_VarShift: Boolean; //розпізнавання var&param у
верхньому/нижньому/обох регістрах.
    F_FuncShift: Boolean;
    F_SyntaxErrorCode: Cardinal; //код помилки при розборі
    F_SyntaxErrorString: String; //рядок помилки
    F_CalcErrorCode: Integer; //вчислит. помилка
    F_InternalError: Integer;
    F_ShowException: Boolean; //вмик./вимк. показ виключень
    F_ExternalException: Boolean; //вмик./вимк. зовнішнього оброблювача
виключень
    //BeginLinker: Integer; //0,1 - початок/кінець трансляції - використовується
при обробці помилок
    F_BeginCalc: Integer; //0,1 - початок/кінець трансляції - використовується
при обробці помилок

    //Alg:
    F_NumberStack: Cardinal;
    F_Address: Cardinal;
    F_NumberArg: Cardinal;
    F_FConst: TFloatType;
    ICOMP: TCode;
    F_SyntaxErrorString1: String;
    F_Id: Cardinal;
    F_ResultType: Cardinal;
    F_ExtNumberArg: Cardinal;

    function FindNumberPR(PR: Cardinal): Cardinal;
    function FindOperation(S: String): Cardinal;
    procedure FindSingleOP(S: String; var NCop: Integer);
    procedure FindCallFunction(var nf: Integer);
    procedure FindLdFunction(var nf: Integer);
    procedure FindConstLdFunction(var nf: Integer);
    procedure FindNumberLevel(var ANL: TArray);
    procedure InsertBracketOperation(S: String; var S1: String);
    procedure IntToHex(N: TAddress; var AH: TArray4);
    procedure AddAddress(Adr: TAddress);
    procedure AddCommand(I: Integer);
    procedure AddByte(B: Byte);
    procedure AddWord(B1: Byte; B2: Byte);
    procedure Compile;
    procedure CALLA;

```

```

procedure CALLB;
procedure RET;
procedure FSTP(N: TAddress);
procedure FLD(N: TAddress);
procedure MOVAM(N: TAddress);
procedure MOVMA(N: TAddress);
procedure MOVA1(N: TAddress);
procedure PUSHA;
procedure POPA;

procedure FindExternalBracket(S: String; var SS: String);
procedure FindMainOperation(S:String; var NCOP: Integer; var NPos:
Cardinal);
procedure StringAnalizator(S:String; var ND: TDataNode; var HS:PLevel);
procedure WriteCode;
procedure WriteLevelParam(H: PNode; HS: PLevel; N: Cardinal);
procedure SubstParam(S1: String; var C: TFloatType; var SB: Integer);
function PointToDec(S: String):String;
procedure CreateInitData;
procedure FindVar(S1: String; var ni: Integer);
procedure SyntaxAnalizator(S:String; var ND: TDataNode; var LV:PLevel; var
EOFN: Boolean);
procedure NumberAnalizator(S: String; var ND: TDataNode; var EOFN:
Boolean);
procedure WriteNodeParam(ND: TDataNode);
procedure InitNode(var ND: TDataNode);
procedure WriteDataCode;
procedure FindFunction(S: String; var ND: TDataNode; var LV: PLevel; var
BH: Boolean);
procedure XchangeAddingBracket(S: String; var S1: String);
procedure PrevTreat(S: String; var S1: String);
//procedure SetExpression(S: String);
procedure FreeTree;
procedure FreeFunction;
procedure Linker(S:String);
procedure SetStackSize;
procedure CodeGeneration;
//function Calculation: TFloatType;
procedure FindManyArg(S: String; var Arg: TArray);
function Calc: TFloatType;

protected
//ExtException: TExternalException; //зовнішній оброблювач виключень
procedure CalcException(Sender: TObject; E: Exception); virtual;

public
//procedure InsertBracketOperation(S: String; var S1: String);
constructor Create;
destructor Destroy;
//function ReadErrorCode: Integer;
//function CalculateExpression(S: String): TFloatType;
procedure SetExtExpression(S: String; var E_Func: TFunction; var Stack:
Cardinal);
procedure CalcExtFunc(E_Func: TFunction);
function CalcExtFunc(Addr: TAddress): TFloatType;
procedure FreeExtFunc(E_Func: TFunction);
property VarShift: Boolean read F_VarShift write F_VarShift default False;
property FuncShift: Boolean read F_FuncShift write F_FuncShift default
False;
property ShowException: Boolean read F_ShowException write F_ShowException
default True;
property SyntaxError: Cardinal read F_SyntaxErrorCode write
F_SyntaxErrorCode default 0;
property SyntaxErrorString: String read F_SyntaxErrorString write
F_SyntaxErrorString;

```

```

    property CalcError: Integer read F_CalcErrorCode write F_CalcErrorCode
default 0;
    property InternalError: Integer read F_InternalError write F_InternalError
default 0;
    property ExternalException: Boolean read F_ExternalException write
F_ExternalException default False;
    //property SetExternalException: TExternalException write ExtException;
    property BeginCalc: Integer read F_BeginCalc write F_BeginCalc default 0;

//Algebra:
property GetStack: Cardinal read F_NumberStack;
property GetAddress: Cardinal read F_Address;
property ResultType: Cardinal read F_ResultType;
property GetNumberArg: Cardinal read F_NumberArg; //для Infinite  $\Phi$ -ій:
число аргументів
property ExtNumberArg: Cardinal read F_ExtNumberArg write F_ExtNumberArg;
//для Infinite  $\Phi$ -ій:
//min число аргументів
для виклику  $\Phi$ -ii
    procedure SetType(IdType: Cardinal);
    procedure SetObject(Name: String; Addr: Cardinal; IdType: Cardinal);
    procedure SetFunction(Name: String; Addr: Cardinal; TF: TArray; IdType:
Cardinal);
    procedure SetFunctionEx(Name: String; Addr: Cardinal; Type: Cardinal; IdType:
Cardinal);
    procedure SetLoadFunction(IdType: Cardinal; Addr: Cardinal);
    procedure SetOperation(Name: String; Func: String; PR: Cardinal; GS:
Integer);
    procedure SetBracketOperation(Br: Integer; Name: String);

    published

    end;

var
    ak,bk: array[1..7] of TFloatType;
    C_2dqrPi: extended;

    Foreval: TForeval;

implementation
{$IFDEF TEXTOUT}
uses
Test8;
{$ENDIF}

procedure TForeval.SetType(IdType: Cardinal);
label endp;
var
    i: Integer;
begin
    begin
    {
    for i:=0 to High(Alg.Types) do
    begin
        if Alg.Types[i] = IdType then goto endp;
    end;
    }
    SetLength(Alg.Types, Length(Alg.Types)+1);
    Alg.Types[High(Alg.Types)] := IdType;

endp:
end;

procedure TForeval.SetObject(Name: String; Addr: Cardinal; IdType: Cardinal);
label endp;
var
    i: Integer;

```

```

S: String;
begin
{
S:=Copy(Name,1,Length(Name));
for i:=0 to High(Alg.Obj) do
begin
if Alg.Obj[i].Type = IdType then
begin
if F_VarShift = False then S:=LowerCase(S);
if S = Alg.Obj[i].Name then goto endp;
end;
end;
}
SetLength(Alg.Obj,Length(Alg.Obj)+1);
Alg.Obj[High(Alg.Obj)].Type:=IdType;
Alg.Obj[High(Alg.Obj)].Addr:=Addr;
Alg.Obj[High(Alg.Obj)].Name:=Copy(Name,1,Length(Name));

endp:
end;

procedure TForeval.SetFunction(Name: String; Addr: Cardinal; TF: TArray; IdType:
Cardinal);
label endp;
var
N,K,i,j: Integer;
begin
for i:=0 to High(Alg.ExFunc) do
begin
if Name = Alg.ExFunc[i].Name then
begin
K:=i;
N:=High(Alg.ExFunc[K].Func);
SetLength(Alg.ExFunc[K].Func,Length(Alg.ExFunc[K].Func)+1);
N:=N+1;
Alg.ExFunc[K].Func[N].Type:=IdType;
Alg.ExFunc[K].Func[N].Addr:=Addr;
Alg.ExFunc[K].Func[N].N:=Length(TF);
SetLength(Alg.ExFunc[K].Func[N].TV,Length(TF));
for j:=0 to High(TF) do
begin
Alg.ExFunc[K].Func[N].TV[j]:=TF[j];
end;

goto endp;
end;
end;

SetLength(Alg.ExFunc,Length(Alg.ExFunc)+1);
Alg.ExFunc[High(Alg.ExFunc)].Name:=Copy(Name,1,Length(Name));

K:=High(Alg.ExFunc);
N:=High(Alg.ExFunc[K].Func);
SetLength(Alg.ExFunc[K].Func,Length(Alg.ExFunc[K].Func)+1);
N:=N+1;
Alg.ExFunc[K].Func[N].Type:=IdType;
Alg.ExFunc[K].Func[N].Addr:=Addr;
Alg.ExFunc[K].Func[N].N:=Length(TF);
SetLength(Alg.ExFunc[K].Func[N].TV,Length(TF));
for j:=0 to High(TF) do
begin
Alg.ExFunc[K].Func[N].TV[j]:=TF[j];
end;

endp:
end;

```

```

procedure TForeval.SetFunctionEx(Name: String; Addr: Cardinal; Type: Cardinal;
  IdType: Cardinal);
label endp;
var
  N,K,i,j: Integer;
begin
  for i:=0 to High(Alg.ExFunc) do
  begin
    if Name = Alg.ExFunc[i].Name then
    begin
      K:=i;
      N:=High(Alg.ExFunc[K].Func);
      SetLength(Alg.ExFunc[K].Func, Length(Alg.ExFunc[K].Func)+1);
      N:=N+1;
      Alg.ExFunc[K].Func[N].Type:=IdType;
      Alg.ExFunc[K].Func[N].Addr:=Addr;
      Alg.ExFunc[K].Func[N].N:=-1;
      SetLength(Alg.ExFunc[K].Func[N].TV, 1);
      Alg.ExFunc[K].Func[N].TV[0]:=Type;
      goto endp;
    end;
  end;

  SetLength(Alg.ExFunc, Length(Alg.ExFunc)+1);
  Alg.ExFunc[High(Alg.ExFunc)].Name:=Copy(Name, 1, Length(Name));

  K:=High(Alg.ExFunc);
  N:=High(Alg.ExFunc[K].Func);
  SetLength(Alg.ExFunc[K].Func, Length(Alg.ExFunc[K].Func)+1);
  N:=N+1;
  Alg.ExFunc[K].Func[N].Type:=IdType;
  Alg.ExFunc[K].Func[N].Addr:=Addr;
  Alg.ExFunc[K].Func[N].N:=-1;
  SetLength(Alg.ExFunc[K].Func[N].TV, 1);
  Alg.ExFunc[K].Func[N].TV[0]:=Type;

endp:
end;

procedure TForeval.SetLoadFunction(IdType: Cardinal; Addr: Cardinal);
begin
  SetLength(Alg.LdFunc, Length(Alg.LdFunc)+1);
  Alg.LdFunc[High(Alg.LdFunc)].Type:=IdType;
  Alg.LdFunc[High(Alg.LdFunc)].Addr:=Addr;
end;

procedure TForeval.SetOperation(Name: String; Func: String; PR: Cardinal; GS:
  Integer);
label 1;
var
  i,nf: Integer;
  - аргументні φ-ii
begin
  nf:=-1;
  for i:=0 to High(Alg.ExFunc) do
  begin
    if Func = Alg.ExFunc[i].Name then
    begin
      nf:=i; goto 1;
    end;
  end;

  1:
  if nf = -1 then begin {Error} end;
  if PR <= 0 then begin {Error} end;

  SetLength(Alg.Opr, Length(Alg.Opr)+1);
  Alg.Opr[High(Alg.Opr)].Name:=Name[1];
  Alg.Opr[High(Alg.Opr)].NFunc:=nf;
  Alg.Opr[High(Alg.Opr)].PR:=PR;

```

//Операції використовують тільки 1,2

```

Alg.Opr[High(Alg.Opr)].GS:=GS;

end;

procedure TForeval.SetBracketOperation(Br: Integer; Name: String);
label 1;
var
i,nf: Integer;
begin
//Br = 1 (); не використовується
!!!
//Br = 2 []; 3 {}; 4 <>;

nf:=-1;
for i:=0 to High(Alg.ExFunc) do
begin
if Name = Alg.ExFunc[i].Name then
begin
nf:=i; goto 1;
end;
end;

1:
if (nf = -1) or (Br < 2) or (Br > 4) then begin {Error} end;

{if Br = 1 then
begin
Alg.Bracket.Br1:=1;
Alg.Bracket.NF1:=nf;
end
else}
if Br = 2 then
begin
Alg.Bracket.Br2:=1;
Alg.Bracket.NF2:=nf;
end
else
if Br = 3 then
begin
Alg.Bracket.Br3:=1;
Alg.Bracket.NF3:=nf;
end
else
if Br = 4 then
begin
Alg.Bracket.Br4:=1;
Alg.Bracket.NF4:=nf;
end;

end;

procedure TForeval.SetStackSize;
label endp;
begin
if F_SyntaxErrorCode <> 0 then goto endp;
Tree:=BeginTree; //не обов'язково, тому що читання дерева - циклічно

while Tree^.Ch <> nil do
begin
Tree:=Tree^.Ch;
end;
StackSize:=Tree^.Stack.S;

while Tree^.Pr <> nil do
begin
while Tree^.Nb <> nil do
begin
Tree:=Tree^.Nb;
while Tree^.Ch <> nil do
begin

```

```

    Tree:=Tree^.Ch;
    end;
if Tree^.Stack.S > StackSize then StackSize:=Tree^.Stack.S;
end;
    Tree:=Tree^.Pr;
if Tree^.Stack.S > StackSize then StackSize:=Tree^.Stack.S;
end;

endp;
end;

procedure TForeval.Linker(S: String);
label endp;
var
    HS: PLevel;
    ND: TDataNode;
    H,H1: PNode;
    EOFN: Boolean;
    i: Cardinal;
begin
    if F_SyntaxErrorCode <> 0 then goto endp;
    New(Tree);
    Tree^.Ch:=nil;
    Tree^.Nb:=nil;
    Tree^.Pr:=nil;
    BeginTree:=Tree;
    //вершина дерева:
    Tree^.Expression:=Copy(S,1,Length(S));
    Tree^.Stack.S:=1;
    EOFN:=True;

    SyntaxAnalizator(Copy(Tree^.Expression,1,Length(Tree^.Expression)),ND,HS,EOFN);
    //спуск:
    //COP<>NOP:
    while EOFN <> True do
    begin
        if F_SyntaxErrorCode <> 0 then goto endp;

        New(H);
        Tree^.Ch:=H;

        WriteNodeParam(ND);
        WriteLevelParam(H,HS,0);

        H^.Stack.S:=Tree^.Stack.S;
        H^.Pr:=Tree;
        H1:=H;
        H^.Ch:=nil;
        H^.Nb:=nil;

        for i:=1 to High(HS) do
        begin
            New(H);
            H^.Ch:=nil;
            H^.Nb:=nil;
            WriteLevelParam(H,HS,i);
            H^.Stack.S:=H1^.Stack.S+1;
            H1.Nb:=H;
            H^.Pr:=Tree;
            //HS:=HS^.Next;
            H1:=H;
        end;
        Tree:=H^.Pr^.Ch;
        HS:=nil; //??? (звільнення масиву рівня)
        SyntaxAnalizator(Copy(Tree^.Expression,1,Length(Tree^.Expression)),ND,HS,EOFN);
        end;
        //EOFN=True: остання команда вузла (нижній лівий)
        WriteNodeParam(ND);

```

```

//підйом:
while Tree^.Pr <> nil do
begin
  while Tree^.Nb <> nil do
  begin
    Tree:=Tree^.Nb;
    SyntaxAnalizator(Copy(Tree^.Expression,1,Length(Tree^.Expression)),ND,HS,EOFN);
//COP<>NOP:
while EOFN<>True do
  begin
    if F_SyntaxErrorCode <> 0 then goto endp;

    New(H);
    Tree^.Ch:=H;
    WriteNodeParam(ND);
    WriteLevelParam(H,HS,0);

    H^.Stack.S:=Tree^.Stack.S;
    H^.Pr:=Tree;
    H1:=H;
    H^.Ch:=nil;
    H^.Nb:=nil;
    for i:=1 to High(HS) do
    begin
      New(H);
      H^.Ch:=nil;
      H^.Nb:=nil;
      WriteLevelParam(H,HS,i);

      H^.Stack.S:=H1^.Stack.S+1;
      H1.Nb:=H;
      H^.Pr:=Tree;
      H1:=H;
    end;
    Tree:=H^.Pr^.Ch;
    HS:=nil; //??? (звільнення масиву рівня)
    SyntaxAnalizator(Copy(Tree^.Expression,1,Length(Tree^.Expression)),ND,HS,EOFN);
  end;
//EOFN=True:остання команда вузла
WriteNodeParam(ND);
end;
Tree:=Tree^.Pr;
end;

Tree:=BeginTree;

endp:
end;

procedure TForeval.FreeFunction;
var
i: Integer;
begin
if F_SyntaxErrorCode = 0 then
for i:=1 to Length(Func.Interp)-1 do
begin
dispose(Func.Interp[i]);
end;
Func.Comp:=nil;
Func.Interp:=nil;

end;

procedure TForeval.WriteNodeParam(ND: TDataNode);
begin
Tree^.Stack.N:=ND.N;
Tree^.NCOP:=ND.NCOP;
Tree^.FD:=ND.FD;

```

```

Tree^.NFunc:=ND.NFunc;
Tree^.COP:=ND.COP;
Tree^.NObj:=ND.NObj;
Tree^.Id:=ND.Id;
end;

procedure TForeval.WriteLevelParam(H: PNode; HS: PLevel; N: Cardinal);
begin
H^.Expression:=Copy(HS[N].Expression,1,Length(HS[N].Expression));
H^.NL:=HS[N].NL;
H^.Id:=HS[N].Id;
end;

procedure TForeval.FindExternalBracket(S: String; var SS: String);
label 1,endp;
var
b,i,z,a:Integer;
begin
SS:=Copy(S,1,Length(S));

1:
if SS = '' then
begin
goto endp;
end;

if (SS[1] = '(' and (SS[Length(SS)] = ')')) then
begin
b:=-1;
for i:=2 to Length(SS)-1 do
begin
if SS[i] = '(' then b:= b-1;
if SS[i] = ')' then b:=b+1;
if b = 0 then goto endp;
end;

Delete(SS,Length(SS),1);
Delete(SS,1,1);
goto 1;
end;

endp:
end;
function TForeval.FindNumberPR(PR: Cardinal): Cardinal;
label 1;
var
i: Integer;
N: Cardinal;
begin
for i:=0 to High(Alg.Opr) do
begin
if PR = Alg.Opr[i].PR then
begin
N:=i; goto 1;
end;
end;
end;

1:
FindNumberPR:=N;
end;

function TForeval.FindOperation(S: String): Cardinal;
label 1;
var
i,NCOP: Integer;
begin
NCOP:=-1;

```

```

for i:=0 to High(Alg.Opr) do
begin
  if S = Alg.Opr[i].Name then
  begin
    NCOP:=i;
    goto 1;
  end;
end;

1:
FindOperation:=NCOP;
end;

procedure TForeval.FindSingleOP(S: String; var NCop: Integer);
var
  i,j,PR1,PR2: Integer;
  NCop1,NCop2: Integer;
begin
  PR1:=0;
  PR2:=0;
  NCop1:=-1; NCop2:=-1; NCop:=-1;

  for i:=0 to High(Alg.Opr) do
  begin
    if Alg.Opr[i].GS < 0 then //тільки одноарг. операції
    begin
      if (S[1] = Alg.Opr[i].Name[1]) and (Alg.Opr[i].GS = -1) then
      begin
        NCop1:=i;
        end;

        if (S[Length(S)] = Alg.Opr[i].Name[1]) and (Alg.Opr[i].GS = -2) then
        begin
          NCop2:=i;
          end;
        end;
      end;
    end;

  if (NCop1 <> -1) and (NCop2 <> -1) then
  begin
    if Alg.Opr[NCop1].PR < Alg.Opr[NCop2].PR then NCop:=NCop1 else NCop:=NCop2;
  end
  else
  if NCop1 <> -1 then NCop:=NCop1
  else
  if NCop2 <> -1 then NCop:=NCop2;

  end;

procedure TForeval.FindMainOperation(S:String; var NCOP: Integer; var NPos:
Cardinal);
label endp;
var
  Pl,Ml: Boolean;
  i,b: Integer;
  NCOP1,NCOP2,NPos1: Integer;
begin
  if S = '' then
  begin
    F_SyntaxErrorCode:=5;
    goto endp;
  end;

  NCOP1:=-1; NCOP2:=-1; NCOP:=-1;
  b:=0; Pl:=False; Ml:=False;

  if S[1] = '(' then b:=-1
  else

```

```

if S[1] = ')' then
begin
  F_SyntaxErrorCode:=2;
  goto endp;
end;

for i:=2 to Length(S)-1 {тому що 1-ий і останній символи м.б. одноарг.
символами} do
begin
  if S[i] = '(' then b:= b-1;
  if S[i] = ')' then b:=b+1;

  if b = 0 then
  begin
    NCOP2:=FindOperation(S[i]);
    if Alg.Opr[NCOP2].GS >= 0 then //тільки багатоарг. операції
    begin
      if (NCOP2 <> -1) and (NCOP1 <> -1) then
      begin
        {обов'язково <=, щоб взяти останній доданок !!!}
        if (Alg.Opr[NCOP2].PR <= Alg.Opr[NCOP1].PR) then
        begin
          NCOP1:=NCOP2;
          NPos1:=i;
        end;
      end
      else
      if (NCOP2 <> -1) and (NCOP1 = -1) then
      begin
        NCOP1:=NCOP2;
        NPos1:=i;
      end;
    end;
  end;

  NPos:=NPos1;
  NCOP:=NCOP1;
end;

//корекція виражень на одноарг. оп. типу -x^y, (neg)
//FindSingleOP(S,NCop,NPos1);
NCop1:=FindOperation(S[1]);
if (NCOP1 <> -1) and (Alg.Opr[NCOP1].PR < Alg.Opr[NCOP].PR) then NCOP:=-1;

endp:
end;

procedure TForeval.StringAnalizator(S:String; var ND: TDataNode; var HS:PLevel);
label endp;
var
  SS,S1: String;
  PR,NPR: Cardinal;
  BH,IH,IH1: PLevel;
  i,p,b: Cardinal;
  PR1,N,NCop: Integer;
  ni: Integer;
  NPos: Cardinal;
begin
  if F_SyntaxErrorCode <> 0 then goto endp;

  FindExternalBracket(S,SS);

  InitNode(ND);
  PR:=0;
  FindMainOperation(SS,NCop,NPos);
  N:=0;
  if NCop <> -1 then

```

```

begin
  inc(F_Id);
  ND.Id:=F_Id;
  SetLength(HS,2);
  HS[0].Expression:=Copy(SS,1, NPos-1);
  HS[0].NL:=1;
  HS[0].Id:=F_Id;
  HS[1].Expression:=Copy(SS,NPos+1,Length(SS)-NPos+1);
  HS[1].NL:=2;
  HS[1].Id:=F_Id;
  ND.N:=2;
  ND.COP:=_COP;
  ND.NFunc:=Alg.Opr[NCop].NFunc;
  ND.NCOP:=NCOP; //для помилок
end
else
begin
  FindSingleOp(SS,NCop);
  if NCop <> -1 then
  begin
    ND.COP:=_FUNC;
    ND.N:=1;
    ND.NFunc:=Alg.Opr[NCop].NFunc;
    SetLength(HS,1);
    if Alg.Opr[NCop].GS = -1 then HS[0].Expression:=Copy(SS,2,Length(SS)-1)
    else
    if Alg.Opr[NCop].GS = -2 then HS[0].Expression:=Copy(SS,1,Length(SS)-1);
    HS[0].NL:=1;
    ND.NCOP:=NCOP; //для помилок
    inc(F_Id);
    ND.Id:=F_Id;
    HS[0].Id:=F_Id;
  end
  else
  begin
    ND.COP:=0;
    SetLength(HS,1);
    ND.N:=1;
    HS[0].NL:=1;
    HS[0].Expression:=Copy(SS,1,Length(SS));
  end;
end;

endp:
end;

procedure TFloatType.SubstParam(S1: String; var C: TFloatType; var SB: Integer);
label endp;
var
  N,i: Cardinal;
  S1h,S3: String;
begin
  if F_SyntaxErrorCode <> 0 then goto endp;
  C:=0;
  if S1 = '' then
  begin
    F_SyntaxErrorCode:=5;
    goto endp;
  end;
  N:=1; S1h:=Copy(S1,1,Length(S1)); SB:=0;

  try
    C:=StrToFloat(PointToDec(S1h));
    SB:=1;
  except
    SB:=0;
  end;

endp:

```

```

end;

procedure TForeval.NumberAnalizator(S: String; var ND: TDataNode; var EOFN:
Boolean);
label 1;
var
S2: String;
ni, SB: Integer;
C: TFloatType;
begin
EOFN:=False;
InitNode (ND);
FindVar (S,ni);
if ni >= 0 then
begin
ND.COP:=_LDSF;
ND.NObj:=ni;
EOFN:=True;
goto 1;
end;

SubstParam (S,C,SB);
if SB = 1 then
begin
ND.FD:=C;
ND.COP:=_LDSC;
EOFN:=True;
inc (F_Id);
ND.Id:=F_Id;
goto 1;
end;

F_SyntaxErrorCode:=1; //невідомий символ
F_SyntaxErrorString:=Copy (S,1,Length (S));
EOFN:=False;

1:
end;

procedure TForeval.FindVar (S1: String; var ni: Integer);
label endp;
var
i: Cardinal;
S2,S3: String;
begin //ni - номер масиву знайденої
змінної
if F_SyntaxErrorCode <> 0 then goto endp; //ni = -1: змінних не знайдена
ni:=-1;

S2:=Copy (S1,1,Length (S1));
if F_VarShift = False then S2:=LowerCase (S2);

for i:=0 to High (Alg.Obj) do
begin
S3:=Alg.Obj [i].Name;
if F_VarShift = False then S3:=LowerCase (S3);
if S2 = S3 then
begin
ni:=i;
goto endp;
end;
end;

endp:
end;

procedure TForeval.FindFunction (S: String; var ND: TDataNode; var LV: PLevel;
var BH: Boolean);
label 1,2,3,4,endp;

```

```

var
i,k,l,bb,pf,pr,b1,b2,nf: Integer;
S1,Sa,Sf,Sf1,Sfn: String;
Neg: Integer;
begin
if F_SyntaxErrorCode <> 0 then goto endp;
BH:=False;

//пошук функції:
nf:=0; bb:=0; pf:=0; b1:=0;
for i:=1 to Length(S) do
begin
if S[i] = '(' then begin b1:=i; goto 1; end;
end;
if b1 = 0 then goto endp;

1:
b2:=0; bb:=-1;
for i:=b1+1 to Length(S) do
begin
if S[i] = '(' then bb:= bb-1;
if S[i] = ')' then bb:=bb+1;
if (i <> Length(S)) and (bb = 0) then begin {Internal Error} end;
end;

Sf:=Copy(S,1,b1-1);
Sa:=Copy(S,b1+1,Length(S)-b1-1);

if F_FuncShift = False then Sf1:=LowerCase(Sf) else Sf1:=Sf;

nf:=-1;
for i:=0 to High(Alg.ExFunc) do
begin
Sfn:=Alg.ExFunc[i].Name;
if F_FuncShift = False then
begin
Sfn:=LowerCase(Sfn);
end;
if Sf1 = Sfn then begin nf:=i; goto 4; end;
end;

F_SyntaxErrorCode:=6; //UnknownFunction
F_SyntaxErrorString:=Copy(Sf,1,Length(Sf));
goto endp;

4:
if nf <> -1 then
begin
FindManyArg(Sa,Arg);
if Length(Arg) <= 1 then
begin
{ ф-ія без аргументів}
F_SyntaxErrorCode:=4;
F_SyntaxErrorString:=Copy(Sf,1,Length(Sf));
goto endp;
end;

inc(F_Id);
ND.Id:=F_Id;

LV[0].Expression:=Copy(Arg[1],1,Length(Arg[1]));
LV[0].NL:=1;
LV[0].Id:=F_Id;

```

```

    for i:= 2 to High(Arg) do
    begin
        SetLength(LV,Length(LV)+1);
        LV[ i-1].Expression:=Copy(Arg[i],1,Length(Arg[i]));
        LV[ i-1].NL:=i;
        LV[ i-1].Id:=F_Id;
    end;

    ND.N:=High(Arg);
    ND.COP:=_Func;
    ND.NFunc:=nf;
    BH:=True;
end;

endp:
end;

procedure TForeval.FindManyArg(S: String; var Arg: TArray);
label endp;
var
    i,N,bb: Cardinal;
    PP: array of Cardinal;
begin
    bb:=0;
    N:=1; SetLength(PP,N+1); PP[1]:=0;

    for i:=1 to Length(S) do
    begin
        if S[i] = '(' then bb:= bb+1;
        if (bb = 0) and (S[i] = ',') then begin inc(N); SetLength(PP,N+1); PP[N]:=i;
        end;
        if S[i] = ')' then bb:=bb-1;
    end;

    inc(N); SetLength(PP,N+1); PP[N]:=Length(S)+1;

    SetLength(Arg,N);

    for i:=1 to N-1 do
    begin
        Arg[i]:=Copy(S,PP[i]+1,PP[i+1]-PP[i]-1);
        if (Arg[i] = '') then
        begin
            F_SyntaxErrorCode:=4; F_SyntaxErrorString:=Copy(S,1,Length(S)); goto endp;
        end;
    end;

endp:
end;

procedure TForeval.XchangeAddingBracket(S: String; var S1: String);
var
    i: Cardinal;
begin
    S1:=Copy(S,1,Length(S));
    for i:=1 to Length(S1) do
    begin
        if (S1[i] = '[') or (S1[i] = '{') or (S1[i] = '<') then
            begin Delete(S1,i,1); Insert('(',S1,i); end;
        if (S1[i] = ']') or (S1[i] = '}') or (S1[i] = '>') then
            begin Delete(S1,i,1); Insert(')',S1,i); end;
    end;
end;

```

```

procedure TForeval.InsertBracketOperation(S: String; var S1: String);
label 1,2,3,4,5, endp;
var
i,b,b1,k,j: Integer;
BH: Boolean;
begin
S1:=Copy(S,1,Length(S));
k:=0;

//круглі дужки для операцій не використовуються !!!
(*)
i:=1;
while i <= Length(S1) do
begin

if (S1[i] = '(') and (Alg.Bracket.Br1 = 1) then
begin
b:=0;
b1:=-1;
BH:=False;
for j:=i+1 to Length(S1) do
begin
if (S1[j] = '(') or (S1[j] = '{') or (S1[j] = '[') or (S1[j] = '<') then b:=
b-1;
if (S1[j] = ')') or (S1[j] = '}') or (S1[j] = ']') or (S1[j] = '>') then
b:=b+1;
if S1[j] = '(' then b1:=b 1-1;
if S1[j] = ')' then b1:=b1+1;
if (b = 0) and (S1[j] = ',') then BH:=True;
if b1 = 0 then begin k:=j; goto 1; end;
end;

F_SyntaxErrorCode:=2;
F_SyntaxErrorString:=')';
goto endp;

1:
if BH = True then
begin
Delete(S1,k,1);
Insert(')',S1,k);
Delete(S1,i,1);
Insert(Alg.ExFunc[Alg.Bracket.NF1].Name+'(',S1,i);
i:=i+Length(Alg.ExFunc[Alg.Bracket.NF1].Name)+1;
end
else inc(i);
end
else inc(i);

end;
*)
5:
for i:=1 to Length(S1) do
begin
if (S1[i] = '[') and (Alg.Bracket.Br2 = 1) then
begin
b:=0;
b1:=-1;
BH:=False;
for j:=i+1 to Length(S1) do
begin
if (S1[j] = '(') or (S1[j] = '{') or (S1[j] = '[') or (S1[j] = '<') then b:=
b-1;
if (S1[j] = ')') or (S1[j] = '}') or (S1[j] = ']') or (S1[j] = '>') then
b:=b+1;
if S1[j] = '[' then b1:=b 1-1;
if S1[j] = ']' then b1:=b1+1;

```

```

    //if (b = 0) and (S1[j] = ',') then BH:=True;    {для будь-якого числа
змінних}
    if b1 = 0 then begin k:=j; goto 2; end;
end;

F_SyntaxErrorCode:=2;
F_SyntaxErrorString:='';
goto endp;

2:
//if BH = True then    {для будь-якого числа змінних}
begin
Delete(S1,k,1);
Insert(',')',S1,k);
Delete(S1,i,1);
Insert(Alg.ExFunc[Alg.Bracket.NF2].Name+'(',S1,i);
goto 5;
end;
end;

if (S1[i] = '{') and (Alg.Bracket.Br3 = 1) then
begin
b:=0;
b1:=-1;
BH:=False;
for j:=i+1 to Length(S1) do
begin
if (S1[j] = '(') or (S1[j] = '{') or (S1[j] = '[') or (S1[j] = '<') then b:=
b-1;
if (S1[j] = ')') or (S1[j] = '}') or (S1[j] = ']') or (S1[j] = '>') then
b:=b+1;
if S1[j] = '{' then b1:=b 1-1;
if S1[j] = '}' then b1:=b1+1;
//if (b = 0) and (S1[j] = ',') then BH:=True;
if b1 = 0 then begin k:=j; goto 3; end;
end;

F_SyntaxErrorCode:=2;
F_SyntaxErrorString:='}';
goto endp;

3:
//if BH = True then
begin
Delete(S1,k,1);
Insert(',')',S1,k);
Delete(S1,i,1);
Insert(Alg.ExFunc[Alg.Bracket.NF3].Name+'(',S1,i);
goto 5;
end;
end;

if (S1[i] = '<') and (Alg.Bracket.Br4 = 1) then
begin
b:=0;
b1:=-1;
BH:=False;
for j:=i+1 to Length(S1) do
begin
if (S1[j] = '(') or (S1[j] = '{') or (S1[j] = '[') or (S1[j] = '<') then b:=
b-1;
if (S1[j] = ')') or (S1[j] = '}') or (S1[j] = ']') or (S1[j] = '>') then
b:=b+1;
if S1[j] = '<' then b1:=b 1-1;
if S1[j] = '>' then b1:=b1+1;
//if (b = 0) and (S1[j] = ',') then BH:=True;
if b1 = 0 then begin k:=j; goto 4; end;
end;
end;

```

```

F_SyntaxErrorCode:=2;
F_SyntaxErrorString:='>';
goto endp;

4:
//if BH = True then
begin
Delete(S1,k,1);
Insert(')',S1,k);
Delete(S1,i,1);
Insert(Alg.ExFunc[Alg.Bracket.NF4].Name+'(',S1,i);
goto 5;
end;
end;

end;

endp:
end;

procedure TForeval.PrevTreat(S: String; var S1: String);
label endp;
var
b,i: Integer;
begin
if S = '' then S:='0';
//S:=LowerCase(S);

InsertBracketOperation(S,S);

if F_SyntaxErrorCode <> 0 then goto endp;

//додаткові дужки можуть використовуватися тільки в операціях !!!
{if F_SyntaxExtension = True then
begin
XchangeAddingBracket(S,S);
end;
}
F_SyntaxErrorCode:=0;
b:=0;
for i:=1 to Length(S)-1 do
begin
if S[i] = '(' then b:= b-1;
if S[i] = ')' then b:=b+1;
end;

if S[Length(S)] = ')' then b:=b+1;
if b <> 0 then
begin // внутрішня не відповідність відкр. і закр. дужок
F_SyntaxErrorCode:=2;
if b > 0 then F_SyntaxErrorString:='('
else
if b < 0 then F_SyntaxErrorString:=')';
end;

//L1.Caption:=S[3];
S1:=Copy(S,1,Length(S));
endp:
end;

procedure TForeval.SyntaxAnalizator(S:String; var ND: TDataNode; var LV:PLevel;
var EOFN: Boolean);
label endp;
var
COP,Arg1,Arg2: String;
nf: Cardinal;
BH: Boolean;

```

```

a,b: TFloatType;
begin
if F_SyntaxErrorCode <> 0 then goto endp;

EOFN:=False;
InitNode (ND);

StringAnalizator (Copy (S, 1, Length (S) ), ND, LV);
If ND.COP = 0 then
begin
FindFunction (LV[0].Expression, ND, LV, BH);
if BH = True then
begin
EOFN:=False;
end
else
NumberAnalizator (LV[0].Expression, ND, EOFN);
end;

endp:
end;

procedure TForeval.CodeGeneration;
label endp;
var
E: Exception;
begin
if F_SyntaxErrorCode <> 0 then goto endp;

LFunc:=1;
SetLength (Func.Interp, LFunc);
new (Func.Interp[0]);
Tree:=BeginTree; //не обов'язково, тому що читання дерева - циклічно

while Tree^.Ch <> nil do
begin
Tree:=Tree^.Ch;
end;
//COP=NOP:
WriteCode;

while Tree^.Pr <> nil do
begin
while Tree^.Nb <> nil do
begin
Tree:=Tree^.Nb;
while Tree^.Ch <> nil do
begin
Tree:=Tree^.Ch;
end;
//COP=NOP:
WriteCode;
end;
//COP<>NOP:
Tree:=Tree^.Pr;
WriteCode;
end;
// 1-ий елемент в WriteCode випадає

endp:

if F_SyntaxErrorCode <> 0 then
begin
CalcException (Foreval as TObject, E);
end
end;

function TForeval.Calc: TFloatType; assembler;

```

```

asm
  call [eax+Func.ICode];
end;

procedure TForeval.WriteCode;
label endp;
var
i: Cardinal;
begin
SetLength(Func.Interp,LFunc+1);
new(Func.Interp[LFunc]);
WriteDataCode;

endp:
{-----}
//out:
{$IFDEF TEXTOUT}
if Test8.Form1.CB_Cod.Checked = True then
begin
if Tree^.COP = _LDSC then
begin
  Test8.Form1.M2.Lines[ LFunc-1]:='LDSC:
'+String(StrUpper(PChar(FloatToStr(Tree^.FD))));
end
else
if Tree^.COP = _LDSF then
begin
  Test8.Form1.M2.Lines[ LFunc-1]:='LDSF:
'+String(StrUpper(PChar(Alg.Obj[Tree^.NObj].Name)));
end
else
if (Tree^.COP = _Func) or (Tree^.COP = _COP) then
begin
  Test8.Form1.M2.Lines[ LFunc-
1]:=String(StrUpper(PChar(Alg.ExFunc[Tree^.NFunc].Name)));
end;

end;
{$ENDIF}
{-----}
inc(LFunc);

end;

procedure TForeval.WriteDataCode;
label endp;
var
fa,i,L,j: Cardinal;
S: String;
nf: Integer;
ANL: TArray;
begin
if F_SyntaxErrorCode <> 0 then goto endp;
if Tree^.COP = _LDSC then
begin
  FindConstLdFunction(nf);
  if nf = -1 then
begin {Error: немає ф-ії завантаження типу}
  F_SyntaxErrorCode:=7;
  F_SyntaxErrorString:='1';
end;
Func.Interp[LFunc]^ .FD:=Tree^.FD;
Func.Interp[LFunc]^ .NF:=NF;
Func.Interp[LFunc]^ .COP:=_LDSC;

```

```

Func.Interp[LFunc]^ .Type:=1;
end
else
if Tree^.COP = _LDSF then
begin
FindLdFunction(nf);
if nf = -1 then
begin {Error: немає ф-ії завантаження типу}
F_SyntaxErrorCode:=7;
F_SyntaxErrorString:=IntToStr(Alg.Obj[Tree^.NObj].Type);
end;
Func.Interp[LFunc]^ .NObj:=Tree^.NObj;
Func.Interp[LFunc]^ .NF:=nf;
Func.Interp[LFunc]^ .Type:=Alg.Obj[Tree^.NObj].Type;
Func.Interp[LFunc]^ .COP:=_LDSF;
end
else
if (Tree^.COP = _FUNC) or (Tree^.COP = _COP) then
begin
FindCallFunction(nf);
if nf = -1 then
begin {Error: немає ф-ії}
//для операцій
if Tree^.COP = _COP then
begin
F_SyntaxErrorCode:=8;
F_SyntaxErrorString1:=Alg.Opr[Tree^.NCOP].Name;
end
else
//для ф-ій
begin
F_SyntaxErrorCode:=9;
F_SyntaxErrorString1:=Alg.ExFunc[Tree^.NFunc].Name;
end;

F_SyntaxErrorString:='';
FindNumberLevel(ANL);
for j:=0 to Tree^.Stack.N-1 do
begin
if j > 0 then F_SyntaxErrorString:=F_SyntaxErrorString+', ';
F_SyntaxErrorString:=F_SyntaxErrorString+IntToStr(Func.Interp[ANL[j]].Type);
end;

end;

Func.Interp[LFunc]^ .COP:=_FUNC;
Func.Interp[LFunc]^ .NFunc:=Tree^.NFunc;
Func.Interp[LFunc]^ .Arg:=Tree^.Stack.N;
Func.Interp[LFunc]^ .NF:=nf;
Func.Interp[LFunc]^ .Type:=Alg.ExFunc[Tree^.NFunc].Func[nf].Type;
end;

Func.Interp[LFunc]^ .S:=Tree^.Stack.S-1; //починається з '0'
Func.Interp[LFunc]^ .N:=Tree^.Stack.N;
Func.Interp[LFunc]^ .NL:=Tree^.NL;
Func.Interp[LFunc]^ .Id:=Tree^.Id;
Func.Interp[LFunc]^ .Id:=Tree^.Id;

endp:
end;

procedure TForeval.FindCallFunction(var nf: Integer);
label 1,2,endp;
var
Fnc: TIntFunc;
i,N,T,j,S: Integer;
ANL: TArray;
begin
nf:=-1;

```

```

i:=Tree^.NFunc;
N:=Tree^.Stack.N;
S:=Tree^.Stack.S;
Fnc:=Alg.ExFunc[Tree^.NFunc].Func;

FindNumberLevel (ANL);

//спочатку шукати серед Infinite
if (F_ExtNumberArg > 0) and (Length(ANL) >= F_ExtNumberArg) then
begin

  for i:=0 to High(Fnc) do
  begin

    if Fnc[i].N = -1 then //перевірити збіг типів:
    begin

      for j:=0 to High(ANL) do
      begin
        if Func.Interp[ANL[j]].Type <> Fnc[i].TV[0] then goto 2;
      end;

      nf:=i;
      goto endp;
    end;

  2:
  end;

end;

for i:=0 to High(Fnc) do
begin

  if N = Fnc[i].N then //перевірити збіг типів:
  begin

    for j:=0 to N-1 do
    begin
      if Func.Interp[ANL[j]].Type <> Fnc[i].TV[j] then goto 1;
    end;

    nf:=i;
    goto endp;
  end;

  1:
  end;

endp:
end;

procedure TForeval.FindNumberLevel (var ANL: TArray);
var //визначення номерів рівнів, що впливають із даного вузла
i,j: Integer;
E: Exception;
begin

  for j:=1 to LFunc-1 do
  begin
    if Tree^.Id = Func.Interp[j]^Id then
    begin
      SetLength (ANL, Length (ANL) +1);
      ANL[High (ANL) ]:=j;
    end;
  end;

  if Tree^.Stack.N <> Length(ANL) then

```

```

begin {InternalError}
  F_InternalError:=1;
  CalcException(Foreval as TObject,E);
end;

end;

procedure TForeval.FindLdFunction(var nf: Integer);
label endp;
var
i: Integer;
begin
nf:=-1;
  for i:=0 to High(Alg.LdFunc) do
  begin
    if Alg.Obj[Tree^.NObj].Type = Alg.LdFunc[i].Type then
    begin
      nf:=i;
      goto endp;
    end;
  end;
end;

endp:
end;

procedure TForeval.FindConstLdFunction(var nf: Integer);
label endp;
var
i: Integer;
begin
nf:=-1;
  for i:=0 to High(Alg.LdFunc) do
  begin
    if Alg.LdFunc[i].Type = 1 then
    begin
      nf:=i;
      goto endp;
    end;
  end;
end;

endp:
end;

procedure TForeval.InitNode(var ND: TDataNode);
var
i: Integer;
begin
ND.N:=0;
ND.NCOP:=0;
ND.NFunc:=0;
ND.NObj:=0;
ND.COP:=0;
ND.FD:=0;
ND.FX:=0;
ND.Id:=0;
end;

function TForeval.PointToDec(S: String):String;
label
1;
var
i: Cardinal;
SS: String;
begin
SS:=Copy(S,1,Length(S));
for i:=1 to Length(S) do
begin
  if SS[i] = '.' then
  begin

```

```

Delete(SS,i,1);
Insert(', ',SS,i);
goto 1;
end
end;

1:
PointToDec:=Copy(SS,1,Length(SS));
end;

procedure TForeval.CreateInitData;
begin
//порядок не міняти

F_SyntaxErrorCode:=0;
LFunc:=0;

F_ShowException:=False;
F_ExternalException:=False;
F_VarShift:=False;
F_FuncShift:=False;
Alg.Bracket.Br1:=0;
Alg.Bracket.Br2:=0;
Alg.Bracket.Br3:=0;
Alg.Bracket.Br4:=0;

Application.OnException:=CalcException;

SetType(1);
end;

procedure TForeval.SetExtExpression(S: String; var E_Func: TFunction; var
Stack: Cardinal);
var
I_Func: TFunction;
I_LFunc: Cardinal;
E: Exception;
begin
try
I_LFunc:=LFunc;
I_Func:=Func;
Func:=E_Func;
F_SyntaxErrorCode:=0;
F_CalcErrorCode:=0;
F_InternalError:=0;
F_SyntaxErrorString:='';
F_Id:=0;
CurrentExpression:=Copy(S,1,Length(S));
PrevTreat(S,S);
Linker(S);
FreeFunction;
CodeGeneration;
if Length(Func.Interp) <> 0 then
F_ResultType:=Func.Interp[High(Func.Interp)]^.Type;
SetStackSize;
Compile;
Func.ICode:=TAddress(@Func.Comp[0]);
FreeTree;
Stack:=StackSize;
E_Func:=Func;
LFunc:=I_LFunc;
Func:=I_Func;

except
on E: EZeroDivide do F_InternalError:=1;
on E: EInvalidOp do F_InternalError:=2;
on E: EOverflow do F_InternalError:=3;
on E: EUnderFlow do F_InternalError:=4;
on E: EIntOverflow do F_InternalError:=5;
on E: EAccessViolation do F_InternalError:=6;

```

```

    on E: EOutOfMemory      do F_InternalError:=7;
    on E: EStackOverFlow    do F_InternalError:=8;
end;
if F_InternalError <> 0 then CalcException(Foreval as TObject,E);
//BeginLinker:=1;
end;

procedure TForeval.CalcExtFunc(E_Func: TFunction); assembler;
asm
    //call [E_Func.ICode]
end;

function TForeval.CalcExtFunc(Addr: TAddress): TFloatType; assembler;
asm
    call Addr
    //fstp @Result
end;

procedure TForeval.FreeExtFunc(E_Func: TFunction);
var
    i: Cardinal;
begin
    for i:=1 to Length(E_Func.Interp)-1 do
    begin
        dispose(E_Func.Interp[i]);
    end;
    E_Func.Comp:=nil;
    E_Func.Interp:=nil;

end;

constructor TForeval.Create;
begin
    //inherited Create(AOwner);
    CreateInitData;
end;

destructor TForeval.Destroy;
begin
    FreeFunction;
    inherited Destroy;
end;

procedure TForeval.FreeTree;
label endp;
var
    H: PNode;
begin
    if F_SyntaxErrorCode <> 0 then goto endp;
    Tree:=BeginTree;

    while Tree^.Ch <> nil do
    begin
        Tree:=Tree^.Ch;
    end;
    //COP=NOP

    while Tree^.Pr <> nil do
    begin
        while Tree^.Nb <> nil do
        begin
            H:=Tree; Tree:=Tree^.Nb; H^.Nb:=nil;
            dispose(H);
            while Tree^.Ch <> nil do
            begin
                Tree:=Tree^.Ch;
            end;
        end;
    end;
end;

```

```

//COP=NOP
end;

//COP<>NOP
H:=Tree; Tree:=Tree^.Pr; H^.Pr:=nil;
dispose(H);
end;
dispose(Tree);

endp:
end;

procedure TForeval.CalcException(Sender: TObject; E: Exception);
var
S: String;
T: Integer;
begin
if F_BeginCalc = 0 then
begin
{if E.ClassType = EZeroDivide      then F_InternalError:=1
else
if E.ClassType = EInvalidOp        then F_InternalError:=2
else
if E.ClassType = EOverFlow         then F_InternalError:=3
else
if E.ClassType = EUnderFlow        then F_InternalError:=4
else
if E.ClassType = EIntOverFlow      then F_InternalError:=5
else
if E.ClassType = EAccessViolation then F_InternalError:=6
else
if E.ClassType = EOutOfMemory      then F_InternalError:=7
else
if E.ClassType = EStackOverFlow    then F_InternalError:=8;
}
}
end

else
if F_SyntaxErrorCode = 0 then
begin
if E.ClassType = EZeroDivide      then F_CalcErrorCode:=1
else
if E.ClassType = EInvalidOp        then F_CalcErrorCode:=2
else
if E.ClassType = EOverFlow         then F_CalcErrorCode:=3
else
if E.ClassType = EUnderFlow        then F_CalcErrorCode:=4
else
if E.ClassType = EIntOverFlow      then F_CalcErrorCode:=5
else
if E.ClassType = EAccessViolation then F_CalcErrorCode:=6;
F_BeginCalc:=0;
end;

if F_ShowException then
begin
MessageBeep(MB_ICONHAND);
if F_CalcErrorCode <> 0 then
begin
MessageDlg('Calculation Error',mtError,[mbOk],0)
end
else
if F_SyntaxErrorCode <> 0 then
begin
if F_SyntaxErrorCode = 1 then S:='UNKNOWN SYMBOL: ';
if F_SyntaxErrorCode = 2 then S:='MISSING BRACKET: ';
if F_SyntaxErrorCode = 3 then S:='MISSING OPERATION: ';
if F_SyntaxErrorCode = 4 then S:='INCOINCIDENCE NUMBER OF ARGUMENTS: ';

```

```

    if F_SyntaxErrorCode = 5 then S:='MISSING EXPRESSION: ';
    if F_SyntaxErrorCode = 6 then S:='UNKNOWN FUNCTION: ';
    if F_SyntaxErrorCode = 7 then S:='ABSENT FUNCTION FOR LOAD TYPE: ';
    if F_SyntaxErrorCode = 8 then S:='NOT DEFINED OPERATION
'+'''+F_SyntaxErrorString1+'''+ ' FOR TYPES: ';
    if F_SyntaxErrorCode = 9 then S:='NOT DEFINED FUNCTION
'+'''+F_SyntaxErrorString1+'''+ ' FOR TYPES: ';

    MessageDlg(S+#13+#10+'''+F_SyntaxErrorString+''',mtError,[mbOk],0)
end
else
if F_InternalError <> 0 then
MessageDlg('Internal Error',mtError,[mbOk],0)
end;

{
if F_ExternalException then
begin
if F_SyntaxErrorCode <> 0 then T:=1000
else
if F_CalcErrorCode <> 0 then T:=1001
else
if F_InternalError <> 0 then T:=1002;

ExtException(T);
end;
}

end;

procedure TForeval.Compile;
label 1;
var
i,j,N: Integer;
AH,AB: TArray4;
R: TFloatType;
P: Pointer;
X: TFloatType;
begin

LFunc:=Length(Func.Interp);

SetLength(ICOMP,0);

PUSHA;
for i:=1 to LFunc-1 do
begin
AddCommand(i);
end;
POPA;
RET;

SetLength(Func.Comp,Length(ICOMP));
Func.Comp:=Copy(ICOMP,0,Length(ICOMP));
ICOMP:=nil;
end;

procedure TForeval.RET;
begin
AddByte($C3);
end;

procedure TForeval.AddCommand(I: Integer);
var
N,NF: Integer;
begin
//стек:
MOVMA(TAddress(@Func.Interp[I]^S));
MOVAM(TAddress(@F_NumberStack));

```

```

if Func.Interp[I]^COP = _LDSC then
begin
  MOVA1 (TAddress (@Func.Interp[I]^FD));
  MOVAM (TAddress (@F_Address));
  N:=Func.Interp[I]^NF;
  MOVA1 (TAddress (Alg.LdFunc[N].Addr));
  CALLA;
end
else
if Func.Interp[I]^COP = _LDSF then
begin
  N:=Func.Interp[I]^NObj;
  MOVA1 (Alg.Obj[N].Addr);
  MOVAM (TAddress (@F_Address));
  N:=Func.Interp[I]^NF;
  MOVA1 (TAddress (Alg.LdFunc[N].Addr));
  CALLA;
end
else
if Func.Interp[I]^COP = _FUNC then
begin
  N:=Func.Interp[I]^NFunc;
  NF:=Func.Interp[I]^NF;

  //infinite:
  if Alg.ExFunc[N].Func[NF].N = -1 then
  begin
    MOVA1 (TAddress (Func.Interp[I]^Arg));
    MOVAM (TAddress (@F_NumberArg));
  end;

  MOVA1 (Alg.ExFunc[N].Func[NF].Addr);
  CALLA;
end;

end;

procedure TForeval.IntToHex(N: TAddress; var AH: TArray4);
var
i,K,N1,j: Integer;
A8: array [1..8] of Integer;
begin
N1:=N; j:=1;
for i:=7 downto 1 do
begin
  K:=Trunc (N1/IntPower (2,i*4));
  A8[j]:=K;
  N1:=N 1-Trunc (IntPower (2,i*4) ) *K;
  inc(j);
end;
A8[j]:=N1;

//у зворотному порядку:
j:=1;
for i:=4 downto 1 do
begin
  AH[i]:=A8[j]*16+A8[j+1];
  j:=j+2;
end;
end;

procedure TForeval.AddByte(B: Byte);
var
j: Integer;
begin
j:=High(ICOMP)+1;
SetLength(ICOMP,Length(ICOMP)+1);
ICOMP[j]:=B;

```

```

end;

procedure TForeval.AddWord(B1: Byte; B2: Byte);
var
j: Integer;
begin
j:=High(ICOMP)+1;
SetLength(ICOMP,Length(ICOMP)+2);
ICOMP[j]:=B1; ICOMP[j+1]:=B2;
end;

procedure TForeval.AddAddress(Adr: TAddress);
var
AB: TArray4;
j: Integer;
begin
IntToHex(Adr,AB);
j:=High(ICOMP)+1;
SetLength(ICOMP,Length(ICOMP)+4);
ICOMP[j]:=AB[1]; ICOMP[j+1]:=AB[2]; ICOMP[j+2]:=AB[3]; ICOMP[j+3]:=AB[4];
end;

procedure TForeval.CALLA;
begin
AddWord($FF,$D0);
end;
procedure TForeval.CALLB;
begin
AddWord($FF,$D3);
end;
procedure TForeval.MOVAM(N: TAddress);
begin
//EAX->MEM
AddWord($89,$05);
AddAddress(N);
end;
procedure TForeval.MOVMA(N: TAddress);
begin
//MEM->EAX
AddWord($8B,$05);
AddAddress(N);
end;
procedure TForeval.MOVAL(N: TAddress);
begin
//N - число
AddByte($B8);
AddAddress(N);
end;
procedure TForeval.FLD(N: TAddress);
begin
AddWord($DD,$05);
AddAddress(N);
end;
procedure TForeval.FSTP(N: TAddress);
begin
AddWord($DD,$1D);
AddAddress(N);
end;
procedure TForeval.PUSHA;
begin
AddByte($50);
end;
procedure TForeval.POPA;
begin
AddByte($58);
end;
end.

```

## Файл About.pas - вікно «Про програму...»

```
unit uAbout;

interface
uses
  Classes,
  Graphics,
  Controls,
  Forms,
  StdCtrls,
  ExtCtrls,
  ShellApi,
  jpeg, Buttons;

type
  TfmAbout = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    BitBtn1: TBitBtn;
    Image1: TImage;
    procedure Label7Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private { Private declarations }
  public { Public declarations }
  end;

var fmAbout: TfmAbout;

implementation

{$R *.DFM}
procedure TfmAbout.BitBtn1Click(Sender: TObject);
begin
  fmAbout.Close;
end;

procedure TfmAbout.FormCreate(Sender: TObject);
begin
  Label1.Caption:='МАГІСТЕРСЬКА РОБОТА';
  Label2.Caption:='на тему:';
  Label3.Caption:='Дослідження та програмна реалізація системи big data наукових досліджень';
  Label4.Caption:='';
  Label5.Caption:='Розробив: студент Пилипенко Андрій Андрійович';
  Label6.Caption:='гр. КН-21М-1,4';
  Label7.Caption:='Керівник: Коваленко А.С.';
  Label8.Caption:='м. Кропивницький 2022';
end;
end.
```