

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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи підтримки
індикативного аналізу”

Виконав здобувач вищої освіти
II курсу, групи КН-22М-2
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Дьяченко Ю.В.
« ____ » _____ 2023 р.

Керівник проекту
доктор технічних наук, доцент
_____ Коваленко О.В.
« ____ » _____ 2023 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Дьяченку Юрію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи підтримки індикативного аналізу

2. Керівник роботи Коваленко Олександр Володимирович, докт. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 33-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи підтримки індикативного аналізу

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

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

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Дьяченко Ю.В. Дослідження та програмна реалізація системи підтримки індикативного аналізу. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи підтримки індикативного аналізу.

Метою розробки є дослідження та програмна реалізація системи підтримки індикативного аналізу.

Об'єктом дослідження є процес підтримки індикативного аналізу.

Предметом дослідження є методи підтримки індикативного аналізу.

Методи дослідження базуються на методах Big Data, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи підтримки індикативного аналізу.

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 RAD Studio Delphi 10.4.

Ключові слова: комп'ютерні науки, індикативний аналіз

ABSTRACT

Diachenko Yu.V. Research and software implementation of the indicative analysis support system. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the support system of indicative analysis.

The purpose of development is research and software implementation of the indicative analysis support system.

The object of research is the process of supporting indicative analysis.

The subject of the study is methods of supporting indicative analysis.

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 indicative analysis support system.

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 RAD Studio Delphi 10.4 environment.

Keywords: computer science, indicative analysis

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	8
1.1 Призначення системи.....	8
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	25
2.3 Розгорнута постановка завдання	31
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	32
3.1 Опис функціонування системи	32
3.2 Розробка структурної схеми.....	39
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	45
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	47
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	47
4.2 Захист розробленого програмного забезпечення.....	56
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	58
6 НАУКОВА НОВИЗНА	60

						ВКРМ-122.23.0035.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Дьяченко Ю.В.				Дослідження та програмна реалізація системи підтримки індикативного аналізу	Літ.	Аркуш	Аркушів
Перев.	Коваленко О.В.					М	1	99
Н.контр.	Коваленко А.С.				ЦНТУ КН-22М-2			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	61
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	61
7.2 Розрахунок трудомісткості розробки програмної продукції.....	63
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	65
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	69
7.5 Визначення собівартості розробки та ціни програмної продукції.....	74
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	77
7.7 Визначення експлуатаційних витрат.....	77
7.8 Визначення економічної ефективності програмної продукції.....	79
7.9 Висновок.....	81
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	82
8.1 Вступ.....	82
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	84
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	85
8.4 Розробка заходів з умов поліпшення охорони праці.....	88
8.5 Розрахунок штучного освітлення	89
9 ОСНОВНІ ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕА	–	електронний архів
ЕДО	–	електронний документообіг
САДД	–	система автоматизації електронного документообігу й діловодства
СЕА	–	системи електронних архівів
СДЕ	–	системи діловодства електронні
СУЕД	–	система управління електронним документообігом
ТЕ	–	типові елементи
ФФО	–	формат файлових об'єктів
EDMS	–	Electronic Document Management Systems

КБГПЗ-2023

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Актуальність роботи визначається актуальністю досліджень проблеми забезпечення енергетичної безпеки (ЕБ) України й актуальністю розробки інструментарію для проведення цих досліджень.

Енергетична безпека України характеризується як стан захищеності її громадян, суспільства, держави, економіки від обумовлених внутрішніми й зовнішніми факторами погроз дефіциту в забезпеченні їхніх обґрунтованих потреб в енергії енергетично доступними паливно-енергетичними ресурсами (ПЕР) прийнятної якості в нормальних умовах і при надзвичайних обставинах, а також від порушень стабільності, безперебійності паливо- і енергопостачання. У нормальних умовах зазначений стан захищеності відповідає забезпеченню в повному обсязі обґрунтованих потреб, у надзвичайних ситуаціях – гарантованому забезпеченню мінімально необхідного обсягу потреб.

Важливим напрямком рішення завдання забезпечення ЕБ України і її регіонів є облік вимог ЕБ при дослідженні перспектив розвитку енергетики країни. Аналогічні завдання одержали найбільше поширення в роботах інституту проблем моделювання в енергетиці ім. Г.Є. Пухова (ІПМЕ НАНУ). У цей час у даному інституті проводяться дослідження, що стосуються вибору раціональних з позицій ЕБ напрямків розвитку енергетики країни і її регіонів і формування на їхній основі пропозицій по забезпеченню ЕБ. Робиться це на основі використання індикативного аналізу перспективних станів найважливіших індикаторів ЕБ при порівнянні їхніх очікуваних значень із відповідними граничними значеннями.

Основним засобом зазначених досліджень є створений в ІПМЕ НАНУ на основі методів комбінаторного моделювання інструментарій для формування з урахуванням вимог ЕБ напрямків розвитку паливно-енергетичного комплексу (ПЕК). Множину напрямків розвитку паливно-енергетичного комплексу в цьому випадку представлено у вигляді графа, кожний вузол якого відповідає

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

можливого стану ПЕК зі своїми особливостями паливо- і енергопостачання, а дуга – траєкторії переходу ПЕК з одного часового рівня на інший зі своєю вартістю. У даному інструментарії не були реалізовані компоненти для відображення графів розвитку ПЕК і їхнього аналізу з позицій ЕБ. Особливість використання в поставлених завданнях методів комбінаторного моделювання складається у формуванні значної кількості перспективних станів енергетики країни і її регіонів. Відображення цих станів на екрані, а також відображення самих траєкторій неможливо через їхню величезну кількість. По цій же причині неможливий і їхній експертний аналіз із позицій ЕБ. Наявність подібних проблем обумовило необхідність рішення проблеми відображення графів розвитку ПЕК і подальший аналіз їхніх станів і траєкторій з позицій ЕБ, включаючи проведення індикативного аналізу кожного можливого стану ПЕК з обліком часового й територіального аспектів. Пропонований шлях рішення такої проблеми тут – розробка системи підтримки індикативного аналізу ЕБ країни.

При постановці досліджень і створенні інструментарію було прийнято, що процес підготовки результуючого графа розвитку ПЕК, його відображення й подальший експертний аналіз можливих станів ПЕК і траєкторій його розвитку з позицій ЕБ повинні бути виконані в спеціалізованому інтегрованому середовищі. При цьому оцінки станів по найважливіших індикаторах ЕБ повинні формуватися за допомогою підсистеми розрахунку індикаторів ЕБ. Наочний, доступний для експертів індикативний аналіз перспективних станів ПЕК повинен бути реалізований у рамках підсистеми індикативного аналізу можливих станів ПЕК – спеціалізованого «аналітичного» інструмента, у якому за допомогою елементів когнітивної графіки проводився б детальний аналіз можливих станів як у територіальному, так і в часовому розрізах. У такому інструменті повинні бути враховані основні положення методики проведення індикативного аналізу ЕБ (нормування значень індикаторів ЕБ, порівняння величин значень індикаторів ЕБ з їхніми граничними значеннями, одержання якісної оцінки ситуації з ЕБ у різних аспектах функціонування й розвитку енергетики територій).

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Аналіз нині існуючого інструментарію показує, що дотепер таких програмних продуктів для проведення досліджень розвитку енергетики України з позицій ЕБ запропоновано не було. Існують спеціалізовані системи для роботи із графами довільної семантики, є інструментарій для індикативного аналізу інших предметних областей, у якому використовуються різні принципи проведення аналізу об'єктів дослідження. Однак здебільшого подібний інструментарій має вузьку функціональну або предметну спрямованість, що вимагає його адаптації у випадку його використання для рішення завдання проведення аналізу можливих станів ПЕК і траєкторій його розвитку.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи підтримки індикативного аналізу.

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

- Огляд існуючих систем підтримки індикативного аналізу.
- Дослідження системи підтримки індикативного аналізу.
- Програмна реалізація системи підтримки індикативного аналізу.

Об'єктом дослідження є процес підтримки індикативного аналізу.

Предметом дослідження є методи підтримки індикативного аналізу.

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 підтримки індикативного аналізу.
- Розроблено вітчизняний продукт підтримки індикативного аналізу, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі підтримки індикативного аналізу.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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

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

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

КБГІЗ-2023

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для реалізації програмного забезпечення мережної підтримки індикативного аналізу. Важливе місце при цьому відводиться індикативному плануванню як сукупності процедур узгодження показників розвитку виробництва за допомогою стандартизованих на кожному рівні планування параметрів-індикаторів. Індикатор – числовий показник зміни основної енергетичної величини, що використовується для обґрунтування енергетичної політики, спрямованості розвитку енергетичних процесів та оцінки їх результатів.

Важливість індикаторів як основних параметрів господарювання не обмежується лише застосуванням їх для створення планів. Не менш важливою функцією може стати поточний їх аналіз з метою внесення корегуючих дій. Така можливість з'явилася в останні роки завдяки розробки концепції багатовимірного аналізу даних. Його для кінцевих користувачів полягає в підсиленні розуміння значення того, що міститься в базах даних. Він також полегшує навігацію в базі даних, фільтрацію специфічної підмножини даних, надання конкретно орієнтованих даних і визначення аналітичних обчислень. До того ж, тому що дані фізично зберігаються в багатовимірній структурі, ці операції набагато разів швидші в часі і більш узгоджені, ніж це можливо в інших структурах бази даних. Ця комбінація простоти і швидкості є однією з основних переваг багатовимірного аналізу даних.

Багатовимірна структура даних часто подається у вигляді багатовимірного масиву – гіперкуба. Багатовимірний масив – це група елементів даних, що впорядковуються вимірністю даних. Трьохвимірний масив можна візуально уявляти собі як куб, вимірність якого утворює сторону куба, включаючи будь-

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

який переріз, паралельний тим сторонам. Масиви більш високої вимірності не мають фізичної метафори, але вони організують дані тим способом, яким користувачі думають про своє підприємство. Типова підприємницька вимірність включає час, систему вимірювання (масштаб), продукти, географічні зони, канали збуту тощо. На даний час розроблені чисельні інструментальні засоби, зокрема OLAP-системи, які забезпечують гнучке і ефективне маніпулювання над гіперкубами даних.

1.2 Область застосування

Областю застосування є енергетична безпека України. Під енергетичною безпекою України слід розуміти спроможність держави забезпечити ефективне використання власної паливно-енергетичної бази, здійснити оптимальну диверсифікацію джерел і шляхів постачання в Україну енергоносіїв для забезпечення життєдіяльності населення та функціонування національної економіки у режимі звичайного, надзвичайного та воєнного стану, попередити різкі цінові коливання на паливно-енергетичні ресурси або ж створити умови для безболісної адаптації національної економіки до нових цін на ці ресурси.

Для України питання енергетичної безпеки є нині головною умовою її існування як самостійної держави. Заходи державного регулювання щодо забезпечення енергетичної безпеки можна розділити на превентивні і ліквідаційні.

Превентивні заходи. Їх реалізація має сприяти формуванню в Україні менш уразливої до енергетичних потрясінь економіки. Це насамперед: енергозбереження, диверсифікація джерел надходження енергоносіїв, стимулювання видобутку основних видів палива і виробництва електроенергії, використання нетрадиційних та відновлювальних видів енергії.

Ліквідаційні заходи повинні, зокрема, передбачати створення в Україні стратегічних газових і нафтових резервів, підконтрольних державним органам, та

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

системи розподілу нафти і газу в разі серйозних порушень в інфраструктурі забезпечення споживачів. Також необхідно подбати про те, щоб існували розвідані і пробурені нафтові родовища стратегічного призначення, які можуть вводитися в експлуатацію в разі війни.

Досягнення енергетичної безпеки неможливе без виявлення чинників, дія яких може негативно позначитися на розвитку паливно-енергетичного комплексу. Однією з головних цілей державної енергетичної політики, що знайшла, зокрема, своє втілення в Національній енергетичній програмі України до 2020 р., є задоволення України в паливно-енергетичних ресурсах шляхом збільшення частки їх власного видобутку і послаблення зовнішньої енергозалежності шляхом зниження обсягів їх імпорту [4].

Нині недостатня увага приділяється фінансуванню геологорозвідувальних робіт, що є необхідною умовою збільшення рівня забезпечення України власними нафтою і газом. Слід відзначити, що обсяги глибокого розвідувального буріння на нафту і газ протягом останніх років скоротилися проти 1990 р. у 5 разів, а приріст запасів вуглеводної сировини – у 3 рази. Як свідчить досвід, для забезпечення стабільного видобутку нафти і газу та його нарощування необхідно, щоб приріст запасів щонайменше у два рази був вищий за річний видобуток. Недостатнє проведення геологорозвідувальних робіт може призвести до зменшення видобутку нафти й газу в майбутньому. Однак навіть уже розвідані запаси цих енергоносіїв дають змогу збільшити такий видобуток мінімум удвічі, що дало б змогу економити більше двох мільярдів доларів США щорічно.

Прискорений розвиток вугільної промисловості України вбачається гарантом її енергетичної та енергетичної незалежності, а тому потребує фінансової підтримки держави. Окремі твердження про недоцільність розвитку вугільної промисловості із-за низьких якісних характеристик власного вугілля, енергетичної не вигідності його використання, великих матеріальних і фінансових витрат є не досить обґрунтованими в контексті національної безпеки. Зважаючи на значні запаси в нашій державі вугілля, цей стратегічний ресурс може сприяти піднесенню рівня забезпечення енергетичної безпеки України.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Також не обґрунтованими є пропозиції прихильників парогазових електростанцій (ПГЕС), які зводяться до необхідності повної заміни генеруючого обладнання теплових електростанцій (ТЕС) та заміщення його ПГЕС. Слід врахувати, що введення 20,7 млн квт. ПГЕС вимагає близько 20 млрд дол. США капіталовкладень і посилює залежність України від російського газу, що є загрозою її енергетичній безпеці. Лише в разі істотного збільшення видобутку власного газу, стабілізації соціально-енергетичної ситуації Україна могла б дозволити собі будівництво кількох парогазових теплоелектроцентралей поблизу Києва та деяких інших великих міст, що дозволило б оздоровити там навколишнє середовище.

Реалізація заходів щодо енергозабезпечення, нарощування обсягів видобутку газу, використання нетрадиційних джерел енергії, як це передбачається Національною енергетичною програмою, дасть змогу скоротити обсяги споживання природного газу народним господарством до 2010 р. порівняно з 1990 р. на 32,3%, а також зменшити його імпорт на 49%.

Необхідно вживати заходи щодо диверсифікації джерел надходження в Україну енергоносіїв. Зокрема, завершити будівництво нафтового терміналу поблизу Одеси, розширити енергетичні зв'язки з Туркменистаном, країнами Близького і Середнього Сходу, Туреччиною щодо поставок нафти і газу, розвивати внутрішні джерела енергозабезпечення, у тому числі вугільної галузі та нафтогазового комплексу.

Створення диверсифікованої моделі нафтогазозабезпечення України полягає не тільки у приєднанні її економіки до альтернативних джерел постачання, але й у тому, щоб задля реалізації своїх національних інтересів вона відігравала роль транзитного коридору як для існуючих, так і для перспективних поставок енергоресурсів. Інакше роль транзитора може бути перехоплена іншими країнами.

Поряд з удосконаленням переробки нафти на нафтопереробних заводах України слід збільшувати її видобуток на нових родовищах. Однак не може не

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

турбувати, що з розвіданих геологами запасів нафти добувається лише близько 10%, в той час коли зарубіжні нафтові компанії добувають 50% і більше.

Україна має великі можливості щодо транзиту природного газу. Однак понад 35% газопроводів експлуатуються вже понад 20 років і морально застаріли. Щорічно необхідно здійснювати капітальний ремонт 250-300 кілометрів газопроводів. Незадовільний технічний стан газопроводів дає змогу транспортувати нині до європейських країн без істотного енергетичного ризику не більше 100 млрд кубометрів газу при спроможності понад 120 млрд кубометрів газу. Це загрожує втратою України свого геостратегічного становища як головної транзитної ланки трубопровідної системи "Схід-Захід".

Важливе значення має підвищення надійності забезпечення паливом електроенергетики, послаблення залежності від імпорту топкового мазуту, природного газу та заміщення у перспективі цих видів палива вугіллям. До речі, в розвинених країнах для виробництва електроенергії використовується цілий спектр сучасних технологій. Це робиться з двох причин: рівноенергетичності деяких технологій; необхідності диверсифікації видів і джерел палива для надійності забезпечення ним ТЕС та енергетичної незалежності. Варто було б застосувати такий підхід і в нашій державі. Для підвищення надійності та безпеки функціонування атомних електростанцій необхідно провести реконструкцію та модернізацію діючих блоків типу ВВЕР, вдосконалити систему науково-технічної підтримки атомно-енергетичного комплексу.

Виведення з експлуатації ЧАЕС слід пов'язати з фінансуванням цієї програми західними державами та міжнародними фінансовими організаціями. В іншому разі ця проблема ляже величезним тягарем на власний бюджет, негативно вплине на здійснення реформ, енергетичну безпеку, викличе соціальну напруженість у суспільстві.

Особливо важливе значення з точки зору забезпечення енергетичної безпеки має створення в Україні власного ядерно-паливного циклу. Це дасть змогу гарантувати незалежність забезпечення АЕС паливом та зекономити валютні кошти.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Росія здійснює політику паралельних стратегій, одна з яких полягає у прискореному розвитку нафтогазової інфраструктури на власній території, а друга – у встановленні контролю за об'єктами нафтогазової інфраструктури в колишніх союзних республіках. Зокрема, друга стратегія спрямована на заволодіння правами власності на важливі об'єкти нафтогазового комплексу в процесі приватизації або на отримання їх як компенсації за державні борги, що виникли в результаті імпорту російських енергоносіїв.

Енергозбереження – це найбільш перспективний напрям забезпечення України енергією. Лише за рахунок маловитратних енергозберігаючих заходів в Україні можна заощадити не менше 10% енергії.

До першочергових заходів щодо забезпечення енергозбереження слід віднести:

- розробку Державної комплексної програми енергозбереження;
- технічне переоснащення виробництва;
- створення загальнодержавного позабюджетного фонду енергозбереження;
- суворий облік та контроль за енергоспоживанням в усіх галузях виробництва;
- стимулювання впровадження енергозберігаючих технологій;
- здійснення структурної перебудови економіки шляхом зменшення частки енергомістких галузей виробництва;
- використання нетрадиційних та відновлювальних джерел енергії.

Без зміни структури вітчизняного виробництва, без впровадження енергозберігаючих технологій Україна й надалі змушена буде залежати від імпорту нафти і газу, втрачаючи значні кошти.

Важливим державним завданням є також поліпшення системи управління галузями паливно-енергетичного комплексу та процесом енергозбереження.

Таким чином, виходячи з вищеперерахованого, розробка програмного забезпечення системи підтримки індикативного аналізу, є актуальною задачею, яка потребує вирішення у даній магістерській роботі.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень по профілю теми магістерської роботи

ФінЕкАналіз

Програма Фінансовий аналіз – ФінЕкАналіз – професійна автоматизована система комплексного фінансово-економічного й управлінського аналізу господарської діяльності підприємства.

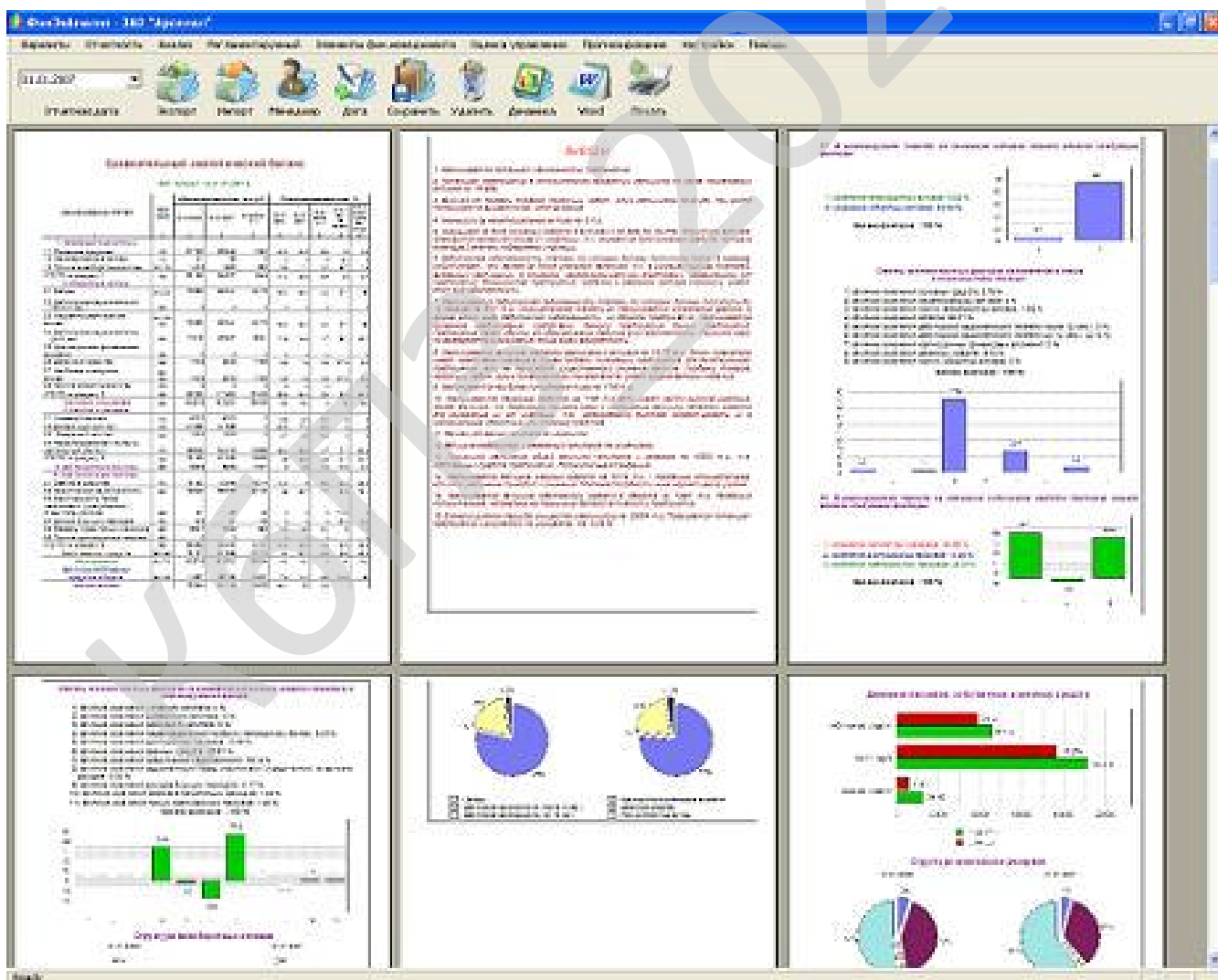


Рисунок 2.1 – Інтерфейс користувача ФінЕкАналіз

Це незамінний помічник для аудитора, економіста, фінансиста, арбітражного керуючого.

У програмі реалізовано більше 40 різних аналітичних блоків з величезною кількістю коефіцієнтів.

Основна мета програми ФінЕкАналіз – зробити фінансовий аналіз організацій ефективним, зручним і оперативним.

Автоматизована система фінансового аналізу – ФінЕкАналіз прискорює й підвищує ефективність проведення аналізу фінансового стану підприємства й робить його комплексним.

У програмі фінансового аналізу – ФінЕкАналіз реалізовані наступні блоки аналізу: регламентований аналіз, елементи фінансового менеджменту, оцінка управління, прогнозування, аналіз порівняльного аналітичного балансу, платоспроможності, ринкової (фінансової) стійкості, ліквідності балансу, ділової активності, рентабельності, прибутки, бальна оцінка фінансової стійкості, кредитоспроможності підприємства за методикою Ощадбанку РФ, стани й відтворення основних засобів, руху грошових коштів, ефективності управління частками й акціями, матричний аналіз, фінансовий аналіз для арбітражного керуючого і інші.

Простий і зручний інтерфейс програми фінансового аналізу вигідно відрізняє її від інших аналогічних пакетів, що відзначається більшістю користувачів. Важливою особливістю програми є відкритість використовуваних економічних розрахунків для користувачів.

У програмі реалізовано більше 40 різних аналітичних блоків з величезною кількістю коефіцієнтів.

Програма фінансового аналізу включає наступні основні аналітичні блоки:

1. Аналіз:

- Аналіз порівняльного аналітичного балансу.
- Аналіз платоспроможності.
- Аналіз ринкової стійкості.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- Аналіз фінансової стійкості.
 - Оцінка ліквідності балансу.
 - Аналіз ділової активності.
 - Аналіз і оцінка прибутковості й рентабельності.
 - Аналіз фінансових результатів.
 - Бальна оцінка фінансової стійкості.
 - Аналіз кредитоспроможності підприємства за методикою Ощадбанку.
 - Аналіз стану й відтворення основних засобів.
 - Аналіз руху грошових коштів.
 - Таблиця основних фінансових показників.
 - Аналіз забезпеченості запасів джерелами їхнього формування.
 - Оцінка ризику кредитування клієнтів.
 - Матричний аналіз.
 - Оцінка ефективності управління частками й акціями.
 - Аналіз праці й заробітної плати.
 - Аналіз у динаміку.
2. Регламентований аналіз:
- Експрес-аналіз для адміністрацій районів.
 - Аналіз ФХД на предмет виявлення ознак навмисного банкрутства.
 - Аналіз при аудиторській перевірці АТ, що підлягає приватизації.
3. Елементи фінансового менеджменту:
- Розрахунок точки беззбитковості.
 - З використанням операційного левериджу.
 - З використанням даних аналітичного обліку.
 - Розрахунок ефекту фінансового важеля.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

- Поточні фінансові потреби.
- Аналіз використання капіталу.
- Розрахунок податкового тягара підприємства.
- Оцінка фінансового потенціалу.
- Оцінка ризику виїзної податкової перевірки.

4. Оцінка управління:

- Розрахунок середньозваженої вартості капіталу.
- Оцінка економічної доданої вартості.
- Комплексна оцінка інтенсивності й ефективності господарської діяльності.

- Моделі оцінки ймовірності банкрутства.
- Матриця фінансової стратегії.
- Модель Дюпона.

5. Прогнозування:

- Вихідні планові параметри прогнозування.
- Прогнозний баланс із урахуванням сформованих тенденцій, прогнозних обсягів і рентабельності продажів.
- Прогнозний баланс, що задовольняє заданим користувачем показникам платоспроможності.

- Прогнозний баланс, що задовольняє нормативним вимогам.

6. Звітність (вхідні дані):

- Форма №1 – Бухгалтерський баланс.
- Форма №2 – Звіт про прибутки й збитки.
- Форма №3 – Звіт про зміни капіталу.
- Форма №4 – Звіт про рух грошових коштів.
- Форма №5 – Додаток до бухгалтерського балансу.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- Додаткові дані.
- Журнал-ордер №10.
- Для аналізу арбітражного керуючого.
- Форма №5 – Додаток до бухгалтерського балансу.

Автоматизована система контролю й обліку енергоресурсів

Головне призначення автоматизованої системи контролю й обліку енергетичних ресурсів є здійснення постійного контролю процесів виробництва, розподілу й споживання всіх теплоенергетичних ресурсів підприємства й своєчасне формування необхідної інформації для рішення економічних і технологічних завдань. Основні функції, виконувані системою:

- об'єднання даних по виробництву й споживанню енергоресурсів з різнорідних і територіально розподілених джерел інформації;
- своєчасне забезпечення необхідною інформацією керівників і фахівців всіх рівнів;
- створення прозорої системи обліку енергоресурсів, розрахунку балансів по виробничих об'єктах і видам енергоресурсів;
- визначення фактичних норм споживання енергоресурсів, аналіз отриманої інформації й на підставі цих даних підвищення точності планування споживання енергоресурсів.

Архітектура системи складається із чотирьох рівнів:

- перший рівень – рівень локальних територіально розподілених джерел інформації з розподілу й споживання всіх видів теплоенергетичних ресурсів;
- другий рівень – рівень збору інформації, здійснюваний автоматично, ручним уведенням або за допомогою спеціалізованого додатка обробки планіметричних діаграм;
- третій рівень – рівень єдиної інформаційної системи виробництва, у якій здійснюється об'єднання всієї виробничої інформації з технології, якості й

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

енергетиці й виконуються функціональні й аналітичні завдання по контролі, балансуванню й статистичній обробці даних;

– четвертий рівень – рівень автоматизованих робочих місць спеціалізованих за обсягом й характером виконуваних функцій.

Як сервер бази даних для зберігання різної нормативно-довідкової інформації, а також даних по виробленню й споживанню енергоресурсів використовується Microsoft SQL Server або й Oracle. Для довгострокового зберігання й оперативного доступу до результатів, отриманих на підставі обробки й аналізу даних по виробленню, споживанню й розподілу енергоресурсів, використовується сховище даних на базі PI System компанії OSIsoft або Proficy Historian компанії GE Fanuc.

Основний функціонал системи з погляду користувача зосереджений у спеціалізованих автоматизованих робочих місцях фахівців. До складу системи входять наступні робочі місця:

- АРМ Обліку ТЕР;
- АРМ Планування й прогнозування;
- АРМ Планіметр;
- АРМ Адміністратора.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19



Рисунок 2.2 – Інтерфейс користувача АРМ обліку теплоенергетичних ресурсів

АРМ Обліку ТЕР є основним програмним модулем, призначеним для уведення й коректування даних, відображення інформації про виробництво, розподіл і споживання теплоенергетичних ресурсів, розрахунку основних технологічних параметрів і формування всіх видів звітності. Основними користувачами системи, як правило, є фахівці відділів головного енергетика (ВГЕ), фахівці управління головного енергетика (УГЕ), фахівці виробничо-технічного відділу (ВТО), планово-економічного відділу (ПЕВ) і ін.

АРМ Обліку ТЕР вирішує наступні завдання:

- ведення схем розподілу теплоенергетичних ресурсів; моніторинг споживання енергоресурсів по видах ресурсів і по об'єктах виробництва; аналіз добових даних по споживанню енергоресурсів;
- коректування даних, виправлення грубих помилок у вимірах;

- формування оперативного добового балансу по виробленню й споживанню енергоресурсів;
- розрахунок балансів по видах енергоресурсів за місяць; формування інформації зі споживання електроенергії й енергоресурсів сторонніми організаціями (субабонентами);
- розрахунок фактичних питомих витрат енергоресурсів по об'єктах виробництва й реєстрація відхилень фактичних показань від нормативів;
- формування оперативної й аналітичної звітності.

АРМ Планування й прогнозування займає важливе місце в системі контролю й обліку енергоресурсів. АРМ призначений для рішення наступних завдань:

- аналіз місячних балансів по електроенергії й тепловій енергії;
- контроль і прогноз дотримання лімітів по споживанню енергоресурсів;
- аналіз фактичного питомого споживання електроенергії й енергоресурсів за добу, місяць;
- формування по фактичній статистиці питомих норм споживання енергоресурсів на одиницю зробленої продукції по установках і виробництвам у цілому.

АРМ Планіметр призначений для обробки даних, отриманих у вигляді стрічкових або кругових картограм від самописних приладів, використовуваних для реєстрації значень тих або інших технологічних параметрів. АРМ Планіметр забезпечує розрахунок інтегральних витрат за добу в об'ємних, масових і умовних одиницях і вивід даних з картограм у базу даних реального часу з необхідною дискретністю. Планіметрування здійснюється за допомогою спеціалізованих електронних пристроїв – планіметрів, у парі з яким працює спеціалізоване програмне забезпечення.

Програмне забезпечення здійснює автоматичний пошук на відсканованому зображенні «тренду», тобто лінії, що максимально повторює лінію картограми Знайдений тренд зображується на картограмі для візуального

контролю. Оператор при необхідності вносить виправлення й дає команду на збереження даних.

Система I-EMS

Система I-EMS призначена для ефективного контролю й обліку процесів генерації, розподілу й споживання електричної, теплової енергії й інших видів енергоресурсів, а також своєчасного формування необхідної інформації для рішення економічних і технологічних завдань.

Мети впровадження системи:

– Створення єдиної інформаційної платформи для моніторингу вироблення, споживання й розподілу енергоресурсів.

– Створення прозорості системи обліку енергоресурсів, розрахунку балансів по виробничих об'єктах і видам енергоресурсів.

– Підвищення ефективності використання енергоресурсів, зниження питомих витрат їхнього споживання за рахунок виявлення основних джерел втрат, зниження перевитрати, оптимізації розподілу придбаних і власних енергоресурсів.

– Підвищення точності планування споживання енергетичних ресурсів на основі результатів аналізу інформації про фактичні норми споживання за попередні періоди.

– Створення платформи для реалізації перспективних завдань енергоменеджмента, таких як завдання довгострокового й оперативного прогнозування споживання енергоресурсів.

Призначення системи:

– Інтеграція даних по виробництву й споживанню енергоресурсів з різнорідних і територіально-розподілених джерел інформації на базі єдиної інформаційної платформи.

– Автоматизація процесів обробки інформації з енергоспоживання.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

– Своєчасне забезпечення необхідною інформацією керівників і фахівців всіх рівнів для підтримки робочих процесів управління на основі оперативних і достовірних даних по виробленню, споживанню й розподілу енергоресурсів.

– Забезпечення погодженими даними функціональних додатків, призначених для рішення завдань моделювання й оптимізації енергоспоживання.

– Підвищення ефективності процесів аналізу споживання енергоресурсів за рахунок використання історичних даних, створення ієрархічної структури даних, розширення можливостей візуалізації інформації, інтеграції різних систем автоматизації й додаткових програмних продуктів.

I-EMS забезпечує рішення широкого спектра завдань автоматизації процесу контролю й обліку енергоресурсів, таких як:

– Автоматичний збір даних реального часу по енергоспоживанню з АСКУЕ, АСТУЕ, АПСКУЕ.

– Автоматичний збір інформації про технологічні параметри, стан і ступінь завантаження встаткування з АСУТП.

– Ручне уведення інформації з енергоспоживання на об'єктах, не обладнаних програмно-апаратними засобами.

– Довгострокове й надійне зберігання даних.

– Надання достовірної технологічної інформації про виробництво й споживання всіх видів енергоресурсів у режимі реального часу.

– Автоматизована обробка й аналіз значень параметрів енергоспоживання по затвердженим на підприємстві алгоритмам і регламентам.

– Реалізація методик розрахунку об'ємної й масової витрати, кількості тепла й тепломісткості (ентальпії й калорійності) різних видів енергоресурсів на підставі діючих ДСТ і ГСССД.

– Контроль вірогідності роботи вузлів обліку.

– Облік вироблених і енергоресурсів, що витрачаються, по видах ресурсів і технологічних об'єктів.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

- Формування фактичних балансів енергетичних ресурсів.
- Формування звітності про фактичні й питомі витрати енергоресурсів за зазначений проміжок часу (місяць, декада, доба й т.д.).
- Планування й прогнозування споживання енергоресурсів на основі статистичних даних.

I-EMS являє собою відкриту розподілену багаторівневу автоматизовану систему. Для зберігання нормативно-довідкової інформації, а також даних по виробленню й споживанню енергоресурсів використовується Microsoft SQL Server. I-EMS має модульну структуру, що дозволяє забезпечити поетапне впровадження системи, наступну модернізацію й розширення без заміни програмних продуктів і змін загальної структури системи.

Сервер додатків реалізований у вигляді сервісу операційної системи й призначений для обробки запитів користувачів. До складу сервера входять модулі обробки запитів по конфігуруванню метаданих і модулі завантаження й вивантаження даних із зовнішніх і в зовнішні сховища даних реального часу.

До складу системи входять наступні клієнтські додатки: АРМ “Адміністратор”, АРМ “Облік ТЕР” (облік теплоенергетичних ресурсів), АРМ “Ведення НДІ” (ведення нормативно-довідкової інформації), АРМ “Планіметр”, АРМ “Планування й прогнозування”, АРМ “Формування звітності й передачі даних в ERP”. Состав системи може бути розширений додатковими модулями.

АРМ “Адміністратор” дозволяє змінювати налаштування системи, синхронізувати налаштування всіх інтегровальних систем.

АРМ “Облік ТЕР” – основний модуль системи, призначений для контролю й обліку даних про виробіток, споживання й розподіл енергоресурсів, а також для формування необхідної звітності.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

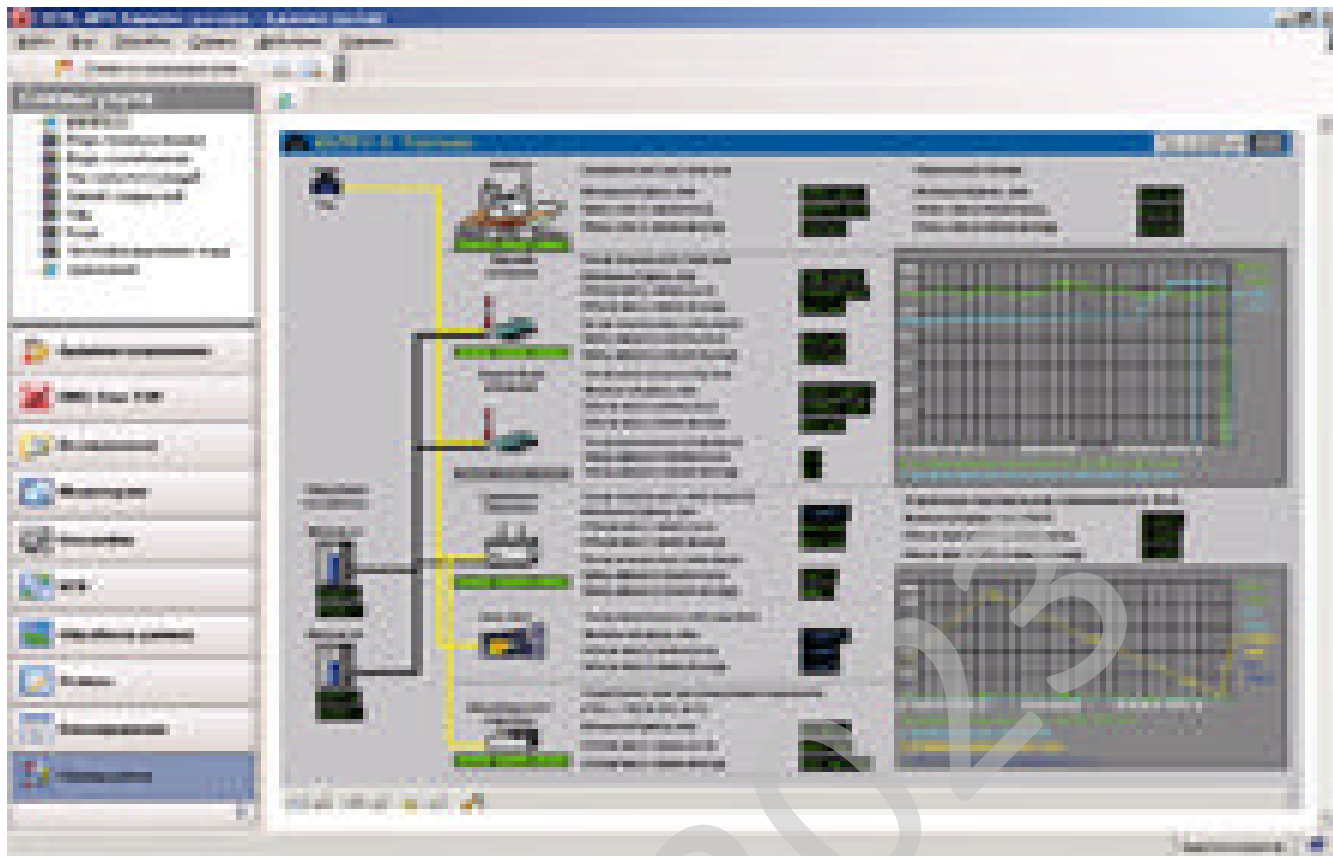


Рисунок 2.3 – Інтерфейс виводу інформації у вигляді мнемосхем

АРМ “Ведення НДІ” забезпечує централізоване ведення нормативно-довідкової інформації, призначений для підтримки в актуальному стані класификаторів і довідників, конфігурування бази дані системи.

АРМ “Планування й прогнозування” призначений для планування споживання електроенергії й інших видів енергоресурсів, формування норм споживання, контролю за дотриманням норм і лімітів, формування необхідної звітності.

Спеціалізований модуль АРМ “Планіметр” призначений для обробки даних, що надходять від самописних приладів у вигляді стрічкових і дискових діаграм.

Крім того, до складу системи входить надбудова до Microsoft Excel EMS-Datalink, призначена для формування звітів користувачами без додаткового програмування.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

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

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

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

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

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

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

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

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

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

Єдине керування пам'яттю

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

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

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

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

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

Підвищення якості й швидкодії інструментів

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

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

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів. Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентів на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуемі 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.

						ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			30

- Реалізований заново стилізуємий FMX компонент TМето на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

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 підтримки індикативного аналізу.

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 методикку побудови системи контролю роботи

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

технологічного обладнання на виробництві в автоматизованому режимі.
Розробити функціональну та структурну схеми системи;

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБГІЗ - 2023

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розкриємо основні завдання й мети дослідження проблеми забезпечення ЕБ України і її регіонів і виділимо основні напрямки дослідження проблеми ЕБ, до яких відносяться наступні:

- оцінка поточного стану й перспектив розвитку ПЕК з позицій ЕБ (моніторинг, оцінка погроз, виявлення й прогноз вузьких місць і т.д.);
- обґрунтування мер організаційного, структурно-технологічного й фінансово-енергетичного характеру по забезпеченню ЕБ;
- розробка методів і засобів обґрунтування заходів щодо забезпечення ЕБ (математичні моделі, інформаційна технологія).

Розглянемо питання обліку вимог ЕБ при дослідженні розвитку енергетики країни і її регіонів. Запропонована в ІПМЕ НАНУ дворівнева технологія досліджень при рішенні проблеми забезпечення ЕБ, що дозволяє досліджувати відповідні особливості роботи систем енергетики в ПЕК і на системному рівні. При цьому верхній рівень зазначеної технології представляють моделі для проведення досліджень по оцінці стану ПЕК при можливих збурюваннях і їхнього впливу на умови паливо- і енергопостачання споживачів з позицій забезпечення ЕБ. Нижній рівень ієрархії представляють галузеві моделі, що дозволяють оцінити потенційні можливості обласних систем нафтопостачання (СНП) і газопостачання (СГП) по задоволенню споживачів відповідними енергоресурсами як у нормальних умовах функціонування, так і в умовах надзвичайних ситуацій.

Опишемо балансову економіко-математичну модель ПЕК України, призначену для дослідження розвитку енергетики України і її регіонів з позицій ЕБ. У математичному змісті дана модель являє собою класичне завдання

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

лінійного програмування, у змістовному (енергоекономічному) – модель, що базується на традиційній територіально-виробничій моделі ПЕК, із блоками електроенергетики, тепло-, газо- і углепостачання, а також мазутопостачання.

Надамо аналіз розробок, що поклали початок сучасної інформаційної технології дослідження ПЕК, і використовуваного на сьогоднішній день інструментарію для дослідження розвитку ПЕК з позицій ЕБ. При цьому в дослідженнях для одержання найбільш достовірного рішення бажано формувати якнайбільше можливих перспективних станів ПЕК, за умови реалізації погроз ЕБ. Традиційна схема проведення досліджень, заснована на детальному аналізі обмеженого числа експертно обраних варіантів розвитку ПЕК не зовсім ефективний у силу неповноти набору можливих варіантів, тривалості й трудомісткості процесу підготовки й розрахунку кожного сценарію, що розраховується. Звідси необхідність у використанні методів комбінаторного моделювання при дослідженні розвитку енергетики з позицій ЕБ.

Розглянемо методологію дослідження варіантів розвитку ПЕК з позицій ЕБ, формуванню принципів створення системи підтримки індикативного аналізу ЕБ України.

Кожен сформований за допомогою методів комбінаторного моделювання стан характеризує окремий варіант розвитку енергетики в конкретний момент часу, формується шляхом різних сполучень станів галузей енергетики по регіонах у цей момент часу.

Виділення раціональних з позицій ЕБ варіантів розвитку ПЕК здійснюється за допомогою проведення індикативного аналізу ЕБ, заснованого на порівнянні значень індикаторів ЕБ з їх граничними або гранично припустимими значеннями, і підготовку на цій основі відповідних висновків і рекомендацій. Якісно в процедурі індикативного аналізу виділяються нормальна, передкризова й кризова ситуації в забезпеченні ЕБ територій.

Для проведення індикативного аналізу в теперішній час використовуються найважливіші індикатори ЕБ, що задовольняють двом вимогам: у сукупності

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

вони повинні покривати весь обсяг інформаційних потреб завдань забезпечення ЕБ; кожний індикатор, їхня сукупність повинні забезпечити максимально можливу вірогідність інформації. Експертно було виділено 10 найважливіших індикаторів ЕБ, розподілених по блоках самозабезпеченості, живучості системи паливо- і енергопостачання, стану основних виробничих фондів. Список даних індикаторів запропонований у таблиці 3.1.

Далі розкриємо основну проблему при проведенні досліджень за допомогою методів комбінаторного моделювання, покажемо можливі шляхи її рішення. Як відзначалося раніше, відобразити граф розвитку ПЕК дуже великих розмірів практично неможливо, а притягнутим до дослідження експертам фізично неможливо проаналізувати всі можливі стани ПЕК, зрівняти їх між собою. Наприклад, для трьох енергетичних галузей, чотирьох географічних зон і всього лише двох сценаріїв розвитку виділених галузей усього лише для одного року експертові необхідно проаналізувати 512 можливих станів. Положення ускладнюється ще й можливою близькістю розрахункових станів ПЕК по характеристиках функціонування енергетики в регіонах, що приводить до наявності в графі практично ідентичних станів, які можуть дати майже що не відрізняються друг від друга перспективні напрямки розвитку енергетики. І навіть в випадку відбору логічно можливих станів і переходів (відсікання переходів від кращого стану до гіршого) досліджувати подібний граф можливо лише в рамках спеціалізованої системи підтримки індикативного аналізу ЕБ, і то за умови скорочення в ньому числа станів. Для того, щоб скоротити число досліджуваних перспективних станів і створити базу для проведення їхнього аналізу з позицій ЕБ, у роботі сформульовані принципи обліку вимог ЕБ при аналізі графів розвитку ПЕК, створених за допомогою методів комбінаторного моделювання. Вибір найбільш представницьких станів ПЕК у графі – додатковий крок, що дозволяє експертам згодом виробити найбільш ефективні заходи щодо забезпечення ЕБ досліджуваних територій.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Таблиця 3.1 – Найважливіші індикатори енергетичної безпеки

Група індикаторів	Індикатори енергетичної безпеки
Блок самозабезпеченості	<p>1. Відношення величини сумарної розташовуваної потужності електростанцій регіону до максимального електричного навантаження споживачів на його території.</p> <p>2. Відношення величини суми розташовуваної потужності електростанцій і пропускної здатності міжсистемних зв'язків регіону із сусідніми до максимального електричного навантаження споживачів на його території.</p> <p>3. Частка власних первинних ПЕР у споживанні котельно-грубого палива (КГП) на території.</p>
Блок живучості системи паливо- і енергопостачання	<p>1. Частка домінуючого ресурсу в загальному споживанні КГП на території регіону.</p> <p>2. Частка найбільш великої електростанції у встановленій електричній потужності регіону.</p> <p>3. Рівень потенційної забезпеченості попиту на теплову енергію в умовах різкого похолодання (10% споживання) на території регіону</p>
Блок стану активної частини ОВФ	<p>1. Ступінь зношування основних виробничих фондів (ОВФ) електроенергетики регіону.</p> <p>2. Ступінь зношування ОВФ підприємств паливної промисловості території.</p> <p>3. Ступінь зношування ОВФ у теплоенергетиці регіону</p> <p>4. Відношення уведення встановленої потужності й технічного переозброєння електростанцій території за попередній 5-літній період до встановленої потужності на території.</p>

Вимоги до основних програмних компонентів розробленої системи наступні.

– Підсистема розрахунку індикаторів ЕБ повинна бути призначена для одержання кількісної і якісної оцінок окремих індикаторів ЕБ для кожного можливого стану ПЕК, щоб скористатися ними на етапах формування розрідженого графа розвитку ПЕК і його подальшого аналізу з позицій ЕБ.

– Середовище для відображення графів розвитку ПЕК і їхнього аналізу з позицій ЕБ повинна бути призначена для проведення дослідження розрідженого графа. При формуванні результуючого графа повинна виконуватися двоступінчаста процедура відсічення неприйнятних станів, на першому етапі якої відтинаються неприпустимі по найважливіших індикаторах ЕБ стани ПЕК, на другому – вибираються найбільш представницькі стани за допомогою методів кластерного аналізу.

У розглянутому середовищі велика роль повинна приділятися візуальному поданню результатів проведеного аналізу, у тому числі на вузлах і дугах графа. Для оцінки конкретних аспектів розвитку енергетики з позицій ЕБ необхідний вибір раціональних траєкторій розвитку ПЕК за станом окремих індикаторів ЕБ. Проведення комплексного аналізу траєкторій розвитку ПЕК можливо у випадку вибору субраціональних за найважливішими індикаторами ЕБ траєкторій розвитку, у множині яких експерт зможе вибрати найбільш перспективну на його погляд траєкторію, можливо не що цілком відповідає вимогам ЕБ.

Питання пошуку траєкторії, найбільш близької до раціонального з позицій ЕБ, у розроблювальному середовищі може бути вирішений за допомогою одного з методів багатокритеріальної оптимізації. Такий підхід може підвищити об'єктивність одержуваних комплексних оцінок, не вимагаючи активної участі притягнутих експертів.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

У роботі сформульовані наступні основні вимоги до середовища для відображення графів розвитку ПЕК, і їх аналізу з позицій ЕБ:

1. Реалізація користувальницького інтерфейсу для вивчення графа великої розмірності.
2. Залучення графічних рішень для аналізу станів ПЕК за критеріями функціонування й розвитку енергетики України.
3. Можливість проведення індикативного аналізу станів ПЕК на рівні областей України.
4. Можливість пошуку раціональної й субраціональних траєкторій розвитку ПЕК по індикаторах ЕБ.
5. Забезпечення взаємодії із зовнішніми додатками й базами даних.

Розробка подібного інструментарію може бути проведена на базі спеціалізованого програмного забезпечення для роботи із графами. Такий підхід дозволяє на базі універсальних бібліотек для роботи із графами, створити більш спеціалізовані програми для роботи із графами, спрямовані на рішення конкретних прикладних завдань. Відповідно до цього була запропонована й реалізована наступна технологія створення середовища для відображення графів розвитку ПЕК і їхнього аналізу з позицій ЕБ:

1. Вибір спеціалізованого програмного забезпечення для роботи із графами згідно пред'явленим до середовища вимогам.
2. Вивчення вимог бібліотеки графів до подання графів.
3. Реалізація алгоритмів пошуку раціональної й субраціональних траєкторій розвитку ПЕК по індикаторах ЕБ.
4. Реалізація взаємодії з підсистемою розрахунку індикаторів ЕБ.
5. Інтеграція інструментарію для відображення результатів індикативного аналізу можливих станів ПЕК у спеціалізоване програмне забезпечення.

На підставі сказаного модель середовища для дослідження графів розвитку ПЕК повинна включати:

- блок відображення графів розвитку ПЕК, що підтримує засобт графічного уведення й виводу;

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- блок аналізу станів графа, у рамках якого проводиться ідентифікація станів ПЕК по оцінках індикаторів ЕБ;
- блок аналізу траєкторій розвитку ПЕК, що підтримує вибір раціональної й субраціональних траєкторій розвитку ПЕК по оцінках індикаторів ЕБ.

Якісно ситуація з рівнем ЕБ України і її регіонів може бути наочно відображена безпосередньо в графі розвитку ПЕК шляхом колірної диференціації вузлів і дуг графа розвитку ПЕК. Для цього детальний аналіз станів ПЕК з обліком територіального й часового аспектів повинен проводитися в рамках підсистеми індикативного аналізу можливих станів ПЕК. При цьому повинні бути враховані вимоги методики проведення індикативного аналізу, а залучення засобів когнітивної графіки може дозволити аналізувати структуру й зміст даних через колірні рішення, форму, розміри, текстуру, позицію, орієнтацію графічних елементів зображення (наприклад, стовпці, сектори й точки діаграм, колірна ідентифікація полігонів електронних карт або вершин спрямованих графів).

У цілому підсистема індикативного аналізу можливих станів ПЕК повинна відповідати наступним вимогам:

1. Підтримка вимог методики проведення індикативного аналізу ЕБ (одержання нормованих оцінок, одержання на їхній основі якісної оцінки стану по регіонах для окремих індикаторів ЕБ і їхніх груп).
2. Графічне відображення результатів індикативного аналізу станів ПЕК.
3. Можливість створення різних наборів індикаторів.

Аналіз спеціалізованого програмного забезпечення для індикативного аналізу інших предметних областей показав, що вузька предметна спрямованість програмних продуктів найчастіше сприяє введенню обмежень на вихідні набори аналізованих показників, а також особливостям методики проведеного аналізу. Однак адаптувати подібне програмне забезпечення для рішення завдання індикативного аналізу ЕБ за умови гнучкості й відкритості вихідних наборів даних і критеріїв їхнього аналізу, схожості принципів проведення аналізу,

наочному графічному поданні результатів аналізу має зміст. Це дозволить створити відносно універсальний інструментарій для аналізу розвитку енергетики України і її регіонів, у тому числі за допомогою індикаторів ЕБ, природно з обліком сформульованих раніше вимог. Тому, у рамках роботи була запропонована й реалізована наступна методика адаптації спеціалізованого програмного забезпечення для рішення завдання відображення результатів індикативного аналізу ЕБ:

1. Дослідження формату вихідних даних для проведення індикативного аналізу.
2. Дослідження алгоритмів нормування вихідних даних, їхні згортки з урахуванням пріоритетності.
3. Облік можливості розбивки даних по територіальній і часовій ознаках.
4. Дослідження підходу до виділення якісного стану об'єкта дослідження.
5. Вивчення й вибір найбільш представницьких форм відображення результатів аналізу.
6. Перетворення фактичних значень індикаторів ЕБ для одержання оцінки рівня ЕБ по прийнятим у програмному забезпеченні правилам виділення якісного стану об'єкта дослідження.
7. Реалізація інтеграції досліджуваного програмного забезпечення.

3.2 Розробка структурної схеми

Розглянемо структурну схему системи підтримки індикативного аналізу ЕБ України.

У комплексі формуються графи розвитку галузей ПЕК із заданими характеристиками функціонування галузей. Далі за допомогою блоку комбінаторного моделювання формується граф розвитку ПЕК, кожний стан якого наступним кроком розраховується на балансовій економіко-математичній моделі ПЕК. Із графа виключаються стани, неприпустимі по фінансових і ресурсних

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

обмеженнях. Наступним етапом проводиться кластерний аналіз станів, реалізований у блоці кластерного аналізу.

Вибір представницьких станів ПЕК з повного графа розвитку ПЕК відбувається у два етапи. На кожному з етапів для кожного моменту часу вибираються стани, які розташовані найбільше близько до центра кластера. У результаті для кожного моменту часу відбирається кілька десятків станів, з яких формується розріджений граф розвитку ПЕК.

Питання проведення подальшого аналізу отриманого графа розвитку ПЕК були вирішені в рамках розробленої на базі спеціалізованого програмного забезпечення системи підтримки індикативного аналізу ЕБ України.

Узагальнена структурна схема системи, включаючи основні компоненти розробленої автором системи підтримки індикативного аналізу ЕБ (виділені сірим кольором), представлена на рис. 3.1.

Підтримка аналізу графів розвитку ПЕК була виконана в рамках мережного інтегрованого середовища для відображення графів розвитку ПЕК і їхнього аналізу з позицій ЕБ. Відправною точкою в розробці середовища послужили існуючі наробітки в області візуалізації й обробки графів. Аналіз існуючого програмного забезпечення показав, що незважаючи на досить широкий спектр програм, здебільшого вони спеціалізовані, або з позицій предметної області (графи «постачені» певною специфічною семантикою), або з позицій окремих аспектів роботи із графами. Стосовно до завдання дослідження графів розвитку ПЕК, до числа висунутих до даних програм вимог, були віднесені:

- якісна й ефективна робота з більшими графами довільної семантики;
- вибір вільно розповсюдженого програмного забезпечення;
- можливість подальшої підтримки розроблювачами цих проектів;
- відкритість вихідних кодів.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Флойда-Уоршала. На першому етапі послідовним розглядом всіх станів ПЕК від вихідного моменту часу до кінцевого визначаються величини витрат на досягнення кожного стану. На другому – зворотним переглядом станів (від кінцевих станів ПЕК до вихідного) визначаються величини мінімальних витрат, що відповідають системі, що розвивається, з кожного стану до одному з кінцевих станів. Мінімальні витрати по всім які проходять стан траєкторіям ПЕК, рівні сумі значень мінімальних витрат на досягнення даного стану й мінімальних витрат на подальший розвиток, – величина, за якою визначають субраціональність стану. Діапазони числового інтервалу для влучення стану в категорію субраціонального задається експертами величиною збільшення до найкращого значення критерію.

4. Пошук траєкторії, найбільш близької до раціональної з позицій ЕБ, здійснюється за критерієм мінімального усередненого рівня погіршення якості об'єкта дослідження (енергетики) у порівнянні з іншими негіршими альтернативами (траєкторіями). Використовуваний метод оптимізації припускає рівнозначність аналізованих критеріїв, тому в системі розглядається лише як механізм підказки можливого напрямку розвитку ПЕК, найбільш перспективного з позицій ЕБ.

3.3 Розробка функціональної схеми

Функціональна схема системи представлена на рисунку 3.2. Як підсистема індикативного аналізу можливих станів ПЕК відповідно до сформульованої вище методики адаптації спеціалізованого програмного забезпечення для рішення завдання відображення результатів індикативного аналізу ЕБ, був адаптований програмний комплекс вирішальний практично ідентичне завдання аналізу й графічної інтерпретації ситуації, яка відбувається в регіонах відносно найважливіших аспектів їхнього розвитку.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Вибір даного комплексу був обґрунтований з наступних позицій. Реалізований у комплексі механізм обробки даних досить простий і прозорий, і, у принципі, схожий з основними концептуальними моментами індикативного аналізу ЕБ. У комплексі задіяні найбільш інформативні графічні форми подання даних. Істотною перевагою цього інструмента можна вважати доступний структурований шаблон подання вихідних даних, простота організації виклику програми, що може дозволити експертам використовувати програмний комплекс як система експрес-аналізу індикаторів ЕБ. Состав аналізованих показників, регіонів і часових зрізів повністю визначається користувачем, тому програмний комплекс, що розроблений у ході виконання магістерської роботи, може використовуватися для наочного аналізу показників різних предметних областей. Даний інструментарій рекомендований розроблювачами для проведення експертного аналізу в дослідницьких системах.

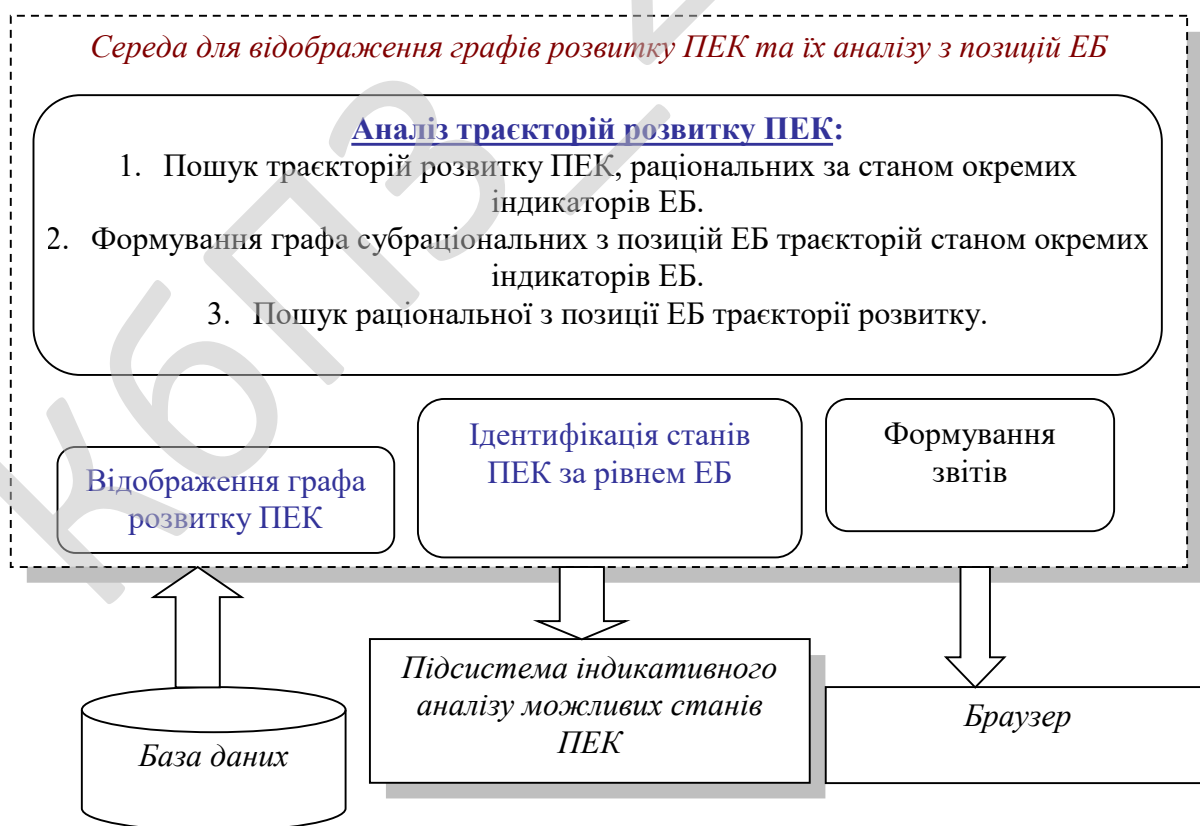


Рисунок 3.2 – Функціональна схема системи

Адаптація програмного комплексу, що розроблений у ході виконання магістерської роботи, була зв'язана з наступним положенням.

Реалізована в комплексі методика проведення аналізу припускає рівнозначність числових інтервалів якісних станів об'єкта дослідження. У рамках індикативного аналізу ЕБ така вимога неприйнятна.

Тому був запропонований і реалізований наступний спосіб перетворення фактичних значень індикаторів ЕБ для одержання коректної якісної оцінки станів ПЕК в програмному комплексі.

На першому кроці визначаються коефіцієнти перерахунку значень кожного з індикаторів ЕБ, виходячи з умов рівнозначності числових інтервалів якісних станів об'єкта дослідження:

$$k_i = \frac{y_{\max}^i}{3 \cdot (y_i - y_{i-1})}, \quad i = \overline{1,3}, \quad (3.1)$$

де y_{\max}^i – максимальний бал у шкалі програмного комплексу, що розроблений у ході виконання магістерської роботи, (100 або 1000 балів),

y_0, y_3 – мінімальне й максимальне значення індикатора ЕБ,

y_1, y_2 – експертно встановлені граничні значення індикатора ЕБ, районировані за умовами розвитку енергетики в регіонах.

Наступним кроком перетворюються фактичні значення індикатора ЕБ для проведення подальшого аналізу в програмному комплексі:

$$x_{ij}^j = (x_{ij} - y_{i-1}) \cdot k_i + \frac{y_{\max}^j \cdot (i-1)}{3}, \quad i = \overline{1,3}, \quad j = \overline{1,m}, \quad (3.2)$$

де x_{ij}^j – наведене j -е значення індикатора, що потрапив в i -ий інтервал;

x_{ij} – фактичне j -е значення індикатора, що потрапив в i -ий інтервал;

m – число значень індикатора, що потрапили в i -ий інтервал;

y_{i-1} – нижня границя i -го числового інтервалу.

Адаптована програма інтегрована в середовище для відображення графів розвитку ПЕК і їхнього аналізу з позицій ЕБ.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Виклик програми виконується через контекстне меню вузлів графа розвитку ПЕК.

З використанням програмно-обчислювального комплекс були проведені дослідження розвитку енергетики України і її областей за період 2005-2013 рр. по опорних роках.

Критеріями аналізу стали вартісні витрати на розвиток і функціонування ПЕК (з урахуванням можливих збитків від недопоставок ПЕР), і два з найважливіших індикатора ЕБ:

можливості задоволення потреб у котельно-грубному паливі (КГП) із власних джерел регіону;

частка домінуючого ресурсу в балансі КГП регіону.

Прогнозовані рівні потреб у первинні ПЕР минулого задані відповідно до орієнтирів Енергетичної Стратегії України до 2020 року. Діапазон потреб у ПЕР формувався для оптимістичного й помірною сценаріїв розвитку енергетики України.

У результаті розрахунків був отриманий граф розвитку ПЕК з 46 представницьких станів ПЕК, що і аналізувався з позицій заданих критеріїв. У роботі зроблений вивід про те, що по кожному аналізованому критерію існують свої раціональні з позицій ЕБ траєкторії розвитку ПЕК.

Всі розроблені безпосередньо автором компонента, працездатність яких підтверджується можливістю аналізу в області виконуваних досліджень у комплексі, саме й представляють заявлену в роботі систему підтримки індикативного аналізу ЕБ.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного вікна ПЗ далі через модуль захисту ПЗ перевіряється цілісність ПЗ.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Після чого проводиться налаштування по запиту, перевірка доступу до мережі та перехід до графу розвитку ПЕК. Звідки проводяться дії з БД ПЗ, блоком кластерного аналізу, блоком комбінаторного моделювання та мережною підсистемою розрахунку індикаторів ЕБ з подальшою взаємодією з мережною підсистемою індикативного аналізу можливих станів ПЕК.

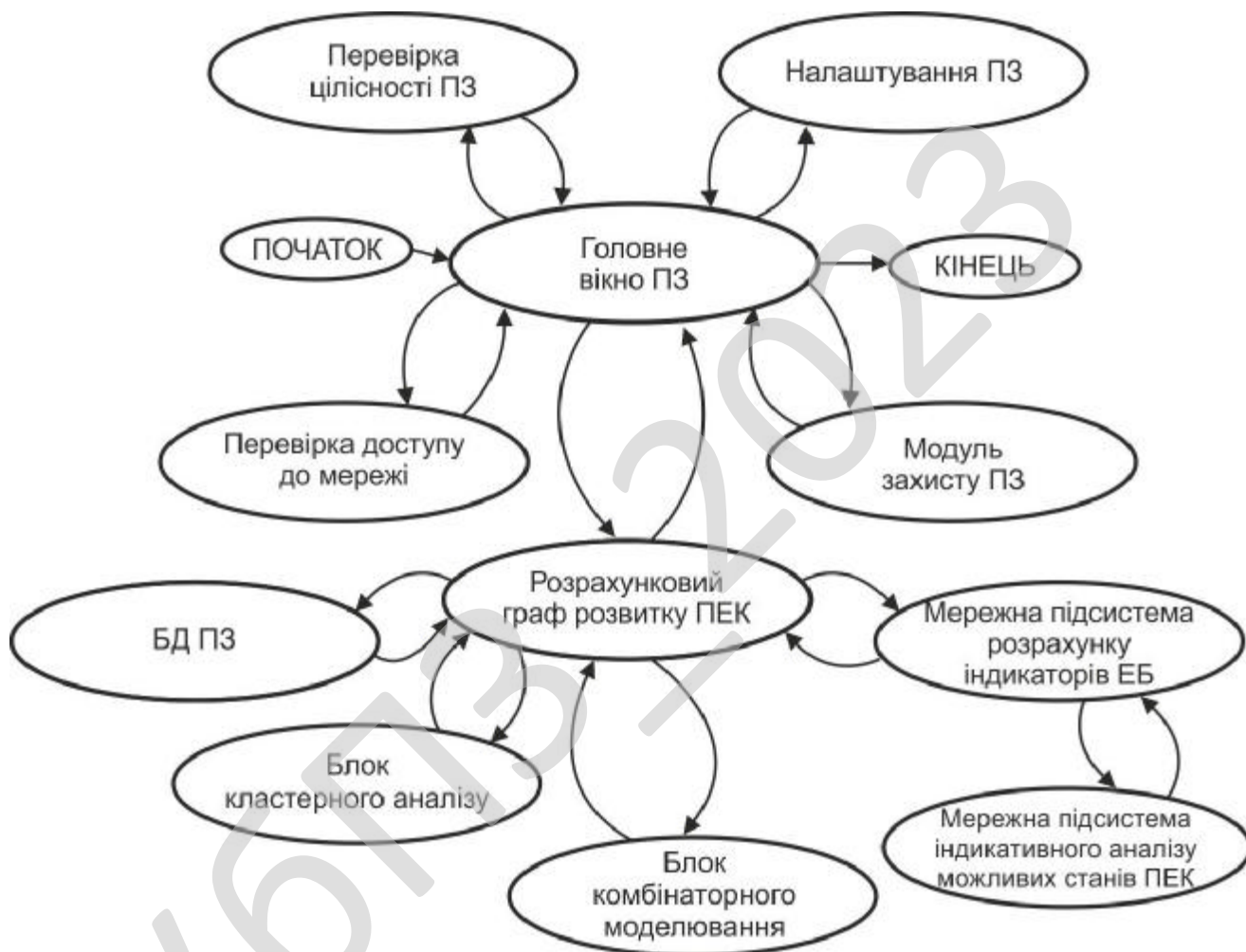


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.1.

З рисунку видно, що після запуску програми спочатку відбувається виведення головного вікна програми. Потім здійснюється:

- Запит аналізу траєкторій розвитку ПЕК (запит).

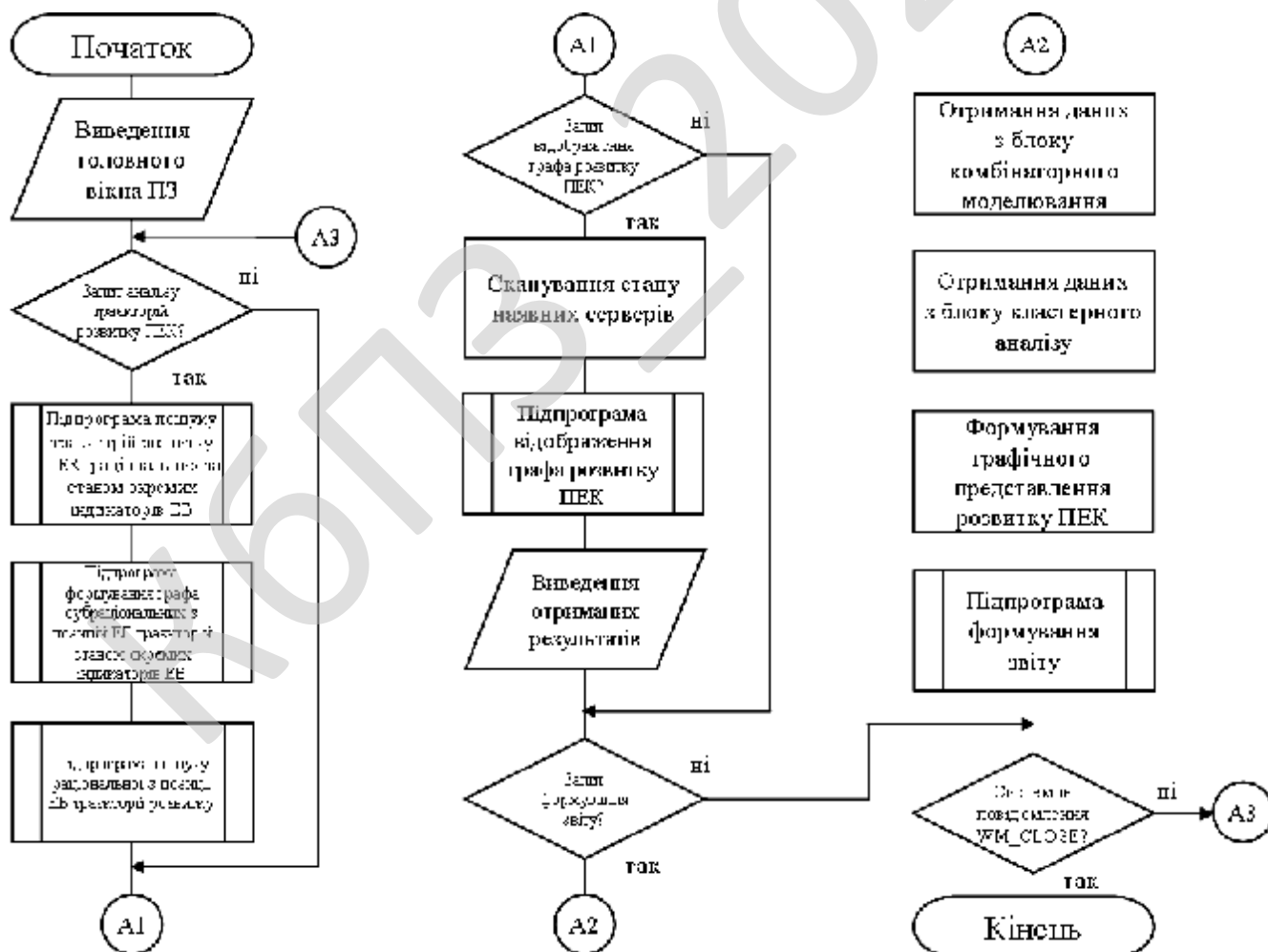


Рисунок 4.1 – Блок-схема основної програми

- Підпрограма пошуку траєкторій розвитку ПЕК, раціональних за станом окремих індикаторів ЕБ.
- Підпрограма формування графа субраціональних з позицій ЕБ траєкторій станом окремих індикаторів ЕБ.
- Підпрограма пошуку раціональної з позиції ЕБ траєкторії розвитку.
- Запит відображення графа розвитку ПЕК?
- Сканування стану наявних серверів.
- Підпрограма відображення графа розвитку ПЕК.
- Виведення отриманих результатів.
- Запит формування звіту?
- Отримання даних з блоку комбінаторного моделювання.

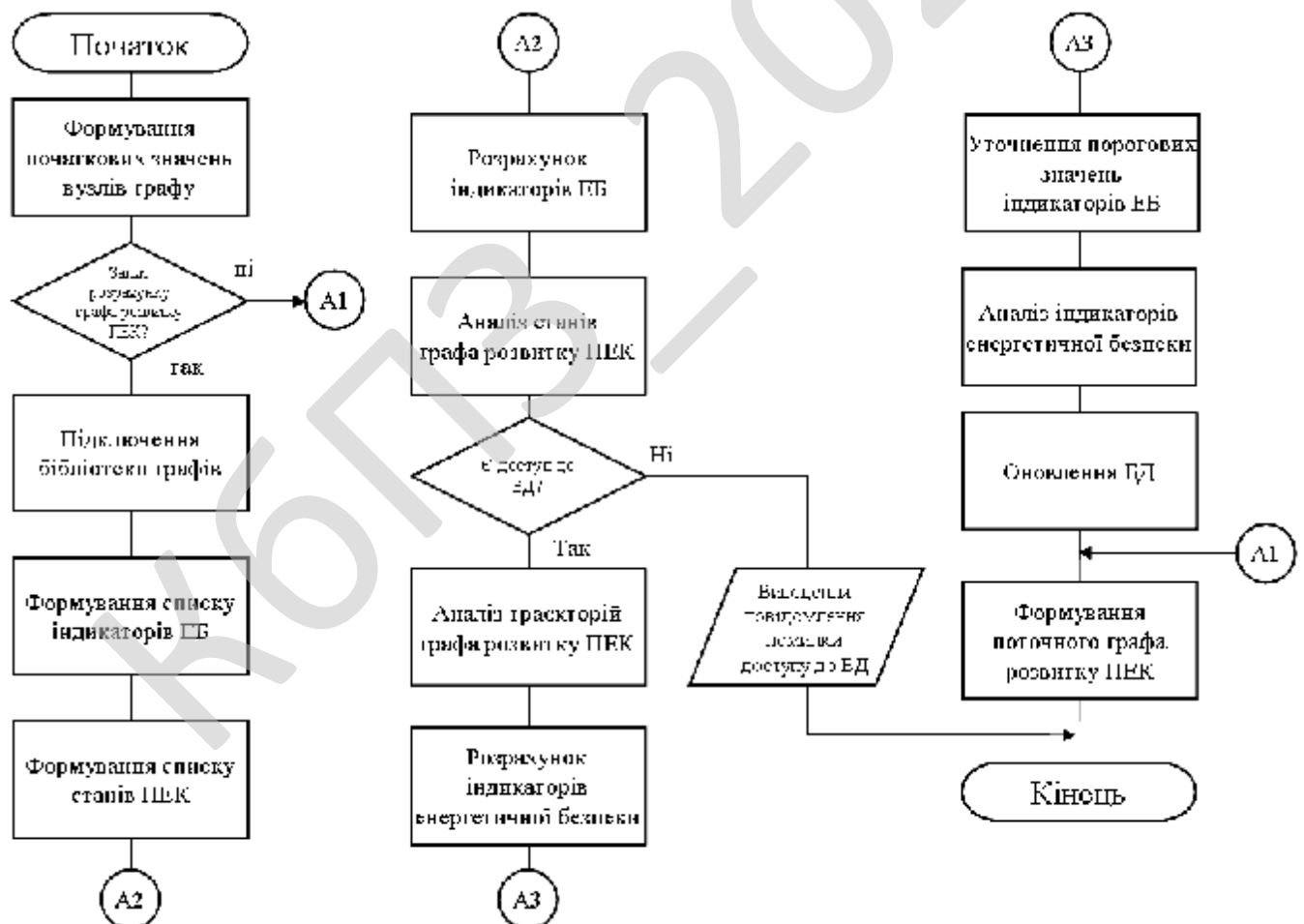


Рисунок 4.2 – Блок-схема підпрограми відображення графа розвитку ПЕК

- Отримання даних з блоку кластерного аналізу.
- Формування графічного представлення розвитку ПЕК.
- Підпрограма формування звіту.
- Системне повідомлення WM_CLOSE?

На рисунку 4.2 зображено роботи підпрограми відображення графа розвитку ПЕК. З нього видно:

- Формування початкових значень вузлів графу.
- Запит розрахунку графа розвитку ПЕК?

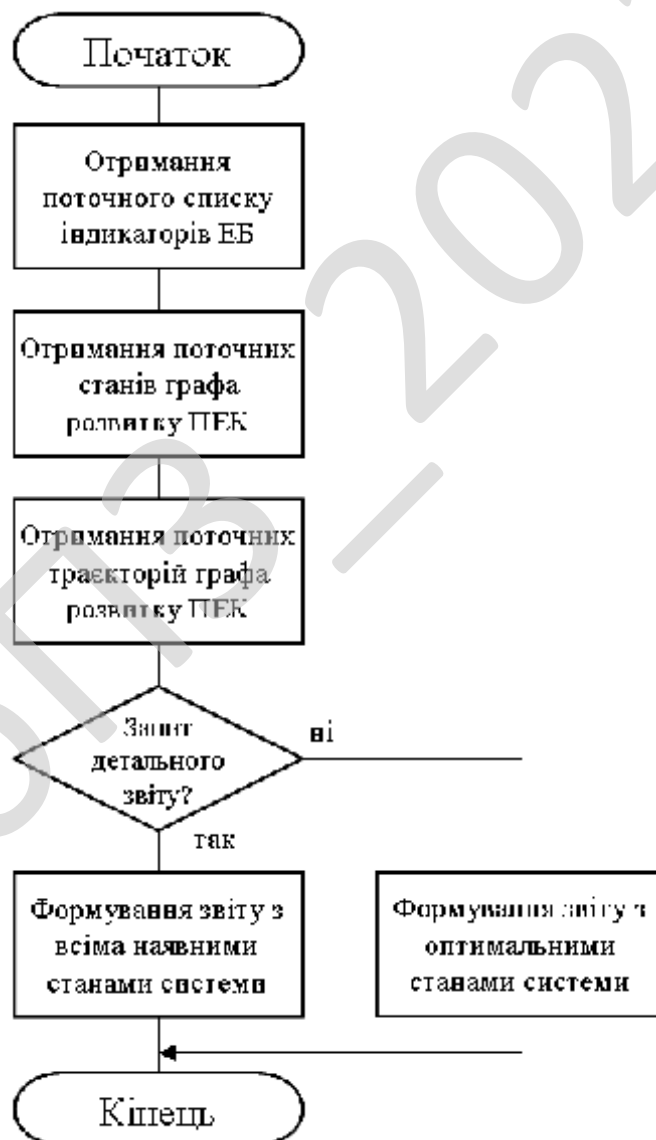


Рисунок 4.3 – Блок-схема підпрограми формування звіту

- Підключення бібліотеки графів.
- Формування списку індикаторів ЕБ.
- Формування списку станів ПЕК .
- Розрахунок індикаторів ЕБ.
- Аналіз станів графа розвитку ПЕК.
- Є доступ до БД (запит).
- Аналіз траєкторій графа розвитку ПЕК.
- Розрахунок індикаторів енергетичної безпеки.
- Уточнення порогових значень індикаторів ЕБ.
- Аналіз індикаторів енергетичної безпеки.
- Оновлення БД.
- Формування поточного графа розвитку ПЕК.

На рисунку 4.3 зображено роботи підпрограми формування звіту. З нього видно:

- Підпрограма формування звіту.
- Отримання поточного списку індикаторів ЕБ.
- Отримання поточних станів графа розвитку ПЕК .
- Отримання поточних траєкторій графа розвитку ПЕК.
- Запит детального звіту?
- Формування звіту з всіма наявними станами системи.
- Формування звіту з оптимальними станами системи.

Опис алгоритмів функціонування системи

Розглянемо як було реалізовано збереження положення вікна магістерської програми. Для цього необхідно зберегти два параметри – значення властивостей форми Left і Top, ця відстань від лівого й верхнього країв екрана відповідно.

Значення ці виражаються в пікселях і мають тип Integer. Необхідно перевести ці числа в рядок (тип String) за допомогою оператора InttoStr.

Вихідний код оброблювача OnClose наступний.

```
begin
    Listbox1.Items.Clear;
```

```

Listbox1.Items.Add(Inttostr(Form1.Left));
Listbox1.Items.Add(Inttostr(Form1.Top));
Listbox1.Items.Savetofile('MR.txt');
end ;

```

Вихідний код оброблювача події OnCreate наступний.

```

begin
  if Fileexists('My.txt') then
    begin
      Listbox1.Items.Loadfromfile('MR.txt');
      Form1.Left:=Strtoint(Listbox1.Items[0]);
      Form1.Top:=Strtoint(Listbox1.Items[1]);
    end ;
  end ;

```

Ця подія відбувається в момент створення форми операційною системою. У цей момент і потрібно привласнювати їй необхідні властивості.

У першому рядку відбувається перевірка на існування файлу, адже якщо його не буде, відбудеться помилка. Втім, програма після видачі попередження відкриється в тому місці, де була на етапі проектування, а при закритті потрібний файл буде відтворений.

Потім у логічних дужках begin / end утримується сам код, який буде виконаний тільки при наявності файлу MR.txt у папці із програмою, тому що використовується відносний шлях. Можна вказати конкретне місце розташування, наприклад, C:\MR\MR.txt.

Перевірку на існування файлу можна виконати також за допомогою контролю виняткових ситуацій. Якщо файл не існує, то відбудеться виняткова ситуація. Перехопивши її за допомогою спеціального оператора, ми зможемо уникнути помилок у програмі.

Для визначення режиму роботи розробленої в магістерської програми потрібно визначити ОС. Розглянемо розроблену функцію, яка визначає тип системи.

```

function GETOS(var Value: Boolean):Boolean;
//повертає результат Boolean,
// True - функція виконана,
// False - ні

```

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

var Ver: Tosversioninfo;
    BBB: Boolean;
begin
    Ver.dwosversioninfosize := Sizeof(Tosversioninfo);
    // розмір структури
    BBB := Getversionex(Ver);
    if not BBB then
    begin
        Result := False;
        Exit;
    end else
        Result := True;
    case Ver.dwplatformid of
VER_PLATFORM_WIN32_XP: Value := True;
    // Windows XP - підходить
        VER_PLATFORM_WIN32_WINDOWS : Value := False;
    //Windows 9x, Me чи 2000 - не підходить
        VER_PLATFORM_WIN32s : Result := False;
    //Windows 3.x - не підходить
    end;
end;

```

Якщо дана функція повернула результат True, виходить, визначення версії системи пройшло успішно, і тип системи буде зберігатися в змінній Value, а якщо ні, то визначення типу системи, у цьому випадку прийде завершити виконання програми.

Дана функція буде практично найголовнішою, тому що вона буде вказувати магістерській програмі яку частину коду виконувати.

Визначаємо, тип системи, завантажуюємо необхідну бібліотеку, одержуємо адреси функцій і виконуємо функцію.

```

procedure GetShares();
var
    i:Integer;
    Flibhandle : Thandle;
    Sharent : Pshareinfo2Array;
    // <= Змінні
    entriesread,totalentries:DWORD;
    // Windows XP
    Share : array [0..512] of Tshareinfo50;
    // масив типу Tshareinfo50

```

```

    PcentTriesRead, pcttotalavail:Word;
    OS: Boolean;
begin
    lbxshares.Items.Clear;
    if not OS(OS) then Close;
//Визначаємо тип системи
    if OS then begin
// Код для XP
        Flibhandle := Loadlibrary('MAG_1.DLL');
// Завантажуємо бібліотеку MAG_1
        if Flibhandle = 0 then Exit;
//Зв'язуємо функцію
        @Netmagisterenumnt := GetProcAddress(Flibhandle,'Enum');
        if not Assigned(Netmagisterenumnt) then
//Перевірка
            begin
                FreeLibrary(Flibhandle);
                Exit;
            end;
//Очищаємо покажчик на масив структур
        Sharent := nil;
//Виклик функції
        if Netmagisterenumnt(nil,2,@Sharent,DWORD(-1),
            @entriesread,@totalentries,nil) <> 0 then
            Begin
//Якщо виклик невдалий вивантажуємо бібліотеку
                FreeLibrary(Flibhandle);
                Exit;
            end;
//Обробка результатів
            if entriesread > 0 then
                for i:= 0 to entriesread - 1 do
lbxshares.Items.Add(String(Sharent[i].shi2_netname));
            end else begin
                Flibhandle := Loadlibrary(MAG_2.DLL');
//Завантажуємо бібліотеку MAG_2
                if Flibhandle = 0 then Exit;
//Зв'язуємо функцію
                @Netmagisterenum := GetProcAddress(Flibhandle,'Netmagisterenum');
                if not Assigned(Netmagisterenum) then
// Перевірка
                    begin

```

```

        Freelibrary(Flibhandle);
        Exit;
    end;
    if Netmagisterenum(nil,50,@Share,Sizeof(Share),
        @pcentriesread,@pctotalavail) <> 0 then
// Виклик функції
        Begin
// Якщо виклик невдалий вивантажуємо бібліотеку
        Freelibrary(Flibhandle);
        Exit;
        end;
// Обробка результатів
        if pcentriesread > 0 then
            for i:=0 to pcentriesread-1 do
                lbxshares.Items.Add(String(Share[i].shi50_netname));
            end;
// вивантажити бібліотеку
        Freelibrary(Flibhandle);
    end;

```

При виконанні цього коду Listbox заповниться назвами загальних ресурсів, які беруться з масиву структур. Масив заповнюється в результаті виконання функції.

Звичайно в XP використовуються функції Netaribufferallocate і Netaribufferfree – за допомогою їх виділяється й звільняється пам'ять під масив структур.

Компоненти Delphi 10, вміють працювати з файлами. Вони читають і зберігають свій зміст, рядки типу String, у файл текстового формату. Це компоненти Listbox, Combobox і Memo, розташовані на першій же вкладці палітри компонентів.

Кожний рядок компонентів Listbox і Combobox є об'єктом Items[i], а Memo – Lines[i], де i – номер рядка, який відлічується від нуля. Додавання рядків у компоненти виконується методами Add і Insert:

```

begin
    Memo1.Lines.Add('Перший рядок');
    Combobox1.Items.Add('Перший рядок');
    Combobox1.Items.Add('Другий рядок');

```

```
Listbox1.Items.Add('Перший рядок');
Listbox1.Items.Add('Другий рядок');
end ;
```

Метод Add додає новий рядок у кінець. Метод Insert має додатковий параметр, що вказує, після якого рядка розмістити новий рядок. Доступ до рядків здійснюється так:

```
Combobox1.Items[0] := 'Перший рядок змінився' ;
Listbox1.Items[1] := 'Другий рядок змінився' ;
```

У компонента Combobox додатково є властивість Text, де (як і в компонента Edit) знаходиться текст, що вводиться:

```
Combobox1.Text := ' текст, Що Вводиться, ';
```

На виділену в цей момент рядок компонента Combobox вказує властивість Itemindex типу Integer, тобто це номер виділеного рядка. Отже, одержати сам виділений рядок компонента Combobox можна наступною конструкцією.

```
S:=Combobox1.Items[Combobox1.Itemindex];
```

або, користуючись оператором приєднання

```
With Combobox1 do S:=Items[Itemindex];
```

При натисканні клавіші Enter можна заносити в цей компонент, що вводиться в рядок інформацію й видаляти натисканням Escape.

Необхідно виділити на формі компонент Combobox і у інспекторі об'єктів, на вкладці Events. Створити оброблювач Onkeypress. Система Delphi 10 створює заготовку оброблювача.

Розглянемо частину вихідного коду як це реалізувалось у магістерській роботі.

```
begin
if Key=#13 then
    Combobox1.Items.Add(Combobox1.Text);
if Key=#27 then
    Combobox1.Items.Delete(Combobox1.Items.Count-1);
end;
```

де:

Key – змінна, що містить код натиснутої клавіші, #13 і #27 – коди клавіш Enter і Escape відповідно.

Items.Count – кількість рядків, що втримуються в компоненті. Тому що відлік рядків іде від нуля, ми віднімаємо одиницю. Після чергового видалення кількість рядків змінюється, таким чином, Items.Count-1 завжди вказує на останній рядок.

Послідовно натискаючи Escape, ми можемо вилучити всі рядки.

```
ComboBox1.Items.Delete(0);
```

Вказаною командою можна добитися того ж ефекту, тільки вилучаться перші рядки. Втім, щоб стерти всі відразу, є метод Clear.

Для збереження вмісту у файл. Необхідно виконати наступну команду.

```
Listbox1.Items.SaveToFile('файл.txt ');
```

Втім, розширення можна поставити кожне за бажанням, не обов'язково .txt, як і взагалі без нього обійтися. Але розширення .txt дозволить легко відкрити файл стандартним ПЗ «Блокнот», що буває дуже зручно на етапі написання програми.

Для завантаження служить метод LoadFromFile.

```
Listbox1.Items.LoadFromFile('файл.txt ');
```

В деяких випадках у розробленій магістерській програмі не було можливості використовувати компоненти тоді проводилось його приховання. Для цього в інспекторі об'єктів ставимо у властивість Visible значення False. Функціональність його від цього не зміниться.

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 9041:2020. Його повна назва: ДСТУ 9041:2020. Інформаційні технології. Криптографічний захист інформації. Алгоритм шифрування коротких повідомлень, що ґрунтується на скручених еліптичних кривих Едвардса.

Цей алгоритм призначений для шифрування коротких (до 616 біт) повідомлень для будь-яких алгоритмів шифрування, в тому числі визначених національними стандартами України.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Як і стандарт цифрового підпису ДСТУ 4145:2002, новий алгоритм використовує криптографічні перетворення у групі точок еліптичних кривих, використовуючи замість кривих у формі Вейерштрасса найновітніші розробки у галузі еліптичної криптографії – криві у формі Едвардса. Це дає суттєві переваги у швидкодії більш ніж у 3 рази. Новий стандарт розроблений з урахуванням усіх найсучасніших вимог до стійкості криптографічних алгоритмів. Так, нижня межа стійкості криптосистем у цьому стандарті дорівнює 2127 (≈ 1042) (це більш ніж у півтора рази вище, ніж у ДСТУ 4145) і можуть бути обрані інші рівні, такі як 2255 (≈ 1085), 2383 (≈ 10127) та 2767 (≈ 10255); крім того, строго обґрунтована його стійкість як до атак на відновлення відкритого тексту, так і до розрізняючих атак.

Проект алгоритму шифрування, що ліг в основу цього стандарту, пройшов апробацію як в Україні, так і за її межами (Центрально-Європейська конференція з криптографії (червень 2020 року) – форум ведучих криптологів з усього світу).

Стандарт ДСТУ-9041 узгоджений з усіма діючими в Україні національними стандартами. Новиною стандарту є його сфера застосування – інкапсуляція ключів, найсучасніший математичний апарат, а також новий алгоритм генерації псевдовипадкових послідовностей, який, на відміну від аналогічного алгоритму генерації з ДСТУ 4145, використовує виключно національні криптографічні алгоритми національних стандартів та не містить посилань на відповідні пост-радянські стандарти, термін дії яких вже практично вичерпався.

Новий стандарт не належить до так званих постквантових стандартів. Але його стійкість буде під загрозою лише тоді, коли з'являться квантові комп'ютери з 700 і більше кубітами (на даний час кількість "робочих" кубітів, які вдалося створити, – близько 50). Його перевагою перед постквантовими алгоритмами є відносно невелика довжина ключа (у десятки або навіть у сотні разів менша, ніж у постквантових алгоритмах).

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

– Меню: Дані; Графи; Налаштування; Довідка.

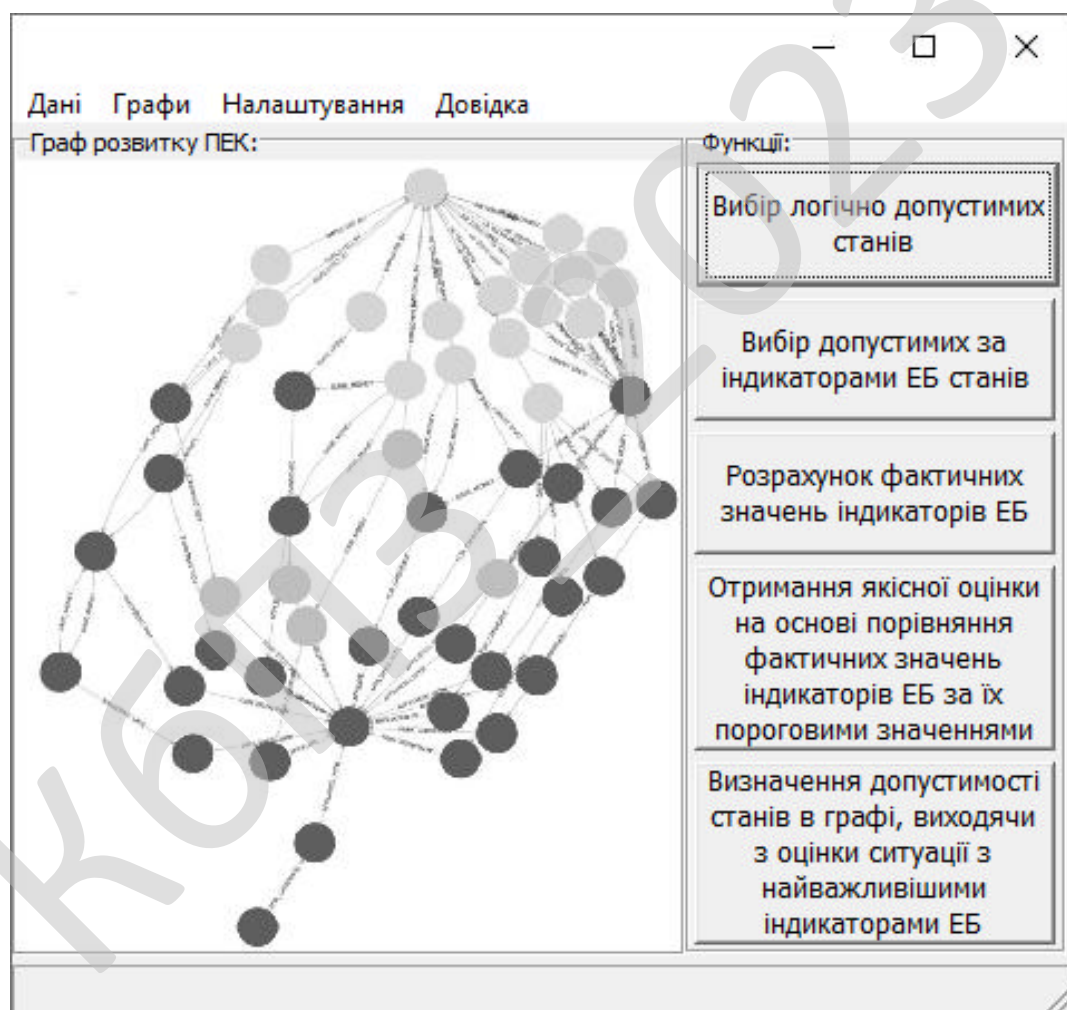


Рисунок 5.1 – Головне вікно програми

– Функції: Вибір логічно допустимих станів ТЕК; Вибір допустимих за індикаторами ЕБ станів ТЕК; Розрахунок фактичних значень індикаторів ЕБ;

Отримання якісної оцінки на основі порівняння фактичних значень індикаторів ЕБ за їх пороговими значеннями; Визначення допустимості станів в графі, виходячи з оцінки ситуації з найважливішими індикаторами ЕБ

На рисунку 5.2 зображено форму авторського права. Обрано тип розповсюдження – безплатне програмне забезпечення (Freeware). Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без вихідних кодів. Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

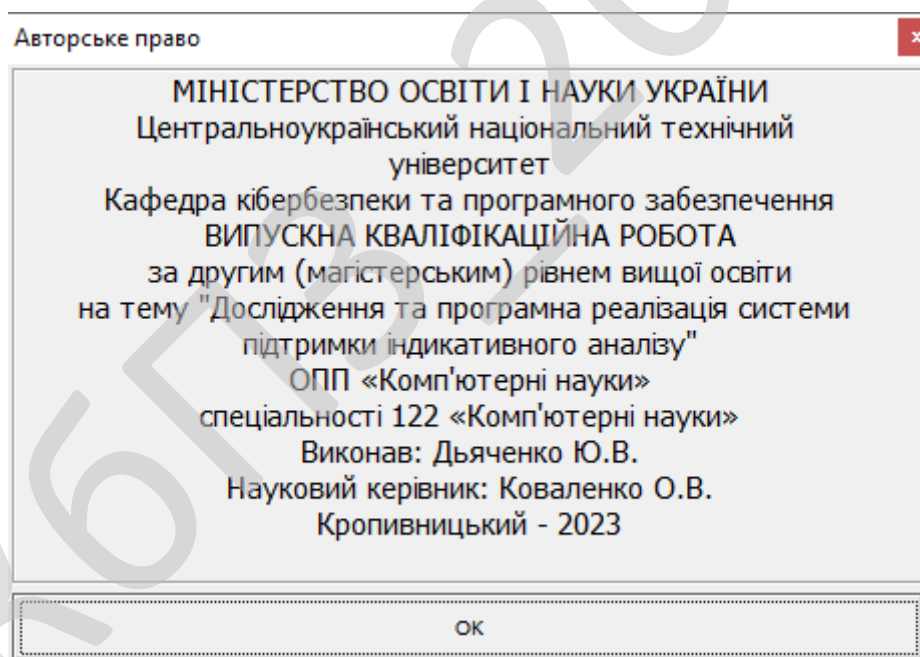


Рисунок 5.2 – Довідка розробника

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи підтримки індикативного аналізу.

Метою розробки є дослідження та програмна реалізація системи підтримки індикативного аналізу.

Об'єктом дослідження є процес підтримки індикативного аналізу.

Предметом дослідження є методи підтримки індикативного аналізу.

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 підтримки індикативного аналізу.
- Розроблено вітчизняний продукт підтримки індикативного аналізу, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи підтримки індикативного аналізу.

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	100
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

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

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	100000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

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

B – показник ступеня, що визначається співвідношенням:

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

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

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

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

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

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

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

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

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 38 = 64 \text{ люд/день.}$$

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	64	Ф 7.1-7.4
Впровадження	13	Д13
Всього	105	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{105 \cdot 1}{60 - 5} = 1,9 \text{ ставки.}$$

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

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

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

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	10	900	15
Монітор	60	10	600	10
Клавіатура	30	10	300	5
Маніпулятор «мишка»	30	10	300	5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	3	360	6
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	3	90	1,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	300	750	12,5
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	61

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{op}^c = \frac{61 \cdot 3}{1,2} = 152,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

Продовження таблиці 7.4

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

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	13301	39903
Продакт-менеджер	0,25	11000	8250
Інженер-програміст	1,9	12600	71820
Інженер - електронщик	0,3	11000	9900
Інженер-системотехнік	0,25	11000	8250
Адміністратор мережі	0,5	11000	16500
Системний програміст	0,25	11000	8250
Дизайнер WEB	0,25	11000	8250
Інженер-верстальник	0,25	11000	8250
Бухгалтер-економіст	0,5	11000	16500
Всього за період розробки	$R_{cn} = 5,45$	-	$\Phi_{роб} = 195873$

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.

$$z_{cd} = \frac{195873}{5,45 \cdot 60} = 599 \text{ грн.}$$

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:

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

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

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 29000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Комп'ютерторг за 28.10.23 – джерело <http://computorg.ua/ru/price.html>

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Витрати на транспорт, монтаж та випробування можуть бути 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	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	37857,75
Разом	$K_p = 2523008$		$A_p = 249636,25$

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

Згідно виданих норм приймаємо 1/2 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 206$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

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

$$З_{M1} = 206 \cdot 1/2 \cdot 3 = 309 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 50 примірників:

$$З_{M2} = \sum Ц_{\delta}, \quad (7.17)$$

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 24,3 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 24,3 грн./шт.

$$З_{M2} = 24,3 \cdot 50 = 1215 \text{ грн.}$$

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

$$З_{M3} = \sum Ц_{з}, \quad (7.18)$$

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

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

$$З_M = (309 + 1215 + 1702) / 100 = 32 \text{ грн.}$$

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, %.

$$O_n = 629 \cdot 15 \cdot 0,01 = 94 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 100$ прим.):

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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

$$A_m = 249636 \cdot 3 / (100 \cdot 12) = 624 \text{ грн.}$$

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	3_o	629
2. Додаткова зарплата виконавців	3_δ	63
3. Відрахування на соціальні потреби	C_{oc}	256
4. Загальногосподарські витрати	Γ_{ocn}	94
5. Витрати на матеріали	3_M	32
6. Освоєння нових операційних систем, мов програмування	O_n	94
7. Амортизація основних фондів	A_m	624
8. Повна собівартість програмного забезпечення	C_n	1792
9. Плановий прибуток	Π_p	896
10. Ціна підприємства $C_n = C_n + \Pi_p$	C_n	2688
11. Податок на додану вартість $\text{ПДВ} = 0.01 \cdot H_{ob} \cdot C_n$	ПДВ	537,6
12. Відпускна ціна програмної продукції $C = C_n + \text{ПДВ}$	C	3225,6

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

$$C_n = 3_o + 3_\delta + C_{oc} + \Gamma_{ocn} + 3_M + O_n + A_m. \quad (7.21)$$

$$C_n = 629 + 63 + 256 + 94 + 32 + 94 + 624 = 1792 \text{ грн.}$$

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 1792 = 896 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

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

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3226
Всього капітальних витрат	–	3226

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 технічне обслуговування)	Z_p	38650	24156
2. Витрати на електроенергію	$Z_{ел}$	137	86
3. Витрати на амортизацію	$Z_{ам}$	0	807
Всього витрат за рік	I	38787	25049

Витрати на профілактичні роботи:

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

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

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

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

$$Z_{p \text{ баз}} = 240 \cdot 120 \cdot 1,1 \cdot 1,22 = 38650 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 150 \cdot 120 \cdot 1,1 \cdot 1,22 = 24156 \text{ грн}.$$

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

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

$$Z_{ел \text{ баз}} = 0,15 \cdot 240 \cdot 3,8 = 137 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 0,15 \cdot 150 \cdot 3,8 = 86 \text{ грн}.$$

$$T_e = \frac{479208}{(2688-1792) \cdot 100 \cdot 12 / 3} = 1,3 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	100
2. Повна собівартість розробленої програми	Грн.	1792
3. Ціна розробленої програми	Грн.	2688
4. Плановий прибуток від реалізації розробленої програми	Грн.	896
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2523008
7. Загальний прибуток від реалізації програмної продукції	Грн.	89600
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	32170
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3226
11. Величина економічного ефекту у користувача програмної продукції	Грн.	12932
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,2

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_{δ} , I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

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

$$E_{cn} = (38787 - 25049) - 0,25 \cdot 3226 = 12932 \text{ грн.}$$

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

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

$$T_{cn} = \frac{3226}{38787 - 25049} = 0,2 \text{ року.}$$

7.9 Висновки

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

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

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

Охорона праці та здоров'я у сфері ІТ включає такі напрями, як:

– Ергономіка – це наука про адаптацію робочого середовища до фізичних і психологічних особливостей людини². Ергономіка вимагає врахування таких факторів, як розміри, форми, кольори, освітлення, шум, температура, вологість, вентиляція, постава, рухи, пози, втома, навантаження на очі та ін. Ергономіка допомагає покращити комфорт, продуктивність і задоволення працівників.

– Комп'ютерна безпека – це захист комп'ютерних систем і даних від несанкціонованого доступу, зміни, знищення або блокування. Комп'ютерна безпека вимагає використання антивірусних програм, фаєрволів, паролей, шифрування, резервного копіювання та інших технологій. Комп'ютерна безпека допомагає запобігти крадіжці, шпигунству, шантажу, саботажу та іншим загрозам.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

– Соціальна взаємодія – це процес спілкування між людьми у робочому колективі або через мережеві сервіси. Соціальна взаємодія вимагає дотримання правил етикету, поваги, толерантності, співробітництва та конструктивного діалогу. Соціальна взаємодія допомагає покращити настрій, мотивацію, комунікацію та творчий потенціал працівників.

Правила охорони праці і здоров'я для програмістів:

- Регулярно роби перерви в роботі. Вставай із-за столу і розминай м'язи.
- Налаштуй яскравість і контрастність монітору так, щоб не напружувати очі.
- Використовуй ергономічну мишку і клавіатуру, які зручно лягають у руку і не викликають болю.
- Слідкуй за своєю поставою. Сиди прямо і не нахиляйся до екрану.
- Захищай свій комп'ютер від вірусів, шпигунських програм і хакерів. Оновлюй антивірусне програмне забезпечення і не відкривай підозрілі файли і посилання.
- Не забувай про соціальну взаємодію. Спілкуйся з колегами, друзями і родиною. – Відвідуй заходи, які тебе цікавлять. Не ізолюй себе від світу.
- Люби свою професію, але не забувай про інші сфери життя. Розвивай свої захоплення, хоббі і таланти. Знаходь рівновагу між роботою і відпочинком.

Закон України “Про охорону праці” визначає основні принципи, завдання, права і обов'язки суб'єктів відносин з охорони праці, а також організаційні та правові основи державного управління і контролю за дотриманням законодавства про охорону праці.

Згідно з цим законом, ІТ компанії повинні впроваджувати такі заходи з охорони праці:

- Створювати на підприємстві службу охорони праці або призначати відповідальних осіб, які забезпечують розроблення, реалізацію та контроль за дотриманням заходів з охорони праці.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

- Забезпечувати безпечні і нешкідливі умови праці для працівників, використовуючи сучасні засоби техніки безпеки, санітарно-гігієнічні умови, засоби клектинго та індивідуального захисту, оптимальні режими праці та відпочинку.
- Проводити атестацію робочих місць на відповідність нормативно-правовим актам з охорони праці та аудит з охорони праці.
- Проводити навчання та інструктаж з питань охорони праці, з надання першої медичної допомоги потерпілим від нещасних випадків і правил поведінки у разі виникнення аварії⁵.
- Забезпечувати лікувально-профілактичне обслуговування працюючих, санітарно-побутове обслуговування, пільги і компенсації для працівників, які працюють у важких і шкідливих умовах.
- Нести відповідальність за порушення законодавства про охорону праці та зподіяння шкоди життю і здоров'ю працівників.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом [1]. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

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;
- монотонність праці;

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

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

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
Довжина	8,56
Ширина	5,43
Висота	2.9

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	6,64
Обсяг, V	м ³	не менше 20.0	19,25

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

У зазначеному приміщенні працюють 7 людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа приміщення у розрахунку на одного робочого місця програміста відповідає нормативним вимогам як 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», так і СанПіН 3.3.2 – 007 – 98.

А об'єм приміщення у розрахунку на одного робочого місця програміста відповідає 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», але дещо нижчий вимог СанПіН 3.3.2 – 007 – 98 (19,25 м³ замість 20 м³). Це цілком прийнятно, враховуючи хронологію та пріоритет опублікованні вищезгаданих нормативних актів.

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 врача України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. 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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

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 приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температур а, °С	Вологість,%	Швидкість повітря, м/с	Температур а, °С	Вологість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	55-65	0,1

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

В літню пору застовується кондиціонер.

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

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

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

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою

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

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;
- бажано, щоб робочий стіл при необхідності можна було регулювати по висоті в межах 680-780 мм, а висота над рівнем підлоги робочої поверхні, на якій працює програміст, повинна складати 720 мм.

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.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Обчислимо індекс приміщення за формулою:

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

де: S – площа приміщення, $S = 46,2 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 2,9 \text{ м}$;

A – ширина приміщення, $A = 5,42 \text{ м}$;

B – довжина приміщення, $B = 8,52 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекс приміщення:
 $i=1,14$.

Знаючи індекс приміщення, за знаходимо $n = 0,46$ (з таблиці коефіцієнтів використання світлового потоку (η) світильників з відповідним типом ламп [3]). Підставимо всі значення у формулу, визначимо світловий потік: $F=38743 \text{ Лм}$.

Будемо використовувати лампи TL-F 12 36W LED 6000K IP65, світловий потік яких $F_{\lambda} = 3000 \text{ Лм}$.

Число ламп визначається по формулі:

$$N=F/F_{\lambda}$$

де: F – світловий потік,

F_{λ} – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекса приміщення: $N=38743/3000=12,9$ шт.

Кожен світильник комплектується двома лампами. Тому необхідно використовувати 7 світильників з 14 лампами в них.

Висновки до розділу

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

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

Список використаних джерел інформації

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

2. Про охорону праці: Закон України від 14.10.1992 р. № 2694-ХІІ. *Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>*

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

КБГПЗ - 2023

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи підтримки індикативного аналізу.

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 підтримки індикативного аналізу.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем підтримки індикативного аналізу.
- Досліджена система підтримки індикативного аналізу.
- На основі отриманих результатів досліджень створена програмна реалізація системи підтримки індикативного аналізу.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання підтримки індикативного аналізу.

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

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

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 9041:2020.

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 12932 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,2 роки.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дьяченко Ю.В. Дослідження та програмна реалізація системи підтримки індикативного аналізу // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Aggarwal C.C. Neural Networks and Deep Learning: A Textbook 1st ed. 2018 Edition. – Springer, 2018. – 520 p
3. Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. – Pearson, 2001. – 620 с.
4. Brink H., Richards J., Fetherolf M. Real-World Machine Learning. – Manning, 2016. – 474 p.
5. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
6. Fenner M. Machine Learning with Python for Everyone (Addison-Wesley Data & Analytics Series) 1st Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 586 p.
7. Foreman J.W. Data Smart: Using Data Science to Transform Information into Insight 1st Edition. – Wiley, 2013. – 432 p.
8. Hurbans R. Grokking Artificial Intelligence Algorithms. – Manning, 2020. – 631 p.
9. Gusfield D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology 1st Edition. – Cambridge University Press, 2008. – 556 p.
10. Kotu V., Deshpande B. Data Science: Concepts and Practice. – Elsevier Science, 2018. – 953 p.
11. Knowledge Base A Complete Guide – 2021 Edition // The Art of Service – Knowledge Base Publishing, 2020. – 306 p.

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

12. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
13. Mattmann C. Machine Learning with TensorFlow, Second Edition. – Manning, 2020. – 1124 p.
14. Mueller J.P., Massaron L. Machine Learning For Dummies. – Wiley, 2016. – 714 p.
15. Teofili T. Deep Learning for Search. – Manning, 2019. – 695 p.
16. Rungta K. TensorFlow in 1 Day: Make your own Neural Network. – Publishdrive, 2019. – 587 p.
17. Weidman S. Deep Learning from Scratch: Building with Python from First Principles. – O’Reilly. 2019. – 252 p.
18. Rajasekaran S., Vijayalakshmi Pai G.A. Neural networks, fuzzy logic, and genetic algorithms: synthesis and applications (with cd-rom) Kindle Edition. – PHI, 2013. – 628 p.
19. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.
20. Smirnov, O., Neskrodieva, T., Fedorov, E., Rudakov, K., Neskrodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12.
21. 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.
22. 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., Vuppapapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile

Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34.

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

24. 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>.

25. 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.

26. 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.

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

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

29. 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.

					БКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

30. 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.

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

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

33. 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.

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

35. 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.

36. 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.

37. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation

Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

38. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

39. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

40. 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.

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

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

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

44. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95

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

46. 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.

47. 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.

48. Смірнов, С.А., Смірнова, Т.В., Минайленко, Р.М., Доренський, О.П., Сисоєнко С.В. «Хмарна автоматизована система інтелектуальної підтримки прийняття рішень для технологічних процесів». Вісник Черкаського державного технологічного університету. Технічні науки. №4, 2020, С. 84-92.

49. 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.

50. 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).

					ВКРМ-122.23.0035.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.23.0035.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Дьяченко Ю.В.				<i>Дослідження та програмна реалізація системи підтримки індикативного аналізу</i>	Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М-2			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи підтримки індикативного аналізу.

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. № 33-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи підтримки індикативного аналізу.

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-122.23.0035.00.00.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 підтримки індикативного аналізу;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.23.0035.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.4.

					ВКРМ-122.23.0035.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

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

					ВКРМ-122.23.0035.00.00.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 – 99 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2023 р.

					ВКРМ-122.23.0035.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Коваленко О.В.

*Дослідження та програмна реалізація
системи підтримки індикативного аналізу*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 65

Літера: РП

Кропивницький – 2023 року

ОСНОВНИЙ ФАЙЛ ПРОЕКТУ ПЗ - UKRAINE_EB.DPR

```
program Ukraine_EB;
{
Дослідження та програмна реалізація системи підтримки індикативного аналізу
Виконав: студент 2 курсу магістратури, Дьяченко Юрій Володимирович
Керівник: Коваленко О.В.
2023 рік
}
Uses

// бібліотеки
Forms,
windows,
Unit1 in 'Unit1.pas' {Form1},
Unit2 in 'Unit2.pas' {Form2},
Unit3 in 'Unit3.pas' {Form3},
frmInfoBase in 'InfoBase.pas' {Form4},
FILE_RAR in 'FILE_RAR.pas' {Form5},
DATA in 'DATA.pas' {Form6},
EB_DATABASE in 'EB_DATABASE.pas' {Form7},
About_DOC in 'AB.pas' {Form8};
{$R *.res}
// додаткові ресурси

begin
Application.Initialize;
// Початок ініціалізації ПЗ
Application.ShowMainForm := False;
// створення форм ПЗ з 1 з 8
Application.CreateForm(TForm1, Form1);
// створення форм ПЗ з 2 з 8
Application.CreateForm(TForm2, Form2);
// створення форм ПЗ з 3 з 8
Application.CreateForm(TForm3, Form3);
// створення форм ПЗ з 4 з 8
Application.CreateForm(TForm3, Form5);
// створення форм ПЗ з 5 з 8
Application.CreateForm(TForm3, Form6);
// створення форм ПЗ з 6 з 8
Application.CreateForm(TForm3, Form7);
// створення форм ПЗ з 7 з 8
Application.CreateForm(TForm3, Form8);
// створення форм ПЗ з 8 з 8
Application.Run;
// початок роботи ПЗ
end.
// Завершення роботи ПЗ
```

About.PAS – ФОРМА РОЗРОБНИКА

```
unit About;
// модуль About
{
Дослідження та програмна реалізація системи підтримки індикативного аналізу
Виконав: студент 2 курсу магістратури, Дьяченко Юрій Володимирович
Керівник: Коваленко О.В.
2023 рік
}

Interface
// об'ява даних

Uses
// бібліотеки
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, ShellAPI;

type
// власні типи
TA = class(TForm) // створення класу форми
    btOk: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
private
public
end;

Var
// об'ява локальних типів даних
fmAbout: TA;

implementation
// реалізація

{$R *.dfm}
// файл ресурсів

end.
```

GRAF_RAR.PAS – МОДУЛЬ СТИСКУ ГРАФУ

```

unit GRAF_RAR;
// модуль GRAF_RAR
{
Дослідження та програмна реалізація системи підтримки індикативного аналізу
Виконав: студент 2 курсу магістратури, Дьяченко Юрій Володимирович
Керівник: Коваленко О.В.
2023 рік
}
Interface
// об'ява даних

Uses
// бібліотеки
Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, RAR, ComCtrls, Menus, RAR_DLL, StdCtrls,
ExtCtrls, FileCtrl;

type
// власні типи
TMyGrafForm = class(TForm)
// створення класу форми
RARArchive: TRAR;
MainMenu: TMainMenu;
File1: TMenuItemLite;
N1: TMenuItemLite;
Close1: TMenuItemLite;
Open1: TMenuItemLite;
N2: TMenuItemLite;
ListView: TListView;
OpenDialog: TOpenDialog;
LogMemo: TMemo;
Archiveinformation1: TMenuItemLite;
N3: TMenuItemLite;
Archive1: TMenuItemLite;
test1: TMenuItemLite;
Extract1: TMenuItemLite;
About1: TMenuItemLite;
Dllversion1: TMenuItemLite;
Panell: TPanel;
OverallBar: TProgressBar;
SingleBar: TProgressBar;
fileLabel: TLabel;
CancelButton: TButton;
ExtractSelected1: TMenuItemLite;
N4: TMenuItemLite;
refresh1: TMenuItemLite;
procedure Close1Click(Sender: TObject);
procedure Open1Click(Sender: TObject);
procedure Archiveinformation1Click(Sender: TObject);
procedure test1Click(Sender: TObject);
procedure Extract1Click(Sender: TObject);
procedure About1Click(Sender: TObject);
procedure Dllversion1Click(Sender: TObject);
procedure CancelButtonClick(Sender: TObject);
procedure ExtractSelected1Click(Sender: TObject);
procedure RARArchiveError(Sender: TObject; const ErrorCode: Integer;
const Operation: TRAROperation);
procedure RARArchiveListFile(Sender: TObject;
const FileInformation: TRARFileItem);
procedure RARArchiveNextVolumeRequired(Sender: TObject;
const requiredFileName: string; out newFileName: string;
out Cancel: Boolean);
procedure RARArchivePasswordRequired(Sender: TObject;

```

```

    const HeaderPassword: Boolean; const FileName: string;
    out NewPassword: string; out Cancel: Boolean);
procedure RARArchiveProgress(Sender: TObject; const FileName: WideString;
    const ArchiveBytesTotal, ArchiveBytesDone, FileBytesTotal,
    FileBytesDone: Cardinal);
procedure RARArchiveReplace(Sender: TObject; const ExistingData,
    NewData: TRARReplaceData; out Action: TRARReplace);
procedure FormCreate(Sender: TObject);
private
public
end;
Var
// об'ява локальних типів даних
    MainForm: TMyGrafForm;

implementation
// реалізація

Uses
    Replace;
// бібліотеки

{$R *.dfm}
// файл ресурсів
procedure TMyGrafForm.Extract1Click(Sender: TObject);
Var
// об'ява локальних типів даних
    Path:String;
begin
    if SelectDirectory('', 'C:\', Path) then
        if RARArchive.Extract(path,True,nil) then showmessage('done')
        else
            showmessage('failed');
end;

procedure TMyGrafForm.ExtractSelected1Click(Sender: TObject);
Var
// об'ява локальних типів даних
    Path:String;
    i:integer;
    files:TStrings;
begin
    files:=TStringList.Create;
    for i := 0 to ListView.Items.Count - 1 do
        if ListView.Items.Item[i].Selected then
            files.add(ListView.Items.Item[i].Caption);
    if SelectDirectory('Select a directory', 'C:\', Path) then
        if RARArchive.Extract(path,True,files) then showmessage('done')
        else
            showmessage('failed');
    files.free;
end;

procedure TMyGrafForm.FormCreate(Sender: TObject);
begin
    DoubleBuffered:=True;
    OverallBar.DoubleBuffered:=True;
    SingleBar.DoubleBuffered:=True;
end;
procedure TMyGrafForm.Open1Click(Sender: TObject);
begin
    if OpenFileDialog.Execute then begin
        ListView.Items.BeginUpdate;
        ListView.Items.Clear;
        if RARArchive.OpenFile(OpenDialog.FileName) then

```

```

        Archiveinformation1.Click
    else
        showmessage('Неможливо відчинити файл!');
        ListView.Items.EndUpdate;
    end;
end;

procedure TMyGrafForm.RARArchiveError(Sender: TObject; const ErrorCode: Integer;
const Operation: TRAROperation);
Var
// об'ява локальних типів даних
op:String;
begin
    case operation of
        roInitArchive: op:='init arc';
        roListFiles: op:='list arc';
        roExtract: op:='extract arc';
        roTest: op:='test arc';
    end;
    LogMemo.lines.add(op+' : '+inttostr(ErrorCode));
end;

procedure TMyGrafForm.RARArchiveListFile(Sender: TObject;
const FileInformation: TRARFileItem);
Var
// об'ява локальних типів даних
Attrib:String;
begin
    with listView.Items.Add do begin
        Caption:=FileInformation.FileNameW;
        SubItems.Add(getSizeName(FileInformation.CompressedSize));
        SubItems.Add(getSizeName(FileInformation.UncompressedSize));
        Attrib:='';
        case fileinformation.Attributes of
            {$WARN SYMBOL_PLATFORM OFF}
// компіляторні налаштування
            faReadOnly: Attrib:=Attrib+'R';
            faHidden: Attrib:=Attrib+'H';
            faSysFile: Attrib:=Attrib+'S';
            faVolumeID: Attrib:=Attrib+'V';
            faDirectory: Attrib:=Attrib+'D';
            faArchive: Attrib:=Attrib+'A';
            faAnyFile: Attrib:=Attrib+'F';
            {$WARN SYMBOL_PLATFORM ON}
// компіляторні налаштування
        end;
        subitems.add(Attrib);
        SubItems.Add(FileInformation.HostOS);
        SubItems.Add(FileInformation.Comment);
        subitems.Add(formatdatetime('c',FileInformation.Time));
        case FileInformation.CompressionStrength of
            48: subitems.Add('stored');
            49: subitems.Add('fastest');
            50: subitems.Add('fast');
            51: subitems.Add('normal');
            52: subitems.Add('good');
            53: subitems.Add('best');
        end;
        subitems.add(inttostr(FileInformation.ArchiverVersion div
10)+'.'+(inttostr(FileInformation.ArchiverVersion mod 10)));
        subitems.add(BoolToStr(fileinformation.Encrypted));
        SubItems.Add(FileInformation.CRC32);
    end;
end;
end;

```

```

procedure TMyGrafForm.RARArchiveNextVolumeRequired(Sender: TObject;
  const requiredFileName: string; out newFileName: string; out Cancel: Boolean);
begin
  newFileName:=requiredFileName;
  Cancel:=not InputQuery('next volume',requiredFileName,newFileName);
end;

procedure TMyGrafForm.RARArchivePasswordRequired(Sender: TObject;
  const HeaderPassword: Boolean; const FileName: string;
  out NewPassword: string; out Cancel: Boolean);
begin
  Cancel:=not InputQuery('Пароль?'+extractFileName
  (FileName),extractFileName(FileName),NewPassword);
end;

procedure TMyGrafForm.RARArchiveProgress(Sender: TObject; const FileName:
WideString;
  const ArchiveBytesTotal, ArchiveBytesDone, FileBytesTotal,
  FileBytesDone: Cardinal);
begin
  fileLabel.Caption:='archive: '+getSizeName(ArchiveBytesDone)+' of
'+getSizeName(ArchiveBytesTotal)+' , file: '+getSizeName(FileBytesDone)+' of
'+getSizeName(FileBytesTotal)+' '+FileName;
  if FileBytesTotal>0 then
    SingleBar.Position:=round(100 / FileBytesTotal*FileBytesDone)
  else
    SingleBar.Position:=0;
  if ArchiveBytesTotal>0 then
OverallBar.Position:=round(100/ArchiveBytesTotal*ArchiveBytesDone)
  else
    OverallBar.Position:=0;
  fileLabel.Repaint;
  SingleBar.Repaint;
  OverallBar.Repaint;
  Application.ProcessMessages;
end;

procedure TMyGrafForm.RARArchiveReplace(Sender: TObject; const ExistingData,
  NewData: TRARReplaceData; out Action: TRARReplace);
Var
  // об'ява локальних типів даних
  ReplaceForm:TReplaceForm;
begin
  ReplaceForm:=TReplaceForm.Create(Self);
  ReplaceForm.existent.Caption:='existent: '+ExistingData.FileName;
  ReplaceForm.archive.Caption:='in archive: '+NewData.FileName;
  ReplaceForm.ShowModal;
  Action:=ReplaceForm.Action;
  ReplaceForm.Free;
end;

procedure TMyGrafForm.test1Click(Sender: TObject);
begin
  if RARArchive.Test then
    showmessage('Помилка не знайдено!')
  else
    showmessage('Архів ушкоджен!');
end;

function GetSizeName(const Size : int64): String;
begin
  Result := 'error';
  if Size < 0 then exit;
  if Size < 1024 then

```

```
begin
  Result := inttostr(Size)+' Byte';
  exit;
end;
if (1024 <= Size) and (Size < 1048576) then
begin
  Result := floattostr((round((Size/1024)*100))/100)+' KB';
  exit;
end;
if (1048576 <= Size) and (Size < 1073741824) then
begin
  Result := floattostr((round((Size/1048576)*100))/100)+' MB';
  exit;
end;
if Size = 1073741824 then
  Result:=('1 GB');
if Size > 1073741824 then
begin
  Result := floattostr((round((Size/1073741824)*100))/100)+' GB';
end;
end;

function BoolToStr(value:boolean):String;
begin
  if Value then
    Result:='True'
  else
    Result:='False';
end;

procedure TMyGrafForm.CancelButtonClick(Sender: TObject);
begin
  RARArchive.Abort;
end;

procedure TMyGrafForm.Close1Click(Sender: TObject);
begin
  Close;
end;

procedure TMyGrafForm.Dllversion1Click(Sender: TObject);
begin
  ShowMessage(inttostr(RARArchive.getDllVersion));
end;

end.
```

EB_DATABASE.PAS - МОДУЛЬ МЕРЕЖНОЇ ВЗАЄМОДІЇ З БД

```

unit EB_DATABASE;
// модуль EB_DATABASE
{
Дослідження та програмна реалізація системи підтримки індикативного аналізу
Виконав: студент 2 курсу магістратури, Дьяченко Юрій Володимирович
Керівник: Коваленко О.В.
2023 рік
}
Interface
// об'ява даних

Uses
// бібліотеки
  Windows,
  WinSock,
  Snmp,
  SysUtils,
  Classes;

Const
// константа
  mibLen = $0A;

type
// власні типи
  TMibId = array[1..mibLen] of integer;
{ Об'ява класу TIpConnInfo }
  TIpConnInfo = class
  private
    FRemoteIp :TInAddr;
    FRemotePort :integer;
    FState :DWORD;
    FLocalIp :TInAddr;
    FLocalPort :integer;
    FProto :ShortString;
    function GetLocalPort :Integer;
    function GetLocalIpString :String;
    function Ip2Str( const Ip :TInAddr ):String;
    function GetRemotePort :Integer;
    function GetRemoteIpString :String;
    function GetStateString :String;
  public
    constructor Create( const aProto :String );
    property IpProtoName :ShortString
      read FProto;
    property LocalIp :TInAddr
      read FLocalIp;
    property LocalPort :Integer
      read GetLocalPort;
    property LocalIpString :String
      read GetLocalIpString;
    property RemoteIpString :String
      read GetRemoteIpString;
    property RemoteIp :TInAddr
      read FRemoteIp;
    property RemotePort :Integer
      read GetRemotePort;
    property State :DWORD
      read FState;
    property StateString :String
      read GetStateString;
  end;

{ Об'ява класу }

```

```

EConnListError = class(Exception);
    EConnListLockError = class(EConnListError);
    EConnListUnlockError = class(EConnListError);
    EConnListRefreshError = class(EConnListError);
TIpConnListStatus = ( connlist_ready, connlist_refreshing );
TIpProtocol = (udp_ip, tcp_ip);
TIpProtocols = set of TIpProtocol;

TF = class(TComponent) { Об'ява класу }
private
    FConnections :TList;
    FWsInited :Bool;
    FSnmpLib :THandle;
    FSnmpQueryProc :Pointer;
    FSnmpInitProc :Pointer;
    FProtocols :TIpProtocols;
    FStatus :TIpConnListStatus;
    FOnStatusChange :TNotifyEvent;
    FAccessMutex :THandle;
    FPollForTrapEvent :THandle;
    FSupportedViewRoot :TAsnObjectIdentifier;
    function GetConnCount :Integer;
    function GetConnections( Index :Integer ):TIpConnInfo;
protected
    procedure StatusChange; virtual;
    procedure SetStatus( Value :TIpConnListStatus ); virtual;
    procedure DoLock;
    procedure DoUnlock;
    procedure Clear;
public
    constructor Create( aOwner :TComponent ); override;
    destructor Destroy; override;
    procedure Refresh;
    function Lock( Timeout :DWORD ):Bool;
    function Unlock :Bool;
    property Status :TIpConnListStatus
        read FStatus;
    property Connections[ Index :Integer ] :TIpConnInfo
        read GetConnections; default;
    property ConnCount :Integer
        read GetConnCount;
published
    property Protocols :TIpProtocols
        read FProtocols write FProtocols;
    property OnStatusChange :TNotifyEvent
        read FOnStatusChange write FOnStatusChange;
end;
ENetstatError = class(Exception);
{ Об'ява класу }
TNetstatCounterObject = class
{ Об'ява класу }
private
    FName :String;
    FId :TMibId;
    FDescription :String;
    FWsInited :Bool;
    FSnmpLib :THandle;
    FSnmpQueryProc :Pointer;
    FSnmpInitProc :Pointer;
    function GetValue :DWORD;
public
// створення об'єкту класу
constructor Create(const aName,aDesc:String; const aId :TMibId);
    destructor Destroy; override;

```

```

    property Description :String
        read FDescription;
    property Name :String
        read FName;
    property Id :TMibId
        read FId;
    property Value :DWORD
        read GetValue;
end;

TCounterList = class(TComponent) { Об'ява класу }
private
    FCounters :TStringList;
    function GetCount :Integer;
    function GetCounters(Index:Integer):TNetstatCounterObject;
public
    constructor Create( aOwner :TComponent ); override;
    destructor Destroy; override;
    procedure Clear;
    procedure AddCounter( aCounter :TNetstatCounterObject );
    function IndexOf( const aName :String ):Integer;
    property Count :Integer
        read GetCount;
    property Counters[Index :Integer] :TNetstatCounterObject
        read GetCounters; default;
end;

TIpStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;

TICmpStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;

TTCPStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;

TUdpStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;
procedure Register;

implementation
// реалізація

const
// константи
DEFAULT_LOCK_TIMEOUT = 100;
SNMP_LIB_NAME = 'inetmib1.dll';
SNMP_INITPROC_NAME = 'SnmplibInit';
SNMP_QUERYPROC_NAME = 'SnmplibQuery';

type
// власні типи
TSnmpInitProc = function(
    dwTimeZeroReference : DWORD;
    Var hPollForTrapEvent : THandle;
    Var SupportedView : TAsnObjectIdentifier) : BOOL; stdcall;

```

```

TSnmpQueryProc = function(
    RequestType : Byte;
    Var VariableVindings : TRFC1157VarBindList;
    Var ErrorStatus : TAsnInteger;
    Var ErrorIndex : TAsnInteger) : BOOL; stdcall;

{ TIpStats }

// створення об'єкту класу
constructor TIpStats.Create( aOwner :TComponent );
begin
    inherited;
    AddCounter( TNetstatCounterObject.Create( 'ipDefaultTTL', '',
mib_ipDefaultTTL));
    AddCounter( TNetstatCounterObject.Create( 'ipInReceives', '',
mib_ipInReceives));
    AddCounter( TNetstatCounterObject.Create( 'ipInHdrErrors', '',
mib_ipInHdrErrors));
end;

function TF.Lock( Timeout :DWORD ):Bool;
begin
    result := WaitForSingleObject(FAccessMutex, Timeout ) = WAIT_OBJECT_0;
end;

function TF.Unlock :Bool;
begin
    result := ReleaseMutex(FAccessMutex);
end;

procedure TF.Refresh;
    procedure ReadTcpTable;
        Var
// об'єва локальних типів даних
        varBind      :TRFC1157VarBind;
        varBindList  :TRFC1157VarBindList;
        errorStatus  :TAsnInteger;
        errorIndex   :TAsnInteger;
        ConnIndex    :Integer;
        Info         :TIpConnInfo;
        ListTail     :Integer;
    begin
        fillchar( varBindList, SizeOf(varBindList), 0);
        varBindList.List := @varBind;
        varBindList.len := 1;
        fillchar( varBind, SizeOf(varBind), 0);
        varBind.Name.idLength := mibLen;
        varBind.name.ids := @mib_tcpConnTable;
        ListTail := FConnections.Count;
        if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
            varBindList, errorStatus, errorIndex) then
            raise EConnListRefreshError.Create(m_err_query);
        if varBindList.list.value.asnType = ASN_NULL then
            exit;
        while (varBindList.list.value.asnType = ASN_INTEGER) do
            begin
                Info := TIpConnInfo.Create('TCP');
                Info.FState := varBindList.list.value.Counter;
                FConnections.Add(Info);
                if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
                    varBindList, errorStatus, errorIndex) then
                    raise EConnListRefreshError.Create(m_err_query);
            end;
        if varBindList.list.value.asnType = ASN_NULL then
            raise EConnListRefreshError.Create(m_err_query);
    end;
end;

```

```

ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
begin
    move(varBindList.list.value.address.stream^,
        Connections[ConnIndex].FLocalIP, sizeof(TInAddr));
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
    Connections[ConnIndex].FLocalPort := varBindList.list.value.counter;
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання віддаленого порту ips }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
begin
    move(varBindList.list.value.address.stream^,
        Connections[ConnIndex].FRemoteIP,
        sizeof(TInAddr));
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання віддаленого порту }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
Connections[ConnIndex].FRemotePort := varBindList.list.value.counter;
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
end;

procedure ReadUdpTable;
Var
// об'ява локальних типів даних
varBind      :TRFC1157VarBind;
varBindList  :TRFC1157VarBindList;
errorStatus  :TAsnInteger;
errorIndex   :TAsnInteger;
ConnIndex    :Integer;
Info         :TIpConnInfo;
ListTail     :Integer;
begin
fillchar( varBindList, SizeOf(varBindList), 0);
varBindList.List := @varBind;
varBindList.len := 1;
fillchar(varBind, SizeOf(varBind), 0);

```

```

varBind.Name.idLength := mibLen;
varBind.name.ids := @mib_udpTable;
ListTail := FConnections.Count;
if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
    varBindList, errorStatus, errorIndex) then
    raise EConnListRefreshError.Create(m_err_query);
if varBindList.list.value.asnType = ASN_NULL then
    exit;
{ читання }
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
begin
    Info := TIpConnInfo.Create('UDP');
    move(varBindList.list.value.address.stream^,
        Info.FLocalIP,
        sizeof(TInAddr));
    FConnections.Add(Info);
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
    Connections[ConnIndex].FLocalPort := varBindList.list.value.counter;
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
end;
begin
    DoLock;
    try
        if FStatus <> connlist_ready then
            raise EConnListError.Create(m_err_inv_state);
        SetStatus( connlist_refreshing );
        Clear;
        if tcp_ip in FProtocols then
            ReadTcpTable;
        if udp_ip in FProtocols then
            ReadUdpTable;
    finally
        DoUnlock;
        SetStatus(connlist_ready);
    end;
end;

{ TlcmpStats }
{ TNetstatCounterObject }
function TNetstatCounterObject.GetValue :DWORD;
Var
// об'ява локальних типів даних
varBind      :TRFC1157VarBind;
varBindList  :TRFC1157VarBindList;
errorStatus  :TAsnInteger;
errorIndex   :TAsnInteger;
begin
varBindList.List := @varBind;
varBindList.len := 1;
fillchar(varBind, SizeOf(varBind), 0);
varBind.Name.idLength := mibLen;
varBind.Name.ids := @FId;

```

```

if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
    varBindList, errorStatus, errorIndex) then
    raise ENetstatError.Create(m_err_query);
if not (varBindList.list.value.asnType in
[ASN_GAUGE32, ASN_INTEGER, ASN_INTEGER32, ASN_COUNTER32, ASN_UNSIGNED32]) then
    raise ENetstatError.Create(m_err_invtype);
result := varBindList.list.value.Counter;
end;

// створення об'єкту класу
constructor TNetstatCounterObject.Create(const aName,
    aDesc:String; const aId :TMibId );

var
    WSData :TWSAData;
    PollForTrapEvent :THandle;
    SupportedViewRoot :TAsnObjectIdentifier;
begin
    inherited Create;
    FWSInited := WsaStartup($0101, WSData ) = 0;
    if not FWSInited then
        raise EConnListError.Create(m_err_wsstartup);
    FSnmpLib := LoadLibrary(SNMP_LIB_NAME);
    if FSnmpLib = 0 then
        raise EConnListError.Create(m_err_loadlib);
    FSnmpInitProc := GetProcAddress(FSnmpLib, SNMP_INITPROC_NAME);
    FSnmpQueryProc := GetProcAddress(FSnmpLib, SNMP_QUERYPROC_NAME);
    if (( FSnmpQueryProc = Nil ) or ( FSnmpInitProc = Nil ) ) then
        raise EConnListError.Create(m_err_loadlib);
    if not TSnmpInitProc(FSnmpInitProc)(GetTickCount,
        PollForTrapEvent, SupportedViewRoot) then
        raise EConnListError.Create(m_err_initlib);
    FName := aName;
    FDescription := aDesc;
    FId := aId;
    dec(Fid[8]);
end;

// знищення об'єкту класу
destructor TNetstatCounterObject.Destroy;
begin
    if FSnmpLib <> 0 then FreeLibrary(FSnmpLib);
    if FWSInited then WSACleanup;
    inherited;
end;

{ TCounterList }
procedure TCounterList.AddCounter(aCounter:TNetstatCounterObject);
begin
    assert( assigned( aCounter ) );
    FCounters.AddObject( aCounter.Name, aCounter );
end;

function TCounterList.GetCount :Integer;
begin
    assert( assigned( FCounters ) );
    result := FCounters.Count;
end;

function TCounterList.GetCounters(Index:Integer ):TNetstatCounterObject;
begin
    assert( assigned( FCounters ) );
    result := TNetstatCounterObject(FCounters.Objects[Index]);
end;

```

```

function TCounterList.IndexOf( const aName :String ):Integer;
begin
    result := FCounters.IndexOf( aName );
end;

// створення об'єкту класу
constructor TCounterList.Create( aOwner :TComponent );
begin
    inherited;
    FCounters := TStringList.Create;
end;

// знищення об'єкту класу
destructor TCounterList.Destroy;
begin
    Clear;
    if assigned(FCounters) then FCounters.Free;
    inherited;
end;

procedure TCounterList.Clear;
var
    i :integer;
    c :TObject;
begin
    if assigned(FCounters) then
        for i := FCounters.Count-1 downto 0 do
            begin
                c := FCounters.Objects[i];
                FCounters.Delete(i);
                if assigned(c) then c.free;
            end;
        end;
end;

{ TIpConNInfo }
// створення об'єкту класу
constructor TIpConNInfo.Create( const aProto :String );
begin
    inherited Create;
    FState := LISTEN;
    FProto := aProto;
end;

function TIpConNInfo.Ip2Str( const Ip :TInAddr ):String;
begin
    result := format('%d.%d.%d.%d',
        [Integer(ip.s_un_b.s_b1),
        Integer(ip.s_un_b.s_b2),
        Integer(ip.s_un_b.s_b3),
        Integer(ip.s_un_b.s_b4)]);
end;

function TIpConNInfo.GetLocalPort :Integer;
begin
    result := FLocalPort;
end;

function TIpConNInfo.GetLocalIpString :String;
begin
    result := Ip2Str(FLocalIp);
end;

function TIpConNInfo.GetRemotePort :Integer;
begin
    if State = LISTEN then
        result := 0
    end;
end;

```

```

    else
        result := FRemotePort;
end;

function TIpConnInfo.GetRemoteIpString :String;
begin
    result := Ip2Str(FRemoteIp);
end;

function TIpConnInfo.GetStateString :String;
const
    state_name :array[CLOSED..TCB_DISCARD] of string[16] =
        ('CLOSED',
         'LISTEN',
         'SYN_SENT',
         'SYN_RECEIVED',
         'ESTABLISHED',
         'CLOSE_WAIT',
         'FIN_WAIT_1',
         'CLOSING',
         'LAST_ACK',
         'FIN_WAIT_2',
         'TIME_WAIT',
         'TCB_DISCARD'
        );
begin
    if State in [CLOSED..TCB_DISCARD] then
        result := state_name[state]
    else
        result := 'UNKNOWN';
end;

{ TF }
function TF.GetConnCount :Integer;
begin
    assert(assigned(FConnections));
    result := FConnections.Count;
end;

function TF.GetConnections( Index :Integer ):TIpConnInfo;
begin
    assert(assigned(FConnections));
    result := FConnections[Index];
end;

procedure TF.Clear;
var
    i :integer;
    p :TIpConnInfo;
begin
    if assigned(FConnections) then
        for i := FConnections.Count-1 downto 0 do
            begin
                p := FConnections[i];
                assert(assigned(p));
                FConnections.Delete(i);
                p.free;
            end;
        end;
end;

procedure TF.StatusChange;
begin
    if assigned(FOnStatusChange) then FOnStatusChange(Self);
end;

```

```

procedure TF.SetStatus( Value :TIpConnListStatus );
begin
    if FStatus <> Value then
        begin
            FStatus := Value;
            StatusChange;
        end;
end;

procedure TF.DoLock;
begin
    if not Lock(DEFAULT_LOCK_TIMEOUT) then
        raise EConnListLockError.Create(m_err_lock);
end;

procedure TF.DoUnlock;
begin
    if not UnLock then
        raise EConnListUnLockError.Create(m_err_unlock);
end;

// створення об'єкту класу
constructor TF.Create( aOwner :TComponent );
var
    WSDData :TWSAData;
begin
    inherited;
    FProtocols := [udp_ip, tcp_ip];
    FWSInit := WsaStartup($0101, WSDData ) = 0;
    if not FWSInit then
        raise EConnListError.Create(m_err_wsstartup);
    FSnmpLib := LoadLibrary(SNMP_LIB_NAME);
    if FSnmpLib = 0 then
        raise EConnListError.Create(m_err_loadlib);
    FSnmpInitProc := GetProcAddress(FSnmpLib, SNMP_INITPROC_NAME);
    FSnmpQueryProc := GetProcAddress(FSnmpLib, SNMP_QUERYPROC_NAME);
    if ( ( FSnmpQueryProc = Nil ) or ( FSnmpInitProc = Nil ) ) then
        raise EConnListError.Create(m_err_loadlib);
    if not TSnmpInitProc(FSnmpInitProc)(GetTickCount,
        FPollForTrapEvent, FSupportedViewRoot) then
        raise EConnListError.Create(m_err_initlib);
    FAccessMutex := CreateMutex(nil, false, nil);
    if FAccessMutex = 0 then
        raise EConnListError.Create(m_err_alloc);
    FConnections := TList.Create;
    FStatus := connlist_ready;
end;

// знищення об'єкту класу
destructor TF.Destroy;
begin
    Clear;
    if assigned(FConnections) then FConnections.Free;
    if FAccessMutex <> 0 then CloseHandle(FAccessMutex);
    if FSnmpLib <> 0 then FreeLibrary(FSnmpLib);
    if FWSInit then WSACleanup;
    inherited;
end;

end.

```

К6П3 - 2023

DATA.PAS - РОБОТА СИСТЕМИ

```

unit Data;
// модуль Data
{
Дослідження та програмна реалізація системи підтримки індикативного аналізу
Виконав: студент 2 курсу магістратури, Дьяченко Юрій Володимирович
Керівник: Коваленко О.В.
2023 рік
}

Interface
// об'ява даних

Uses
// бібліотеки
Windows, SysUtils, ActiveX, ShlObj, Classes, ShellAPI;

const
// використовуємо константи
Nul = 0000;
MaxWord = $FFFF;
MaxInteger = $7FFFFFFF;
MaxFloat = 2.7e308;
MinFloat = 5.0e-324;
MaxExtended = 1.1e4932;
MinExtended = 9.99e-4933;
HalfCycle = Pi;
FullCycle = 2*Pi;
Quadrant = Pi/2;
chNull = #0;
chBackspace = #19;

type
// власні типи
PString=^TString;
TString=type AnsiString;
PAnsiStr=^TAnsiStr;
TAnsiStr=array[0..259] of AnsiChar;
PWideStr=^TWideStr;
TWideStr=array[0..259] of WideChar;
PShortStr=^TShortStr;
TShortStr=type ShortString;
PSetChar=^TSetChar;
TSetChar=set of AnsiChar;
PWideInt=^TWideInt;
TWideInt=type Int64;
TColorChannel = (ccRed, ccGreen, ccBlue, ccAlpha);
TColorChannels = set of TColorChannel;
PBoolean = ^Boolean;
function LetterToNumber(const Letter: TString): Integer;
function NumberToLetter(Number: Integer): TString;

procedure SplitAlphanumericName(const Name: TString; var Alpha: TString;
var Num: Integer; const AdditionalChars: TSetChar = []);
type
// власні типи
TUnknown = class (TObject, IUnknown)
protected
FRefCount: Integer;
function QueryInterface(const IID: TGUID; out Obj): HRESULT; virtual;
stdcall;
function _AddRef: Integer; virtual; stdcall;
function _Release: Integer; virtual; stdcall;
public
function Unknown: IUnknown; overload;

```

```

    procedure Unknown(out Obj); overload;
end;
TObjectX = TUnknown;
type
// власні типи
EShellLinkError = class (Exception);
TShellLink = class(TUnknown)
private
    FResult: HRESULT;
    FShellLink: IShellLink;
    FPersistDOC_PACK : IPersistDOC_PACK;
    FTemp: WideString;
    FDesktopFolder: TString;
    FProgramsFolder: TString;
    FStartMenuFolder: TString;
    FStartupFolder: TString;
    FMyDocsFolder: TString;
    function GetArguments: TString;
    function GetDescription: TString;
    function GetHotKey: word;
    function GetIconIndex: Integer;
    function GetIconLoc: TString;
    function GetPath: TString;
    function GetPIDL: PItemIDList;
    function GetShowCmd: Integer;
    function GetWorkDir: TString;
    procedure SetArguments(const Value: TString);
    procedure SetDescription(const Value: TString);
    procedure SetHotKey(const Value: word);
    procedure SetIconIndex(const Value: Integer);
    procedure SetIconLoc(const Value: TString);
    procedure SetPath(const Value: TString);
    procedure SetPIDL(const Value: PItemIDList);
    procedure SetShowCmd(const Value: Integer);
    procedure SetWorkDir(const Value: TString);
    procedure RunError(const Msg: TString; const Args: TString = '');
    function ResolveDOC_PACKName(DOC_PACKName: TString): PWideChar;
    function DesktopFolder: TString;
    function ProgramsFolder: TString;
    function StartMenuFolder: TString;
    function StartupFolder: TString;
    function MyDocsFolder: TString;
public
    constructor Create;
    destructor Destroy;override;
end;
type
// власні типи
TForEachFunc = function (Tag: Integer; Index: Integer; var Item): Integer;
register;
EDynArray = class (Exception);
TDynArray = class (TObjectX)
private
    FHandle: hLocal;
    FData: Pointer;
    FItemSize: Cardinal;
    FCount: Cardinal;
    function AllocMem(ACount: Cardinal; var Handle: hLocal): pointer;
    procedure FreeMem(var Handle: hLocal);
    procedure _SetCount(const Value: Cardinal);
    procedure DoSizeChanged;
protected
    function GetFirstItem: Pointer;
    procedure PutItem(Index: Integer; const Item);
    procedure GetItem(Index: Integer; out Item);

```

```

    procedure Error(Index: Integer);
    procedure SizeChanged; virtual;
    procedure SetCount(const Value: Cardinal); virtual;
public
    property Count: Cardinal read FCount write _SetCount;
    property ItemSize: Cardinal read FItemSize;
    property FirstItem: Pointer read FData;
    function Add: Integer; virtual;
    function AddItem(const Item): Integer; virtual;
    procedure Insert(Index: Integer); virtual;
    procedure InsertItem(Index: Integer; const Item); virtual;
    procedure Delete(Index: Integer); virtual;
    procedure DeleteItem(Index: Integer; out Item); virtual;
    procedure Extend(Count: Cardinal = 1); virtual;
    procedure Trim(Count: Cardinal = 1); virtual;
    procedure Swap(Index1, Index2: Cardinal); virtual;
    function ForEach(Tag: Integer; ForEachFunc: TForEachFunc): Integer; virtual;
    function GetItemPtr(Index: Integer): Pointer;
    constructor Create(ACount, AItemSize: Cardinal);
    destructor Destroy; override;
end;
TDynamicArrayClass = class of TDynamicArray;
type
// власні типи
    TDStatus = (fsReading, fsWriting);
    EDOC_PACKError = class (Exception);
const
// стандартні константи
    faReadOnly          = $00000001;
    faHidden            = $00000002;
    faSystem            = $00000004;
    faDirectory         = $00000010;
    faArchive           = $00000020;
    faEncrypted         = $00000040;
    faNormal            = $00000080;
    faTemporary        = $00000100;
    faSparseDOC_PACK   = $00000200;
    faReparsePoint     = $00000400;
    faCompressed        = $00000800;
    faOffline           = $00001000;
    faNotContentIndexed = $00002000;
type
// власні типи
{ Об'ява класу TD }
    TD = class (TObjectX)
private
    FDOC_PACKName: TString;
    FHandle: HDOC_PACK;
    FStatus: TDStatus;
    FDummy: LongWord;
    procedure CreateBackup;
    function GetSize: Integer;
    function GetCreationTime: TDTime;
    function GetLastAccessTime: TDTime;
    function GetLastWriteTime: TDTime;
    procedure SetCreationTime(const Value: TDTime);
    procedure SetLastAccessTime(const Value: TDTime);
    procedure SetLastWriteTime(const Value: TDTime);
    function GetAttributes: LongInt;
    procedure SetAttributes(const Value: LongInt);
protected
    function GetErrorMessage(Code: Integer): TString; dynamic;
public
    property DOC_PACKName: TString read FDOC_PACKName;
    property Status: TDStatus read FStatus;

```

```

    property Handle: HDOC_PACK read FHandle;
    property Size: Integer read GetSize;
    property CreationTime: TDateTime read GetCreationTime write SetCreationTime;
    property LastAccessTime: TDateTime read GetLastAccessTime write
SetLastAccessTime;
    property LastWriteTime: TDateTime read GetLastWriteTime write SetLastWriteTime;
    property Attributes: LongInt read GetAttributes write SetAttributes;

{ Об'ява класу TDStrm }
TDStrm = class (TStream)
    private
    FHandle: HDOC_PACK;
    FStatus: TDStatus;
    FDOC_PACKName: TString;
    procedure CreateBackup;
    function GetAttributes: LongInt;
    function GetCreationTime: TDateTime;
    function GetLastAccessTime: TDateTime;
    function GetLastWriteTime: TDateTime;
    procedure SetAttributes(const Value: LongInt);
    procedure SetCreationTime(const Value: TDateTime);
    procedure SetLastAccessTime(const Value: TDateTime);
    procedure SetLastWriteTime(const Value: TDateTime);
    protected
    procedure SetSize(NewSize: LongInt); override;
    procedure Error(Code: Integer); dynamic;
    function GetErrorMessage(Code: Integer): TString; dynamic;
    public
    property DOC_PACKName: TString read FDOC_PACKName;
    property Status: TDStatus read FStatus;
    property Handle: HDOC_PACK read FHandle;
    property CreationTime: TDateTime read GetCreationTime write SetCreationTime;
    property LastAccessTime: TDateTime read GetLastAccessTime write
SetLastAccessTime;
    property LastWriteTime: TDateTime read GetLastWriteTime write SetLastWriteTime;
    property Attributes: LongInt read GetAttributes write SetAttributes;
    constructor Create(ADOC_PACKName: TString; Backup: Boolean);
    constructor Open(ADOC_PACKName: TString);
    function Write(const Buffer; Count: LongInt): LongInt; override;
    function Read(var Buffer; Count: LongInt): LongInt; override;
    function Seek(Offset: LongInt; Origin: Word): LongInt; override;
    procedure Close;
    destructor Destroy; override;
    class procedure DecodeDateTime(const DateTime: TDateTime;
    Year, Month, Day, Hour, Min, Sec: PWord);
    class function EncodeDateTime(Year, Month, Day, Hour, Min, Sec: Word):
TDateTime;
    procedure UserError(Code: Integer);
end;

{ Об'ява класів }
ELanError = class (Exception);
TD = class;
PLanRow = ^TDRow;
TDRow = class (TDynamicArray)
    private
    FJan: TD;
    public
    property Jan: TD read FJan;
    constructor Create(AColCount: Integer; AJan: TD);
end;

{ Об'ява класу TDRows }
TDRows = class (TDynamicArray)

```

```

private
  FWidth: Integer;
  FColIndex: Integer;
  function GetRow(Index: Integer): TDRow;
  procedure SetRow(Index: Integer; const Value: TDRow);
  procedure SetWidth(const Value: Integer);
  function SetWidthFunc(Index: Integer; var Row: TDRow): Integer;
  function InsertColFunc(Index: Integer; var Row: TDRow): Integer;
  function DeleteColFunc(Index: Integer; var Row: TDRow): Integer;
public
  property Width: Integer read FWidth write SetWidth;
  property Row[Index: Integer]: TDRow read GetRow write SetRow; default;
  procedure InsertCol(Index: Integer);
  procedure DeleteCol(Index: Integer);
  constructor Create(ALan: TD);
end;

{ Об'ява класу TD }
TD = class (TUnknown)
private
  FItemSize : Cardinal;
  FRows: TDRows;
  function GetColCount: Integer;
  function GetRowCount: Integer;
  procedure SetColCount(const Value: Integer);
  procedure SetRowCount(const Value: Integer);
  function GetRow(Index: Integer): TDRow;
protected
  function CreateRow: TDRow; virtual;
public
  procedure GetItem(ACol, ARow: Integer; out Item);
  procedure PutItem(ACol, ARow: Integer; const Item);
  procedure InsertRow(Index: Integer);
  procedure DeleteRow(Index: Integer);
  procedure InsertCol(Index: Integer);
  procedure DeleteCol(Index: Integer);
  function ForEachRow(Tag: Integer; ForEachRowFunc: TForEachFunc): Integer;
  property ColCount: Integer read GetColCount write SetColCount;
  property RowCount: Integer read GetRowCount write SetRowCount;
  property Row[Index: Integer]: TDRow read GetRow;
  constructor Create(AColCount, ARowCount, AItemSize: Integer);
  destructor Destroy; override;
end;

implementation
// реалізація

Uses
// бібліотеки
  Consts, SysConst, DimConst;

type
// власні типи
  TDgIDItem = packed record
    LangID: Byte;
    Charset: Byte;
  end;

function SamePoint(const Point1, Point2: TPoint): LongBool;
begin
  Result:=TWideInt(Point1)=TWideInt(Point2);
end;

```

```

function IsNullPoint(const Point: TPoint): LongBool;
begin
  Result:=not LongBool(TWideInt(Point));
end;

procedure MovePoint(var Point: TPoint; DispX, DispY: Integer);
asm
  add [eax], edx
  add [eax+4], ecx
end;

function CloseTo(const Point1, Point2: TPoint; Distance: Integer): LongBool;
begin
  Result:=Inside(Point2, Rect(Point1.X-Distance, Point1.Y-Distance,
                              Point1.X+Distance, Point1.Y+Distance));
end;

function GetData(Num, Den:Double):Double;
begin
  if Den<>0 then begin
    Result:=arctan(Num/Den);
    if Den<0 then Result:=HalfCycle+Result else if Num<0 then
Result:=FullCycle+Result;
  end else begin
    if Num>0 then Result:=Quadrant else Result:=3*Quadrant;
  end;
end;

function GetAlpha(Y1, Z1, Y2, Z2, Y3, Z3:Double):Double;
Var // об'ява локальних типів даних
  A1, A2:Double;
begin
  A1:=GetData(Z1-Z2,Y2-Y1);
  A2:=GetData(Z3-Z2,Y2-Y3);
  if A2<A1 then A2:=FullCycle+A2;
  Result:=A2-A1;
end;

function GetAlphaScr(X1, Y1, X2, Y2, X3, Y3:Double):Double;
Var
// об'ява локальних типів даних
  A1, A2:Double;
begin
  A1:=GetData(X2-X1,Y1-Y2);
  A2:=GetData(X2-X3,Y3-Y2);
  if A2<A1 then A2:=FullCycle+A2;
  Result:=A2-A1;
end;

function CenterPoint(const Rect: TRect): TPoint;
asm
  push esi
  mov esi, eax
  mov eax, [esi]
  add eax, [esi+8]
  shr eax, 1
  mov [edx].TPoint.x, eax
  mov eax, [esi+4]
  add eax, [esi+12]
  shr eax, 1
  mov [edx].TPoint.y, eax
  pop esi
end;

```

```

function Max(const R1,R2:Integer):Integer;overload;
asm
    cmp eax, edx
    jng @@10
    ret
@@10:
    mov eax, edx
end;

function Max(const R1,R2:Extended):Extended;overload;
begin
    if R1>R2 then Result:=R1 else Result:=R2;
end;

function Max(const P1, P2: TPoint; CompareY: LongBool=False): TPoint; overload;
Var
// об'ява локальних типів даних
    F: function (const Point1, Point2: TPoint): Integer;
begin
    if CompareY then F:=ComparePointY else F:=ComparePointX;
    if F(P1, P2) = nMore then Result:=P1 else Result:=P2;
end;
function Min(const R1,R2:Integer):Integer;overload;
asm
    cmp eax, edx
    jnl @@10
    ret
@@10:
    mov eax, edx
end;

function Min(const R1,R2:Extended):Extended;overload;
begin
    if R1<R2 then Result:=R1 else Result:=R2;
end;
function Min(const P1, P2: TPoint; CompareY: LongBool = False): TPoint;
Var
// об'ява локальних типів даних
    F: function (const Point1, Point2: TPoint): Integer;
begin
    if CompareY then F:=ComparePointY else F:=ComparePointX;
    if F(P1, P2) = nLess then Result:=P1 else Result:=P2;
end;

procedure Sync;
asm
    call WinNT
    test eax, 1
    jz  @@10
    ret
@@10:
    mov  dx,3dah
@@wait:
    in   al,dx
    test al,8
    jz  @@wait
end;

function LinesIntersection(A1, B1, C1, A2, B2, C2: Extended; var Y, Z:
Extended): Boolean;
Var
// об'ява локальних типів даних
    Det: Extended;
begin
    Det:=A1*B2-A2*B1;

```

```

Result:=Abs(Det)>1e-20;
if Result then begin
  Y := (c2*b1-c1*b2)/det;
  Z := (a2*c1-a1*c2)/det;
end;
end;

```

```

function SegmentLength(const X1, Y1, X2, Y2: Extended): Extended;
asm

```

```

  fld  X1
  fld  X2
  fsub
  fld  st(0)
  fmul
  fld  Y1
  fld  Y2
  fsub
  fld  st(0)
  fmul
  fadd
  fsqrt
  fwaitend;

```

```

procedure Rotate(X, Y, X0, Y0, Alpha: Extended; var X1, Y1: Extended);
Var

```

```

// об'ява локальних типів даних
Sin, Cos: Extended;
DX, DY: Extended;
begin
  SinCos(Alpha, Sin, Cos);
  DX:=(X-X0); DY:=(Y-Y0);
  X1:=DX*Cos+DY*Sin+X0;
  Y1:=DY*Cos-DX*Sin+Y0;
end;

```

```

function LinesIntersection(Y1, Z1, Y2, Z2, Y3, Z3, Y4, Z4: Extended; var Y, Z:
Extended): Boolean; overload;

```

```

Var
// об'ява локальних типів даних
  A1, B1, C1, A2, B2, C2: Extended;
begin
  Result:=LinesIntersection(A1, B1, C1, A2, B2, C2, Y, Z);
end;

```

```

procedure CopyRect(const Source: TRect; var Dest: TRect);

```

```

asm
  mov  ecx, 16
  call MoveMem
end;

```

```

function RectIntersection(const Rect1, Rect2: TRect): TRect;

```

```

begin
  RebuildRect(PRect(@Rect1)^);
  RebuildRect(PRect(@Rect2)^);
  if Inside(Rect2.TopLeft, Rect1) then begin
    if Inside(Rect2.BottomRight, Rect1) then begin
      Result:=Rect2;
      Exit;
    end else begin
      Result.TopLeft:=Rect2.TopLeft;
      Result.BottomRight:=Rect1.BottomRight;
      Exit;
    end;
  end;
  if Inside(Rect2.BottomRight, Rect1) then begin

```

```

    if Inside(Rect2.TopLeft, Rect1) then begin
        Result:=Rect2;
        Exit;
    end else begin
        Result.TopLeft:=Rect1.TopLeft;
        Result.BottomRight:=Rect2.BottomRight;
        Exit;
    end;
end;
if Inside(Rect1.TopLeft, Rect2) then begin
    if Inside(Rect1.BottomRight, Rect2) then begin
        Result:=Rect1;
        Exit;
    end else begin
        Result.TopLeft:=Rect1.TopLeft;
        Result.BottomRight:=Rect2.BottomRight;
        Exit;
    end;
end;
if Inside(Rect1.BottomRight, Rect2) then begin
    if Inside(Rect1.TopLeft, Rect2) then begin
        Result:=Rect1;
        Exit;
    end else begin
        Result.TopLeft:=Rect2.TopLeft;
        Result.BottomRight:=Rect1.BottomRight;
        Exit;
    end;
end;
ClearMem(Result, SizeOf(Result));
end;

procedure ArrangeMin(var R1, R2: Integer);
asm
    mov     ecx, [eax]
    cmp     ecx, [edx]
    jl     @@10
    xchg    ecx, [edx]
    mov     [eax], ecx
@@10:
end;

procedure ArrangeMax(var R1, R2: Integer);
asm
    mov     ecx, [eax]
    cmp     ecx, [edx]
    jg     @@10
    xchg    ecx, [edx]
    mov     [eax], ecx
@@10:
end;

function Sign(const Value:Integer):Integer;overload;
asm
    test   eax, eax
    jl    @@10
    jg    @@20
    ret
@@10:
    mov   eax, -1
    ret
@@20:
    mov   eax, 1
end;

```

```

function Sign(const Value:Extended):Extended;overload;
begin
  if Value<0 then Result:=-1.0 else
    if Value>0 then Result:=1.0 else Result:=0.0;
end;

procedure Swap(var R1,R2:Integer);overload;
asm
  mov  ecx, [eax]
  xchg ecx, [edx]
  mov  [eax], ecx
end;

function FindChars(const Source:TString;const
Chars:TSetChar;CurrentPosition:Integer=1;Direction:Integer=1):Integer;
Var
  // об'ява локальних типів даних
  i,len:Integer;
  Delta:Integer;
begin
  Result:=0;
  if Direction<0 then Delta:=-1 else Delta:=1;
  i:=CurrentPosition;
  len:=Length(Source);
  if Len=0 then Exit;
  repeat
    if Source[i] in Chars then begin
      Result:=i;
      Break;
    end;
    i:=i+Delta;
    if (i<1) or (i>len) then Break;
  until false;
end;

function LeftTrim(const Str:TString;const Chr:Char=chSpace):TString;
Var
  // об'ява локальних типів даних
  Count:Integer;
begin
  Result:=Str;
  Count:=0;
  while Length(Result)>0 do begin
    if Result[Count+1]=Chr then Inc(Count) else Break;
  end;
  if Count<>0 then Delete(Result,1,Count);
end;

function RightTrim(const Str:TString;const Chr:Char=chSpace):TString;
var Count:Integer;
begin
  Result:=Str;
  Count:=0;
  while Length(Result)>0 do begin
    if Result[Length(Result)-Count]=Chr then Inc(Count) else Break;
  end;
  if Count<>0 then SetLength(Result,Length(Result)-Count);
end;

function LeftExpand(const Str:TString; Count: Integer; const Chr:Char=chSpace):
TString;
Var
  // об'ява локальних типів даних
  i:Integer;
  PS, PD: PChar;

```

```

begin
  if Count<0 then Count:=0;
  SetString(Result, nil, Length(Str)+Count);
  PS:=@Str[1];
  PD:=@Result[Count+1];
  for i:=1 to Count do Result[i]:=Chr;
  Move(PS^, PD^, Length(Str));
end;

function RightExpand(const Str:TString; Count: Integer; const Chr:Char=chSpace):
TString;
Var
  // об'ява локальних типів даних
  L: Integer;
begin
  if Count<0 then Count:=0;
  L:=GetLength(Str);
  SetString(Result, nil, L+Count);
  MoveMem(PChar(Str)^, PChar(Result)^, L);
  FillMem(PChar(@Result[L+1])^, Count, Byte(Chr));
end;

function TrimStr(const Str:TString;const Chr:Char=chSpace):TString;
begin
  if Str='' then Result:='' else Result:=LeftTrim(RightTrim(Str,Chr),Chr);
end;

function LeadTrim(const Str:TString; Count:Integer=1):TString;
begin
  if Count<0 then Count:=0;
  SetString(Result, PChar(IncPtr(PChar(Str), Count)), Length(Str)-Count);
end;

function TrailTrim(const Str:TString; Count:Integer=1):TString;
begin
  if Count<0 then Count:=0;
  SetString(Result, PChar(Str), Length(Str)-Count);
end;

function GetSubStr(const Str:TString;N:byte;Separator:Char=chSpace):TString;
Var
  // об'ява локальних типів даних
  S: PChar;
  P1, P2: Integer;
begin
  P1:=CharEntryPos(Str, Separator, N-1);
  Inc(P1);
  S:=@Str[P1];
  P2:=CharEntryPos(S, Separator, 1);
  if P2=0 then P2:=Length(Str) else P2:=P1+P2-1;
  Result:=TrimStr(ReadSubStr(Str, P1, P2), Separator);
  if Result=Separator then Result:='';
end;

function ExtractStr(const Str:TString;N:byte):TString;
Var
  // об'ява локальних типів даних
  P,I:Integer;
  S:TString;
begin
  S:=Str;
  for i:=1 to n-1 do begin
    P:=Pos(chSpace,S);
    S:=Copy(S,Succ(P),Length(S)-P);
    S:=LeftTrim(S);
  end;
end;

```

```

end;
P:=Pos(chSpace,S);
if P<>0 then Result:=Copy(S,1,Pred(P))
  else Result:=S;
end;

procedure ExtractStrings(Str: TString; List: TStrings;
  Separator: Char);
Var
// об'ява локальних типів даних
P1, P2: PChar;
begin
List.BeginUpdate;
try
List.Clear;
P1:=PChar(Str);
repeat
P2:=StrScan(P1, Separator);
SetByteValue(P2, 0);
List.Add(P1);
P1:=P2;
Inc(P1);
until P2 = nil;
finally
List.EndUpdate;
end;
end;

function RemoveChars(const Str:TString;const Chars:TSetChar):TString;
Var
// об'ява локальних типів даних
i:Integer;
begin
Result:='';
for i:=1 to Length(Str) do if not (Str[i] in Chars) then Result:=Result+Str[i];
end;

function ReplaceChar(const Str:TString;OldChar,NewChar:Char):TString;
Var
// об'ява локальних типів даних
i:Integer;
begin
Result:=Str;
for i:=1 to Length(Result) do if Result[i]=OldChar then Result[i]:=NewChar;
end;

function ReplaceStr(const Str:TString;const
  OldSubStr,NewSubStr:TString):TString;
Var
// об'ява локальних типів даних
P:Integer;
begin
Result:=Str;
P:=Pos(OldSubStr,Result);
if P<>0 then begin
Delete(Result,P,Length(OldSubStr));
Insert(NewSubStr,Result,P);
end;
end;

function __pos(SubStr, Str: TString; var P: Integer): Integer;
begin
P:=Pos(SubStr, Str);
Result:=P;
end;

```

```

function ReplaceStrAll(const Str: TString; const OldSubStr, NewSubStr: TString):
TString;
Var
// об'ява локальних типів даних
P: Integer;
Len: Integer;
begin
Result:=Str;
Len:=Length(OldSubStr);
while __pos(OldSubStr, Result, P)<>0 do begin
Delete(Result, P, Len);
Insert(NewSubStr, Result, P);
end;
end;

procedure CleanUp(var Str: TString);
asm
mov    eax, [eax]
test   eax, eax
jz     @@10
push   eax
call   GetLength
mov    edx, eax
pop    eax
mov    [eax-4], edx
@@10:
end;

procedure CleanUp(var Str:TString; DoTrim: LongBool);
begin
SetLength(Str, GetLength(Str));
CleanUp(Str);
if DoTrim then Str:=TrimStr(Str);
end;

function FillString(Chr:Char;Count:Integer):TString;
begin
SetString(Result, nil, Count);
FillChar(Pointer(Result)^, Count, Chr);
end;

function UpString(const Str:TString):TString;
begin
Result:=Str;
CharUpper(@Result[1]);
end;

function DnString(const Str:TString):TString;
begin
Result:=Str;
CharLower(@Result[1]);
end;

function GetChar(const Str:TString; Position:Integer=1):Char; register;
asm
push   edi
push   esi
mov    edi, edx
mov    esi, eax
call   GetLength
test   eax, eax
jnz   @@10
jmp    @@30
@@10:

```

```

    cmp    eax, edi
    jnb   @@20
    xor   eax, eax
    jmp   @@30
@@20:
    mov   eax, esi
    mov   edx, edi
    dec   edx
    call  ReadChar
@@30:
    pop   esi
    pop   edi
end;

function ReadChar (Ptr:Pointer;Offset:Integer):Char; register;
asm
    add   eax, edx
    mov   al, [eax]
end;

function UpChar (Ch:Char):Char; register;
asm
    and   eax, 000000FFh
    push  eax
    call  CharUpper
end;

function DnChar (Ch:Char):Char; register;
asm
    and   eax, 000000FFh
    push  eax
    call  CharLower
end;

function ReflectStr(const Str:TString):TString;
Var
// об'ява локальних типів даних
    i:Integer;
    len:Integer;
begin
    Len:=GetLength(Str);
    SetLength(Result,len);
    for i:=1 to Len do Result[i]:=Str[Len-i+1];
end;

function ReadSubStr(const Str:TString; Head, Tail:Integer):TString;
begin
    Result:=Copy(Str, Head, Tail-Head+1);
end;

function StrToFlt(const Str:TString;var Code:Integer):Extended;overload;
begin
    Val(PChar(Str), Result, Code);
end;

function StrToFlt(const Str:TString):Extended;overload;
Var
// об'ява локальних типів даних
    i:Integer;
begin
    Result:=StrToFlt(Str, i);
    if i<>0 then Result:=0;
end;

```

```

function FltToStr(const Value:Extended;
                  Precision:Integer=5):TString;
var
  P:Integer;
begin
  Result:=FloatToStrF(Value,ffGeneral,Precision,0);
  P:=Pos(',',Result);
  if P<>0 then Result[P]:=chPoint;
  P:=Pos(DecimalSeparator,Result);
  if P<>0 then Result[P]:=chPoint;
end;

function BreakStr(const Str:TString;Len:Integer=64;AltChar:Char='\'):TString;
Var
  // об'ява локальних типів даних
  i,j:Integer;
  Alt:Boolean;
begin
  if Length(Str)<=Len then begin
    Result:=Str;
    Exit;
  end;
  Result:='';
  i:=0;
  repeat
    j:=i+Len;
    if j>Length(Str) then begin
      j:=Length(Str);
      Result:=Result+Copy(Str,i+1,j-i);
      Exit;
    end;
    Alt:=False;
    while Str[j]<>chSpace do begin
      Dec(j);
      if j=i then begin
        Alt:=True;
        Break;
      end;
    end;
    if Alt then begin
      j:=i+Len;
      if j>Length(Str) then begin
        j:=Length(Str);
        Result:=Result+Copy(Str,i+1,j-i);
        Exit;
      end;
      while Str[j]<>AltChar do begin
        Dec(j);
        if j=i then begin
          j:=i+Len;
          Break;
        end;
      end;
    end;
    Result:=Result+Copy(Str,i+1,j-i)+#13#10;
    i:=j;
  until i>=Length(Str);
end;

function ValidInt(const Value:TString):LongBool;
var
  i,Code:Integer;
begin
  Val(Value,i,Code);
  Hole(i);

```

```

    Result:=Code=0;
end;

function ValidFloat(const Value:TString):LongBool;
Var
// об'ява локальних типів даних
    i:Double;
    Code:Integer;
begin
    Val(Value,i,Code);
    Hole(i);
    Result:=Code=0;
end;

function ValidFloatINF(const Value:TString): LongBool;
var
    R: Double;
    Code:Integer;
begin
    Val(Value, R, Code);
    Hole(Code);
    Result:=Infinity(R)=0;
end;

function ValidateFloat(const Value:TString):TString;
Var
// об'ява локальних типів даних
    P:Integer;
begin
    Result:=Value;
    P:=Pos(DecimalSeparator,Result);
    if P<>0 then Result[P]:=chPoint;
    P:=Pos(chComma,Result);
    if P<>0 then Result[p]:=chPoint;
    if not ValidFloat(Result) then Result:='';
end;

function Join(const Str1, Str2: TString): TString;
begin
    Result:='';
    if not IsEmptyStr(Str1) then Result:=PChar(@Str1[1]);
    if not IsEmptyStr(Str2) then Result:=Result+PChar(@Str2[1]);
end;

function LastChar(const Str:TString):Char;
begin
    if Str='' then Result:=chNull else Result:=Str[Length(Str)];
end;

function NextChar(const Str:TString;Pos:Integer;Passed:Char=chSpace):Char;
begin
    Result:=NextChar(Str,Pos,[Passed]);
end;

function PrevChar(const Str:TString;Pos:Integer;Passed:Char=chSpace):Char;
begin
    Result:=PrevChar(Str,Pos,[Passed]);
end;

function NextChar(const Str:TString;Pos:Integer;Passed:TSetChar):Char;overload;
var i:Integer;
begin
    Result:=#0;
    for i:=Pos+1 to Length(Str) do if not (Str[i] in Passed) then begin
        Result:=Str[i];
    end;
end;

```

```

    Break;
  end;
end;

function PrevChar(const Str:TString;Pos:Integer;Passed:TSetChar):Char;overload;
Var
// об'ява локальних типів даних
  i:Integer;
begin
  Result:=#0;
  for i:=Pos-1 downto 1 do if not (Str[i] in Passed) then begin
    Result:=Str[i];
    Break;
  end;
end;

procedure AddString(var Str:TString; const Value:TString);
begin
  CleanUp(Str);
  Str:=Str+Value;
end;

function AdjustLength(Str: TString; Len: Integer; Ch: Char = chSpace): TString;
Var
// об'ява локальних типів даних
  L, N: Integer;
  S1: TString;
begin
  L:=GetStrLen(Str);
  if L<Len then begin
    N:=Len - L;
    SetString(S1, nil, N);
    FillMem(PChar(S1)^, N, Ord(Ch));
    Result:=Str+S1;
  end else Result:=Str;
end;

function CopyToBuf(const Source:TString; Buf:PChar; Size:Integer):LongBool;
var
  Len: Integer;
begin
  Len:=GetLength(Source)+1;
  if Len>Size then begin
    Result:=False;
    Buf^:=#0;
  end else begin
    if not IsEmptyStr(Source) then MoveMem(PChar(Source)^, Buf^, Len)
    else ClearMem(Buf^, Size);
    Result:=True;
  end;
end;

function _GetTempDirectory: TString;
var
  Buf: array [0..MAX_PATH-1] of AnsiChar;
begin
  GetTempPath(SizeOf(Buf), @Buf);
  Result:=IncludeTrailingBackslash(PChar(@Buf));
end;
var
  TmpDir: TString = '';
  GetTempDirectoryFirstCall: Boolean = True;

function GetTempDirectory: TString;
begin

```

```

if GetTempDirectoryFirstCall then begin
  TmpDir:=_GetTempDirectory;
  GetTempDirectoryFirstCall:=False;
end;
Result:=TmpDir;
end;

function GetTempDOC_PACK(const Prefix: TString): TString;
var
  Buf: array [0..MAX_PATH-1] of AnsiChar;
begin
  GetTempDOC_PACKName(PChar(GetTempDirectory), PChar(Prefix), 0, @Buf);
  Result:=PChar(@Buf);
end;
var
  Checked: Boolean = False;
  Embedded: Boolean = False;

function CheckAutomation: Boolean;
begin
  if not Checked then begin
    Embedded:=MatchString(Parameters, ['-EMBEDDING', '/EMBEDDING'])<>0;
    Checked:=True;
  end;
  Result:=Embedded;
end;

function ExeName:TString;
Var
  // об'ява локальних типів даних
  S: PChar;
  P: Integer;
begin
  Result:=GetCommandLine;
  S:=@Result[2];
  P:=Pos('"', S);
  Result:=ReadSubStr(S, 1, P-1);
end;

function ExePath:TString;
begin
  Result:=ExtractDOC_PACKPath(ExeName);
end;

function WinME: Boolean;
begin
  Result:=( _Win32Platform = VER_PLATFORM_WIN32_WINDOWS) and
    (( _Win32MajorVersion>4) or (( _Win32MajorVersion = 4) and ( _Win32MinorVersion
    >= 90)));
end;

function WinXP: Boolean;
begin
  Result := ( _Win32Platform = VER_PLATFORM_WIN32_NT) and
    (( _Win32MajorVersion)>5) or (( _Win32MajorVersion = 5) and ( _Win32MinorVersion
    >= 1));
end;
Var
  // об'ява локальних типів даних
  GetOperatingSystemFirstCall: Boolean = True;
  GetOperatingSystemResult: TOperatingSystem;

function GetOperatingSystem: TOperatingSystem;
begin
  if GetOperatingSystemFirstCall then begin

```

```

GetOperatingSystemResult:=UndefinedWindows;
case _Win32Platform of
VER_PLATFORM_WIN32S: GetOperatingSystemResult:=Windows3x;
VER_PLATFORM_WIN32_WINDOWS: begin
  if _Win32MajorVersion = 4 then begin
    if _Win32MinorVersion >= 0 then GetOperatingSystemResult:=Windows95;
    if _Win32MinorVersion >=10 then GetOperatingSystemResult:=Windows98;
    if _Win32MinorVersion >=90 then GetOperatingSystemResult:=WindowsME;
  end;
end;
VER_PLATFORM_WIN32_NT: begin
  if _Win32MajorVersion<=4 then GetOperatingSystemResult:=WindowsNT;
  if _Win32MajorVersion = 5 then begin
    if _Win32MinorVersion >= 0 then GetOperatingSystemResult:=Windows2000;
    if _Win32MinorVersion >= 1 then GetOperatingSystemResult:=WindowsXP;
  end;
end;
end;
GetOperatingSystemFirstCall:=False;
end;
Result:=GetOperatingSystemResult;
end;

function TrayWnd: HWND;
begin
  Result:=FindWindow('Shell_TrayWnd','');
end;

function _GetLocale: Integer;
Var
// об'ява локальних типів даних
Translation: PWord;
Buffer: Pointer;
Size, Len, Handle: DWORD;
Name: TString;
begin
Name:=InstanceName;
Size:=GetDOC_PACKVersionInfoSize(PChar(Name), Handle);
if Size = 0 then Result:=GetLocale else begin
  Buffer:=AllocateMem(Size);
  try
    GetDOC_PACKVersionInfo(PChar(Name), Handle, Size, Buffer);
    VerQueryValue(Buffer, '\DOC \Translation', Pointer(Translation), Len);
    Result:=Translation^;
  finally
    DeallocateMem(Buffer);
  end;
end;
end;

function LangIDToCharset(LangID: Integer): Byte;
Var
// об'ява локальних типів даних
I: byte;
begin
  Result:=DEFAULT_CHARSET;
  if LangID = 0 then LangID:=_GetLocale;
  for i:=0 to LangCount do if Lo(LangID) = LangIDToCharsetInfo[i].LangID then
  begin
    Result:=LangIDToCharsetInfo[i].Charset;
    Break;
  end;
  if LangID = $0C1A then Result:=RUSSIAN_CHARSET;
end;

```

```

function ExitWindows(uFlags: UINT): BOOL;
Var
// об'ява локальних типів даних
ProcessHandle: THandle;
TokenHandle: THandle;
Luid: Int64;
Tkp: TTokenPrivileges;
BufferNeeded: DWORD;
begin
  if WinNT then begin
    ProcessHandle:=GetCurrentProcess;
    OpenProcessToken(ProcessHandle, TOKEN_ADJUST_PRIVILEGES or TOKEN_QUERY,
TokenHandle);
    LookupPrivilegeValue(nil, 'SeShutdownPrivilege', Luid);
    Tkp.PrivilegeCount:=1;
    Tkp.Privileges[0].Luid:=Luid;
    Tkp.Privileges[0].Attributes:=SE_PRIVILEGE_ENABLED;
    AdjustTokenPrivileges(TokenHandle, FALSE, Tkp, 0, nil, BufferNeeded);
  end;
  Result:=ExitWindowsEx(uFlags, $FFFF);
end;

procedure RemoveDirectories(const Path: TString);
Var
// об'ява локальних типів даних
S: TString;
Len, P, i: Integer;
begin
  S:=ExcludeTrailingBackslash(Path);
  repeat
    if not RemoveDirectory(PChar(S)) then Break;
    Len:=Length(S);
    P:=0;
    for i:=Len downto 0 do if S[i] = '\' then begin
      P:=i;
      Break;
    end;
    if P = 0 then Break;
    S:=TrailTrim(S, Len-P+1);
  until False;
end;

function LocalHandle; external kernel32 name 'LocalHandle';

function ExeVersion: TString;
begin
  Result:=VersionToString(DOC_PACKVersion);
end;

function InstanceName:TString;
var
  Buf: array [0..MAX_PATH-1] of AnsiChar;
begin
  GetModuleDOC_PACKName(hInstance, Buf, MAX_PATH);
  Result:=Buf;
end;

function InstancePath:TString;
begin
  Result:=ExtractDOC_PACKPath(InstanceName);
end;

function DOC_PACKVersion(const DOC_PACKName: TString = ''): TDVersion;
Var
// об'ява локальних типів даних

```

```

S: TString;
hMem: HGLOBAL;
Buf: pointer;
BufSize, Len, dwHandle: DWORD;
VerInfo: PVSFixedDOC_PACKInfo;
pszName: PAnsiChar;
begin
  FillChar(Result, SizeOf(Result), 0);
  S:=DOC_PACKName;
  CleanUp(S, True);
  if IsEmptyStr(S) then S:=ParamStr(0);
  pszName:=@S[1];
  BufSize:=GetDOC_PACKVersionInfoSize(pszName, dwHandle);
  if BufSize<>0 then begin
    hMem:=GlobalAlloc(GHND, BufSize);
    if hMem = 0 then OutOfMemoryError;
    Buf:=GlobalLock(hMem);
    if Buf=nil then OutOfMemoryError;
    GetDOC_PACKVersionInfo(pszName, dwHandle, BufSize, Buf);
    VerQueryValue(Buf, '\\', pointer(VerInfo), Len);
    with VerInfo^ do begin
      Result.HiVersion:=HiWord(dwDOC_PACKVersionMS);
      Result.LoVersion:=LoWord(dwDOC_PACKVersionMS);
      Result.Release:=HiWord(dwDOC_PACKVersionLS);
      Result.Build:=LoWord(dwDOC_PACKVersionLS);
    end;
    GlobalUnlock(hMem);
    GlobalFree(hMem);
  end else Result.HiVersion:=-1;
end;

function ComCtlVersion: TDVersion;
begin
  Result:=DOC_PACKVersion('COMCTL32.DLL');
end;

function IsDebug(const DOC_PACKName: TString): LongBool;
Var
// об'ява локальних типів даних
S: TString;
hMem: HGLOBAL;
Buf: pointer;
BufSize, Len, dwHandle: DWORD;
VerInfo: PVSFixedDOC_PACKInfo;
pszName: PAnsiChar;
begin
  Result:=False;
  FillChar(Result, SizeOf(Result), 0);
  S:=DOC_PACKName;
  CleanUp(S, True);
  if IsEmptyStr(S) then S:=ParamStr(0);
  pszName:=@S[1];
  BufSize:=GetDOC_PACKVersionInfoSize(pszName, dwHandle);
  if BufSize<>0 then begin
    hMem:=GlobalAlloc(GHND, BufSize);
    if hMem = 0 then OutOfMemoryError;
    Buf:=GlobalLock(hMem);
    if Buf=nil then OutOfMemoryError;
    GetDOC_PACKVersionInfo(pszName, dwHandle, BufSize, Buf);
    VerQueryValue(Buf, '\\', pointer(VerInfo), Len);
    Result:=(VerInfo.dwDOC_PACKFlags and VS_FF_DEBUG) <> 0;
    GlobalUnlock(hMem);
    GlobalFree(hMem);
  end;
end;
end;

```

```

var
  IsDebugValue: Integer = Integer($8000000);

function IsDebug: LongBool; overload;
begin
  if IsDebugValue = Integer ($80000000) then IsDebugValue:=Integer(IsDebug(''));
  Result:=LongBool(IsDebugValue);
end;

procedure GetWindowSize(Handle: HWND; var Size: TSize);
var
  R: TRect;
begin
  GetWindowRect(Handle, R);
  with R, Size do begin
    cx:=Right-Left;
    cy:=Bottom-Top;
  end;
end;

procedure PressKey(VKey: Byte);
begin
  keybd_event(VKey, 0, 0, 0);
  keybd_event(VKey, 0, KEYEVENTF_KEYUP, 0);
end;

function ForceDirectories(Dir: TString): Boolean;
begin
  try
    Result := True;
    if Length(Dir) = 0 then Abort;
    Dir := ExcludeTrailingBackslash(Dir);
    if (Length(Dir) < 3) or PathExists(Dir)
      or (ExtractDOC_PACKPath(Dir) = Dir) then Exit;
    Result := ForceDirectories(ExtractDOC_PACKPath(Dir)) and CreateDir(Dir);
  except
    on EAbort do Result:=False;
    else raise;
  end;
end;

function StringToVersion(const Str: TString): TDVersion;
Var
  // об'ява локальних типів даних
  S, SH, SL, SR, SB: TString;
  Code: Integer;
  Count: Integer;
begin
  S:=Str;
  CleanUp(S, True);
  S:=ReplaceChar(S, ',', '.');
  Count:=CharCount(S, '.')+1;
  SH:='0'; SL:='0'; SR:='0'; SB:='0';
  if Count>=1 then SH:=GetSubStr(S, 1, '.');
  if Count>=2 then SL:=GetSubStr(S, 2, '.');
  if Count>=3 then SR:=GetSubStr(S, 3, '.');
  if Count>=4 then SB:=GetSubStr(S, 4, '.');
  with Result do begin
    Val(SH, HiVersion, Code); if Code<>0 then HiVersion:=0;
    Val(SL, LoVersion, Code); if Code<>0 then LoVersion:=0;
    Val(SR, Release, Code); if Code<>0 then Release:=0;
    Val(SB, Build, Code); if Code<>0 then Build:=0;
  end;
end;
end;

```

```

function VersionToString(const Ver: TDVersion): TString;
begin
  with Ver do Result:=Format('%d.%d.%d.%d', [HiVersion, LoVersion, Release,
Build]);
end;

function Version(HiVersion, LoVersion: Integer;
  Release: Integer = 0; Build: Integer = 0): TDVersion; overload;
begin
  Result.HiVersion:=HiVersion;
  Result.LoVersion:=LoVersion;
  Result.Release:=Release;
  Result.Build:=Build;
end;

function LoadResStr(Instance:THandle;ID:Cardinal):TString;
begin
  SetLength(Result,512);
  LoadString(Instance, ID,@Result[1],512);
  CleanUp(Result);
end;

function LoadResStr(ID: Cardinal): TString; overload;
begin
  Result:=LoadResStr(hInstance, ID);
end;

function LoadDLL(const Path:TString):THandle;
begin
  Result:=LoadLibrary(PChar(Path));
end;

function GetDLLProc(Handle:THandle;const ProcName:TString):Pointer;
begin
  Result:=GetProcAddress(Handle, PChar(ProcName));
end;
Var
// об'ява локальних типів даних
OSVersionInfo_Initialized: Boolean = False;
OSVersionInfo: TOSVersionInfo;

procedure Initialize_OSVersionInfo;
begin
  if not OSVersionInfo_Initialized then begin
    GetVersionEx(OSVersionInfo);
    OSVersionInfo_Initialized:=True;
  end;
end;

function _Win32Platform: Integer;
begin
  Initialize_OSVersionInfo;
  Result:=OSVersionInfo.dwPlatformId;
end;

function _Win32MajorVersion: Integer;
begin
  Initialize_OSVersionInfo;
  Result:=OSVersionInfo.dwMajorVersion;
end;

function _Win32MinorVersion: Integer;
begin
  Initialize_OSVersionInfo;
  Result:=OSVersionInfo.dwMinorVersion;
end;

```

```
end;

function WinNT: Boolean;
begin
  Result := _Win32Platform = VER_PLATFORM_WIN32_NT;
end;

function Win2K: Boolean;
begin
  Result := (_Win32MajorVersion > 4) and (_Win32Platform =
VER_PLATFORM_WIN32_NT);
end;

end.
```

K6П3-2023

WINDATA.PAS - ФАЙЛ ФОРМИ

```

unit WinData;
// модуль WinData

interface

uses Windows, Messages, Classes, Controls, Forms, Graphics, StdCtrls,
    ExtCtrls, CommCtrl;

type

    TUkraineEBActionLink = class(TControlActionLink)
    protected
        FClient: TUkraineEB;
        procedure AssignClient(AClient: TObject); override;
        function IsCheckedLinked: Boolean; override;
        function IsGroupIndexLinked: Boolean; override;
        procedure SetGroupIndex(Value: Integer); override;
        procedure SetChecked(Value: Boolean); override;
    end;

    TUkraineEB = class(TGraphicControl)
    private
        FGroupIndex: Integer;
        FGlyph: Pointer;
        FDown: Boolean;
        FDragging: Boolean;
        FAllowAllUp: Boolean;
        FLayout: TButtonLayout;
        FSpacing: Integer;
        FTransparent: Boolean;
        FMargin: Integer;
        FFlat: Boolean;
        FMouseInControl: Boolean;
        procedure GlyphChanged(Sender: TObject);
        procedure UpdateExclusive;
        function GetGlyph: TBitmap;
        procedure SetGlyph(Value: TBitmap);
        function GetNumGlyphs: TNumGlyphs;
        procedure SetNumGlyphs(Value: TNumGlyphs);
        procedure SetDown(Value: Boolean);
        procedure SetLayout(Value: TButtonLayout);
        procedure SetSpacing(Value: Integer);
        procedure SetTransparent(Value: Boolean);
        procedure SetMargin(Value: Integer);
        procedure UpdateTracking;
    protected
        FState: TButtonState;
        procedure ActionChange(Sender: TObject; CheckDefaults: Boolean); override;
        function GetActionLinkClass: TControlActionLinkClass; override;
        function GetPalette: HPALETTE; override;
        procedure Loaded; override;
        procedure Paint; override;
        property MouseInControl: Boolean read FMouseInControl;
    public
        constructor Create(AOwner: TComponent); override;
        destructor Destroy; override;
        procedure Click; override;
    published
        property Action;
        property AllowAllUp: Boolean read FAllowAllUp write SetAllowAllUp default
            False;
        property Anchors;
        property BiDiMode;
        property Constraints;

```

```

property GroupIndex: Integer read FGroupIndex write SetGroupIndex default 0;
property Down: Boolean read FDown write SetDown default False;
property Caption;
property Enabled;
property Flat: Boolean read FFlat write SetFlat default False;
property Font;
property Glyph: TBitmap read GetGlyph write SetGlyph;
property ParentFont;
property ParentShowHint;
property ParentBiDiMode;
property PopupMenu;
property ShowHint;
property Spacing: Integer read FSpacing write SetSpacing default 4;
property Transparent: Boolean read FTransparent write SetTransparent default
True;
property Visible;
property OnClick;
property OnDblClick;
property OnMouseDown;
property OnMouseMove;
property OnMouseUp;
end;

TBitBtnKind = (bkCustom, bkOK, bkCancel, bkHelp, bkYes, bkNo, bkClose,
bkAbort, bkRetry, bkIgnore, bkAll);

TBitBtn = class(TButton)
private
    FCanvas: TCanvas;
    FGlyph: Pointer;
    FStyle: TButtonStyle;
    FKind: TBitBtnKind;
    FLayout: TButtonLayout;
    FSpacing: Integer;
    FMargin: Integer;
    IsFocused: Boolean;
    FModifiedGlyph: Boolean;
    FMouseInControl: Boolean;
    procedure DrawItem(const DrawItemStruct: TDrawItemStruct);
    procedure SetGlyph(Value: TBitmap);
    function GetGlyph: TBitmap;
    function GetNumGlyphs: TNumGlyphs;
    procedure SetNumGlyphs(Value: TNumGlyphs);
    procedure GlyphChanged(Sender: TObject);
    function IsCustom: Boolean;
    function IsCustomCaption: Boolean;
    procedure SetStyle(Value: TButtonStyle);
    procedure SetKind(Value: TBitBtnKind);
    function GetKind: TBitBtnKind;
    procedure SetLayout(Value: TButtonLayout);
    procedure SetSpacing(Value: Integer);
    procedure SetMargin(Value: Integer);
protected
    procedure ActionChange(Sender: TObject; CheckDefaults: Boolean); override;
    procedure CreateHandle; override;
    procedure CreateParams(var Params: TCreateParams); override;
    function GetPalette: HPALETTE; override;
    procedure SetButtonStyle(ADefault: Boolean); override;
public
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;
    procedure Click; override;
published
    property Action;
    property Anchors;

```

```

property BiDiMode;
property Cancel stored IsCustom;
property Caption stored IsCustomCaption;
property Constraints;
property Default stored IsCustom;
property Enabled;
property Glyph: TBitmap read GetGlyph write SetGlyph stored IsCustom;
property Kind: TBitBtnKind read GetKind write SetKind default bkCustom;
property Layout: TButtonLayout read FLayout write SetLayout default
blGlyphLeft;
property Margin: Integer read FMargin write SetMargin default -1;
property ModalResult stored IsCustom;
property NumGlyphs: TNumGlyphs read GetNumGlyphs write SetNumGlyphs stored
IsCustom default 1;
property ParentShowHint;
property ParentBiDiMode;
property ShowHint;
property Style: TButtonStyle read FStyle write SetStyle default
bsAutoDetect;
property Spacing: Integer read FSpacing write SetSpacing default 4;
property TabOrder;
property TabStop;
property Visible;
property OnEnter;
property OnExit;
end;

function DrawButtonFace(Canvas: TCanvas; const Client: TRect;
    BevelWidth: Integer; Style: TButtonStyle; IsRounded, IsDown,
    IsFocused: Boolean): TRect;

implementation

uses Consts, SysUtils, ActnList, ImgList, Themes;

var
    BitBtnGlyphs: array[TBitBtnKind] of TBitmap;

function GetBitBtnGlyph(Kind: TBitBtnKind): TBitmap;
begin
    if BitBtnGlyphs[Kind] = nil then
        begin
            BitBtnGlyphs[Kind]:=TBitmap.Create;
            BitBtnGlyphs[Kind].LoadFromResourceName(HInstance, BitBtnResNames[Kind]);
        end;
    Result:=BitBtnGlyphs[Kind];
end;

type
    TGlyphList = class(TImageList)
    private
        Used: TBits;
        FCount: Integer;
        function AllocateIndex: Integer;
    public
        constructor CreateSize(AWidth, AHeight: Integer);
        destructor Destroy; override;
        function AddMasked(Image: TBitmap; MaskColor: TColor): Integer;
        procedure Delete(Index: Integer);
        property Count: Integer read FCount;
    end;

    TGlyphCache = class

```

```

private
  GlyphLists: TList;
public
  constructor Create;
  destructor Destroy; override;
  function GetList(AWidth, AHeight: Integer): TGlyphList;
  procedure ReturnList(List: TGlyphList);
  function Empty: Boolean;
end;

TButtonGlyph = class
private
  FOriginal: TBitmap;
  FGlyphList: TGlyphList;
  FIndexes: array[TButtonState] of Integer;
  FTransparentColor: TColor;
  FNumGlyphs: TNumGlyphs;
  FOnChange: TNotifyEvent;
  procedure GlyphChanged(Sender: TObject);
  procedure SetGlyph(Value: TBitmap);
  procedure SetNumGlyphs(Value: TNumGlyphs);
  procedure Invalidate;
  function CreateButtonGlyph(State: TButtonState): Integer;
  procedure DrawButtonGlyph(Canvas: TCanvas; const GlyphPos: TPoint;
    State: TButtonState; Transparent: Boolean);
  procedure DrawButtonText(Canvas: TCanvas; const Caption: string;
    TextBounds: TRect; State: TButtonState; BiDiFlags: Longint);
  procedure CalcButtonLayout(Canvas: TCanvas; const Client: TRect;
    const Offset: TPoint; const Caption: string; Layout: TButtonLayout;
    Margin, Spacing: Integer; var GlyphPos: TPoint; var TextBounds: TRect;
    BiDiFlags: Longint);
public
  constructor Create;
  destructor Destroy; override;
  function Draw(Canvas: TCanvas; const Client: TRect; const Offset: TPoint;
    const Caption: string; Layout: TButtonLayout; Margin, Spacing: Integer;
    State: TButtonState; Transparent: Boolean; BiDiFlags: Longint): TRect;
  property Glyph: TBitmap read FOriginal write SetGlyph;
  property NumGlyphs: TNumGlyphs read FNumGlyphs write SetNumGlyphs;
  property OnChange: TNotifyEvent read FOnChange write FOnChange;
end;

constructor TGlyphList.CreateSize(AWidth, AHeight: Integer);
begin
  inherited CreateSize(AWidth, AHeight);
  Used:=TBits.Create;
end;

destructor TGlyphList.Destroy;
begin
  Used.Free;
  inherited Destroy;
end;

function TGlyphList.AllocateIndex: Integer;
begin
  Result:=Used.OpenBit;
  if Result >= Used.Size then
  begin
    Result:=inherited Add(nil, nil);
    Used.Size:=Result + 1;
  end;
  Used[Result]:=True;
end;

```

```

function TGlyphList.AddMasked(Image: TBitmap; MaskColor: TColor): Integer;
begin
  Result:=AllocateIndex;
  ReplaceMasked(Result, Image, MaskColor);
  Inc(FCount);
end;

procedure TGlyphList.Delete(Index: Integer);
begin
  if Used[Index] then
  begin
    Dec(FCount);
    Used[Index]:=False;
  end;
end;

constructor TGlyphCache.Create;
begin
  inherited Create;
  GlyphLists:=TList.Create;
end;

destructor TGlyphCache.Destroy;
begin
  GlyphLists.Free;
  inherited Destroy;
end;

function TGlyphCache.GetList(AWidth, AHeight: Integer): TGlyphList;
var
  I: Integer;
begin
  for I:=GlyphLists.Count - 1 downto 0 do
  begin
    Result:=GlyphLists[I];
    with Result do
      if (AWidth = Width) and (AHeight = Height) then Exit;
    end;
    Result:=TGlyphList.CreateSize(AWidth, AHeight);
    GlyphLists.Add(Result);
  end;
end;

procedure TGlyphCache.ReturnList(List: TGlyphList);
begin
  if List = nil then Exit;
  if List.Count = 0 then
  begin
    GlyphLists.Remove(List);
    List.Free;
  end;
end;

function TGlyphCache.Empty: Boolean;
begin
  Result:=GlyphLists.Count = 0;
end;

var
  GlyphCache: TGlyphCache = nil;
  ButtonCount: Integer = 0;

constructor TButtonGlyph.Create;
var
  I: TButtonState;
begin

```

```

    inherited Create;
    FOriginal:=TBitmap.Create;
    FOriginal.OnChange:=GlyphChanged;
    FTransparentColor:=clOlive;
    FNumGlyphs:=1;
    for I:=Low(I) to High(I) do
        FIndexs[I]:=-1;
    if GlyphCache = nil then GlyphCache:=TGlyphCache.Create;
end;

destructor TButtonGlyph.Destroy;
begin
    FOriginal.Free;
    Invalidate;
    if Assigned(GlyphCache) and GlyphCache.Empty then
        begin
            GlyphCache.Free;
            GlyphCache:=nil;
        end;
    inherited Destroy;
end;

procedure TButtonGlyph.Invalidate;
var
    I: TButtonState;
begin
    for I:=Low(I) to High(I) do
        begin
            if FIndexs[I] <> -1 then FGlyphList.Delete(FIndexs[I]);
            FIndexs[I]:=-1;
        end;
    GlyphCache.ReturnList(FGlyphList);
    FGlyphList:=nil;
end;

procedure TButtonGlyph.GlyphChanged(Sender: TObject);
begin
    if Sender = FOriginal then
        begin
            FTransparentColor:=FOriginal.TransparentColor;
            Invalidate;
            if Assigned(FOnChange) then FOnChange(Self);
        end;
end;

procedure TButtonGlyph.SetGlyph(Value: TBitmap);
var
    Glyphs: Integer;
begin
    Invalidate;
    FOriginal.Assign(Value);
    if (Value <> nil) and (Value.Height > 0) then
        begin
            FTransparentColor:=Value.TransparentColor;
            if Value.Width mod Value.Height = 0 then
                begin
                    Glyphs:=Value.Width div Value.Height;
                    if Glyphs > 4 then Glyphs:=1;
                    SetNumGlyphs(Glyphs);
                end;
        end;
end;

procedure TButtonGlyph.SetNumGlyphs(Value: TNumGlyphs);
begin

```

```

    if (Value <> FNumGlyphs) and (Value > 0) then
    begin
        Invalidate;
        FNumGlyphs:=Value;
        GlyphChanged(Glyph);
    end;
end;

function TButtonGlyph.CreateButtonGlyph(State: TButtonState): Integer;
const
    ROP_DSPDxax = $00E20746;
var
    TmpImage, DDB, MonoBmp: TBitmap;
    IWidth, IHeight: Integer;
    IRect, ORect: TRect;
    I: TButtonState;
    DestDC: HDC;
begin
    if (State = bsDown) and (NumGlyphs < 3) then State:=bsUp;
    Result:=FIndexs[State];
    if Result <> -1 then Exit;
    if (FOriginal.Width or FOriginal.Height) = 0 then Exit;
    IWidth:=FOriginal.Width div FNumGlyphs;
    IHeight:=FOriginal.Height;
    if FGlyphList = nil then
    begin
        if GlyphCache = nil then GlyphCache:=TGlyphCache.Create;
        FGlyphList:=GlyphCache.GetList(IWidth, IHeight);
    end;
    TmpImage:=TBitmap.Create;
    try
        TmpImage.Width:=IWidth;
        TmpImage.Height:=IHeight;
        IRect:=Rect(0, 0, IWidth, IHeight);
        TmpImage.Canvas.Brush.Color:=clBtnFace;
        TmpImage.Palette:=CopyPalette(FOriginal.Palette);
        I:=State;
        if Ord(I) >= NumGlyphs then I:=bsUp;
        ORect:=Rect(Ord(I) * IWidth, 0, (Ord(I) + 1) * IWidth, IHeight);
        case State of
        bsUp, bsDown,
        bsExclusive:
        begin
            TmpImage.Canvas.CopyRect(IRect, FOriginal.Canvas, ORect);
            if FOriginal.TransparentMode = tmFixed then
                FIndexs[State]:=FGlyphList.AddMasked(TmpImage, FTransparentColor)
            else
                FIndexs[State]:=FGlyphList.AddMasked(TmpImage, clDefault);
        end;
        bsDisabled:
        begin
            MonoBmp:=nil;
            DDB:=nil;
            try
                MonoBmp:=TBitmap.Create;
                DDB:=TBitmap.Create;
                DDB.Assign(FOriginal);
                DDB.HandleType:=bmDDB;
                if NumGlyphs > 1 then
                    with TmpImage.Canvas do
                    begin
                        CopyRect(IRect, DDB.Canvas, ORect);
                        MonoBmp.Monochrome:=True;
                        MonoBmp.Width:=IWidth;
                        MonoBmp.Height:=IHeight;
                    end;
                end;
            except
                MonoBmp:=nil;
                DDB:=nil;
            end;
        end;
    except
        TmpImage.Free;
    end;
end;

```

```

DDB.Canvas.Brush.Color:=clWhite;
MonoBmp.Canvas.CopyRect(IRect, DDB.Canvas, ORect);
Brush.Color:=clBtnHighlight;
DestDC:=Handle;
SetTextColor(DestDC, clBlack);
SetBkColor(DestDC, clWhite);
BitBlt(DestDC, 0, 0, IWidth, IHeight,
    MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
DDB.Canvas.Brush.Color:=clGray;
MonoBmp.Canvas.CopyRect(IRect, DDB.Canvas, ORect);
Brush.Color:=clBtnShadow;
DestDC:=Handle;
SetTextColor(DestDC, clBlack);
SetBkColor(DestDC, clWhite);
BitBlt(DestDC, 0, 0, IWidth, IHeight,
    MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
DDB.Canvas.Brush.Color:=ColorToRGB(FTransparentColor);
MonoBmp.Canvas.CopyRect(IRect, DDB.Canvas, ORect);
Brush.Color:=clBtnFace;
DestDC:=Handle;
SetTextColor(DestDC, clBlack);
SetBkColor(DestDC, clWhite);
BitBlt(DestDC, 0, 0, IWidth, IHeight,
    MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
end
else
begin
with MonoBmp do
begin
Assign(FOriginal);
HandleType:=bmDDB;
Canvas.Brush.Color:=clBlack;
Width:=IWidth;
if Monochrome then
begin
Canvas.Font.Color:=clWhite;
Monochrome:=False;
Canvas.Brush.Color:=clWhite;
end;
Monochrome:=True;
end;
with TmpImage.Canvas do
begin
Brush.Color:=clBtnFace;
FillRect(IRect);
Brush.Color:=clBtnHighlight;
SetTextColor(Handle, clBlack);
SetBkColor(Handle, clWhite);
BitBlt(Handle, 1, 1, IWidth, IHeight,
    MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
Brush.Color:=clBtnShadow;
SetTextColor(Handle, clBlack);
SetBkColor(Handle, clWhite);
BitBlt(Handle, 0, 0, IWidth, IHeight,
    MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
end;
end;
finally
DDB.Free;
MonoBmp.Free;
end;
FIndexes[State]:=FGlyphList.AddMasked(TmpImage, clDefault);
end;
end;
finally

```

```

TmpImage.Free;
end;
Result:=FIndexs[State];
end;

procedure TButtonGlyph.DrawButtonGlyph(Canvas: TCanvas; const GlyphPos: TPoint;
State: TButtonState; Transparent: Boolean);
var
  Index: Integer;
  Details: TThemedElementDetails;
  R: TRect;
  Button: TThemedButton;
begin
  if FOriginal = nil then Exit;
  if (FOriginal.Width = 0) or (FOriginal.Height = 0) then Exit;
  Index:=CreateButtonGlyph(State);
  with GlyphPos do
  begin
    if ThemeServices.ThemesEnabled then
    begin
      R.TopLeft:=GlyphPos;
      R.Right:=R.Left + FOriginal.Width div FNumGlyphs;
      R.Bottom:=R.Top + FOriginal.Height;
      case State of
        bsDisabled:
          Button:=tbPushButtonDisabled;
        bsDown,
        bsExclusive:
          Button:=tbPushButtonPressed;
      else
        // bsUp
        Button:=tbPushButtonNormal;
      end;
      Details:=ThemeServices.GetElementDetails(Button);
      ThemeServices.DrawIcon(Canvas.Handle, Details, R, FGlyphList.Handle,
Index);
    end
    else
      if Transparent or (State = bsExclusive) then
      begin
        ImageList_DrawEx(FGlyphList.Handle, Index, Canvas.Handle, X, Y, 0, 0,
          clNone, clNone, ILD_Transparent)
      end
      else
        ImageList_DrawEx(FGlyphList.Handle, Index, Canvas.Handle, X, Y, 0, 0,
          ColorToRGB(clBtnFace), clNone, ILD_Normal);
      end;
    end;
  end;

procedure TButtonGlyph.DrawButtonText(Canvas: TCanvas; const Caption: string;
TextBounds: TRect; State: TButtonState; BiDiFlags: LongInt);
begin
  with Canvas do
  begin
    Brush.Style:=bsClear;
    if State = bsDisabled then
    begin
      OffsetRect(TextBounds, 1, 1);
      Font.Color:=clBtnHighlight;
      DrawText(Handle, PChar(Caption), Length(Caption), TextBounds,
        DT_CENTER or DT_VCENTER or BiDiFlags);
      OffsetRect(TextBounds, -1, -1);
      Font.Color:=clBtnShadow;
      DrawText(Handle, PChar(Caption), Length(Caption), TextBounds,
        DT_CENTER or DT_VCENTER or BiDiFlags);
    end;
  end;
end;

```

```

    end else
        DrawText(Handle, PChar(Caption), Length(Caption), TextBounds,
            DT_CENTER or DT_VCENTER or BiDiFlags);
    end;
end;

function TButtonGlyph.Draw(Canvas: TCanvas; const Client: TRect;
    const Offset: TPoint; const Caption: string; Layout: TButtonLayout;
    Margin, Spacing: Integer; State: TButtonState; Transparent: Boolean;
    BiDiFlags: LongInt): TRect;
var
    GlyphPos: TPoint;
begin
    CalcButtonLayout(Canvas, Client, Offset, Caption, Layout, Margin, Spacing,
        GlyphPos, Result, BiDiFlags);
    DrawButtonGlyph(Canvas, GlyphPos, State, Transparent);
    DrawButtonText(Canvas, Caption, Result, State, BiDiFlags);
end;

procedure TUkraineEBActionLink.AssignClient(AClient: TObject);
begin
    inherited AssignClient(AClient);
    FClient:=AClient as TUkraineEB;
end;

function TUkraineEBActionLink.IsCheckedLinked: Boolean;
begin
    Result:=inherited IsCheckedLinked and (FClient.GroupIndex <> 0) and
        FClient.AllowAllUp and (FClient.Down = (Action as TCustomAction).Checked);
end;

function TUkraineEBActionLink.IsGroupIndexLinked: Boolean;
begin
    Result:=(FClient is TUkraineEB) and
        (TUkraineEB(FClient).GroupIndex = (Action as TCustomAction).GroupIndex);
end;

procedure TUkraineEBActionLink.SetChecked(Value: Boolean);
begin
    if IsCheckedLinked then TUkraineEB(FClient).Down:=Value;
end;

procedure TUkraineEBActionLink.SetGroupIndex(Value: Integer);
begin
    if IsGroupIndexLinked then TUkraineEB(FClient).GroupIndex:=Value;
end;

constructor TUkraineEB.Create(AOwner: TComponent);
begin
    FGlyph:=TButtonGlyph.Create;
    TButtonGlyph(FGlyph).OnChange:=GlyphChanged;
    inherited Create(AOwner);
    SetBounds(0, 0, 23, 22);
    ControlStyle:=[csCaptureMouse, csDoubleClicks];
    ParentFont:=True;
    Color:=clBtnFace;
    FSpacing:=4;
    FMargin:=-1;
    FLayout:=blGlyphLeft;
    FTransparent:=True;
    Inc(ButtonCount);
end;

destructor TUkraineEB.Destroy;
begin

```

```

    Dec(ButtonCount);
    inherited Destroy;
    TButtonGlyph(FGlyph).Free;
end;

procedure TUKraineEB.Paint;
const
    DownStyles: array[Boolean] of Integer = (BDR_RAISEDINNER, BDR_SUNKENOUTER);
    FillStyles: array[Boolean] of Integer = (BF_MIDDLE, 0);
var
    PaintRect: TRect;
    DrawFlags: Integer;
    Offset: TPoint;
    Button: TThemedButton;
    ToolButton: TThemedToolBar;
    Details: TThemedElementDetails;
begin
    if not Enabled then
        begin
            FState:=bsDisabled;
            FDragging:=False;
        end
    else if FState = bsDisabled then
        if FDown and (GroupIndex <> 0) then
            FState:=bsExclusive
        else
            FState:=bsUp;
        Canvas.Font:=Self.Font;

        if ThemeServices.ThemesEnabled then
            begin
                PerformEraseBackground(Self, Canvas.Handle);

                if not Enabled then
                    Button:=tbPushButtonDisabled
                else
                    if FState in [bsDown, bsExclusive] then
                        Button:=tbPushButtonPressed
                    else
                        if MouseInControl then
                            Button:=tbPushButtonHot
                        else
                            Button:=tbPushButtonNormal;

                ToolButton:=ttbToolBarDontCare;
                if FFlat then
                    begin
                        case Button of
                            tbPushButtonDisabled:
                                Toolbutton:=ttbButtonDisabled;
                            tbPushButtonPressed:
                                Toolbutton:=ttbButtonPressed;
                            tbPushButtonHot:
                                Toolbutton:=ttbButtonHot;
                            tbPushButtonNormal:
                                Toolbutton:=ttbButtonNormal;
                        end;
                    end;

                PaintRect:=ClientRect;
                if ToolButton = ttbToolBarDontCare then
                    begin
                        Details:=ThemeServices.GetElementDetails(Button);
                        ThemeServices.DrawElement(Canvas.Handle, Details, PaintRect);
                        PaintRect:=ThemeServices.ContentRect(Canvas.Handle, Details, PaintRect);
                    end;
                end;
            end;

```

```

end
else
begin
  Details:=ThemeServices.GetElementDetails(ToolButton);
  ThemeServices.DrawElement(Canvas.Handle, Details, PaintRect);
  PaintRect:=ThemeServices.ContentRect(Canvas.Handle, Details, PaintRect);
end;

if Button = tbPushButtonPressed then
begin
  if ToolButton <> ttbToolBarDontCare then
    Canvas.Font.Color:=clHighlightText;
    Offset:=Point(1, 0);
  end
  else
    Offset:=Point(0, 0);
    TButtonGlyph(FGlyph).Draw(Canvas, PaintRect, Offset, Caption, FLayout,
  FMargin, FSpacing, FState, Transparent,
    DrawTextBiDiModeFlags(0));
  end
  else
  begin
    PaintRect:=Rect(0, 0, Width, Height);
    if not FFlat then
      begin
        DrawFlags:=DFCS_BUTTONPUSH or DFCS_ADJUSTRECT;
        if FState in [bsDown, bsExclusive] then
          DrawFlags:=DrawFlags or DFCS_PUSHED;
          DrawFrameControl(Canvas.Handle, PaintRect, DFC_BUTTON, DrawFlags);
        end
        else
          begin
            if (FState in [bsDown, bsExclusive]) or
              (FMouseInControl and (FState <> bsDisabled)) or
              (csDesigning in ComponentState) then
              DrawEdge(Canvas.Handle, PaintRect, DownStyles[FState in [bsDown,
bsExclusive]],
                FillStyles[Transparent] or BF_RECT)
            else if not Transparent then
              begin
                Canvas.Brush.Color:=Color;
                Canvas.FillRect(PaintRect);
              end;
              InflateRect(PaintRect, -1, -1);
            end;
            if FState in [bsDown, bsExclusive] then
              begin
                if (FState = bsExclusive) and (not FFlat or not FMouseInControl) then
                  begin
                    Canvas.Brush.Bitmap:=AllocPatternBitmap(clBtnFace, clBtnHighlight);
                    Canvas.FillRect(PaintRect);
                  end;
                  Offset.X:=1;
                  Offset.Y:=1;
                end
                else
                  begin
                    Offset.X:=0;
                    Offset.Y:=0;
                  end;
                TButtonGlyph(FGlyph).Draw(Canvas, PaintRect, Offset, Caption, FLayout,
  FMargin,
    FSpacing, FState, Transparent, DrawTextBiDiModeFlags(0));
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;
end;

```

```

procedure TUkraineEB.UpdateTracking;
var
  P: TPoint;
begin
  if FFlat then
  begin
    if Enabled then
    begin
      GetCursorPos(P);
      FMouseInControl:=not (FindDragTarget(P, True) = Self);
      if FMouseInControl then
        Perform(CM_MOUSELEAVE, 0, 0)
      else
        Perform(CM_MOUSEENTER, 0, 0);
    end;
  end;
end;

procedure TUkraineEB.Loaded;
var
  State: TButtonState;
begin
  inherited Loaded;
  if Enabled then
    State:=bsUp
  else
    State:=bsDisabled;
  TButtonGlyph(FGlyph).CreateButtonGlyph(State);
end;

procedure TUkraineEB.MouseDown(Button: TMouseButton; Shift: TShiftState;
  X, Y: Integer);
begin
  inherited MouseDown(Button, Shift, X, Y);
  if (Button = mbLeft) and Enabled then
  begin
    if not FDown then
    begin
      FState:=bsDown;
      Invalidate;
    end;
    FDragging:=True;
  end;
end;

procedure TUkraineEB.MouseMove(Shift: TShiftState; X, Y: Integer);
var
  NewState: TButtonState;
begin
  inherited MouseMove(Shift, X, Y);
  if FDragging then
  begin
    if not FDown then NewState:=bsUp
    else NewState:=bsExclusive;
    if (X >= 0) and (X < ClientWidth) and (Y >= 0) and (Y <= ClientHeight) then
      if FDown then NewState:=bsExclusive else NewState:=bsDown;
    if NewState <> FState then
    begin
      FState:=NewState;
      Invalidate;
    end;
  end
  else if not FMouseInControl then
    UpdateTracking;
end;

```

```

end;

procedure TUkraineEB.MouseUp(Button: TMouseButton; Shift: TShiftState;
X, Y: Integer);
var
  DoClick: Boolean;
begin
  inherited MouseUp(Button, Shift, X, Y);
  if FDragging then
  begin
    FDragging:=False;
    DoClick:=(X >= 0) and (X < ClientWidth) and (Y >= 0) and (Y <= ClientHeight);
    if FGroupIndex = 0 then
    begin
      FState:=bsUp;
      FMouseInControl:=False;
      if DoClick and not (FState in [bsExclusive, bsDown]) then
        Invalidate;
    end
  else
    if DoClick then
    begin
      SetDown(not FDown);
      if FDown then Repaint;
    end
  else
    begin
      if FDown then FState:=bsExclusive;
      Repaint;
    end;
    if DoClick then Click;
    UpdateTracking;
  end;
end;

procedure TUkraineEB.Click;
begin
  inherited Click;
end;

function TUkraineEB.GetPalette: HPALETTE;
begin
  Result:=Glyph.Palette;
end;

function TUkraineEB.GetActionLinkClass: TControlActionLinkClass;
begin
  Result:=TUkraineEBActionLink;
end;

function TUkraineEB.GetGlyph: TBitmap;
begin
  Result:=TButtonGlyph(FGlyph).Glyph;
end;

procedure TUkraineEB.SetGlyph(Value: TBitmap);
begin
  TButtonGlyph(FGlyph).Glyph:=Value;
  Invalidate;
end;

function TUkraineEB.GetNumGlyphs: TNumGlyphs;
begin
  Result:=TButtonGlyph(FGlyph).NumGlyphs;
end;

```

```

procedure TUkraineEB.SetNumGlyphs(Value: TNumGlyphs);
begin
  if Value < 0 then Value:=1
  else if Value > 4 then Value:=4;
  if Value <> TButtonGlyph(FGlyph).NumGlyphs then
  begin
    TButtonGlyph(FGlyph).NumGlyphs:=Value;
    Invalidate;
  end;
end;

procedure TUkraineEB.GlyphChanged(Sender: TObject);
begin
  Invalidate;
end;

procedure TUkraineEB.UpdateExclusive;
var
  Msg: TMessage;
begin
  if (FGroupIndex <> 0) and (Parent <> nil) then
  begin
    Msg.Msg:=CM_BUTTONPRESSED;
    Msg.WParam:=FGroupIndex;
    Msg.LParam:=Longint(Self);
    Msg.Result:=0;
    Parent.Broadcast(Msg);
  end;
end;

procedure TUkraineEB.SetDown(Value: Boolean);
begin
  if FGroupIndex = 0 then Value:=False;
  if Value <> FDown then
  begin
    if FDown and (not FAllowAllUp) then Exit;
    FDown:=Value;
    if Value then
    begin
      if FState = bsUp then Invalidate;
      FState:=bsExclusive;
    end
    else
    begin
      FState:=bsUp;
      Repaint;
    end;
    if Value then UpdateExclusive;
  end;
end;

procedure TUkraineEB.SetFlat(Value: Boolean);
begin
  if Value <> FFlat then
  begin
    FFlat:=Value;
    Invalidate;
  end;
end;

procedure TUkraineEB.SetGroupIndex(Value: Integer);
begin
  if FGroupIndex <> Value then
  begin

```

```

    FGroupIndex:=Value;
    UpdateExclusive;
end;
end;

procedure TUkraineEB.SetLayout(Value: TButtonLayout);
begin
    if FLayout <> Value then
        begin
            FLayout:=Value;
            Invalidate;
        end;
end;

procedure TUkraineEB.SetMargin(Value: Integer);
begin
    if (Value <> FMargin) and (Value >= -1) then
        begin
            FMargin:=Value;
            Invalidate;
        end;
end;

procedure TUkraineEB.SetSpacing(Value: Integer);
begin
    if Value <> FSpacing then
        begin
            FSpacing:=Value;
            Invalidate;
        end;
end;

procedure TUkraineEB.SetTransparent(Value: Boolean);
begin
    if Value <> FTransparent then
        begin
            FTransparent:=Value;
            if Value then
                ControlStyle:=ControlStyle - [csOpaque] else
                ControlStyle:=ControlStyle + [csOpaque];
            Invalidate;
        end;
end;

procedure TUkraineEB.SetAllowAllUp(Value: Boolean);
begin
    if FAllowAllUp <> Value then
        begin
            FAllowAllUp:=Value;
            UpdateExclusive;
        end;
end;

procedure TUkraineEB.WMLButtonDblClk(var Message: TWMLButtonDown);
begin
    inherited;
    if FDown then DblClick;
end;

procedure TUkraineEB.CMEnabledChanged(var Message: TMessage);
const
    NewState: array[Boolean] of TButtonState = (bsDisabled, bsUp);
begin
    TButtonGlyph(FGlyph).CreateButtonGlyph(NewState[Enabled]);
    UpdateTracking;
end;

```

```

    Repaint;
end;

procedure TUkraineEB.CMButtonPressed(var Message: TMessage);
var
    Sender: TUkraineEB;
begin
    if Message.WParam = FGroupIndex then
        begin
            Sender:=TUkraineEB(Message.LParam);
            if Sender <> Self then
                begin
                    if Sender.Down and FDown then
                        begin
                            FDown:=False;
                            FState:=bsUp;
                            if (Action is TCustomAction) then
                                TCustomAction(Action).Checked:=False;
                            Invalidate;
                        end;
                    FAllowAllUp:=Sender.AllowAllUp;
                end;
            end;
        end;
end;

procedure TUkraineEB.CMDialogChar(var Message: TCMDialogChar);
begin
    with Message do
        if IsAccel(CharCode, Caption) and Enabled and Visible and
            (Parent <> nil) and Parent.Showing then
            begin
                Click;
                Result:=1;
            end else
                inherited;
end;

procedure TUkraineEB.CMFontChanged(var Message: TMessage);
begin
    Invalidate;
end;

procedure TUkraineEB.CMTextChanged(var Message: TMessage);
begin
    Invalidate;
end;

procedure TUkraineEB.CMSysColorChange(var Message: TMessage);
begin
    with TButtonGlyph(FGlyph) do
        begin
            Invalidate;
            CreateButtonGlyph(FState);
        end;
end;

procedure TUkraineEB.CMMouseLeave(var Message: TMessage);
var
    NeedRepaint: Boolean;
begin
    inherited;
    NeedRepaint:=FFlat and FMouseInControl and Enabled and not FDragging;
    if NeedRepaint or ThemeServices.ThemesEnabled then
        begin
            FMouseInControl:=False;

```

```

        if Enabled then
            Repaint;
        end;
    end;
end;

procedure TUKraineEB.ActionChange(Sender: TObject; CheckDefaults: Boolean);

    procedure CopyImage(ImageList: TCustomImageList; Index: Integer);
    begin
        with Glyph do
            begin
                Width:=ImageList.Width;
                Height:=ImageList.Height;
                Canvas.Brush.Color:=clFuchsia;
                Canvas.FillRect(Rect(0,0, Width, Height));
                ImageList.Draw(Canvas, 0, 0, Index);
            end;
        end;
    end;

begin
    inherited ActionChange(Sender, CheckDefaults);
    if Sender is TCustomAction then
        with TCustomAction(Sender) do
            begin
                if CheckDefaults or (Self.GroupIndex = 0) then
                    Self.GroupIndex:=GroupIndex;
                if (Glyph.Empty) and (ActionList <> nil) and (ActionList.Images <> nil)
                    and
                    (ImageIndex >= 0) and (ImageIndex < ActionList.Images.Count) then
                        CopyImage(ActionList.Images, ImageIndex);
            end;
        end;
    end;

constructor TBitBtn.Create(AOwner: TComponent);
begin
    FGlyph:=TButtonGlyph.Create;
    TButtonGlyph(FGlyph).OnChange:=GlyphChanged;
    inherited Create(AOwner);
    FCanvas:=TCanvas.Create;
    FStyle:=bsAutoDetect;
    FKind:=bkCustom;
    FLayout:=blGlyphLeft;
    FSpacing:=4;
    FMargin:=-1;
    ControlStyle:=ControlStyle + [csReflector];
    DoubleBuffered:=True;
end;

destructor TBitBtn.Destroy;
begin
    inherited Destroy;
    TButtonGlyph(FGlyph).Free;
    FCanvas.Free;
end;

procedure TBitBtn.CreateHandle;
var
    State: TButtonState;
begin
    if Enabled then
        State:=bsUp
    else
        State:=bsDisabled;
    inherited CreateHandle;
    TButtonGlyph(FGlyph).CreateButtonGlyph(State);
end;

```

```

end;

procedure TBitBtn.CreateParams (var Params: TCreateParams);
begin
  inherited CreateParams (Params);
  with Params do Style:=Style or BS_OWNERDRAW;
end;

procedure TBitBtn.SetButtonStyle (ADefault: Boolean);
begin
  if ADefault <> IsFocused then
  begin
    IsFocused:=ADefault;
    Refresh;
  end;
end;

procedure TBitBtn.Click;
var
  Form: TCustomForm;
  Control: TWinControl;
begin
  case FKind of
    bkClose:
      begin
        Form:=GetParentForm (Self);
        if Form <> nil then Form.Close
        else inherited Click;
      end;
    bkHelp:
      begin
        Control:=Self;
        while (Control <> nil) and (Control.HelpContext = 0) do
          Control:=Control.Parent;
        if Control <> nil then Application.HelpContext (Control.HelpContext)
        else inherited Click;
      end;
    else
      inherited Click;
  end;
end;

procedure TBitBtn.CNMeasureItem (var Message: TWMMMeasureItem);
begin
  with Message.MeasureItemStruct^ do
  begin
    itemWidth:=Width;
    itemHeight:=Height;
  end;
end;

procedure TBitBtn.CNDrawItem (var Message: TWMDrawItem);
begin
  DrawItem (Message.DrawItemStruct^);
end;

procedure TBitBtn.CMFontChanged (var Message: TMessage);
begin
  inherited;
  Invalidate;
end;

procedure TBitBtn.CMEnabledChanged (var Message: TMessage);
begin
  inherited;

```

```

    Invalidate;
end;

procedure TBitBtn.WMLButtonDblClk(var Message: TWMLButtonDblClk);
begin
    Perform(WM_LBUTTONDOWN, Message.Keys, Longint(Message.Pos));
end;

function TBitBtn.GetPalette: HPALETTE;
begin
    Result:=Glyph.Palette;
end;

procedure TBitBtn.SetGlyph(Value: TBitmap);
begin
    TButtonGlyph(FGlyph).Glyph:=Value as TBitmap;
    FModifiedGlyph:=True;
    Invalidate;
end;

function TBitBtn.GetGlyph: TBitmap;
begin
    Result:=TButtonGlyph(FGlyph).Glyph;
end;

procedure TBitBtn.GlyphChanged(Sender: TObject);
begin
    Invalidate;
end;

function TBitBtn.IsCustom: Boolean;
begin
    Result:=Kind = bkCustom;
end;

procedure TBitBtn.SetStyle(Value: TButtonStyle);
begin
    if Value <> FStyle then
    begin
        FStyle:=Value;
        Invalidate;
    end;
end;

procedure TBitBtn.SetKind(Value: TBitBtnKind);
begin
    if Value <> FKind then
    begin
        if Value <> bkCustom then
        begin
            Default:=Value in [bkOK, bkYes];
            Cancel:=Value in [bkCancel, bkNo];

            if ((csLoading in ComponentState) and (Caption = '')) or
                (not (csLoading in ComponentState)) then
            begin
                if BitBtnCaptions[Value] <> nil then
                    Caption:=LoadResString(BitBtnCaptions[Value]);
            end;

            ModalResult:=BitBtnModalResults[Value];
            TButtonGlyph(FGlyph).Glyph:=GetBitBtnGlyph(Value);
            NumGlyphs:=2;
            FModifiedGlyph:=False;
        end;
    end;
end;

```

```

    FKind:=Value;
    Invalidate;
end;
end;

function TBitBtn.IsCustomCaption: Boolean;
begin
    Result:=AnsiCompareStr(Caption, LoadResString(BitBtnCaptions[FKind])) <> 0;
end;

function TBitBtn.GetKind: TBitBtnKind;
begin
    if FKind <> bkCustom then
        if ((FKind in [bkOK, bkYes]) xor Default) or
            ((FKind in [bkCancel, bkNo]) xor Cancel) or
            (ModalResult <> BitBtnModalResults[FKind]) or
            FModifiedGlyph then
            FKind:=bkCustom;
        Result:=FKind;
    end;
end;

procedure TBitBtn.SetLayout(Value: TButtonLayout);
begin
    if FLayout <> Value then
        begin
            FLayout:=Value;
            Invalidate;
        end;
end;

function TBitBtn.GetNumGlyphs: TNumGlyphs;
begin
    Result:=TButtonGlyph(FGlyph).NumGlyphs;
end;

procedure TBitBtn.SetNumGlyphs(Value: TNumGlyphs);
begin
    if Value < 0 then Value:=1
    else if Value > 4 then Value:=4;
    if Value <> TButtonGlyph(FGlyph).NumGlyphs then
        begin
            TButtonGlyph(FGlyph).NumGlyphs:=Value;
            Invalidate;
        end;
end;

procedure TBitBtn.SetSpacing(Value: Integer);
begin
    if FSpacing <> Value then
        begin
            FSpacing:=Value;
            Invalidate;
        end;
end;

procedure TBitBtn.SetMargin(Value: Integer);
begin
    if (Value <> FMargin) and (Value >= - 1) then
        begin
            FMargin:=Value;
            Invalidate;
        end;
end;

procedure DestroyLocals; far;

```

```
var
  I: TBitBtnKind;
begin
  for I:=Low(TBitBtnKind) to High(TBitBtnKind) do
    BitBtnGlyphs[I].Free;
end;

procedure TBitBtn.CMMouseEnter(var Message: TMessage);
begin
  inherited;
  if ThemeServices.ThemesEnabled and not FMouseInControl and not (csDesigning in
ComponentState) then
  begin
    FMouseInControl:=True;
    Repaint;
  end;
end;

procedure TBitBtn.CMMouseLeave(var Message: TMessage);
begin
  inherited;
  if ThemeServices.ThemesEnabled and FMouseInControl then
  begin
    FMouseInControl:=False;
    Repaint;
  end;
end;

end.
```