

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 оновлення
цифрових топографічних карт геоінформаційної хмари”

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

Керівник проекту
кандидат технічних наук, доцент
_____ Минайленко Р.М.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 "Інформаційні технології"
Спеціальність 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

Шепелю Артему Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари*

2. Керівник роботи *Минайленко Роман Миколайович, канд. 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 оновлення цифрових топографічних карт геоінформаційної хмари*

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

Шепель А.С. Дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари. 122 Комп'ютерні науки. Центральнoукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи оновлення цифрових топографічних карт геоінформаційної хмари.

Метою розробки є дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари.

Об'єктом дослідження є процес оновлення цифрових топографічних карт геоінформаційної хмари.

Предметом дослідження є методи оновлення цифрових топографічних карт геоінформаційної хмари.

Методи дослідження базуються на методах теорії побудови геоінформаційних систем, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари.

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

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

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

Програму розроблено в середовищі Delphi 10.4 Sydney.

Ключові слова: комп'ютерні науки, геоінформаційна хмара

ABSTRACT

Shepel A.S. Research and software implementation of the system for updating digital topographic maps of the geoinformation cloud. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of updating digital topographic maps of the geoinformation cloud.

The purpose of the development is the research and software implementation of the system for updating digital topographic maps of the geoinformation cloud.

The object of research is the process of updating digital topographic maps of the geoinformation cloud.

The subject of research is methods of updating digital topographic maps of the geoinformation cloud.

Research methods are based on methods of the theory of building geoinformation systems, methods of mathematical statistics, and methods of software development.

The result of the work is the software implementation of the system for updating digital topographic maps of the geoinformation cloud.

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

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

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

The program was developed in the Delphi 10.4 Sydney environment.

Keywords: computer science, geoinformation cloud

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	15
2.3 Розгорнута постановка завдання	21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми	30
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	42
4.2 Захист розробленого програмного забезпечення.....	52
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	54
6 НАУКОВА НОВИЗНА	56

					ВКРМ-122.23.0020.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари	Літ.	Аркуш	Аркушів
Розроб.	Шепель А.С.					М	1	96
Перев.	Минайленко Р.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КН-22М-2		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	57
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	57
7.2 Розрахунок трудомісткості розробки програмної продукції.....	59
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	60
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	65
7.5 Визначення собівартості розробки та ціни програмної продукції.....	69
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	73
7.7 Визначення експлуатаційних витрат.....	73
7.8 Визначення економічної ефективності програмної продукції.....	75
7.9 Висновок.....	77
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	78
8.1 Вступ.....	78
8.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	79
8.3 Розробка заходів з умов поліпшення охорони праці.....	81
8.4 Техніка безпеки та протипожежна профілактика.....	82
8.5 Розрахункова частина	84
9 ОСНОВНІ ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ГІС	–	геоінформаційні системи
ГНСС	–	глобальні навігаційні супутникові системи
ЕОМ	–	електронно-обчислювальна машина
ІАСВК	–	інтегрована апаратура супутникового визначення координат
САПР	–	системи автоматизованого проектування
ТК	–	топографічні карти
ТП і К	–	топографічні плани і карти
ЦФС	–	цифрові фотограмметричні станції

КБПЗ – 2023

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. У теперішній час в Україні у зв'язку із глобальними змінами, що відбуваються на всій її території, створення й відновлення топографічних планів і карт стало актуальним завданням в області геодезії й картографії. Сучасний розвиток інформаційних технологій в області геодезії, картографії й інтеграція із суміжними галузями науки заклали нові основи в принципи одержання, обробки й зберігання просторової інформації. В останні роки ринок геоінформаційних систем виходить на лідируючі позиції. Всі ГІС системи базуються на інформаційних технологіях створення, обробки й комплексного аналізу складноструктурованої цифрової картографо-геодезичної інформації. Одним з важливих питань у створенні й відновленні топографічних планів і карт (далі ТП і К) є вибір ефективної методики й технології створення й відновлення ТП і К з застосуванням новітніх інтегрованих комплексів, які привели б до швидкого, абсолютно новому виду збору і якості інформації, зниженню вартості й скорочення строків виконання робіт.

Традиційні технології створення й відновлення, такі як векторизація паперових планів і карт на ЕОМ, не є закінченими рішеннями при створенні й відновленні ТП і К. Таке різноманіття підходів, способів і методів створення й відновлення планів і карт не дозволяє говорити про повне, вичерпне рішення цього завдання.

Для міських територій плани масштабу 1:2000 є основними графічними документами, на основі яких виконуються роботи, пов'язані із землекористуванням, кадастром, містобудуванням, визначенням прав власності й ін. Чим більш інтенсивно ведеться нове будівництво й реконструкція забудованих територій, тим актуальніше стає питання підтримки на належному рівні зміст ТП і К и зручності їхнього використання. Паралельно із цим минулого сформовані тверді, сучасні вимоги до точності й правил збору,

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

зберігання й відновлення просторової інформації. Перехід від традиційних методів на сучасні технології з використанням методів і засобів космічної геодезії привело до розвитку й широкого застосування інтегрованих глобальних навігаційних супутникових систем (ГНСС). Поряд із цим, останнім часом відбулися кардинальні зміни в поняттях і вартості всієї апаратури ГНСС по видах і класам точності. Природно, таке положення справ вимагало доробки й впроваджуваних нових технологій.

Зміст магістерського проекту присвячено реалізації технології створення й відновлення ТП і К з застосуванням новітньої інтегрованої апаратури супутникового визначення координат (ІАСВК) і ГІС технологій з метою одержання ефективного й економічно вигідного рішення даного завдання. Цим і визначається актуальність теми магістерського проекту.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари.

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

- Огляд існуючих систем оновлення цифрових топографічних карт геоінформаційної хмари.
- Дослідження системи оновлення цифрових топографічних карт геоінформаційної хмари.
- Програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари.

Об'єктом дослідження є процес оновлення цифрових топографічних карт геоінформаційної хмари.

Предметом дослідження є методи оновлення цифрових топографічних карт геоінформаційної хмари.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Методи дослідження базуються на методах теорії побудови геоінформаційних систем, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод оновлення цифрових топографічних карт геоінформаційної хмари.

– Розроблено вітчизняний продукт оновлення цифрових топографічних карт геоінформаційної хмари, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі оновлення цифрових топографічних карт геоінформаційної хмари.

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 оновлення цифрових топографічних карт геоінформаційної хмари, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Призначенням системи, що розробляється в ході виконання магістерського проекту, є реалізація технології по створенню й відновленню топографічних карт і планів із застосуванням ІАСВК і ГІС технологій. Дослідження полягає в реалізації методів збору просторової інформації, з обліком сучасних нормативно-технічних вимог до виконання робіт з топографічної зйомки із застосуванням апаратури ГНСС.

Для досягнення цієї мети в магістерському проекті були проведені наступні дослідження.

– виконано аналіз методів проведення робіт з топографічної зйомки, апаратних і програмно-інформаційних засобів для створення й відновлення топографічних планів і карт великих масштабів;

– досліджено технологію проведення топографічної зйомки для створення й відновлення топографічних планів і карт на основі інтегрованої апаратури супутникового визначення координат (ІАСВК) з використанням цифрових зображень карт і знімків;

– розроблено комплекс інформаційно-програмних засобів, що дозволяє використовувати ГІС-технології для обробки даних польових вимірів;

– виконано аналіз економічної ефективності від впровадження запропонованої технології у виробництві.

1.2 Область застосування

Областю застосування системи, яка розробляється у магістерському проекті є геоінформаційні системи. Геоінформаційна система –

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

сучасна комп'ютерна технологія, що дозволяє поєднати модельне зображення території (електронне відображення карт, схем, космо-, аерозображень земної поверхні) з інформацією табличного типу (різноманітні статистичні дані, списки, економічні показники тощо). Також, під геоінформаційною системою розуміють систему управління просторовими даними та асоційованими з ними атрибутами. Конкретніше, це комп'ютерна система, що забезпечує можливість використання, збереження, редагування, аналізу та відображення географічних даних.

Геоінформаційні технології, ГІС-технології – технологічна основа створення географічних інформаційних систем, що дозволяють реалізувати їхні функціональні можливості.

Інформаційно-обчислювальна система, призначена для фіксації, збереження, модифікації, керування, аналізу і відображення усіх форм географічної інформації. ГІС використовується багатьма дослідниками в галузі вивчення проблем навколишнього середовища, для визначення різних показників на географічній сітці.

За територіальним поділом ГІС поділяються на глобальні ГІС, субконтинентальні ГІС, національні ГІС частіше мають статус державних, регіональних ГІС, субрегіональних ГІС та локальних або місцевих ГІС.

ГІС розрізняють за предметною областю інформаційного моделювання, наприклад, міські ГІС, або муніципальні ГІС, природоохоронні ГІС. Найпоширенішими ГІС – земельно-інформаційні системи. Проблема орієнтації ГІС визначається розв'язуваними задачами в ній, серед них інвентаризація ресурсів (в тому числі кадастр), аналіз, оцінка, моніторинг, управління і планування, підтримка прийняття рішень. Інтегровані ГІС, ІГІС (integrated GIS, IGIS) поєднують функціональні можливості ГІС і систем цифрової обробки зображень (даних дистанційного зондування) в єдиному інтегрованому середовищі.

Полімасштабні, або масштабно-незалежні ГІС засновані на множинних, або полімасштабних предсталеннях просторових об'єктів, забезпечуючи графічне

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

або картографічне відтворення даних на будь-якому з обраних рівнів масштабного ряду на основі єдиного набору даних з найбільшою просторовою роздільною здатністю. Просторово-часові ГІС оперують просторово-часовими даними. Реалізація геоінформаційних проектів, створення ГІС в широкому сенсі слова, включає етапи:

– передпроектних досліджень у тому числі вивчення вимог користувача і функціональних можливостей використовуваних програмних засобів ГІС, техніко-економічне обґрунтування, оцінку співвідношення «витрати / прибуток»;

– системне проектування ГІС, включаючи стадію пілот-проекту, розробку ГІС;

– тестування на невеликому територіальному фрагменті, або тестовій ділянці, прототипування, або створення дослідного зразка, або прототипу;

– впровадження ГІС;

– експлуатацію та використання.

Наукові, технічні, технологічні та прикладні аспекти проектування, створення та використання ГІС вивчаються геоінформатикою.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Існуюче ПЗ ГІС можна умовно поділити на 3 класи:

- веб ГІС;
- настільні ГІС;
- просторові бази даних.

У таблиці 2.1 представлені типові стеки відкритого ПЗ для веб і настільних ГІС.

Рівні системного ПЗ в обох випадках містять багато загальних інструментів. Таке тісне переплетення в перспективі дає можливість реалізації різних ГІС функцій як для веб, так і для настільних платформ.

Можна припустити, що майбутні настільні додатки будуть використовувати веб-сервіси, які, у свою чергу, будуть містити в собі функції, що традиційно реалізується в настільних ГІС (наприклад функції аналізу).

У той час як бази даних і картографічні веб-сервера зайняли досить стійку нішу у виробництві, настільні ГІС перебувають у стадії активного пошуку своєї ніші.

Настільна (користувальницька) ГІС – це картографічне ПЗ, яке встановлюється й запускається на персональному комп'ютері й яке дозволяє користувачам відображати, вибирати, обновляти й аналізувати дані про географічні об'єкти й пов'язану з ними атрибутивну інформацію. Розглянемо короткі характеристики основних відкритих користувальницьких ГІС.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Таблиця 2.1 – Інструментальні шари відкритих настільних і веб платформ.

Тип ПЗ	Представники	Група
Додатки	QGIS, GRASS, OSSIM, uDig, MapWindow GIS	Користувальницький інтерфейс
Середовище розробки	Eclipse, QT, OpenGL, SharpDevelop	
Високорівневі утиліти	GeoTools, PostGIS, MapWindow GIS Active	Зберігання даних
Високорівневі скриптові мови програмування	Python, Perl, R	Обробка даних
Низькорівневі утиліти	Shapelib, JTS/GEOS, GDAL/OGR, GMT	Системне ПЗ
Низькорівневі мови програмування	C, C++, Java, Fortran, C#, VB.NET	
Операційна система	Linux, Microsoft Windows	Користувальницький інтерфейс
Браузер	Firefox, Safari	
Клієнтський скриптинг	JavaScript, Java, Perl, Python	Зберігання даних
Серверний скриптинг	PHP, Perl, Python	
Високорівневі утиліти	UMN MapServer, GeoServer	Обробка даних
Високорівневі скриптові мови програмування	PHP, Perl, Python	
Низькорівневі утиліти	Shapelib, JTS/GEOS, GDAL/OGR, GMT, PostGIS	Системне ПЗ
Низькорівневі мови програмування	C, C++, Java, Fortran	
Операційна система	Linux, Microsoft Windows	

Geographic Resources Analysis Support System (GRASS)

Остання версія GRASS 6.4 є модульною системою, що надає доступ до більш ніж 300 модулів для роботи із двомірними й тривимірними растровими й векторними даними й по функціональних можливостях порівнянна із продуктом ESRI ArcGIS рівня ArcInfo. Через відсутність зручного користувальницького графічного інтерфейсу поширеність GRASS обмежена й вона використовується переважно дослідницькими інститутами й університетами. Донедавна другою причиною, що стримує ріст числа користувачів, була неможливість запуску GRASS на платформах MS-Windows без використання емуляторів Linux або Unix платформ (наприклад, Cygwin). Однак, з виходом версії 6.3.0 ця проблема була вирішена.

Quantum GIS (QGIS)

Розробка QGIS почалася в 2002 році групою ентузіастів. Метою розробки було створення простого у використанні й швидкого переглядача географічних даних для операційних систем сімейства Linux. Однак, з ростом проекту з'явилася ідея використовувати QGIS як простий графічний інтерфейс для GRASS, одержуючи в такий спосіб у своє розпорядження його аналітичні й інші функції. На сьогоднішній момент група розроблювачів QGIS вирішила первісні завдання й працює над розширенням функціональних можливостей, що давно вийшли за рамки простого переглядача. За рахунок використання крос-платформного тулкіту QT, QGIS доступний для більшості сучасних платформ (Windows, Mac OS X, Linux) і сполучає в собі підтримку векторних і растрових даних, а також здатна працювати з даними, надавані різними картографічними веб-серверами й багатьма розповсюдженими просторовими базами даних. Функціональність QGIS може бути розвинена за допомогою створення модулів розширення на C++, або Python. QGIS має одне з найбільш розвинених співтовариств у середовищі відкритих ГІС, при цьому кількість розроблювачів постійно збільшується, чому сприяє наявність гарної документації по процесі розробки й зручна архітектура.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

User-friendly Desktop Internet GIS (uDig)

Основною метою створення uDig була розробка ПЗ, що дозволяє переглядати й редагувати дані, що зберігаються в БД прямо або через веб. Розробка uDig почалася між 2004 і 2005 з ініціативи канадської компанії Refrations Research Inc. uDig написана на Java (з використанням платформи Eclipse) і споконвічно була налаштована на роботі з векторними даними. Однак, в 2007 році команда розроблювачів uDig приєдналася до команди розроблювачів Jgrass, що зайнялася роботою з реалізації в uDig можливості роботи з растровими даними. Дуже часто uDig використовується як інтерфейс доступу до бази даних PostGIS. Існує 2 основних недоліки, пов'язаних з використанням Eclipse. Перший – це розмір додатка й другий – це те, що графічний інтерфейс дуже схожий із середовищем розробки для програмування, тому може бути дуже складним для кінцевих користувачів.

Generalitat Valenciana, Sistema d'Informacio Geografica (gvSIG)

Імовірно, самий великий проект, якщо вимірювати розмірами фінансових вкладень. Ціль розробки – створення системи, здатної замінити ESRI ArcView GIS 3.x в органах муніципальної влади. Ініціатор створення – міністерство транспорту Валенсії (Іспанія), що почала розробку у зв'язку з ухваленням рішення про переклад всіх органів регіональної влади на комп'ютери під керуванням ОС Linux. Розробка gvSIG почалася наприкінці 2003 року, основний розроблювач – компанія IVERA S.A. (Іспанія). У роботу над проектом також включені кілька університетів і інші компанії. gvSIG підтримує роботу з растровими і векторними даними, а також здатний працювати з геоданими, що зберігаються в різних БД. Функції по роботі з растровими даними побудовані на основі алгоритмів проекту SAGA. Мова програмування – Java. Ціль створення ПЗ з функціональними можливостями, порівнянними з ESRI ArcView (3.X), була повністю виконана, причому місцями gvSIG перевершив ArcView. Відзначимо, що для даної ГІС існує українськомовна користувальницька документація. Однак, існує ряд мінусів: немає документації для розроблювачів і масивна залежність від більш ніж 100 C++ і Java бібліотек.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

System for Automated Geoscientific Analyses (SAGA)

Як треба з назви, дана ГІС має наукових корінь. Перший модуль для SAGA був розроблений в 2001 році в Департаменті Географії Геттингемського Університету (Німеччина) і був призначений для роботи з растровими даними. Основним призначенням SAGA є аналіз рельєфу, ґрунтове картування й рішення завдань по візуалізації даних. SAGA написана на C++ і надає стороннім розроблювачам зручний API. Основний розроблювач, а трохи пізніше й сам проект недавно перемістилися в Гамбурзький Університет. Користувальницька документація дуже гарна, що сприяє постійному росту міжнародного користувальницького співтовариства.

Integrated Land and Water Information System (ILWIS)

Розробка ILWIS починалася в компанії ITC, м. Енсхеде (Голландія) в 80-х роках. Сполучить у собі функціональність векторної й растрової ГІС призначеної для рішення широкого діапазону завдань, від аналізу зображень до моделювання ерозійних процесів. Версія 3.0 продукту дуже добре документований. Однак нова версія відсилає до документації на версію 3.0. В 2007 році вихідний код, написаний мовою (MS Visual) C був випущений під відкритою ліцензією GPL. У цей час основним координатором проекту є німецька компанія 52° North Gmb і на противагу gvSIG – репозиторій з вихідними кодами вільно доступний. ILWIS працює тільки в ОС сімейства MS-Windows.

MapWindow GIS

Дана ГІС була створена в 1998 році членами Водної Дослідницької Лабораторії в Університеті штату Юта (США). Основною метою була розробка «ядра ГІС», яке б надавало необхідну функціональність ГІС-розроблювачам. MapWindow GIS Active control написаний на MS Visual C і реалізовував функції відображення, пошуку й керування просторовими даними. Пізніше був розроблений графічний інтерфейс, названий MapWindow GIS Desktop і реалізована можливість розширення функціональності шляхом використання системи розширень. Проект очолює команда розроблювачів Університету штату Айдахо. З недавнього часу розробка базується на основі Microsoft. Net Framework, у зв'язку із чим MapWindow доступна тільки для ОС сімейства MS-Windows.

						ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			14

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-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

– 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 TМето 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-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

– 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-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи оновлення цифрових топографічних карт геоінформаційної хмари.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Проведемо аналіз поточного стану технологій проведення топографічних зйомок, існуючих методик, апаратних і програмних засобів для проведення робіт. У цей час для виконання робіт застосовують різні сучасні методи й засоби. Але не всі методики й устаткування, що використовується для досягнення поставлених завдань, завжди економічно й технічно обґрунтовано.

Застосування лазерного 3D сканування обумовлено високою вартістю устаткування, зайвою щільністю й точністю вимірів, однак є перспективним методом.

Стереотопографічний метод має недоліки при зйомці з високою щільністю забудови й густою деревною рослинністю, вимагає певних правил і часових рамок виконання, кваліфікованого персоналу і є не раціональним з погляду вартості робіт і устаткування.

Велике застосування знайшли сучасні методи тахеометричної й супутникової технології. Ці технології динамічно розвиваються при інтеграції з областю високих комп'ютерних технологій і дозволяють вирішувати більший спектр геодезичних і картографічних завдань.

За результатами проведеного порівнянного аналізу на першому етапі можна говорити про те, що на забудованих територіях найбільш кращими є комбінований метод зйомки, при використанні тахеометричного й супутникового, як найбільш оперативний, маневрений і економічно ефективний метод великомасштабної топографічної зйомки. Роботи можуть виконуватися в будь-який час року, практично не залежать від часу доби й погодних умов. При використанні сучасних електронних тахеометрів і ІАСВК практично досяжна максимальна економічна ефективність.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Тому надалі розглянуті технологічні процеси з використанням сучасних приладів і апаратно-програмних засобів. Дане дослідження було б не повним, якщо не розглянути хоча б основні й найпоширеніші апаратно-програмні засоби. На сьогоднішній день дані програми стали невід'ємною частиною технологічного процесу при створенні, зберіганні й відновленні ТП і К усього масштабного ряду. Протягом останнього сторіччя безупинно відбувалися зміни в стані технічних і програмних засобів, використовуваних для виконання польових геодезичних робіт, створення й відновлення картографічної інформації. Ці зміни пов'язані з розвитком ІТ-технологій, вимог автоматизації всіх видів геодезичних робіт і впровадження у виробництво останніх досягнень у галузі науки й техніки.

Актуальні завдання, розв'язувані в теперішній час геодезистами, вимагають застосування нових технологій і високотехнологічної техніки, застосування інтегрованих приладів і програмних засобів, що дозволяють вирішувати їх у найкоротший термін з максимальною економічною ефективністю. У цей час на ринку геодезичного устаткування існують інтегровані геодезичні прилади нового покоління, що дозволяють оперативно вирішувати завдання геодезії й картографії в автоматизованому режимі з абсолютно новим принципом збору просторової інформації про місцевість. Такі вимірювальні прилади постачені убудованими обчислювальними засобами, повнокольоровими екранами й запам'ятовувальними пристроями, що створюють можливість реєстрації й зберігання результатів вимірів для подальшого їхнього використання в технологічному процесі. При камеральній обробці результатів польових робіт необхідно високопродуктивне й багатофункціональне програмне забезпечення, для перекладу растрової інформації в цифровий вид і формування властиво цифрової карти. Робота з векторною інформацією. При топографічному картографуванні на комп'ютері можуть використовуватися:

- цифрові фотограмметричні станції (ЦФС);
- геоінформаційні системи (ГІС);
- видавничі пакети.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Тому програми для роботи із графічними даними в топографії, а саме різні системи автоматизованого проектування (САПР) і ГІС, у край важливі в процесі створення (оформлення) і відновлення цифрової моделі місцевості. На даний момент сучасний ринок супутникової апаратури представлений великим вибором супутникових ГНСС приймачів. Великі фірми виробники оцінили можливість застосування інтегрованих супутникових технологій для виробництва різних топографо-геодезичних і картографічних робіт.

Також велике значення має перспективність даного напрямку для геодезії, картографії й ГІС. Тому при виборі ІАСВК, у першу чергу, спочатку необхідно визначити:

- для яких цілей необхідно дане устаткування;
- тип виконуваних завдань;
- необхідна точність позиціонування;
- вивченість району, де будуть вироблятися роботи й т.д.

Тільки після цього можна приступитися до вибору необхідного устаткування, тому що ціни на супутникові приймачі коливаються в більших межах, і не варто переплачувати за недопрацьовані, непотрібні опції й можливості. Для цього проаналізовані існуючі інтегровані супутникові приймачі, найбільш відомих фірм виробників, їхні технічні характеристики й програмне забезпечення що дозволяють вирішувати великий спектр завдань. Обробка результатів польових вимірів, як правило, виробляється за допомогою спеціального програмного забезпечення. При викладі матеріалу даного розділу розглядаються ГІС, орієнтовані на роботу з ІАСВК і спільного використання растрових і векторних матеріалів (фотопланів, карт, цифрових моделей місцевості), які забезпечують:

- перекодування й перетворення масиву «сирих» даних;
- переклад вихідних матеріалів у векторний вид;
- прив'язку до векторних об'єктів баз даних семантичної інформації;

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

- організацію взаємозв'язків між об'єктами;
- що володіють великими функціями за аналізом спільно й роздільно метричної й семантичної інформації.

Програмні засоби можуть обробляти значні обсяги даних. Однак створення геоінформаційних систем і їхня інтеграція з апаратурою ГНСС при створенні й відновленні топографічної інформації, баз даних, у тому числі в

On-line і режимі пост-обробки, залишають велику кількість питань щодо використання даних видів програмних продуктів. Підбиваючи підсумок аналізу даного етапу можна зробити наступний висновок: основним критерієм для вибору приладового й апаратного засобу стає можливість роботи з більшою кількістю інформації, простота використання й сумісності з ГІС пакетами.

У результаті, мабуть, що існуючі технології й методи не дозволяють говорити про впровадження нової техніки у виробництво. За результатами проведеного аналізу при існуючих приладах, апаратно-програмних засобів необхідно розробити універсальну методичку й технологію для спільного використання, застосування сучасних методів зйомки, сучасних технічних засобів і ГІС. Дана тенденція диктується вимогами до строків виконання робіт і є актуальним завданням в області геодезії й картографії. Надто важливо, з погляду економічної ефективності, використання сучасного комбінованого методу для процесу створення, обробки, зберігання поширення й відновлення цифрових ТП і К.

3.2 Розробка структурної схеми

Проведемо проектування структурної схеми системи створення й відновлення ТП і К із застосуванням ІАСВК і ГІС технологій, що відповідають принципам і вимогам її побудови, виявлених за підсумками аналізу предметної області.

Архітектура системи націлена на рішення наступних завдань:

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

- оперативного моніторингу й відновлення топографічних карт і планів;
- універсальність і простота використання всієї гео– просторової інформації;
- максимальна автоматизація проведення все процесів;
- мінімізація витрат і працездат;
- сумісність форматів і конвертація інформації;
- виключення грубих помилок і контроль якості готової продукції.

Дана структурна схема припускає використання сучасних методів топографічної зйомки й апаратно-програмних засобів. Розроблена структурна схема показана на (рис.3.1).

Технологічний процес містить у собі чотири основних етапи:

- підготовчий;
- польовий;
- камеральний;
- контроль якості й прийом готової продукції.

По суті структурна схема реалізує навігаційну парадигму пошуку рішення, коли за результатами аналізу вихідної інформації приймається рішення по питанню вибору методу зйомки.

На першому етапі з існуючого банку гео– просторових даних збирається вся наявна інформація про місцевість, у якій передбачається вести роботи. Після вивчення або аналізу існуючої інформації приймається рішення про необхідність залучення додаткових даних, отриманих від міністерств і відомств, адміністрації, або інших органів самоврядування. У процесі необхідно з максимальною ймовірністю встановити зміни, що відбулися, для вибору методу зйомки. Якщо дана інформація відсутня або повністю застаріла, приймається рішення про вибір економічно ефективного методу первісного створення карт і планів аерофотознімання. Якщо дана інформація застаріла частково, приймається

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

рішення про вибір комбінованого методу для оперативного відновлення. Після вибору методу вирішується питання фінансування. Розробляється або використовується наявний класифікатор об'єктів, пункти з відомими координатами. Всі зібрані дані завантажуються в прилади.



Рисунок 3.1 – Структурна схема системи

На другому етапі виконується робота у полі. При супутниково-тахеометричному методі бригада може складатися із двох чоловік, що значно скорочує витрати на заробітну плату. У полі виробляється обстеження вихідних реперів, визначаються зміни, що відбулися, із часу останнього відновлення плану або карти. Наступний крок – створення знімальної основи. Для цієї мети

традиційно прокладаються тахеометричні ходи. Однак з появою супутникових методів створення знімального обґрунтування сталася більше простим і швидким. Проблема існуюча в тахеометрів – необхідність у видимості. У супутникових методах не потрібна видимість між пунктами й можливість роботи одночастотної апаратури до 25 км із заявленою точністю фірми виготовлювача. Новий рівень геодезичного устаткування дозволяє зробити максимальну інтеграцію з використовуваною ГІС. Крім геодезичних вимірів приймач дозволяє у полі збирати семантичну інформація, що кодується за допомогою обраного й завантаженого класифікатора. При проведенні польових робіт додатково збираються різні фото- і відеоматеріали, які також можна прив'язати до об'єкта й розмістити в банку даних. Крім того, до будь-якої пікетної точки можна записати інформацію в текстовому або аудіо форматі, використовуючи прилад як диктофон. Для проведення польових робіт з високою точністю необхідно мінімум два пристрої: одне (базова станція) працює на пункті з точно відомими координатами, інше (ровер) – у точці, що підлягає зйомці. Треба відзначити, що при базовій лінії в 10 км час стояння на пункті в середньому становить 20 хв. Більшість статичних спостережень може бути виконане в автоматизованому режимі, так що оператор не потрібно. Однак корисно виконувати перевірку даних протягом сесії, і будь-які відхилення необхідно відзначати в польовому протоколі. Обробка й малювання об'єктів звичайно автоматизовані, так що роботи у полі не займуть багато часу. Запис нових ГІС-даних починається після вибору класифікатора й створення проекту зйомки. При даному виді зйомки можливий контроль створення й відновлення цифрових топографічних карт безпосередньо у полі (у реальному масштабі й часі), у тому числі виявлення помилок, невідповідностей і недоліків (з'єднання лінійних умовних знаків, додавання інформації про матеріал, стан і інші параметри й т.п.). Після дані повинні бути переміщені на ПК для їхній наступного аналізу, обробки урівняння й коректування існуючих цифрових топографічних карт.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

На третьому етапі проводиться аналіз і обробка польових вимірів і їхня конвертація в ГІС середовище для: створення, наповнення, зберігання, відновлення й поширення цифрових топографічних карт, підготовки карт до друку.

По закінченні польових робіт із приладів необхідно скачати інформацію на ПК. У ході камеральної обробки застосовують інтегровано-автоматизовану обробку даних. Ефективність автоматизації й інтеграції полягає в збільшенні продуктивності робіт з аналізу й обробки за рахунок збільшення швидкості їхнього виконання й у багато разів скоротить імовірність появи всякого роду помилок, пов'язаних з польовими й камеральними геодезичними й картографічними роботами. Так само при автоматизації підвищується продуктивність праці й зменшуються витрати за рахунок більше швидкого виконання співробітниками своїх завдань, виключення дублювання інформації. Після завантаження проводиться ручний первинний аналіз. Проглядаються дані по перекриттю часових інтервалів, правильності завантаження, уведених значень, параметрів антени і якості прийому сигналу із супутників за коефіцієнтами: відносини сигнал/шум; висота супутників; фаза несучої. Крім того, попередній аналіз допомагає виявити й розв'язати типових проблеми, наприклад, перевірка ID місця, параметрів висоти антени й інформації про контрольну точку. Якщо інформація про точку реєструвалася в польових умовах за допомогою переносного пристрою або приймача з інтегрованим інтерфейсом користувача й було підтверджено, що ID місця, час спостереження й висоти антени правильні, те, імовірно, редагувати дані не прийде. Далі необхідно провести визначення диференціальних відносин між точками, спостережуваними в процесі збору даних. Для цього вихідна інформація, отримана приймачем, повинна бути оброблена в автоматичному режимі. Результатом обробки вихідних GPS-даних є вектор, що визначає ці відносини. В обчисленні таких векторів і складається роль модуля обробки даних. Фактично весь процес обробки даних зводиться до простого натискання кнопки «Обробити», і можна бути впевненим, що підсумком

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

цієї дії буде одержання найкращого результату. Коли вихідні дані оброблені й отримані задовільні результати обробки, не утримуючих непояснених помилок, проводиться урівняння. Після отримані результати необхідно конвертувати в зрозумілий формат ГІС середовища для їхній подальшого використанні. Тому при даній технології розроблений конвертор імпорту й експорту цифрових даних. Зокрема, перетворення координат повністю відповідають стандарту OpenGIS. Для рішення зазначеного двоєдиного завдання при реалізації програмних засобів імпортування й експортування даних від геодезичних приладів урахувалися об'єктивні й суб'єктивні фактори даного питання. Об'єктивним фактором є широка розмаїтість цифрових геодезичних приладів і, як наслідок, велика кількість форматів зберігання даних. До суб'єктивних факторів можна віднести правила ведення зйомки, або вірніше їхня відсутність, при виконанні польових вимірів багатьма геодезистами. Функції експорту й імпорту даних покликані забезпечити взаємодія ГІС із іншими інформаційними системами. Для підтвердження розробленої методики й технології надалі на підставі експериментальних досліджень будуть наведені обґрунтовані результати по виявленню точнісних характеристик супутникових вимірів на прикладі комплексного дослідження декількох класів супутникової апаратури.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Геоінформаційні системи базуються на кількох основних компонентах:

- проєкційні перетворення;
- класифікація даних;
- система управління базами даних;
- аналітичний апарат.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Основним компонентом будь-якої просторової інформації є дані про положення кожної точки контуру об'єкту на місцевості (метрика об'єктів). При цьому слід враховувати, що реальна місцевість не є плоскою, як екран монітору чи аркуш паперу. Для відтворення земної поверхні на площі в картографії застосовуються спеціальні проєкційні перетворення, різні для різних за формою та місцезнаходженням ділянок місцевості. Тому ГІС, що зберігає дані на значні за площею території, має постійно виконувати операції перетворення метрики. Від швидкості та точності виконання операцій проєкційних перетворень залежить якість роботи всієї системи в цілому.

На незначній за площею території знаходиться значна кількість різноманітних об'єктів. Вони мають різний тип локалізації: ліс займає певну площу, струмок може представлятися як лінійний об'єкт, окреме дерево – просто точка на карті і т.п. Для різних задач певний об'єкт може мати різний тип локалізації. Наприклад, якщо ГІС вирішує завдання по зберіганню та обробці даних про земельні ділянки, то дорога має описуватись площинним об'єктом, який характеризується певним контуром та площею, але якщо ГІС вирішує транспортну задачу, дорога може розглядатися як лінійний об'єкт з певною довжиною та шириною кожної ділянки. Крім того, різні об'єкти місцевості можна розділити за їх призначенням або відношенням до певної категорії (дорожня мережа, рослинність та інше). Тому важливим аспектом ГІС є спосіб класифікації об'єктів за різними ознаками, важливими для задач даної системи. Зазвичай інформація в ГІС поділяється на певні теми (топографічні дан, земельний кадастр...). А в межах теми об'єкти поділяються на шари (гідрографія, рельєф, рослинний покрів...). Крім того, для кожного об'єкта встановлюються: переліки ознак (семантик або атрибутів), якими він повинен або може характеризуватися; правила відображення об'єкту на екрані та при роздруку карти місцевості на папері; діапазон масштабів, в межах якого об'єкт при перегляді карти відображається на екрані. Також можуть призначатися певний набір правил

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

цифрового опису, топологічних відносин та інші аспекти подання інформації про певний об'єкт місцевості.

Для зберігання, пошуку та вибору даних геоінформаційна система повинна мати розвинуті засоби роботи з базами даних. Просторова інформація, класифікована певним чином, структурується і зберігається в спеціалізованих базах даних. Враховуючи, що сформована база даних навіть для невеликої території може мати значні обсяги і при цьому пошук та вибірка інформації по атрибутах та місцезрештуванню мають певні відміни, ГІС часто спираються на власні спеціалізовані бази даних. Крім того геоінформаційна система обробляє не тільки дискретні дані, але й інформацію про безперервні явища, подану растрами, матрицями, різними моделями. Тому на швидкість роботи і можливості системи впливає спосіб зберігання різних за типами даних. Через це ГІС часто використовують власні формати для всіх типів інформації.

Під аналітичним апаратом ГІС слід розуміти набір алгоритмів і задач обробки просторових даних, що включили до складу програмного забезпечення розробники системи. Склад аналітичного апарату ГІС визначається її призначенням. Широкий набір розрахункових і аналітичних операцій розширює можливості ГІС, але ускладнює її інтерфейс і, відповідно, часто впливає на складність роботи користувача, особливо новачка. Тому сучасні геоінформаційні системи мають здебільшого модульний склад. Певна частина операцій включається до базового складу системи, а інші додаються за потреби.

Основні задачі, що вирішують ГІС

Інформація, що включає просторову складову, становить значну частину всіх даних, з якими мають працювати організації та установи. Тому сьогодні геоінформаційні системи вже давно вийшли за рамки поняття системи, що обробляє власно просторові дані. Сучасні ГІС дозволяють працювати не тільки з різними картами та атрибутами об'єктів на них, але і з різними типами документів (текстових, графічних, мультимедійних), пов'язаних з певними об'єктами,

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

здійснювати складні запити до баз даних та перетворювати їх результати у карти, картограми чи діаграми, прив'язані до певних територій та багато інших операцій.

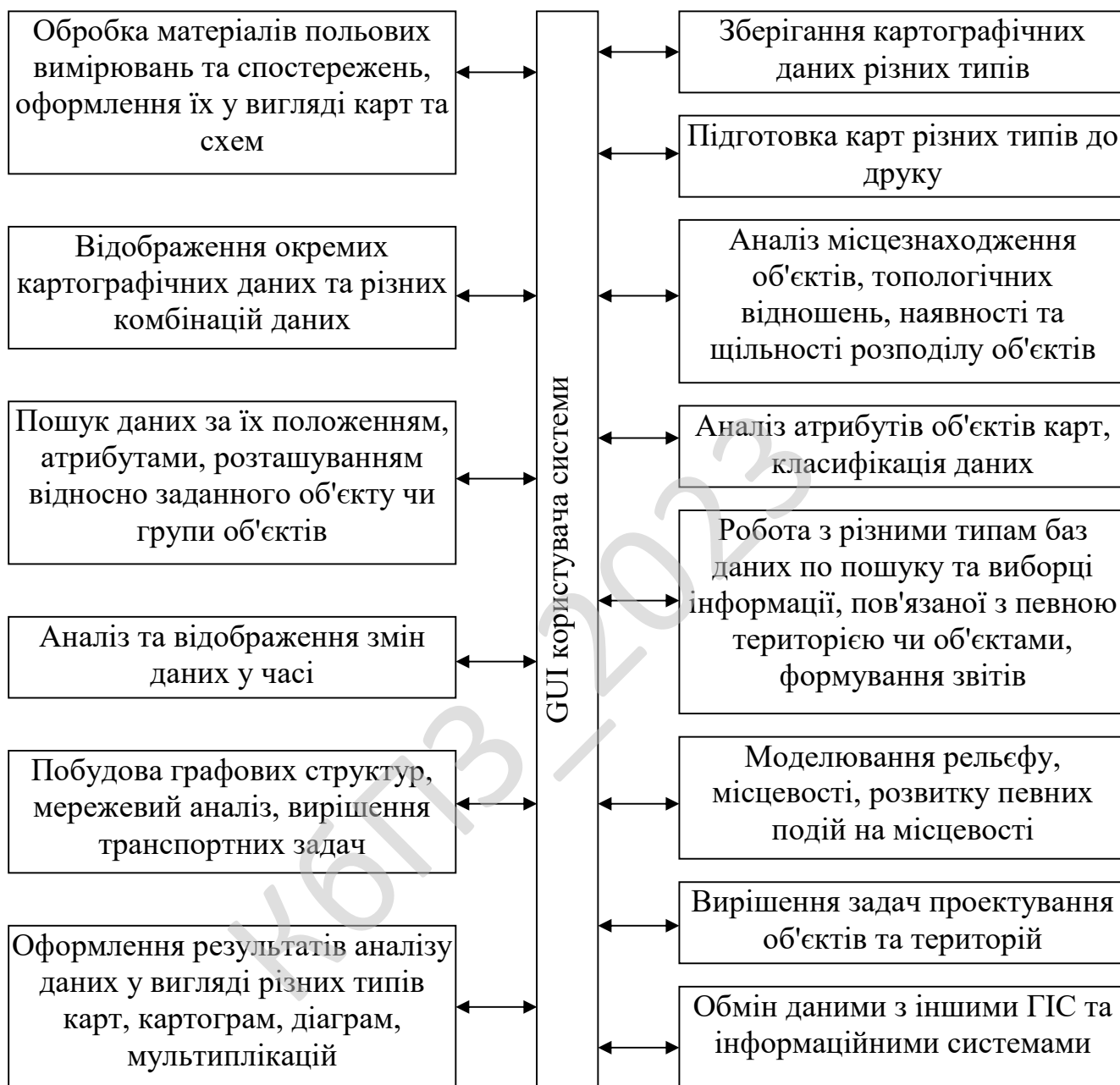


Рисунок 3.2 – Функціональна схема системи

Наведемо основні функції розробленого у магістерському проекті програмного забезпечення:

1. Обробка матеріалів польових вимірювань та спостережень, оформлення їх у вигляді карт та схем.
 2. Зберігання картографічних даних різних типів.
 3. Відображення окремих картографічних даних та різних комбінацій даних.
 4. Підготовка карт різних типів до друку.
 5. Пошук даних за їх положенням, атрибутами, розташуванням відносно заданного об'єкту чи групи об'єктів.
 6. Аналіз місцезнаходження об'єктів, топологічних відношень, наявності та щільності розподілу об'єктів.
 7. Аналіз атрибутів об'єктів карт, класифікація даних.
 8. Аналіз та відображення змін даних у часі.
 9. Робота з різними типам баз даних по пошуку та виборці інформації, пов'язаної з певною територією чи об'єктами, формування звітів.
 10. Побудова графових структур, мережевий аналіз, вирішення транспортних задач.
 11. Моделювання рельєфу, місцевості, розвитку певних подій на місцевості.
 12. Оформлення результатів аналізу даних у вигляді різних типів карт, картограм, діаграм, мультиплікацій.
 13. Вирішення задач проектування об'єктів та територій.
 14. Обмін даними з іншими ГІС та інформаційними системами.
- Дані в ГІС поділяються на позиційні та атрибутивні.

Позиційні дані

Позиційні дані описують просторові характеристики різних об'єктів, таких як дороги, будівлі, водойми, лісові масиви. Реальні об'єкти можна розділити на дві абстрактні категорії: дискретні (будинки, територіальні зони) і безперервні (рельєф, рівень опадів, середньорічна температура). Існує два способи представлення позиційної інформації – векторний та растровий.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Растровий спосіб

Растрові дані зберігаються у вигляді наборів величин, упорядкованих у формі прямокутної сітки. Осередки цієї сітки називаються пікселями. Найпоширенішим способом отримання растрових даних про поверхню Землі є дистанційне зондування, проведене за допомогою супутників. Зберігання растрових даних може здійснюватися в графічних форматах, наприклад TIF або JPEG, або в бінарному вигляді в базах даних. Найчастіше растр використовують для безперервних об'єктів.

Векторний спосіб

Дискретні об'єкти та безперервні поля величин представляють за допомогою сукупності геометричних фігур – векторних об'єктів. Найпоширенішими типами векторних об'єктів є:

– Точки – Використовуються для позначення географічних об'єктів, для яких важливо розташування, а не їхня форма або розміри. Можливість позначення об'єкта точкою залежить від масштабу карти. У той час як на карті світу міста доцільно позначати точковими об'єктами, то на мапі міста саме місто представляється у вигляді безлічі об'єктів. У ГІС точковий об'єкт зображується у вигляді деякої геометричної фігури невеликих розмірів (квадратик, гурток, хрестик), або піктограмою, що передає тип реального об'єкта.

– Полілінії – Служать для зображення лінійних об'єктів. Полілінія – ламана лінія, складена з відрізків прямих. Полілінією зображуються дороги, залізничні колії, річки, вулиці, водопровід. Допустимість зображення об'єктів полілінією також залежить від масштабу карти. Наприклад, велика річка в масштабах континенту цілком може зображуватися лінійним об'єктом, тоді як вже в масштабах міста потрібно її зображати просторовим об'єктом. Характеристикою лінійного об'єкта є довжина.

– Багатокутники (многокутники чи полігони) – Служать для позначення просторових об'єктів з чіткими кордонами. Прикладами можуть служити озера,

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

парки, будівлі, країни, континенти. Характеризуються площею і довжиною периметра.

Векторні дані добре підходять для передачі інформації про дискретні географічні об'єкти, але можуть описувати також безперервні поля величин. Поля при цьому зображуються у вигляді ізолій або контурних ліній. Одним із способів подання рельєфу є нерегулярна триангуляційна сітка. Така сітка формується безліччю точок з прив'язаними значеннями (в даному випадку висота). Значення в довільній точці всередині сітки виходять шляхом інтерполяції значень у вузлах трикутника, в який потрапляє ця точка.

Векторні дані зазвичай мають набагато менший розмір, ніж растрові. Їх легко трансформувати і проводити над ними бінарні операції. Векторні дані легко перетворити на растрові в той час як обернена операція набагато складніша. Векторні дані дозволяють проводити різні типи просторового аналізу, наприклад пошук найкоротшого шляху в дорожній мережі. Проте з растром простіше проводити оверлейний аналіз.

Атрибутивна інформація

У ГІС до векторних об'єктів можуть бути прив'язані семантичні дані. Наприклад, на карті територіального зонування до просторових об'єктів, які становлять зони, може бути прив'язана характеристика типу зони. Структуру і типи даних визначає користувач. На основі атрибутивних значень, присвоєних векторним об'єктам на карті, може будуватися тематична карта, на якій ці значення позначені кольорами відповідно до шкали кольорів або різного роду штриховками чи крапом. Найчастіше атрибутивні дані зберігаються у таблицях реляційної бази даних та є прив'язаними до певних векторних об'єктів. У випадку використання растрового способу позиційна та атрибутивна інформація поєднуються – колір пікселя передає одночасно і розташування і характеристику.

Базова карта

Базова карта (Base map) – карта, що містить основну (базову) топографічну інформацію в цифровому вигляді в одному чи

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

кількох шарах. Використовується як стандартна структура, на яку накладаються додаткові конкретні дані та для контролю інших джерел просторових даних.

Приведемо результати комплексних і окремих випробувань роботи ГНСС апаратури. При виконанні даного експериментального дослідження ставилася мета: на основі аналізу існуючих методів і наукових досліджень, розробленої методики й технології, присвяченої відновленню карт, провести випробування як у цілому, так і окремих процесів, роботи з топографічної зйомки із застосуванням ІАСВК і ГІС технологій по оптимальній технологічній схемі, що включає найбільш раціональної розробки на різних етапах технологічного циклу. У першій частині даної глави були проведені комплексні й окремі випробування роботи ГНСС апаратури. Для цього проводилися роботи з виявлення точнісних характеристик апаратури на різних відстанях. А також комбінація з різними типами й класами ГНСС апаратури.

На другому етапі експериментальних досліджень проведені випробування, необхідні для проведення самої зйомки із застосуванням ІАСВК і різних методів вимірів із застосуванням ГНСС апаратури.

Таблиця 3.1 – Результати випробувань при геодезичному режимі

Вектор	Довжина вектора, м	Тривалість зйомки	СКО, м
MEND – 005	34.705	6:19:02.00	0.002
MEND – STAR	9344.724	4:25:50.00	0.045
MEND – SIGN	898.892	4:41:06.00	0.004
SIGN – STAR	10187.372	4:25:50.00	0.049
005 – SIGN	865.819	4:41:06.00	0.004
005 – STAR	9379.119	4:25:50.00	0.045

З метою одержання даних, на підставі аналізу яких можна було б зробити висновки про точнісні характеристики інтегрованої супутникової апаратури, автором були виконані експериментальні виміри. Виміри проводилися в кілька етапів, при цьому віддалення рухливих приймачів від базового було на відстанях

топографічних карт, вирішувати широкий спектр завдань по інвентаризації об'єктів нерухомості й оперативного моніторингу територій зі зниженням собівартості проведених робіт. На основі виконаних польових випробувань отриманий оновлений цифровий план місцевості, і доведено по точності й можливостям використання ІАСВК для топографічних зйомок усього масштабного ряду.

Наведемо порівняльні результати застосування ІАСВК і традиційних методів зйомки. Основними результатами даного етапу досліджень є докази переваги технологічного процесу із застосуванням ІАСВК і ГІС технологій над найбільш популярним і традиційним методом тахеометричної зйомки, обґрунтування оптимального спільного використання й переваги при окремій роботі в технологічному процесі. Критерії вибору технології, основи на виконанні роботи з максимальною економічною ефективністю.

Фактично від економічної ефективності залежить кінцевий вибір і результат виробництва. Для цього в третій частині проведений розрахунок економічної ефективності від впровадження запропонованої технології відновлення ТП і К із застосуванням ІАСВК і ГІС технологій.

Розрахунки показали, що рентабельність використання ІАСВК нижче, ніж при використанні традиційного набору геодезичних приладів, хоча при впровадженні нової техніки й відбувається значне зниження трудомісткості польових робіт. Це пояснюється тим, що різке збільшення фондоємності робіт, через високу вартість імпортного устаткування, росту курсу ЄВРО не компенсується достатнім зниженням витрат на заробітну плату. Цей приватний приклад ілюструє загальну для української перехідної економіки тенденцію збільшення розриву в співвідношенні вартості двох факторів виробництва – праці (дешева вітчизняна робоча сила) і капіталу (дорогі кредити й реальний капітал). Незважаючи на більше низьку рентабельність сучасного устаткування, при проведенні польових геодезичних робіт зміст впровадження нової техніки полягає в підвищенні якості й скороченні строку виконання робіт, що є

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

немаловажним чинником оперативності надання інформації, збереженні й збільшенні обсягу робіт, за рахунок підвищення якості надаваних замовникам послуг. За результатами даного розділу можна зробити наступні висновки:

1. Швидкість виконання польових робіт в 2-5 разів вище, ніж при використанні традиційних методів.

2. Точність і надійність вимірів дозволяє оновлювати практично весь ряд топографічних карт.

3. Підвищується рентабельність і знижуються працевитрати.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

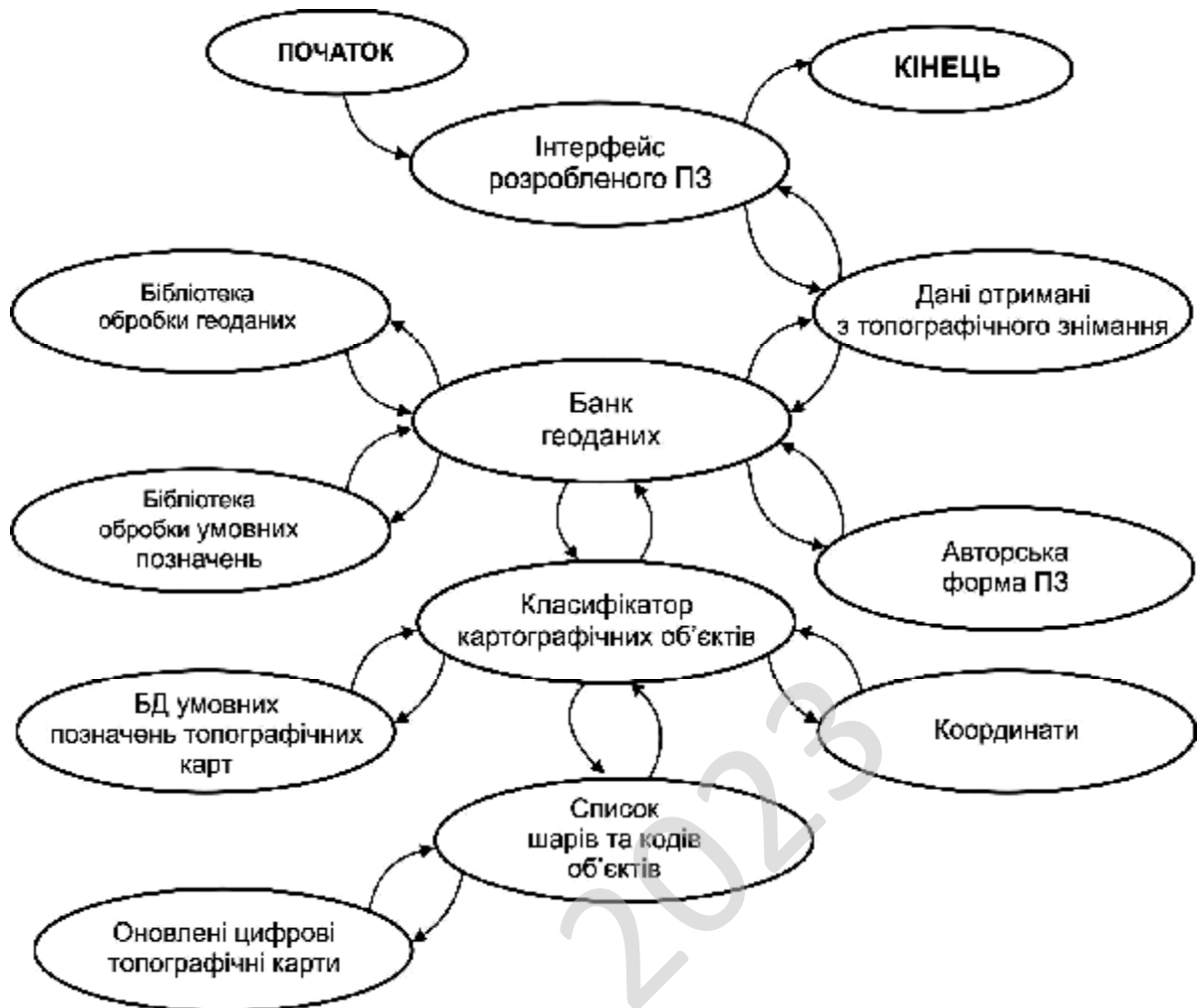


Рисунок 3.3 – Діаграма взаємодії процесів

Після початку роботи розробленого ПЗ ми потрапляємо до інтерфейсу розробленого ПЗ, звідки через дані отримані з топографічного знімання потопляємо до банку геоданих.

З цієї точки підключаються та переглядаються бібліотека обробки геоданих, бібліотека обробки умовних позначень, та авторська форма ПЗ.

Далі через класифікатор картографічних об'єктів можна користуватись БД умовних позначень топографічних карт, координати та через список шарів та кодів об'єктів провести перегляд оновлених цифрових топографічних карт.

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 ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків:

- Виділення пам'яті, підключення модулів ОС.
- Ініціалізація початкових змінних та модулю картографічних об'єктів.
- Підключення файлу налаштування ПЗ.
- Підключення модулю шарів та кодів топографічних об'єктів.
- Перевірка цілісності ПЗ та наявності ресурсів ОС для запуску ПЗ.
- Перевірка пройдена (запит).
- Виведення головного вікна ПЗ.
- Очікування запиту користувача.
- Є запит користувача?
- Обробка дії користувача.
- Є запит налаштування параметрів ПЗ?
- Виведення вікна налаштування доступу.
- Налаштування параметрів доступу до топографічних карт геоінформаційної хмари.
- Є запит обробки введених матеріалів?
- Підпрограма обробки введених матеріалів.
- Виведення системних повідомлень.
- Запит відображення окремих картографічних даних?
- Підпрограма відображення окремих картографічних даних.
- Запит пошуку даних за положенням?

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

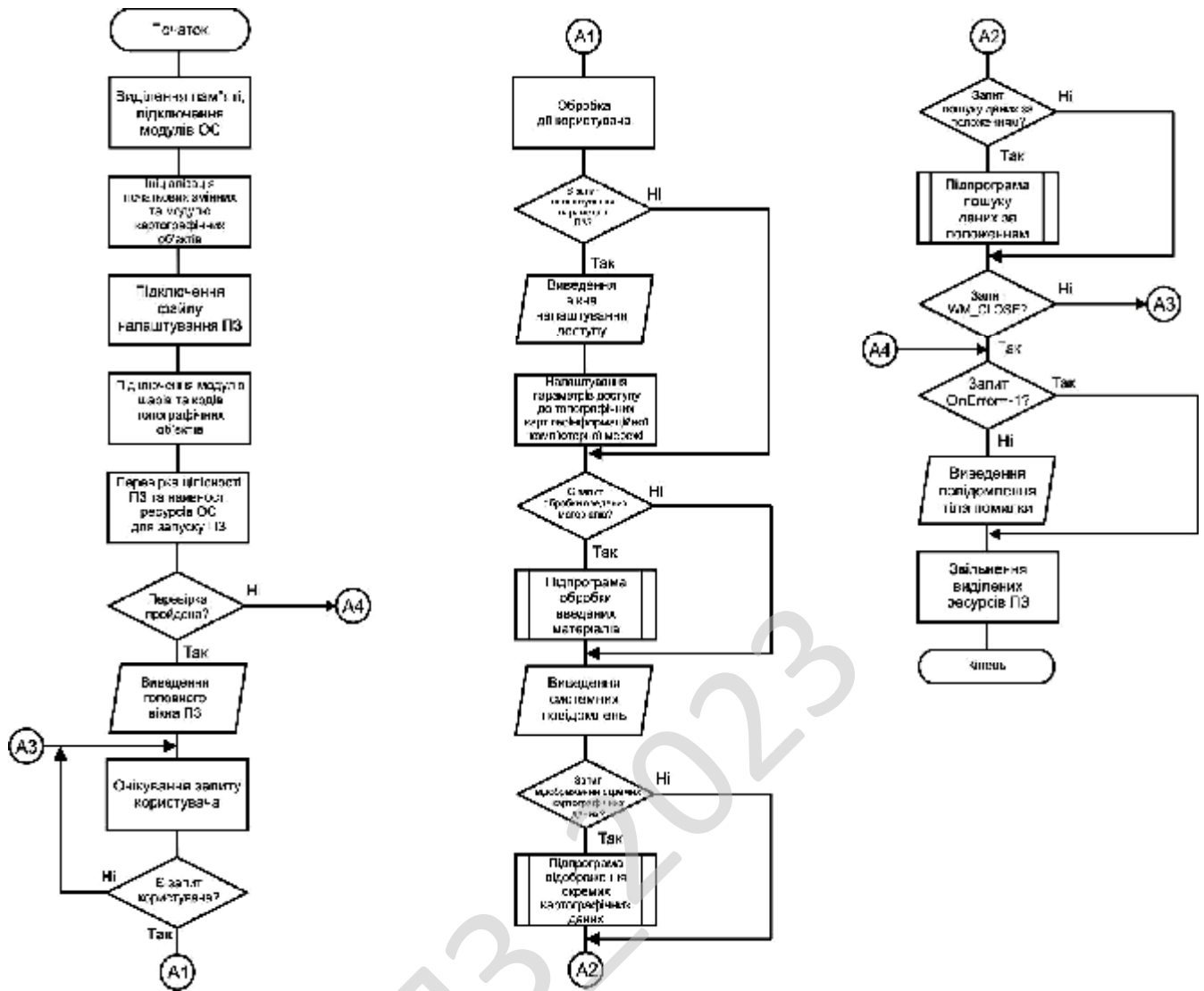


Рисунок 4.1 – Блок-схема основної програми

- Підпрограма пошуку даних за положенням.
- Запит WM_CLOSE?
- Запит OnError=-1?
- Виведення повідомлення тіла помилки.
- Звільнення виділених ресурсів ПЗ.

На рисунку 4.2 наведено блок-схему підпрограми обробки введених матеріалів. Її робота складається з виконання наступних кроків:

- Відображення вікна завантаження та обробки топографічного зображення.

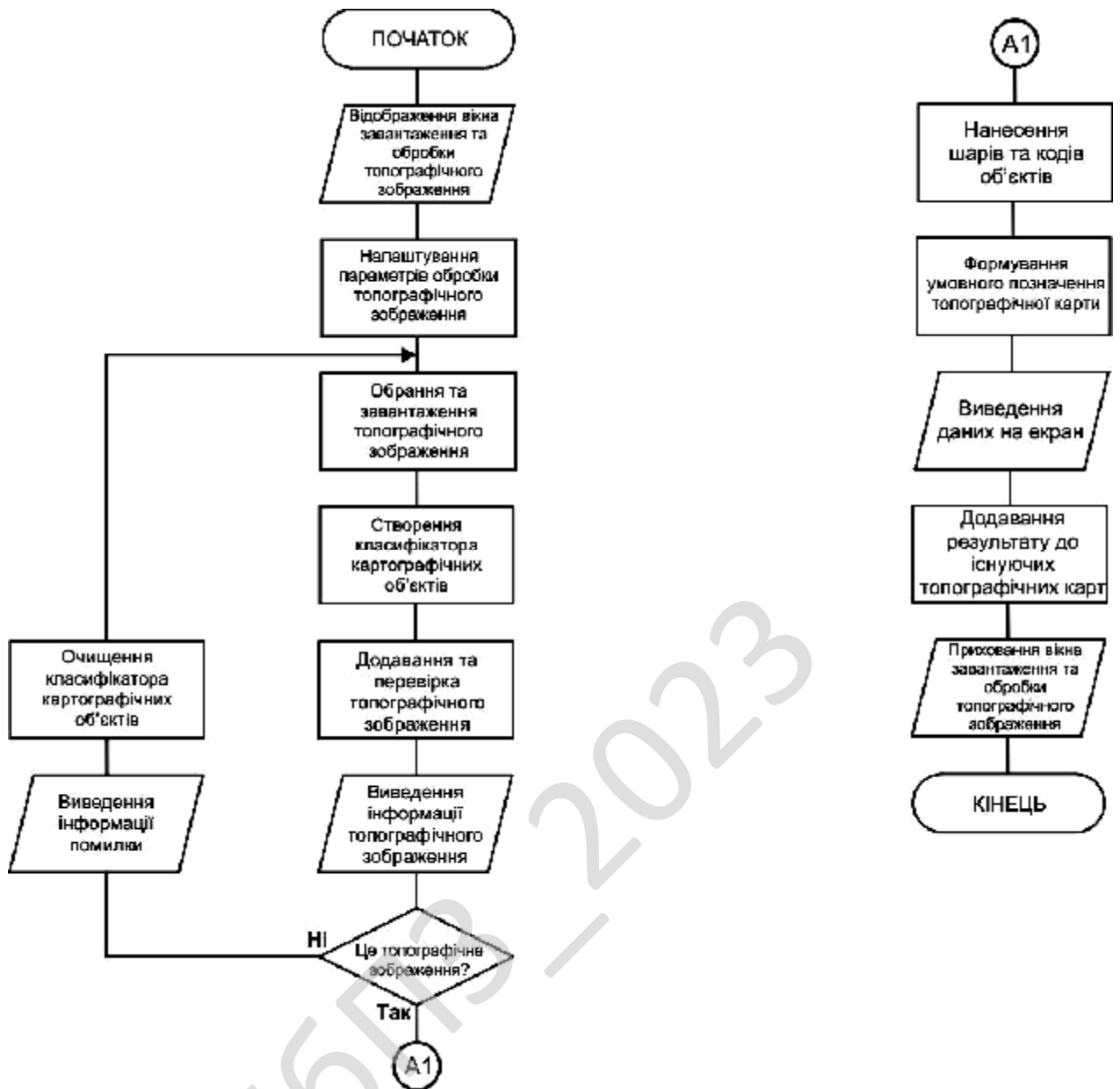


Рисунок 4.2 – Блок-схема підпрограми обробки введених матеріалів

- Налаштування параметрів обробки топографічного зображення.
- Обрання та завантаження топографічного зображення.
- Створення класифікатора картографічних об'єктів.
- Додавання та перевірка топографічного зображення.
- Виведення інформації топографічного зображення.
- Це топографічне зображення (запит).
- Нанесення шарів та кодів об'єктів.

- Формування умовного позначення топографічної карти.
- Виведення даних на екран.
- Додавання результату до існуючих топографічних карт.
- Приховання вікна завантаження та обробки топографічного зображення.

Опис алгоритмів функціонування системи

Розглянемо як саме проходила реалізація алгоритмів. У першу чергу використовувалось Javascript API. Служба «API OsmAnd Maps» має програмний інтерфейс (API) на Javascript, який дозволяє інтегрувати інтерактивні карти у WEB-сторінки.

В API OsmAnd Maps головним об'єктом є карта. Інші об'єкти лише тим або іншим способом взаємодіють із картою, змінюючи її зовнішній вигляд, поведінку, або доповнюють карту новою функціональністю.

Карта складається з одного або декількох накладених один на одного шарів. Кожний шар являє собою область, набрану з безлічі прозорих (або непрозорих) тайлів.

Один або кілька шарів можуть бути об'єднані в тип карти. Тип карти може мати додаткові параметри (наприклад, мінімальне й максимальне значення коефіцієнта масштабування), тип карти може бути доданий у перемикач типів карт.

Географічна карта земної поверхні має три вбудовані типи, які можна перемикати між собою, наприклад, за допомогою перемикача типів карт:

- звичайна карта місцевості (MAP);
- карта, складена із супутникових знімків земної поверхні (SATELLITE);
- гібридна супутникова карта з дорожньою мережею, границями держав і назвами географічних об'єктів (HYBRID).

У якості шарів можуть використовуватися зображення географічних і інших карт Землі й планет, супутникові знімки, малюнки й схеми, плани будинків і споруджень і просто фотографії.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Взаємодія з картою здійснюється за допомогою елементів керування, які дозволяють користувачеві змінювати більшість параметрів карти.

Об'єкти-оверлеї (мітки) додаються на карту поверх усіх шарів. Об'єкти прикріплюються до певного місця на карті й зберігають цю позицію при зміні масштабу й переміщенні карти.

Для навігації й для вказівки місця розташування об'єктів на карті використовується відповідна система координат.

Для географічних карт – це геодезична система координат WGS-84, де для вказівки положення об'єкта на карту досить указати його довготу й широту.

Для навігації по кресленнях і планам місцевості також може використовуватися декартова система координат.

Можливості карти також можуть бути розширені за допомогою сервісів геокодування, прокладки маршруту й візуалізації OsmAndSml.

OsmAndSml (OsmAnd Maps Markup Language) — Xml мови, призначена для опису географічних даних на картах Яндекс.

За допомогою OsmAndSml можливо в текстовому виді задати ділянку карти з нанесеними мітками й іншими об'єктами, а потім відобразити його в браузері засобами JS API OsmAnd Maps.

Сервіс геокодування служби API OsmAnd Maps використовує OsmAndSml для опису знайдених по запиту географічних об'єктів.

Для візуалізації OsmAndSml документів засобами JS API OsmAnd Maps потрібно вказувати API ключ.

OsmAndSml використовується як формат передачі даних між сервісами OsmAnd Maps і сторонніми програмними засобами.

Дані у форматі OsmAndSml можуть бути відображені на OsmAnd Maps, у цьому ж форматі повертає інформацію про місце розташування об'єктів сервіс геокодування.

API Ключ — унікальна цифро-буквена комбінація, по якій сервіс «API OsmAnd Maps» ідентифікує ПЗ, що звертається до API.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Щоб одержати API ключ, слід зареєструватися на Яндексі, у спеціальній формі одержання ключа ознайомитися й прийняти умови користувацької угоди, а також указати програму, на якій буде використовуватися API.

Ключ можна закріпити за будь-якою програмою. Для однієї програми може бути отримано кілька ключів, причому не обов'язково, щоб усі ключі були зареєстровані на один обліковий запис.

За допомогою Javascript API можливо:

- показувати фрагменти карти й супутникових знімків;
- управляти картою й обробляти події;
- показувати на карті різні об'єкти (мітки, лінії).

Щоб мати можливість користуватися API OsmAnd Maps, необхідно одержати унікальний API ключ.

API стабільно працює з наступними браузерами: Microsoft Internet Explorer 4.x; Microsoft Internet Explorer 5.x; Microsoft Internet Explorer 6.x; Microsoft Internet Explorer 7.x; Microsoft Internet Explorer 8; Microsoft Internet Explorer 9; Mozilla Firefox 2.x; Mozilla Firefox 3.x; Opera 9.x і вище; Apple Safari 3.1 і вище; Google Chrome.

Розглянемо як було реалізоване створення карти та пошук у магістерському ПЗ. Щоб додати карту на сторінку, створювався оброблювач події закінчення завантаження сторінки onload якій розмішався у тегу head після скрипта завантаження API.

```
<script type="text/javascript">
// Створює оброблювач події window.onload
    OsmAnds.jquery(function () {
// Створює екземпляр карти й прив'язує його до
// створеного контейнера
        var map = new
            OsmAnds.Мap(OsmAnds.jquery("#OsmAndsId")[0]);
// Встановлює початкові параметри відображення
// карти: центр карти
// і коефіцієнт масштабування
        map.setcenter(new OsmAnds.Geopoint(37.64, 55.76), 10);
    }
}
```

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

</script>

Клас `OsmAnd.Geocoder` дозволяє відправляти запити геокодеру, одержувати інформацію про статус і результат процесу геокодування, а також одержувати результати геокодування у вигляді міток.

Конструктор класу приймає два параметри.

1. `request` – рядок з адресою, яку потрібно геокодувати (для зворотного геокодування – точка з географічними координатами);

2. `options` – параметри:

– `boundedby` – область на карті, у якій здійснюється пошук об'єкта;

– `strictbounds` – прапор, що вказує, що шукати впливає тільки усередині області, заданою опцією `boundedby`;

– `results` – необхідна кількість результатів пошуку (не більш);

– `skip` – вказівка пропустити перші `n` результатів у відповіді;

– `preflang` – мова відповіді.

Щоб визначити координати об'єкта, створювався екземпляр класу `OsmAnd.Geocoder` і передавався йому в якості параметра рядок з адресою об'єкта.

У прикладі нижче проводиться пошук усіх об'єктів з назвою "Знам'янка":

```
var geocoder = new OsmAnd.Geocoder ("Знам'янка");  
map.addoverlay(geocoder);
```

У результаті виконання прикладу, змінна `geocoder` буде містити результати пошуку у вигляді групи спеціальних міток класу `OsmAnd.Geocoderresult`. Ці мітки відрізняються від звичайних міток класу `OsmAnd.Placemark` наявністю чотирьох додаткових полів, у яких передаються додаткові відомості про знайдений об'єкт:

– `kind` – вид топоніма (вулиця, будинок, місто і т.д.);

– `precision` – передбачувана точність відповіді;

– `text` – адреса одним рядком;

– `Addressdetails` – структуроване подання адреси об'єкта у відповідності зі стандартом `hxl`.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Клас `OsmAnds.Geocoderresult` реалізує інтерфейс `OsmAnds.Overlay`, тому мітки `OsmAnds.Geocoderresult` можна додавати на карту точно так само, як і звичайні.

Для обробки результатів геокодування необхідно враховувати, що геокодеру потрібен час на обробку запиту й відповідь. Тому рекомендується завжди використовувати оброблювачі подій для одержання інформації про закінчення геокодування й про його поточний статус.

У класі `OsmAnds.Geocoder` передбачено дві події: `Load` (геокодування пройшло без помилок) і `Fault` (з помилками).

В оброблювачі події `Load` можна оперувати результатами геокодування як групою.

Групування об'єктів. Приклад нижче ілюструє обробку подій геокодера реалізованого у магістерському проекті. У випадку успішного геокодування користувачеві карти буде показане повідомлення про кількість знайдених об'єктів, а самий релевантний (перший) з них буде показаний на карті.

```
OsmAnds.Events.observe(geocoder, geocoder.Events.Load, function () {
    if (this.length()) {
        alert("Знайдене :" + this.length());
        map.addoverlay(this.get(0));
        map.panto(this.get(0).getgeopoint())
    }else {
        alert("Нічого не знайдено")
    }
});
OsmAnds.Events.observe(geocoder, geocoder.Events.Fault,
function (error) { alert("Відбулася помилка: " + error.message) });
```

Параметр `preflang` геокодера дає можливість контролювати мову результатів пошуку.

Цей параметр дозволяє задати мову, на якій буде проводитися видача результатів геокодування. Код мови вказується у відповідності зі стандартом ISO 639.

Список можливих значень:

– `uk` Україна.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Якщо встановити параметр `preflang` у значення "uk", то в результаті пошуку адреси «Знам'янка, площа» буде отриманий відповідний результат.

```
var geocoder = new OsmAnds.Geocoder("Знам'янка",
{ preflang : "uk" } );
OsmAnds.Events.observe(geocoder, geocoder.Events.Load, function (geocoder)
{
    map.addoverlay(geocoder.get(0));
    map.setBounds(geocoder.get(0).getbounds());
});
```

Об'єкт `OsmAnds.Geocoder` підтримує зворотне геокодування, за допомогою якого можна перетворювати координати об'єкта в поштову адресу.

Для зворотного геокодування у параметрі `Request` потрібно передати замість адреси геоточки (`OsmAnds.Geopoint`).

```
// Запускає процес геокодування
var geocoder= new OsmAnds.Geocoder(new OsmAnds.Geopoint(37.588395,
55.762718), {results: 1});
// Оброблювач успішного завершення процесу геокодування
OsmAnds.Events.observe(geocoder, geocoder.Events.Load, function ()
{
    if (this.length()) {
        map.addoverlay(this.get(0));
        map.panto(this.get(0).getgeopoint())
    }else {
        alert("не знайдене")
    }
}
);
// Оброблювач невдалого завершення геокодування
OsmAnds.Events.observe(geocoder, geocoder.Events.Fault,
function (geocoder, error)
{
    alert("Відбулася помилка: " + error.message)
});
```

Результати пошуку, як і у випадку із прямим геокодуванням, вертаються у вигляді групи міток класу `OsmAnds.GeoCoderResult`, які відразу можна нанести на карту.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Об'єкт `OsmAnds.OsmAndSml` дозволяє відобразити `OsmAndSml` документ на карті, використовуючи Javascript.

Щоб відобразити документ необхідно передати на вхід конструктора `OsmAnds.OsmAndSml` URL документ, а потім додати створений об'єкт оверлей на карту. Що завантажується `OsmAndSml` повинен бути доступний привселюдно по протоколу `http`.

```
var ml = new OsmAnds.OsmAndSml ("http://api.OsmAnd.ua/  
maps/OsmAndSml/examples  
/xml/demonstration.xml");  
map.addoverlay(ml);
```

Тому що завантаження `OsmAndSml` документа – асинхронний процес, закінчення завантаження необхідно очікувати за допомогою події `Load`.

Щоб одержати можливість переглядати помилки при завантаженні `OsmAndSml` документа, необхідно прикріпити оброблювач до події `Fault`.

```
OsmAnds.Events.observe(ml, ml.Events.Fault,  
function(ml, error) {  
alert("Відбулася помилка: " + error);});
```

Тому що створений об'єкт оверлей належить класу `Maps.Geoobjectcollection`, з ним можна звертатися як із групою. Для пошуку об'єкта по якому-небудь параметру використовується метод групи `filter()`.

```
var ml = new  
OsmAnds.OsmAndSml ("http://api.OsmAnd.ua/maps/OsmAndSml/examples/xml/mapparams.xml"  
);  
map.addoverlay(ml);  
OsmAnds.Events.observe(ml,  
ml.Events.Load,  
function () {  
var obj = this.filter(function  
(obj) {  
return obj.name == 'Ім'я об'єкта'  
});  
obj[0].openballoon();  
});
```

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм RC5, який являє собою блоковий шифр із безліччю параметрів: розміром блоку, розміром ключа й числом раундів. В алгоритмі RC5 передбачені три операції: XOR, додавання й циклічні зрушення. На більшості процесорів операції циклічного зрушення виконуються за постійний час, змінні циклічні зрушення являють собою нелінійну функцію. Циклічні зрушення залежать як від ключа, так і від даних.

В RC5 використовується блок змінної довжини, але в приводиться прикладі, що буде розглянутий, 64-бітовий блок даних. Шифрування використовує $2r+2$ залежних від ключа 32-бітових слів – $S_0, S_1, S_2, \dots, S_{2r+1}$ – де r – число раундів. Для шифрування спочатку потрібно розділити блок відкритого тексту на два 32-бітових слова: A й B . (При впакуванні байтів у слова в алгоритмі RC5 дотримується угода про прямий порядок (little-endian) байтів: перший байт займає молодші біти регістра A й т. ін.) Потім:

$$A = A + S_0$$

$$B = B + S_0$$

Для i від 1 до r :

$$A = ((A \oplus B) \lll B) + S_{2i}$$

$$B = ((B \oplus A) \lll A) + S_{2i+1}$$

Вихід перебуває в регістрах A й B .

Розшифрування теж нескладно. Потрібно розбити блок відкритого тексту на два слова, A й B , а потім:

Для i від r до 1 із кроком -1:

$$B = ((B - S_{2i+1}) \ggg A) \oplus A$$

$$A = ((A - S_{2i}) \ggg B) \oplus B$$

$$B = B - S_i$$

$$A = A - S_0$$

Символом «>>>» позначене циклічне зрушення вправо. Звичайно ж, всі додавання й вирахування виконуються по модулю 2^{32} .

Створення масиву ключів складніше, але теж прямолінійно. Спочатку байти ключа копіюються в масив L із 32-бітових слів, доповнюючи при необхідності заключне слово нулями. Потім масив S ініціалізується за допомогою лінійного конгруентного генератора по модулю 2^{32} :

$$S_0 = P$$

Для i від 1 до $2(r + 1) - 1$:

$$S_i = (S_{i-1} + Q) \bmod 2^{32}$$

де $P = 0xb7e15163$ і $Q = 0x9e3779b9$.

Нарешті, потрібно підставити L в S :

$$i = j = 0$$

$$A = B = 0$$

Виконати $3n$ раз (де n – максимум від $2(r + 1)$ і c):

$$A = S_i = (S_i + A + B) \lll 3$$

$$B = L_i = (L_i + A + B) \lll (A + B)$$

$$i = (i + 1) \bmod 2(r + 1)$$

$$j = (j + 1) \bmod c$$

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

- Меню:Карти; Плагіни; Налаштування; Довідка.
- Карти: Поле введення карти; Навігаційні клавіші та дії.
- Функції; Завантаження; Обробка матеріалів польових вимірювань;

Відображення окремих картографічних даних; Пошук даних за їх положенням; Аналіз та відображення змін даних у часі; Друкування; Перегляд БД; Налаштування; Авторське право.

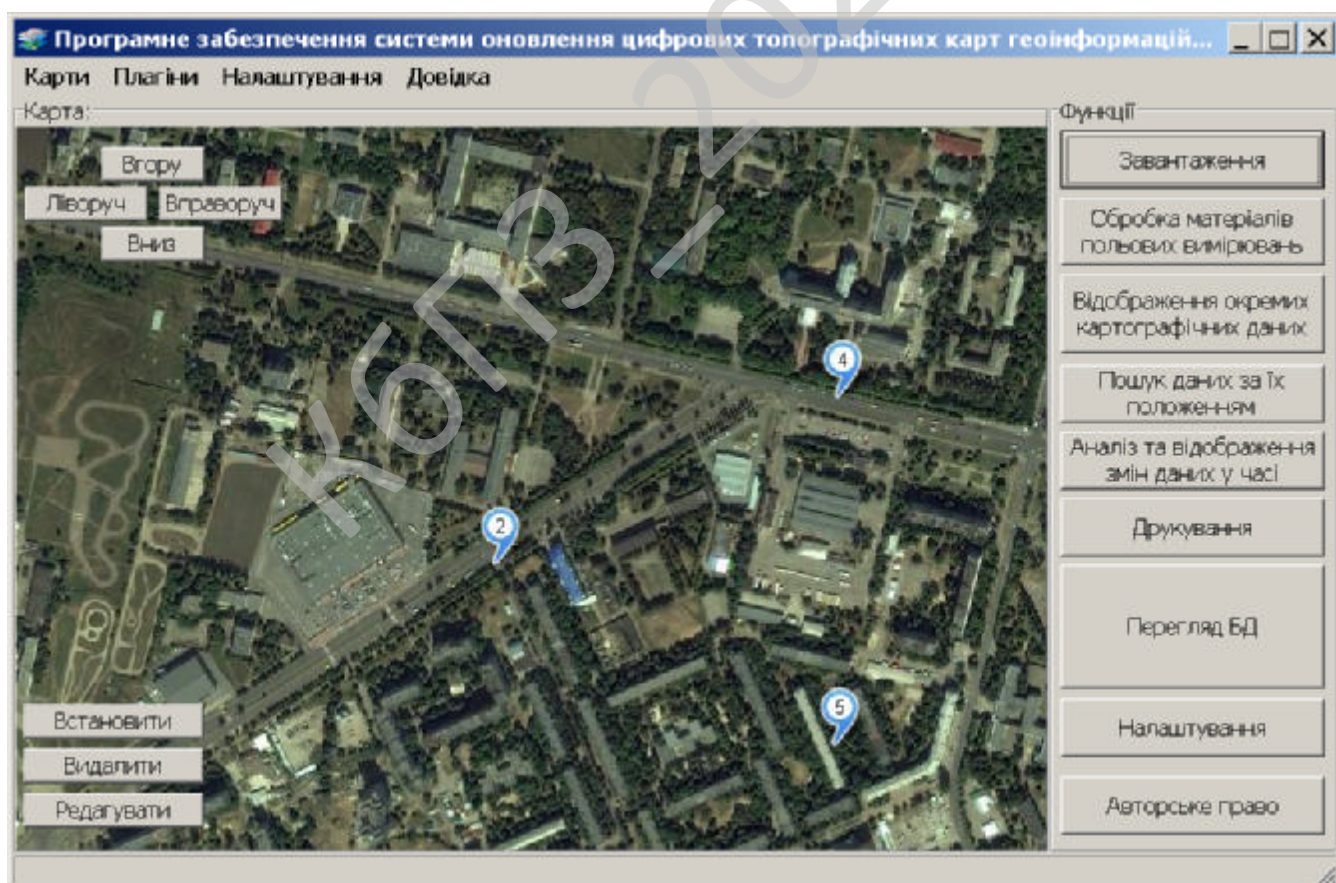


Рисунок 5.1 – Головне вікно програми

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

На рисунку 5.2 зображено форму авторського права. Була вибрана ліцензія Freeware.

При розробці ПЗ була створена форма авторського права з ліцензій на використання програмного забезпечення.

Ліцензія на використання програмного забезпечення це вид ліцензії, що визначає умови використання виробу, яким є комп'ютерне програмне забезпечення.

Ліцензія може надавати дозвіл робити з ним речі, які були б інакше заборонені законом про авторське право. Наприклад, ліцензія на використання програмного забезпечення може дати дозвіл робити копії програмного забезпечення. Власник авторського права може запропонувати ліцензію на використання ПЗ односторонньо, або як частину ліцензійної угоди на використання ПЗ з іншою стороною.

Ліцензія може містити умови. Якщо Ви не дотримуєтесь умов ліцензії, то підпадаєте під дію закону про авторське право.

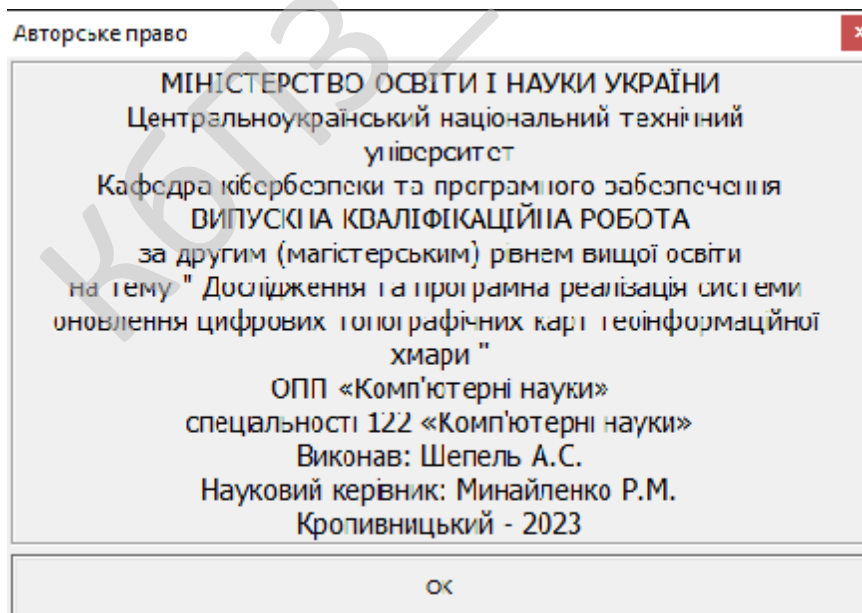


Рисунок 5.2 – Довідка

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи оновлення цифрових топографічних карт геоінформаційної хмари.

Метою розробки є дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари.

Об'єктом дослідження є процес оновлення цифрових топографічних карт геоінформаційної хмари.

Предметом дослідження є методи оновлення цифрових топографічних карт геоінформаційної хмари.

Методи дослідження базуються на методах теорії побудови геоінформаційних систем, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод оновлення цифрових топографічних карт геоінформаційної хмари.
- Розроблено вітчизняний продукт оновлення цифрових топографічних карт геоінформаційної хмари, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

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

В магістерській роботі було проведене дослідження та виконана програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари. 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	120
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.	–	120000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

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.

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

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 62 = 104 \text{ люд/день.}$$

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

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

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	104	Ф 7.1-7.4
Впровадження	13	Д13
Всього	145	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;
 $T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{145 \cdot 1}{60 - 5} = 2,6 \text{ ставки.}$$

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	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

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:

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

$$\Phi_{др}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

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

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

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	12098	36294
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	2,6	12000	93600
Інженер-електронщик	1,2	11000	39600
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	$R_{cn} = 7,05$	-	$\Phi_{роб} = 244494$

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.

$$z_{cd} = \frac{244494}{7,05 \cdot 60} = 578 \text{ грн.}$$

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-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

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

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

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

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

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

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

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

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Brain за 26.10.23 – джерело <http://brain.com.ua>.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	Intel Core™ i3 10105 (BX8070110105) 1200, 4 ядра, 8 потоків, 3.7 GHz, 4.4 GHz, TDP - 65 Вт, 14nm box	-
Системна плата	GIGABYTE H510M S2H V3, DDR4, 3200 MHz, LAN - 1 Гбит/с, D-Sub (VGA), DisplayPort, HDMI, 1 x M.2 2280, 4 x Sata 6.0 Gb/s, Micro-ATX	-
Відеокарта	Intel UHD Graphics 630	-
Жорсткий диск	SSD 2.5" 256GB Mibrand (MI2.5SSD/CA256GBST) 256 GB, 3D TLC NAND, 2.5", SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 16GB 3200 MHz Dato (DT8G4DLND32) DDR4, 8 ГБ, В наборі - 2	-
DVD-привод	-	-
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA-489, PSU 350W(FSP Brand: ATX-350PNR, 12cm), black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design, Thermally Advantaged Chassis	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	—	—
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000:1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Таблиця 7.7 – Балансова вартість обчислювальної техніки

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

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	0	20	0,0
6. Господарський інвентар	28000	25	7000
Всього по групі	33190	-	8297,5
7. Нематеріальні активи	120000	10	12000
Разом	$K_p = 1760367$		$A_p = 190286$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 578 \cdot 145 / 120 = 698 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 698 \cdot 10 \cdot 0,01 = 70 \text{ грн.}$$

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(698+70) = 288 \text{ грн.}$$

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

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

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 698 \cdot 15 \cdot 0,01 = 105 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджів, тонеру, грн.; N_e – кількість екземплярів програм, шт.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Згідно норм приймаємо пів пачки паперу на три місяці розробки ($n_p=0,5$). Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 210$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

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

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

Згідно норм до вартості запам'ятовуваних пристроїв входить вартість CD дисків в кількості, що дорівнює 50 екземплярів програм та одного DVD диска для збереження резервної копії програми:

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

де: $Ц_\delta$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 28 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 40 грн./шт.

$$З_{M2} = 28 \cdot 50 + 40 = 1440 \text{ грн.}$$

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

$$З_{M3} = \sum Ц_\gamma, \quad (7.18)$$

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

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

$$З_M = (105 + 1440 + 1702) / 120 = 27 \text{ грн.}$$

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, %.

$$O_n = 698 \cdot 15 \cdot 0,01 = 105 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 120$ прим.):

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

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

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

$$A_m = 190286 \cdot 3 / (120 \cdot 12) = 396 \text{ грн.}$$

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

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

$$C_n = 698 + 70 + 288 + 105 + 27 + 105 + 396 = 1689 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1. Основна зарплата виконавців	Z_o	698
2. Додаткова зарплата виконавців	Z_d	70
3. Відрахування на соціальні потреби	C_{oc}	288
4. Загальногосподарські витрати	Γ_{ocn}	105
5. Витрати на матеріали	Z_m	27
6. Освоєння нових операційних систем, мов програмування	O_n	105
7. Амортизація основних фондів	A_m	396
8. Повна собівартість програмного забезпечення	C_n	1689
9. Плановий прибуток	P_p	845
10. Ціна підприємства $C_n = C_n + P_p$	C_n	2534
11. Податок на додану вартість $ПДВ = 0.01 \cdot N_{дв} \cdot C_n$	$ПДВ$	506,8
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	3040,8

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, %.

$$P_p = 0,01 \cdot 50 \cdot 1689 = 845 \text{ грн.}$$

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	–	3041
Всього капітальних витрат	–	3041

7.7 Визначення експлуатаційних витрат

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

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	Z_p	29524	24156
2. Витрати на електроенергію	$Z_{ел}$	456	373
3. Витрати на амортизацію	$Z_{ам}$	0	760
Всього витрат за рік	I	29980	25289

Витрати на обслуговування роботи системи:

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

де: T_p – кількість годин обслуговування за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість годин на обслуговування системи зменшилось з 220 год до 180 год на рік.

$$Z_{p \text{ баз}} = 220 \cdot 100 \cdot 1,1 \cdot 1,22 = 29524 \text{ грн.}$$

$$Z_{p \text{ нов}} = 180 \cdot 100 \cdot 1,1 \cdot 1,22 = 24156 \text{ грн.}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	3041	–	760,25
Всього відрахувань	-	–	3041	–	760,25

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 ($C_{ел}$):

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

$$Z_{ел\ баз} = 0,545 \cdot 220 \cdot 3,8 = 456 \text{ грн.}$$

$$Z_{ел\ нов} = 0,545 \cdot 180 \cdot 3,8 = 373 \text{ грн.}$$

7.8 Визначення економічної ефективності програмної продукції

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

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

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

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

$$E_e = (2534 - 1689) \cdot 120 - (0,05 \cdot 1408000 + 0,4 \cdot 199177 + 0,25 \cdot 33190 + 0,1 \cdot 120000) \cdot 3/12 = 53828,5 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

де: K_p – балансова вартість основних фондів розробника.

$$T_e = \frac{1760367}{(2534 - 1689) \cdot 120 \cdot 12 / 3} = 4 \text{ роки}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_o - I_n) - E_n (K_n - K_o), \quad (7.27)$$

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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.

$$E_{en} = (29980 - 25289) - 0,25 \cdot 3041 = 3931 \text{ грн.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	120
2. Повна собівартість розробленої програми	Грн.	1689
3. Ціна розробленої програми	Грн.	2534
4. Плановий прибуток від реалізації розробленої програми	Грн.	845
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1760367
7. Загальний прибуток від реалізації програмної продукції	Грн.	101400
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	53828,5
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Роки	4
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3041
11. Величина економічного ефекту у користувача програмної продукції	Грн.	3931
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,65

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

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

$$T_{cn} = \frac{3041}{29980 - 25289} = 0,65 \text{ року.}$$

7.9 Висновки

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

КБПЗ_2023

					VKPM-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – це: система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності. Охорона праці є складовою частиною безпеки життєдіяльності [1, 2].

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [3], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ.

У розділі даної магістерської роботи висвітлюються основні питання охорони праці працівників, робота яких пов'язана з роботою за комп'ютером, планування робочого приміщення, де працюють користувачі ПК; параметри мікроклімату, освітленість робочих місць та виробничих приміщень; шумові

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

завади.

Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту.

В зв'язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов'язаних з постійною роботою за комп'ютером.

Електробезпека є одним із критичних питань для співробітників, що працюють із технікою, яка одержує живлення з електричної мережі. При невиконанні норм електробезпеки можлива поразка електричним струмом.

8.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

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

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у тому числі високочастотні) електромагнітні

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [3] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

8.3 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.4 Техніка безпеки та протипожежна профілактика

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізольований кабель. Висота проводки становить 2,2 м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1м від підлоги. Усього до складу входять п'ять розеток і дві клема заземлення. Всі обчислювальні машини з'єднані із клемою заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V, а одна, забезпечує подачу напруги в 36 V. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежна безпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки. До небезпечних і шкідливих

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однієї з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

8.5 Розрахункова частина

Запорукою безпечної роботи є виконання вимог електричної безпеки, оскільки все офісне обладнання заживлюється від електричної мережі. Одним з необхідних засобів електричної безпеки є встановлення захисного заземлення.

Початкові дані, необхідні для розрахунку захисного заземлення:

- допустимий опір розповсюдженню струму в землі від заземлювального пристрою $R_{зн} = 10 \text{ Ом}$;
 - питомий опір ґрунту в місці встановлення заземлювача $\rho_3 = 100 \text{ Ом/м}$;
 - тип ґрунту – суглинок;
 - тип заземлювача – труба, діаметром $d=0.05 \text{ м}$ і довжиною $l = 2.5 \text{ м}$;
- конструкція заземлювача – розташування заземлювачів по контуру. Розрахунок проводимо за стандартною методикою [7].

Визначимо розрахунковий опір землі:

$$\rho_{pz} = \phi \cdot \rho_3$$

де ϕ – коефіцієнт сезонності, що враховує коливання питомого опору при зміні вологості ґрунту протягом року; при використанні заземлювача довжиною $l = 2.5 \text{ м}$ при глибині закладання від вершини $h = 0.5 \text{ м}$ $\phi = 1.1$ для четвертої кліматичної зони.

Схема розташування заземлювачів показана на рисунку 8.1.

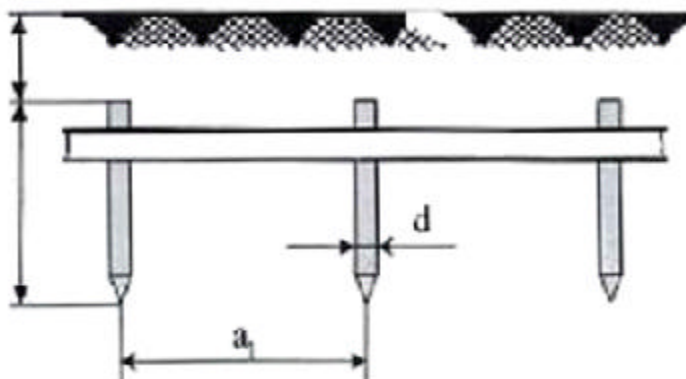


Рисунок 8.1 – Схема розташування заземлювачів

Опір землі:

$$\rho_{pz} = 1,1 \cdot 100 = 110 \text{ Ом}\cdot\text{м}$$

Опір R_B , розповсюдженню струму в землі від одного вертикального заземлювача:

$$R_B = \frac{\rho_{pz}}{2\pi \cdot l} \left(\ln \frac{2 \cdot l}{d} + 0,5 \ln \frac{4t+l}{4t-l} \right)$$

де l – довжина заземлювача ($l = 2,5$ м);

$d = 0,05$ м – діаметр заземлювача при $U < 1$ кВ та при $S < 100$ кВА;

t – відстань від поверхні до середини заземлювача:

$$t = h + l/2 = 0,5 + 2,5/2 = 1,75 \text{ м.}$$

$$R_B = \frac{110}{2 \cdot 3,14 \cdot 2,5} \left(\ln \left(\frac{2 \cdot 2,5}{0,05} \right) + 0,5 \cdot \ln \left(\frac{4 \cdot 1,75 + 2,5}{4 \cdot 1,75 - 2,5} \right) \right) = 34,325 \text{ Ом}$$

Визначаємо потрібну кількість заземлювачів:

$$n = \frac{R_g}{R_{zn}} = \frac{34,325}{10} = 3,4 \approx 4 \text{ шт.}$$

Коефіцієнт використання вертикальних заземлювачів враховує ефект екранування. При вибраному значенні $k = a/l$, де a – відстань між вертикальними заземлювачами, м; $k = 1$ при $a = 2$ м. Таким чином коефіцієнт використання вертикального заземлювача за довідковими даними дорівнює $\eta_B = 0,6$.

Кількість вертикальних заземлювачів з урахуванням коефіцієнту використання η_B приблизно складає

$$n = \frac{R_g}{R_{zn} \cdot \eta_g} = \frac{34,325}{10 \cdot 0,6} = 5,72 \approx 6 \text{ шт.}$$

Довжина горизонтального заземлювача, необхідна для розміщення вертикальних заземлювачів по контуру

$$L = a \cdot n = 2 \cdot 6 = 12 \text{ м}$$

Опір горизонтального заземлювача R_G , Ом, прокладеного на глибині $h = 0,5$ м від поверхні землі буде

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

6. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів – *Режим доступу до ресурсу: https://dnaop.com/html/2029/doc_40.1-1.21-* НПАОА 40.1-1.21-98 (дата звернення 19.09.23).

7. Розрахунки з електробезпеки. Розрахунок захисного заземлення. URL: https://cpo.stu.cn.ua/Oksana/rozrah_rozd_OP_DP_bak_spec_mag/90.html

КБПЗ_2023

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи оновлення цифрових топографічних карт геоінформаційної хмари.

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 оновлення цифрових топографічних карт геоінформаційної хмари.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем оновлення цифрових топографічних карт геоінформаційної хмари.
- Досліджена система оновлення цифрових топографічних карт геоінформаційної хмари.
- На основі отриманих результатів досліджень створена програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання оновлення цифрових топографічних карт геоінформаційної хмари.

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.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

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

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм RC5.

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 3931 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,65 роки.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шепель А.С. Дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Adam Freeman. Pro Go The Complete Guide to Programming Reliable and Efficient Software Using Golang. Apress Media. 2022. 1078 p.
3. Fernando Doglio. Skills of a Successful Software Engineer. Manning. 2022. 182 с.
4. M. Holmes He. Creating Apps with React Native. Apress Media. 2022. 445 p.
5. Maurício Aniche. Effective Software Testing. Manning Publications. 2021. 372 p
6. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
7. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
8. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
9. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
10. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
11. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
12. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.

					БКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

13. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
14. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
15. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
16. 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.
17. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.
18. 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.
19. 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>.
20. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143
21. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 120-184.
22. 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.

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

24. 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.

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

26. 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.

27. 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.

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

29. 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.

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

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

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

33. 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.

34. 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.

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

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

36. 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.

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

38. 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.

39. 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.

40. 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.

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

42. 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).

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

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

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

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

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

48. Смірнов О.А., Коваленко О.В., Мелешко Є.В., Константинова Л.В., Кожанова А.С. Інженерія програмного забезпечення. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки України від 18.03.2013 року № 1/11-5584. – Кіровоград: КНТУ 2013. – 409с.

49. Смірнов О.А., Осадчий С.І., Мелешко Є.В., Іванов С.Г., Павленко М.А., Усачов О.М. Основи технічної експлуатації АСУ. Навчальний

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова, Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки України від 27.02.2013 року № 1/11-4487 . – Кіровоград: КНТУ 2013. – 322с.

50. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. Коваленка О.В., Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки України від 26.02.2013 року № 1/11-4368. – Кіровоград: КНТУ 2013. – 257с.

51. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 17.04.2012 року № 1/11-5249. – Кіровоград: КНТУ 2012. – 250 с.

КБГЗ – 2023

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

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-122.23.0020.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Шепель А.С.				<i>Дослідження та програмна реалізація системи оновлення цифрових топографічних карт геоінформаційної хмари</i>	Літ.	Аркуш	Аркушів
Перевірів	Минайленко Р.М.					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 оновлення цифрових топографічних карт геоінформаційної хмари.

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 оновлення цифрових топографічних карт геоінформаційної хмари.

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-122.23.0020.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		

- 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 оновлення цифрових топографічних карт геоінформаційної хмари;
- 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-122.23.0020.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

					ВКРМ-122.23.0020.00.00.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 повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці програміста.

					ВКРМ-122.23.0020.00.00.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 – 96 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 12.12.2023 р.

					ВКРМ-122.23.0020.00.00.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

_____ Минайленко Р.М.

*Дослідження та програмна реалізація
системи оновлення цифрових топографічних карт геоінформаційної хмари*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 41

Літера: РП

Кропивницький – 2023 року

GPS_MAPS.DPR ТІЛО ОСНОВНОЇ ПРОГРАМИ

```

program GPS_MAPS;
{
Дослідження та програмна реалізація системи оновлення цифрових
топографічних карт геоінформаційної хмари
Виконав: студент 6 курсу,
групи КН-22М-2
напряму підготовки (спеціальності) 122 Комп'ютерні науки
Шепель Артем Сергійович
}
Uses
// об'ява бібліотек
  Forms,
// бібліотека форм ОС
  GPS_About in 'GPS_About.pas' {F2},
// Форма авторського права
  GPS_MainForm in 'GPS_MainForm.pas' {F1},
// Основна форма
  GPS in 'GPS_MAPS.pas';
// Отримання даних
{$R *.RES}

function DoFolderAction(FolderPath, ArchivePath:string;
  ActionFunction :TActionFuntion):boolean;
  Var
//розділ об'яви змінних
  ArchiveFile,CurrFile :THandle;
  Value,i,CurrFileSize,_readed,_writed:DWORD;
  _Files:TStringList;
  path_in_archive_file :string;
  pBuff:Pointer;
begin
  Result:=false;
  if FolderPath[Length(FolderPath)]='\ ' then
    FolderPath:=FolderPath+'\ ';
  _Files:=GetAllFiles('*.*',FolderPath);
  ArchiveFile := CreateFile(pchar(ArchivePath),
  GENERIC_READ+GENERIC_WRITE,FILE_SHARE_READ,0,
  CREATE_ALWAYS,0,0);
  Value:=ArchiveSignature;
  WriteFile(ArchiveFile,value,4,_writed,0);
  Value:=_Files.Count;
  WriteFile(ArchiveFile,value,4,_writed,0);
  for i:=0 to value-1 do
    begin
    if not ActionFunction(_Files.Strings[i],bkpFile) then
      begin
        CloseHandle(ArchiveFile);
        exit;
      end;
  CurrFile := CreateFile(pchar(bkpFile),GENERIC_READ,
    FILE_SHARE_READ,0,OPEN_EXISTING,0,0);
  CurrFileSize := GetFileSize(CurrFile,nil);
  if path_in_archive_file[1] = '\ ' then
    Delete(path_in_archive_file,1,1);
  Write_LS(ArchiveFile,path_in_archive_file);
  WriteFile(ArchiveFile,currFileSize,4,_writed,0);
  pBuff:=VirtualAlloc(0,MainBufferSize,
    MEM_COMMIT+MEM_RESERVE,PAGE_READWRITE);
  repeat
    ReadFile(CurrFile,pBuff^,MainBufferSize,_readed,0);
    WriteFile(ArchiveFile,pBuff^,_readed,_writed,0);
  until _writed < MainBufferSize;
  VirtualFree(pBuff,MEM_RELEASE,0);
  CloseHandle(CurrFile);
  DeleteFile(bkpFile);
end;

```

```

    CloseHandle(ArchiveFile);
    Result:=True;
end;

//пошук файлів
function GetAllFiles (Filter, Folder: string):TFilesList;
    Var
//розділ об'яви змінних
    sr: TSearchRec;
    sDirList,_FilesList,_LST: TStringList;
    i,j: Integer;
begin
    j:=0;
    _FilesList:= TStringList.create;
    _FilesList.Clear;
    if FindFirst (Folder + Filter,faAnyFile , sr) = 0 then
        repeat
            j:=j+1;
            if j 0;
            FindClose(sr);
            sDirList := TStringList.Create;
            try
// спроба з можливим викликом виключення
                GetSubDirs (Folder, sDirList);
                for i := 0 to sDirList.Count - 1 do
                    if (sDirList[i] '.') and (sDirList[i] '..') then
begin _LST:=GetAllFiles(Filter,IncludeTrailingPathDelimiter(
                Folder + sDirList[i]));
                    for j:=0 to _LST.Count-1 do
                        _FilesList.Add(_LST.Strings[j]);
                        _LST.free;
end;
                    finally
// виконати у будь якому випадку навідь при появі виключення
                        sDirList.Free;
                    end;
                    Result:= _FilesList;
end;

// процедура отримання піддиректорій
procedure GetSubDirs (Folder: string; sList: TStringList);
    Var
//розділ об'яви змінних
    sr: TSearchRec;
// структура
begin
//початок
    if FindFirst (Folder+ '*.*', faDirectory, sr) = 0 then
        try
// спроба з можливим викликом виключення
            repeat
                if (sr.Attr and faDirectory) = faDirectory then
                    sList.Add (sr.Name);
                until FindNext(sr) 0;
            finally // виконати у будь якому випадку навідь при появі виключення
                FindClose(sr);
            end;
        end;
    end;

begin // Початок основного потоку ПЗ
    Application.Initialize; // Ініціалізація
    Application.CreateForm(TF1, F1); // Створення форми
    Application.Run;//Запуск ПЗ GPS_MAPS
end.

```

ОБРОБКА КАРТ MAPS.PAS

```

unit MAPS; // Форма отримання геопараметричних даних
{
Дослідження та програмна реалізація системи оновлення цифрових
топографічних карт геоінформаційної хмари
Виконав: студент 6 курсу,
групи КН-22М-2
напряму підготовки (спеціальності) 122 Комп'ютерні науки
Шепель Артем Сергійович
}
Interface
// Інтерфейсна частина - об'ява даних
// об'ява бібліотек
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, ExtCtrls, buttons, grids, stdctrls, clipbrd, Printers;

Type // об'ява власних типів даних
TMouseEvent = procedure (Sender: TObject; Shift: TShiftState; Points: trect)
of object;
TMAPSdataArray=array[0..16]of double;

TMAPS = class(TPanel) // розроблений клас від TPanel
private
MAXELNUM:integer; //об'ява змінних класу
npolypoints:word;
MapLimits:trect;
Titulo: string;
Nbloqs:word;
viewpct,viewext,viewbmp:trect;
FLegendChartPos,FLegendBulletsPos,FLegendShadowsPos:trect;
FZoomBox:trect;
arrfname:string;
bitmap:tbitmap;
backbitmap:tbitmap;
logobitmap:tbitmap;
tometa:boolean;
bmpfname,polfname:string;
Fready:boolean;
sel:TSELARRAY;
BB_anchorx,BB_anchory,BB_curx,BB_cury:integer;
BB_drawing:boolean;
FPaintMode:integer;
BackLink:array[1..5] of string;
nBacks:integer;
fgrid:integer;
paleta:hpalette;
FOnMouseDown:TMouseEvent;
FOnMouseUp:TMouseEvent;
FOnMouseMove:TMouseMoveEvent;
FOnMouseLine:TMouseEvent;
// Обробка сповіщень WM_...
Procedure WMRButtonDown(var msg:TWMRButtonDown); message WM_RBUTTONDOWN;
Procedure WMLButtonUp(var msg:TWMLButtonUp); message WM_LBUTTONUP;
Procedure WMRButtonUp(var msg:TWMRButtonUp); message WM_RBUTTONUP;
Procedure WMMouseMove(var msg:TWMMouseMove); message WM_MOUSEMOVE;
FOnMouseBox:TMouseEvent;
procedure time(sender:tobject);
procedure replaceglis;
Procedure CalcChartRatios;
Procedure SetPBgGPS(k:integer;c:Tmaps);
function GetPBgGPS(k:integer):Tmaps;
Procedure SetPFgGPS(k:integer;c:Tmaps);
function GetPFgGPS(k:integer):Tmaps;
Procedure SetBitMap(fname:string);
Procedure DrawBitMap(h:hdc);
Procedure CleanHDC(h:hdc;col:Tmaps;box:trect);
Procedure Meridians(h:hdc;box:trect);
Procedure SetHDCFrame(h:hdc;r:trect);

```

```

Procedure SetChartValue(element,item:word;value:real);
Function GetChartValue(element,item:word):real;
Procedure WMF_clip2file(wnd:HWND;fname:string);
Procedure SetPPattern(k:integer;c:byte);
function GetPPattern(k:integer):byte;
Procedure SetChartLegend(k:word;s:string);
function GetChartLegend(k:word):string;
procedure ClpWMF2Text(fname:string;textfname:string);
Function ClpWMFRect(fname:string):trect;
Procedure ScaleClpWMF2Text(fname,textfname:string;iso:boolean);
Procedure RectToPolyLine(r2:trect;var cuadro:array of tpoint);
procedure adaptlegrect(r,r1:trect;var r2:trect);
Procedure SetFont(i:integer;f:tfont);
Procedure MyMouseMove(Sender:TObject; Shift:TShiftState; X, Y: Integer);
Procedure WMLButtonDown(var msg:TWMLButtonDown); message WM_LBUTTONDOWN;
Procedure Loaded;Override;

public
drw:tdraws; //об'ява змінних класу
np:TSPOLARRAY;
Arr:TSArrange;
InfoPanel,
CLegPanel,
BLegPanel,
SLegPanel,
ChartPanel:TGMovPan;
UserLine:record
    p:array[1..300] of tpoint;
    np:integer;
    drawing:boolean;
    dist:double;
end;
MyVersion:string;
gli:TGlyphList;
timer:ttimer;
Procedure LoadGlyphsBitmap(fname:string);
procedure addgli(p:tpoint);
procedure placegli(ix:integer;p:tpoint);
function GlyphCount:integer;
Procedure GlyphSettings(ixglyph:integer;imageix:integer;
    animate:boolean;enable:boolean);
constructor create(sender:tcomponent);override;
destructor destroy;override;
Procedure paint;override;
Procedure CreatePanels;
Procedure FromFile(fname:string);
Procedure ToFile(fname:string);
Procedure GetPoint(elem,ix:integer;var c:tcitypoint);
Function RemovePoint(elem,ix:integer):boolean;
Function GetPointCount(elem:integer):integer;
Procedure ClearPoints(elem:integer);
Procedure SetSymbol(f:tfont;c:char);
Procedure ForwardLink(ix:integer);
Procedure BackwardLink;
Procedure LoadPaletteFromFile(filename:string);
Procedure LoadMosaicBitmap(fname:string);
Procedure SetMapOptions(var op:tmapop);
Procedure SetChartOptions(var op:tchop);
Procedure SetBulletOptions(var op:BullOptions);
Procedure SetShadowOptions(var op:tshop);
Procedure SetThemeShadowOptions(var op:TSHThem);
Procedure SetBackgroundOptions(var op:tbkop);
Procedure SetLayerOptions(var op:TLAYOP);
Procedure GetMapOptions(var op:tmapop);
Procedure GetChartOptions(var op:tchop);
Procedure GetBulletOptions(var op:BullOptions);
Procedure GetShadowOptions(var op:tshop);
Procedure GetThemeShadowOptions(var op:TSHThem);
Procedure GetBackgroundOptions(var op:tbkop);
Procedure GetLayerOptions(var op:TLAYOP);

```

```

Procedure Merge(fname:string;layeroffset:integer; x,y:integer);
Procedure Flip(horz,vert:boolean);
Procedure ScaleTo(r:trect);
Procedure AddPoly(closed:boolean);
Procedure DeleteElements;
Procedure JoinElements(orig,dest:integer);
Procedure ToFront(elem:integer);
Procedure ToBack(elem:integer);
Procedure ReMap(var m:array of integer);
// завдання властивості наступним типам даних
Property Ready:boolean read Fready write fready;
Property NumBox:boolean read GetNumBox write SetNumBox;
Property Selected[i:word]:boolean read GetElSelected write SetElSelected;
Property Pol_Layer[i:word;j:word]:byte read GetPolLayer write SetPolLayer;
Property PolCount[i:word]:byte read GetPolCount;
Property PolNames[i:word;j:word]:string read GetPolName write SetPolName;
Property Links[i:word]:string read GetElmLink write SetElmLink;
Property Fonts[i:integer]:Tfont read GetFont write SetFont;
Property canvas;
Property PaintMode:integer read FPaintMode write FPaintMode;
Property ElementCount:word read npolypoints;
Property ElementName[i:word]:string read getElmName write SetElmName;
Property ZoomBox:TRECT read FZoomBox write FZoomBox;
property PBgGPS[i:integer]:Tmaps read GetPBgGPS write SetPBgGPS;
property PFgGPS[i:integer]:Tmaps read GetPFgGPS write SetPFgGPS;

Implementation // Реалізація об'явлених даних у інтерфейс ній частині

// конструктор класу TMAPS
constructor TMAPS_DATA.create(sender:tcomponent);
begin //початок
inherited create(sender);
ready:=false;
fgrid:=8;
MaxElNum:=TMAPS_MAXELEMENTS;
MyVersion:=TMAPS_Version;
BackLink[1]:='';
nbacks:=0;

FPaintMode:=0;
UserLine.drawing:=false;

npolypoints:=0;
Arr:=nil;
bitmap:=nil;
backbitmap:=tbitmap.create;
logobitmap:=tbitmap.create;

drw:=tdraws.create;
tometa:=false;
Align:=AlClient;

paleta:=GetDefaultPalette;
FLegendChartPos:=rect(1,70,30,99);
FLegendBulletsPos:=rect(31,70,60,99);
FLegendShadowsPos:=rect(61,70,98,99);
bmpfname:='';
arrfname:='';
polfname:='';
zoombox:=rect(1,1,100,100);
viewpct:=rect(1,1,99,99);
end;
// завантаження
Procedure TMAPS_DATA.Loaded;
Begin //початок
  Inherited Loaded;
  CreatePanels;
end;

```

```

procedure createmasks (bm:tbitmap);
var t:Tmaps;
    x,y,h:integer;
begin
    h:=bm.height;
    bm.height:=h*2;
    t:=bm.canvas.pixels[0,0];
    for x:=0 to bm.width-1 do for y:=0 to h-1 do
        if bm.canvas.pixels[x,y]=t then bm.canvas.pixels[x,y+h]:=$00FFFFFF
        else bm.canvas.pixels[x,y+h]:=$00000000
    end;
end;

```

```

begin
if not (ComponentState=[]) then exit;
gli:=TGlyphList.create(canvas);
timer:=ttimer.create(self);
timer.interval:=500;
timer.ontimer:=time;
timer.enabled:=false;
InfoPanel:=TGMovPan.create(self);
infopanel.Left:= 32;    infopanel.width:= 64;
infopanel.top:=8;    infopanel.height:= 20;
infopanel.Align:= alnone;    infopanel.marge:=3;
infopanel.borderwidth:=0;    infopanel.bevelwidth:=1;
infopanel.BevelOuter:= bvLowered;    infopanel.BevelInner:= bvNone;
infopanel.Caption:= '';    infopanel.visible:=true;
InfoPanel.parent:=self;
if bitmap<>nil then SetBitmap(bmpfname);
if polfname<>' then Fromfile(polfname);
end;

```

```

// деструктор класу TMAPS
destructor TMAPS_DATA.destroy;
Var //розділ об'яви змінних
i:integer;
begin
gli.Free;
for i:=1 to npolypoints do np[i].free;
npolypoints:=0;
arr.free;
bitmap.free;
backbitmap.free;
logobitmap.free;
drw.free;
DeleteObject(paleta);
timer.free;

```

```

InfoPanel.free;
CLegPanel.free;
BLegPanel.free;
SLegPanel.free;
ChartPanel.free;
inherited destroy;
end;

```

```

//отримання даних з файлу
Procedure TMAPS_DATA.FromFile (fname:string);
var f:file;
i,j:integer;
ax:shortstring;
bloque:record
codigo:word;
nombre:string[50];
numpolis:word;
nummarcos:word;
end;
elemento:record
    fillGPS,borderGPS:longint;

```

```

numpuntos:word;
layer:word;
other:longint;
nombre:str50;
end;
buffer:^SPltSmallPoint;

tfn:tfontname;
rt:trect;
nupu,ipu:word;
c:tsapoint;

begin //початок
if not fileexists(fname) then exit;
polfname:=fname;
if not (ComponentState=[]) then exit;

new(buffer);
assignfile(f,fname); // підключення файлу
reset(f,1); // встановлення типу роботи з файлом

blockread(f,ax,100); // читання даних з фалу
if ax<>MyVersion then begin
closefile(f); // закриття файлу
dispose(buffer);
exit;
end;
screen.cursor:=crHourGlass;
for i:=1 to npolypoints do np[i].free;
npolypoints:=0;
ready:=false;
c:=tsapoint.create;
try
// спроба з можливим викликом виключення
if nbloqs> MaxElNum then nbloqs:=maxelnum;
npolypoints:=nbloqs;
for i:=1 to npolypoints do begin
blockread(f,bloque,sizeof(bloque)); // читання даних з фалу
np[i]:=TSPolyPol.create(bloque.nummarcos,bloque.numpolis,bloque.nombre);
for j:=1 to bloque.nummarcos do begin
blockread(f,elemento,sizeof(elemento)); // читання даних з фалу
blockread(f,buffer^,sizeof(tSmallPoint)*elemento.numpuntos);
np[i].SetMValsSmall(elemento.fillGPS,elemento.borderGPS,j,elemento.numpuntos,buffer^);
np[i].m[j].drawer:=drw;
np[i].m[j].layer:=elemento.layer;
np[i].m[j].nombre:=elemento.nombre;
end;

for j:=1 to bloque.numpolis do begin
blockread(f,elemento,sizeof(elemento)); // читання даних з фалу
blockread(f,buffer^,sizeof(tSmallPoint)*elemento.numpuntos);
np[i].SetPValsSmall(elemento.fillGPS,elemento.borderGPS,j,elemento.numpuntos,buffer^);
np[i].p[j].drawer:=drw;
np[i].p[j].layer:=elemento.layer;
np[i].p[j].nombre:=elemento.nombre;
end;
blockread(f,nupu,sizeof(nupu)); // читання даних з фалу
np[i].npoints:=nupu;
for ipu:=1 to nupu do begin
blockread(f,c.c,sizeof(c.c)); // читання даних з фалу
np[i].points[ipu]:=c;
end;
blockread(f,np[i].eo,sizeof(np[i].eo)); // читання даних з фалу
blockread(f,np[i].link,sizeof(np[i].link)); // читання даних з фалу
blockread(f,np[i].values,sizeof(np[i].values)); // читання даних з фалу
np[i].CalcGPS;
np[i].drawer:=drw;

```

```

end;
finally // виконати у будь якому випадку навідь при появи виключення
closefile(f); // закриття файлу
dispose(buffer);
c.free;
drw.setfonts;
arr.free;
arr:=TSArrange.create(2,npolypoints,'Normal');
if bmpfname<>' ' then setbitmap(bmpfname);
if arrfname<>' ' then ArrFromFile(arrfname);
CalcGPS;
ready:=true;
zoombox:=MapLimits;
infopanel.caption:=title;
adaptlegrect(rect(left,top,left+width,top+height),FLegendChartPos,rt);
CLegPanel.Left:= rt.left; CLegPanel.width:= rt.right-rt.left;
CLegPanel.top:=rt.top; CLegPanel.height:= rt.bottom-rt.top;
adaptlegrect(rect(left,top,left+width,top+height),FLegendBulletsPos,rt);
BLegPanel.Left:= rt.left; BLegPanel.width:= rt.right-rt.left;
BLegPanel.top:=rt.top; BLegPanel.height:= rt.bottom-rt.top;
adaptlegrect(rect(left,top,left+width,top+height),FLegendShadowsPos,rt);
SLegPanel.Left:= rt.left; SLegPanel.width:= rt.right-rt.left;
SLegPanel.top:=rt.top; SLegPanel.height:= rt.bottom-rt.top;
infopanel.Left:= left+4;
infopanel.top:=top+4;
repaint;
screen.cursor:=crdefault;
end;
end;

//збереження даних до вайлу
Procedure TMAPS_DATA.ToFile(fname:string);
var f:file;
i,j:integer;
ax:string;
bloque:record
codigo:word;
nombre:string[50];
numpolis:word;
nummarcos:word;
end;
elemento:record
fillGPS,borderGPS:longint;
numpuntos:word;
layer:word;
other:longint;
nombre:str50;
end;
buffer:^SPltSmallPoint;
nupu,ipu:word;
c:tsapoint;
begin //початок
new(buffer);
c:=tsapoint.create;
assignfile(f,fname); // підключення файлу
rewrite(f,1); // встановлення типу роботи з файлом
fillchar(ax,100,20);
ax:=MyVersion;
i:=0;
nbloqs:=npolypoints;
blockwrite(f,nbloqs,sizeof(nbloqs));
for i:=1 to npolypoints do begin
bloque.nummarcos:=np[i].nmarcos;
bloque.numpolis:=np[i].npolis;
bloque.nombre:=np[i].nombre;
bloque.codigo:=i;
blockwrite(f,bloque,sizeof(bloque));
for j:=1 to bloque.nummarcos do begin
elemento.fillGPS:=np[i].m[j].bkcol;

```

```

elemento.borderGPS:=np[i].m[j].frcol;
elemento.numpuntos:=np[i].m[j].n;
elemento.layer:=np[i].m[j].layer;
elemento.other:=0;
elemento.nombre:=np[i].m[j].nombre;
blockwrite(f,elemento,sizeof(elemento));
Copy32To16(np[i].m[j].pt^,buffer^,elemento.numpuntos);
blockwrite(f,buffer^,sizeof(TSmallPoint)*elemento.numpuntos);
end;
for j:=1 to bloque.numpolis do begin
elemento.fillGPS:=np[i].p[j].bkcol;
elemento.borderGPS:=np[i].p[j].frcol;
elemento.numpuntos:=np[i].p[j].n;
elemento.layer:=np[i].p[j].layer;
elemento.other:=0;
elemento.nombre:=np[i].p[j].nombre;
blockwrite(f,elemento,sizeof(elemento));
Copy32To16(np[i].p[j].pt^,buffer^,elemento.numpuntos);
move( np[i].p[j].pt^,buffer^,elemento.numpuntos*sizeof(tpoint));
blockwrite(f,buffer^,sizeof(tsmallpoint)*elemento.numpuntos);
end;
nupu:=np[i].npoints;
blockwrite(f,nupu,sizeof(nupu));
for ipu:=1 to nupu do begin
c:=np[i].points[ipu];
blockwrite(f,c.c,sizeof(c.c));
end;
  blockwrite(f,np[i].eo,sizeof(np[i].eo));
  blockwrite(f,np[i].link,sizeof(np[i].link));
  blockwrite(f,np[i].values,sizeof(np[i].values));
end;

closefile(f); // закриття файлу
dispose(buffer);
c.free;
end;

//читання даних
Procedure TMAPS_DATA.ReadData(fname:string);
var f:file of TMAPSdataArray; TMAPSdataArray=array[0..16]of double;
ad:TMAPSdataArray;
i,j:integer;
begin
assignfile(f,fname); // підключення файлу
reset(f); // встановлення типу роботи з файлом
i:=1;
try // спроба з можливим викликом виключення
while ((not eof(f)) and (i<elementcount)) do begin
read(f,ad);
for j:=0 to 16 do np[i].values[j]:=ad[j];
inc(i);
end;
finally // виконати у будь якому випадку навідь при появі виключення
closefile(f); // закриття файлу
end;
end;

//зберігання даних
Procedure TMAPS_DATA.SaveData(fname:string);
var f:file of TMAPSdataArray; TMAPSdataArray=array[0..16]of double;
ad:TMAPSdataArray;
i,j:integer;
begin
assignfile(f,fname); // підключення файлу
rewrite(f); // встановлення типу роботи з файлом
try
/ спроба з можливим викликом виключення
for i:=1 to elementcount do begin
for j:=0 to 16 do ad[j]:=np[i].values[j];

```

```

write(f,ad);
end;
finally // виконати у будь якого випадку навідь при появи виключення
closefile(f); // закриття файлу
end;
end;
// отримання даних форми - шрифту
Function TMAPS_DATA.GetFont(i:integer):tfont;
Begin //початок
case i of
1:GetFont:=drw.font1;
3:GetFont:=drw.font3;
4:GetFont:=drw.font4;
5:GetFont:=drw.font5;
else GetFont:=drw.font2;
end;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA ArrCalcBoxes
Procedure TMAPS_DATA.ArrCalcBoxes;
var i,j:integer;
begin
for i:=1 to arr.n do begin
preprect(arr.caja[i]);
for j:=1 to arr.t do if arr.matr.bit[i,j] then begin
comprect(np[j].caja,arr.caja[i]);
end;end;end;

// реалізація внутрішньої процедури класу TMAPS_DATA ArrFromFile
Procedure TMAPS_DATA.ArrFromFile(fname:string);
Begin
arrfname:=fname;
if not fileexists(fname) then exit;
arr.fromfile(fname);
arrcalcboxes;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA getElmName
Function TMAPS_DATA.getElmName(i:word):string;
begin
if ((i>0) and (i<=npolypoints)) then GetElmName:=np[i].nombre
else GetElmName:='';
end;

// реалізація внутрішньої процедури класу TMAPS_DATA SetElmName
Procedure TMAPS_DATA.SetElmName(i:word;s:string);
begin
if ((i>0) and (i<=npolypoints)) then np[i].nombre:=s;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA SetSel
Procedure TMAPS_DATA.SetSel(lb:tlistbox);
var i:integer;
begin
for i:=1 to lb.items.count do sel[i]:=(lb.selected[i-1]);
end;
// реалізація внутрішньої процедури класу TMAPS_DATA IsArrSelected
Function TMAPS_DATA.IsArrSelected(father:word):boolean;
var i:integer;
begin //початок
IsArrSelected:=true;
for i:=1 to npolypoints do if sel[i] then
if arr.Esta(father,i) then exit;
IsArrSelected:=false;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA DoChart
Procedure DoChart(h:hdc;r:trect);
var kx,ky,i,j:integer;

```

```

ox,oy:integer;
v:integer;
x0,y0:integer;
bola:integer;
ax:string;
mx:integer;
maximovalor,total:real;
caxa:trect;
pu:tpoint;
suma:array[0..16] of integer;
ds:array[1..16] of double;
dd:double;
a:array[0..255] of char;
centro:tpoint;
begin
ChartPanel.font:=font;
chartpanel.GPS:=drw.co.boxbackGPS;
centro.x:=(r.left+r.right) div 2;
centro.y:=(r.top+r.bottom) div 2;
selectobject(h,getstockobject(BLACK_PEN));
for i:=1 to 16 do ds[i]:=0;
maximovalor:=0;
total:=0;
for i:=drw.co.nvalues downto 1 do
for j:=1 to elementcount do if sel[j] then begin
ds[i]:=ds[i]+np[j].values[i];
if ds[i]>maximovalor then maximovalor:=ds[i];
end;
if maximovalor=0 then maximovalor:=1;
for i:=1 to drw.co.nvalues do total:=total+ds[i];

SetBkMode(h,1);
setTextAlign(h,TA_CENTER+TA_TOP);
str(total:3:drw.co.ndecimalS,ax);
ax:='Всього: '+ax;
strcpy(a,ax);
textout(h,centro.x,r.top+2,a,length(ax));
caxa:=r;
ox:=(caxa.right-caxa.left) div 5;
oy:=(caxa.bottom-caxa.top) div 5;
kx:=(caxa.right-caxa.left-2*ox) div drw.co.nvalues;
ky:=3*oy;
bola:=4;
mx:=kx div 2;
if ((ox<5) or (oy<5)) then exit;

if drw.co.style<>4 then begin
movetoex(h,caxa.left+ox,caxa.bottom-oy,nil);
lineto(h,caxa.left+ox+drw.co.nvalues*kx,caxa.bottom-oy);
MoveToEx(h,caxa.left+ox,caxa.bottom-oy,nil);
lineto(h,caxa.left+ox,caxa.bottom-oy - ky);
end;

case drw.co.style of
0:begin
for i:=1 to drw.co.nvalues do begin
drw.getbrocha(h,drw.co.patt[i],drw.co.lineGPS,drw.co.GPSs[i]);
v:=trunc(ds[i]*ky/maximovalor);
rectangle(h,
caxa.left+ox+(i-1)*kx,
caxa.bottom-oy,
caxa.left+ox+i*kx,
caxa.bottom-oy - v);
drw.releasebrocha;
end;
end;
3:begin
for i:=1 to drw.co.nvalues do begin
v:=trunc(ds[i]*ky/maximovalor);

```

```

drw.bar3d(h,
          caxa.left+ox+(i-1)*kx,
          caxa.bottom-oy,
          caxa.left+ox+i*kx,
          caxa.bottom-oy - v,i);
end;
end;

1,2:begin
drw.getpen(h,3,drw.co.GPSs[1]);
x0:=caxa.left+ox+mx;y0:=caxa.bottom-oy -trunc(ds[1]*ky/maximovalor);
MoveToEx(h,x0,y0,nil);
for i:=1 to drw.co.nvalues do begin
  v:=trunc(ds[i]*ky/maximovalor);
  lineto(h,caxa.left+ox+i*kx-mx,caxa.bottom-oy - v);
end;
drw.releasepen;

if drw.co.style=2 then begin
drw.getbrocha(h,drw.co.patt[2],drw.co.LineGPS,drw.co.GPSs[2]);
for i:=1 to drw.co.nvalues do begin
  v:=trunc(ds[i]*ky/maximovalor);
end;
drw.releasebrocha;
end;
end;
4:begin
  pu:=centro;
  dd:=0; for i:=1 to drw.co.nvalues do begin
    dd:=dd+ds[i];
    ds[i]:=dd;
  end;
  suma[0]:=0; v:=0;
  for i:=1 to drw.co.nvalues do begin
    v:=v+trunc(ds[i]*360/dd);
    suma[i]:=trunc(ds[i]*360/dd);
  end;
  drw.Queso(h,pu.x,pu.y,drw.co.PieSize,0,360,true,
    drw.co.GPSs[1],drw.co.lineGPS,drw.co.GPSs[1]);
  for i:=1 to drw.co.nvalues do
    drw.Queso(h,pu.x,pu.y,drw.co.PieSize,suma[i-1],suma[i],false,
    drw.co.GPSs[1],drw.co.GPSs[i],drw.co.lineGPS);
  end;
end;
if drw.co.style<>4 then begin
  ox:=ox- kx div 2;
  for i:=1 to drw.co.nvalues do begin
    if drw.co.ndecimal=0 then ax:=inttostr(trunc(ds[i]))
    else str(ds[i]:3:drw.co.ndecimal,ax);
    v:=trunc(ds[i]*ky/maximovalor);
    drw.ContrastText(h,caxa.left+ox+i*kx ,
      caxa.bottom-oy - v,ax,3,false);
  end;
end;

end;
// реалізація внутрішньої процедури класу TMAPS_DATA SelObjects
procedure SelObjects(c:Tmaps;f:tfont);
begin
oldf:=selectobject(h,f.handle);
settexTmaps(h,f.GPS);
drw.getbrocha(h,0,f.GPS,c);
rectangle(h,rto.left,rto.top,rto.right,rto.bottom);
drw.releasebrocha;
end;
Var //розділ об'яви змінних
  mm,ww:double;
begin
chartpanel.visible:=true;

```

```

if not tometa then begin
h:=ChartPanel.canvas.handle;
rto:=rect(1,1,ChartPanel.width,ChartPanel.height);
SelObjects(ChartPanel.GPS,drw.font4);
Dochart(h,rto);
end
else begin
mm:=(r.right-r.left)/width;
ww:=(r.bottom-r.top)/height;
rto.left:=trunc(r.left+ChartPanel.left*mm);
rto.top:=trunc(r.top+ChartPanel.top*ww);
rto.right:=trunc(r.left+(ChartPanel.left+ChartPanel.width)*mm);
rto.bottom:=trunc(r.top+(ChartPanel.top+ChartPanel.height)*ww);
DoChart(h,rto);
end;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA ChartLegend
Procedure ChartLegend(h:hdc;r:trect);
var i:integer;
wy,wb,wm:integer;
a:array[0..255] of char;
begin //початок
wy:=(r.bottom-r.top) div (drw.co.nvalues+1);
wb:=trunc(0.60*wy);
wm:=trunc(0.20*wy);
if abs(font.size) >wy then font.size:=wy;
SetBkMode(h,1);
setTextAlign(h,TA_CENTER+TA_TOP);
strcpy(a,drw.co.title);
textout(h,(r.left+r.right) div 2,r.top+2,a,length(drw.co.title));
setTextAlign(h,TA_LEFT+TA_BOTTOM);
for i:=1 to drw.co.nvalues do begin
Bar(h,r.left+wb,r.top+wy*i+wm,
r.left+wb+wb,r.top+wy*i+wm+wb,i);
strcpy(a,drw.co.LegTexts[i]);
textout(h,r.left+3*wb,r.top+wy*i+wm+wb,a,length(drw.co.LegTexts[i]));
end;
end;
Procedure BulletLegend(h:hdc;rl:trect);
var i,j,k,k0,k2:integer;
wy,wm,wb:integer;
a:array[0..255] of char;
q:real;
ax:string;
pc:pchar;
oldf:hfont;
h1,h2,h3:integer;
f:tfont;
fh1:integer;
begin
case drw.bo.applyAs of
0,2:begin
k0:=trunc(drw.bo.size);
wb:=k0 div drw.bo.legendpoints;
q:=drw.maxval0 / drw.bo.legendpoints;
wy:=(rl.bottom-rl.top) div (drw.bo.legendpoints+1);
wm:=trunc(0.20*wy);
if abs(font.size) >wy then font.size:=wy;
SetBkMode(h,1);
setTextAlign(h,TA_CENTER+TA_TOP);
strcpy(a,drw.co.LegTexts[drw.bo.BulletSource]);
textout(h,(rl.left+rl.right) div
2,rl.top+2,a,length(drw.co.LegTexts[drw.bo.BulletSource]));
setTextAlign(h,TA_LEFT+TA_BOTTOM);
for i:=1 to drw.bo.legendpoints do begin
j:=drw.bo.legendpoints-i+1;
drw.fontsymbol.size:=k*2;
k2:=k ;

```

```

        pc:=@drw.bo.thesymbol;
        settexTmaps(h,drw.bo.lineGPS);
        setbkmode(h,1);
        oldf:=selectobject(h,drw.fontsymbol.handle);
        fh1:=abs(drw.fontsymbol.height) div 2 ;
        setTextAlign(h,TA_LEFT+TA_TOP);
        textout(h,rl.left+wm,rl.top+wy*i+wm-fh1,pc,1);
        selectobject(h,oldf);
    str(q*j:3:drw.mo.decimals,ax);
    strpcopy(a,ax);
    textout(h,rl.left+2*k0+wm,rl.top+wy*i+wm,a,length(ax));
    if drw.bo.applyAs=0 then begin
        movetoex(h,rl.left+wm+k0 ,rl.top + wy * i ,nil);
        lineto(h,rl.left+2*k0+wm,rl.top + wy * i);
    end;
end;
end;
1:begin
    wy:=(rl.bottom-rl.top) div (drw.bo.legendpoints+1);
    wb:=trunc(0.80*wy);
    wm:=trunc(0.10*wy);

    k:=100 div drw.bo.legendpoints;
    q:=drw.maxval0 / drw.bo.legendpoints;

    if abs(font.size) >wy then font.size:=wy;
    SetBkMode(h,1);
    setTextAlign(h,TA_CENTER+TA_TOP);
    ax:=drw.co.LegTexts[drw.bo.BulletSource];
    strpcopy(a,ax);
    textout(h,(rl.left+rl.right) div 2,rl.top+2,a,length(ax));
    setTextAlign(h,TA_LEFT+TA_BOTTOM);

    for i:=1 to drw.bo.legendpoints do begin
        drw.getbrocha(h,0,drw.bo.shadows[k*i],drw.bo.shadows[k*i]);
        ellipse(h,rl.left+wb,rl.top+wy*i+wm,rl.left+wb+wb,rl.top+wy*i+wm+wb);
        drw.releasebrocha;
        str(q*i:3:drw.mo.decimals,ax); strpcopy(a,ax);
        textout(h,rl.left+3*wb,rl.top+wy*i+wm+wb,a,length(ax));
    end;

end;
5:begin
    rectangle(h,rl.left,rl.top,rl.right,rl.bottom);
    h1:=rl.top+font.size*3;
    h2:=rl.top+rl.bottom -h1;
    h2:=h2 div drw.bo.custom.nranges;

    setTextAlign(h,TA_CENTER+TA_TOP);
    strpcopy(a,drw.co.LegTexts[drw.bo.BulletSource]);
    textout(h,(rl.left+rl.right) div
2,rl.top+2,a,length(drw.co.LegTexts[drw.bo.BulletSource]));
    setTextAlign(h,TA_LEFT+TA_TOP);

    for i:=1 to drw.bo.custom.nranges do begin
        strpcopy(a,drw.bo.custom.ranges[i].text);
        textout(h,44,h1+(i-1)*h2 + (drw.bo.custom.ranges[i].size div
2),a,length(drw.bo.custom.ranges[i].text));
    end;
    settexTmaps(h,drw.bo.lineGPS);
    f:=tfont.create;
    f.name:=drw.bo.custom.fontname;
    for i:=1 to drw.bo.custom.nranges do begin
        f.size:=drw.bo.custom.ranges[i].size;
        oldf:=selectobject(h,f.handle);
        pc:=@drw.bo.custom.ranges[i].symbol;
        textout(h,16,h1+(i-1)*h2 ,pc,1);
        selectobject(h,oldf);
    end;
end;

```

```

    f.free;
  end;
end;
end;
// реалізація внутрішньої процедури класу TMAPS_DATA CalcChartRatios
Procedure TMAPS_DATA.CalcChartRatios;
var i,j:integer;
mx,mx0,mx1:real;
begin //початок
mx:=0;mx0:=0;mx1:=0;
for i:=1 to npolypoints do if sel[i] then begin
if np[i].values[drw.bo.bulletsource]>mx0 then
mx0:=np[i].values[drw.bo.bulletsource];
if np[i].values[drw.so.source]>mx1 then mx1:=np[i].values[drw.so.source];
for j:=1 to drw.co.nvalues do
if np[i].values[j]>mx then mx:=np[i].values[j];
end;
drw.maxval0:=mx0;
drw.maxvall:=mx1;
drw.maxvals:=mx;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA SetHDCFrame
Procedure TMAPS_DATA.SetHDCFrame(h:hdc;r:trect);
begin
viewext:=MapLimits;
drw._view:=fzoombox;
drw._back:=r;
adaptlegrect(r,viewpct,drw._frame);
if drw.mo.aspect=1 then begin
if drw._view.right-drw._view.left<drw._view.bottom-drw._view.top
then begin drw._view.right:=drw._view.left+drw._view.bottom-drw._view.top;end
else begin drw._view.bottom:=drw._view.top+drw._view.right-drw._view.left;end;
end;
if (drw._view.right-drw._view.left) >0 then
drw.Rx:=(drw._frame.right-drw._frame.left)/(drw._view.right-drw._view.left)
else drw.Rx:=1;
if (drw._view.bottom-drw._view.top)>0 then
drw.Ry:=(drw._frame.bottom-drw._frame.top)/(drw._view.bottom-drw._view.top)
else drw.Ry:=1;
if drw.rx=0 then drw.rx:=1;
if drw.ry=0 then drw.ry:=1;
end;

Procedure TMAPS_DATA.FillNames(lb:tlistbox);
var i,j,k:integer;
begin //початок
lb.clear;
for i:=1 to npolypoints do lb.items.add(inttostr(i)+':'+np[i].nombre);
end;

Procedure TMAPS_DATA.FillArranges(lb:tlistbox);
begin
if not ready then exit;
arr.alltolistbox(lb);
end;

procedure TMAPS_DATA.ArrSet(father,son:word;ok:boolean);
begin //початок
if not ready then exit;
arr.Poner(father,son,ok);
end;

Procedure TMAPS_DATA.ArrFromListBox(lb:tlistbox;father:word);
begin
if not ready then exit;
arr.FromListBox(lb,father);
ArrCalcBoxes;
end;

```

```

Procedure TMAPS_DATA.ArrToListBox(lb:tlistbox;father:word);
begin
if not ready then exit;
arr.ToListBox(lb,father);
end;

// реалізація внутрішньої процедури класу TMAPS_DATA ArrSetName
Procedure TMAPS_DATA.ArrSetName(name:string;father:word);
begin
if not ready then exit;
if father>arr.n then exit;
arr.titulos[father]^:=name;
end;

Procedure TMAPS_DATA.ArrSetNumber(n:word);
begin
arr.setnumber(n);
end;

Function TMAPS_DATA.ArrGetNumber:word;
begin
ArrGetNumber:=arr.n;
end;

Procedure TMAPS_DATA.SelArrGroup(father:integer;ok:boolean);
var i:integer;
begin
if father>0 then
if father<=arr.n then
for i:=1 to arr.t do if arr.Matr.Bit[father,i] then sel[i]:=ok;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA GetElSelected
Function TMAPS_DATA.GetElSelected(i:word):boolean;
begin
GetElSelected:=false;
if i>0 then if i<=npolypoints then GetElSelected:=Sel[i];
end;

Procedure TMAPS_DATA.SetElSelected(i:word;ok:boolean);
begin
if i>0 then if i <=npolypoints then Sel[i]:=ok;
end;

Procedure TMAPS_DATA.SetAllSelections(ok:boolean);
var i:integer;
begin
for i:=1 to elementcount do selected[i]:= ok;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA SetBitMap
Procedure TMAPS_DATA.SetBitMap(fname:string);
begin
bmpfname:=fname;
bitmap.free;
bitmap:=nil;
if fileexists(fname) then begin
bitmap:=tbitmap.create;
bitmap.loadfromfile(fname);
drw.mo.mapimage:=2;
end;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA DrawBitMap
Procedure TMAPS_DATA.DrawBitMap(h:hdc);
var ew,ow:longint;
w:trect;

```

```

vex,vey,vox,voy:integer;
rwb:record x,y:real;end;
wb:trect;
bm:tbitmap;
i,j:integer;
wx,wy:double;
wf:trect;

begin //початок
if bitmap=nil then exit;
if drw.mo.mapimage=0 then exit;
if drw.mo.mapimage=2 then begin
  wx:=Bitmap.width/(viewext.right-viewext.left);
  wy:=Bitmap.height/(viewext.bottom-viewext.top);

  w:=drw._view;
  if w.left<viewext.left then w.left:=viewext.left;
  if w.top<viewext.top then w.top:=viewext.top;
  if w.right>viewext.right then w.right:=viewext.right;
  if w.bottom>viewext.bottom then w.bottom:=viewext.bottom;
  wb.left:=Trunc((w.left-viewext.left)*wx);
  wb.right:=Trunc((w.right-viewext.left)*wx);
  wb.top:=Trunc((w.top-viewext.top)*wy);
  wb.bottom:=Trunc((w.bottom-viewext.top)*wy);
  drw.ConvertaRect(w,wf);
  if tometa then begin
    bm:=tbitmap.create;
    bm.assign(bitmap);
    StretchBlt(bm.canvas.handle,0,0,wb.right-wb.left,wb.bottom-wb.top,
    bitmap.canvas.handle,wb.left,wb.top,wb.right-wb.left,wb.bottom-
wb.top,SRCCOPY);

    bm.width:=wb.right-wb.left;bm.height:=wb.bottom-wb.top;
    StretchBlt(h,wf.left,wf.top,wf.right-wf.left,wf.bottom-wf.top,
    bm.canvas.handle,0,0,bm.width,bm.height,SRCCOPY);
    bm.destroy;
  end
  else
  with wf do
    StretchBlt(h,left,top,right-left,bottom-top,
    bitmap.canvas.handle,wb.left,wb.top,wb.right-wb.left,wb.bottom-
wb.top,SRCCOPY);
  end
  else begin
  with drw._frame do
    StretchBlt(h,left,top,right-left,bottom-top,
    bitmap.canvas.handle,0,0,bitmap.width,bitmap.height,SRCCOPY);
  end;
end;

Procedure TMAPS_DATA.CleanHDC(h:hdc;col:Tmaps;box:trect);
var b,oldb:hbrush;
p,oldp:hpen;
i,j,k,wy:integer;
gr:TGrArray;
w,hh,x,y:integer;
bm:tbitmap;
begin //початок
CalcColGradients(drw.bko.minGPS,drw.bko.maxGPS,drw.bko.GradItems,gr);
if not tometa then box:=drw._back;
w:=(box.right-box.left) div drw.bko.GradItems;
hh:=(box.bottom-box.top) div drw.bko.GradItems;
b:=drw.GetSolidBrush(gr[drw.bko.GradItems]);
oldb:=selectobject(h,b);

fillrect(h,rect(box.left,box.top,box.right,box.bottom),b);
selectobject(h,oldb);deleteobject(b);

```

```

case drw.bko.effect of
1:for i:=1 to drw.bko.GradItems do begin
b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
fillrect(h,rect(box.left,box.top+(i-1)*hh,box.right,box.top+i*hh),b);
selectobject(h,oldb);deleteobject(b);
end;
2:for i:=1 to drw.bko.GradItems do begin
b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
fillrect(h,rect(box.left+(i-1)*w,box.top,box.left+i*w,box.bottom),b);
selectobject(h,oldb);deleteobject(b);
end;
3:begin
x:=(box.right-box.left) div 2;
y:=(box.bottom-box.top) div 2;
w:=x div 20;hh:=y div 20;
for i:=drw.bko.GradItems downto 1 do begin
b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
p:=createpen(0,0,gr[i]);
oldp:=selectobject(h,p);
ellipse(h,box.left+x-w*i,box.top+y-hh*i,box.left+x+w*i,box.top+y+hh*i);
selectobject(h,oldp);deleteobject(p);
selectobject(h,oldb);deleteobject(b);
end;
end;
4:begin
x:=(box.right-box.left) div 2;
y:=(box.bottom-box.top) div 2;
w:=x div drw.bko.GradItems;hh:=y div drw.bko.GradItems;
for i:=drw.bko.GradItems downto 1 do begin
b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
fillrect(h,rect(box.left+x-w*i,box.top+y-hh*i,box.left+x+w*i,box.top+y+hh*i),b);
selectobject(h,oldb);deleteobject(b);
end;
end;
5:begin
if not BackBitmap.empty then begin
x:=(box.right-box.left) div BackBitmap.width;
y:=(box.bottom-box.top) div BackBitmap.height;
for i:=0 to x do for j:=0 to y do
BitBlt(h,box.left+i*BackBitmap.width,box.top+j*BackBitmap.height,BackBitmap.widht
h,BackBitmap.height,BackBitmap.canvas.handle,0,0 ,SRCCOPY);
end;
end;
end;
end;

```

```

Procedure TMAPS_DATA.Meridians(h:hdc;box:trect);
var i,j,w,hh,x,y:integer;
p,oldp:hpen;
pix:Tmaps;
bm:tbitmap;
begin
if drw.bko.lines then begin
if not tometa then box:=drw._back;
w:=drw.bko.pctw;w:=trunc( w* (box.right-box.left)/ 100.0);
hh:=drw.bko.pctv;hh:=trunc( hh * (box.bottom-box.top)/ 100.0);
x:=(box.right-box.left) div w; y:= (box.bottom -box.top) div hh;
p:=createpen(0,0,drw.bko.lineGPS);
oldp:=selectobject(h,p);
for i:=1 to x do begin
movetoex(h,box.left+i*w,box.top,nil);
lineto(h,box.left+i*w,box.bottom);
end;
end;

```

```

for i:=1 to y do begin
    movetoex(h,box.left,box.top+i*hh,nil);
    lineto(h,box.right,box.top+i*hh);
end;
selectobject(h,oldp);
deleteobject(p);
end;
if drw.bko.logoon then begin
i:=LogoBitmap.width div 2;
j:=LogoBitmap.height div 2;
box.left:=box.left+1;box.right:=box.right-
1;box.top:=box.top+1;box.bottom:=box.bottom-1;
case drw.bko.Logopos of
    1:begin x:=box.left;y:=box.top;end;
    2:begin x:=(box.left+box.right) div 2 - i;y:=box.top;end;
    3:begin x:=box.right-LogoBitmap.width;y:=box.top;end;
    4:begin x:=box.left;y:=(box.top+box.bottom) div 2 -j;end;
    5:begin x:=(box.left+box.right) div 2 - i;y:=(box.top+box.bottom) div 2 -
j;end;
    6:begin x:=box.right-LogoBitmap.width;y:=(box.top+box.bottom) div 2 -j;end;
    7:begin x:=box.left;y:=box.bottom-LogoBitmap.height;end;
    8:begin x:=(box.left+box.right) div 2 - i;y:=box.bottom-LogoBitmap.height;end;
    9:begin x:=box.right-LogoBitmap.width;y:=box.bottom-LogoBitmap.height;end;
    else begin x:=box.left;y:=box.top;end;
end;
end;
end;

Procedure TMAPS_DATA.SeTmapss(t:byte;f,b:Tmaps;ix:word);
begin
np[ix].seTmapss(t,f,b);
end;

Procedure TMAPS_DATA.SetPBgGPS(k:integer;c:Tmaps);
var i:integer;
begin
for i:=1 to np[k].npolis do np[k].p[i].bkcol:=c;
for i:=1 to np[k].nmarcos do np[k].m[i].bkcol:=c;
end;

function TMAPS_DATA.GetPBgGPS(k:integer):Tmaps;
begin
if np[k].npolis>0 then
GetPBgGPS:=np[k].p[1].bkcol
else if np[k].nmarcos>0 then
GetPBgGPS:=np[k].m[1].bkcol
else GetPBgGPS:=clwhite;
end;
// реалізація внутрішньої процедури класу TMAPS_DATA SetPFgGPS
Procedure TMAPS_DATA.SetPFgGPS(k:integer;c:Tmaps);
var i:integer;
begin
for i:=1 to np[k].npolis do np[k].p[i].frcol:=c;
for i:=1 to np[k].nmarcos do np[k].m[i].frcol:=c;
end;

function TMAPS_DATA.GetPFgGPS(k:integer):Tmaps;
begin //початок
if np[k].npolis>0 then
GetPFgGPS:=np[k].p[1].Frcol
else if np[k].nmarcos>0 then
GetPFgGPS:=np[k].m[1].Frcol
else GetPFgGPS:=clBLACK;
end;

Procedure TMAPS_DATA.SetPPattern(k:integer;c:byte);
var i:integer;
begin
for i:=1 to np[k].npolis do np[k].p[i].trama:=c;

```

```

for i:=1 to np[k].nmarcos do np[k].m[i].trama:=c;
end;

function TMAPS_DATA.GetPPattern(k:integer):byte;
begin
if np[k].npolis>0 then
GetPPattern:=np[k].p[1].trama
else if np[k].nmarcos>0 then
GetPPattern:=np[k].m[1].trama
else GetPPattern:=0;
end;

Procedure TMAPS_DATA.SetChartLegend(k:word;s:string);
begin
if k>16 then exit;
drw.co.LegTexts[k]:=s;
end;

function TMAPS_DATA.GetChartLegend(k:word):string;
begin //початок
if k>16 then exit;
GetChartLegend:=drw.co.LegTexts[k];
end;

Procedure TMAPS_DATA.SetShadows(a,b:Tmaps);
begin
drw.calcshadows(a,b);
end;

Procedure TMAPS_DATA.Applyshadows;
var i,j:integer;
aux:double;
c:integer;
pcts:array[0..250] of record
    f,s:double;
    v:double;
end;

Procedure CalcFrecs;
var i,j,k:integer;
d,m:double;
s:double;
nump:integer;
begin //початок
d:=drw.maxvall / 250;
fillchar(pcts,sizeof(pcts),0);
nump:=0;
for i:=1 to 250 do pcts[i].v:=i*drw.maxvall/250;
for i:=1 to npolypoints do if sel[i] then begin
k:=trunc(np[i].values[drw.so.source] *250/drw.maxvall);
if drw.sot.calc=1 then
pcts[k].f:=pcts[k].f+1
else pcts[k].f:=pcts[k].f+np[i].values[drw.so.source];
inc(nump);
end;
s:=0;
for i:=1 to 250 do begin
s:=s+pcts[i].f;
pcts[i].s:=s;
end;
if drw.sot.calc=1 then m:=nump / drw.so.legendpoints
else m:=s / drw.so.legendpoints;
for i:=1 to drw.so.legendpoints do
for j:=1 to 250 do
if pcts[j].s>= m*i then begin
drw.brkshadow[i]:=pcts[j].v;
break;
end;
end;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;

```

```

for i:=1 to npolypoints do if sel[i] then begin
m:=np[i].values[drw.so.source];
for j:=1 to drw.so.legendpoints do
if m<=drw.brkshadow[j] then begin
    np[i].SeTmapss(0,0,drw.sot.colsf[j]);
    break;
begin
calcchartrratios;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
case drw.sot.kind of
0:begin
drw.calcshadows(drw.so.minGPS,drw.so.maxGPS);
for i:=1 to drw.so.legendpoints do
drw.brkshadow[i:=(drw.maxvall/drw.so.legendpoints)*i;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
for i:=1 to npolypoints do if sel[i] then begin
c:=trunc(np[i].values[drw.so.source]*100/drw.maxvall);
if c<1 then c:=1;if c>100 then c:=100;
np[i].SeTmapss(0,0,drw.shadow[c]);
end;
end;
1:begin
if drw.sot.calc>0 then calcfrecs
else begin
for i:=1 to drw.so.legendpoints do
drw.brkshadow[i:=(drw.maxvall/drw.so.legendpoints)*i;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
for i:=1 to npolypoints do if sel[i] then begin
c:=trunc(np[i].values[drw.so.source]*drw.so.legendpoints/drw.maxvall);
if c<1 then c:=1;if c>drw.so.legendpoints then c:=drw.so.legendpoints;
np[i].SeTmapss(0,0,drw.sot.colsf[c]);
end;
end;
end;
2:begin
for i:=1 to npolypoints do if sel[i] then begin
aux:=np[i].values[drw.so.source];
for j:=1 to drw.so.legendpoints do
if ((aux>=drw.sot.vals[j].min) and (aux<=drw.sot.vals[j].max)) then begin
    np[i].SeTmapss(0,0,drw.sot.cols[j]);
    break;
end;
end;

Procedure TMAPS_DATA.RectToPolyLine(r2:trect;var cuadro:array of tpoint);
Begin //початок
cuadro[0].x:=r2.left;cuadro[0].y:=r2.top;
cuadro[1].x:=r2.right;cuadro[1].y:=r2.top;
cuadro[2].x:=r2.right;cuadro[2].y:=r2.bottom;
cuadro[3].x:=r2.left;cuadro[3].y:=r2.bottom;
cuadro[4].x:=r2.left;cuadro[4].y:=r2.top;
end;

procedure TMAPS_DATA.adaptlegrect(r,r1:trect;var r2:trect);
begin //початок
r2.left:=trunc((r1.left/100)*(r.right-r.left)+r.left);
r2.top:=trunc((r1.top/100)*(r.bottom-r.top)+r.top);
r2.right:=trunc((r1.right/100)*(r.right-r.left)+r.left);
r2.bottom:=trunc((r1.bottom/100)*(r.bottom-r.top)+r.top);
end;

Procedure TMAPS_DATA.ClipWMF;
Var //розділ об'яви змінних
hdcmeta:hdc;
hmeta:hmetafile;
r:trect;
twmf:tmetafile;
begin

```

```

tometa:=true;
hdcmeta:=createmetasfile(nil);
r:=rect(0,0,width,height);
SetWindowExtEx(hdcmeta,width,height,nil);
SetWindowOrgEx(hdcmeta,0,0,nil);
SetViewportOrgEx(hdcmeta,0,0,nil);
SetViewportExtEx(hdcmeta,width,height,nil);
Draw(hdcmeta,r);
hmeta:=closemetasfile(hdcmeta);
twmf:=tmetasfile.create;
twmf.width:=width;
twmf.height:=height;
with TMetafileCanvas.Create(twmf, 0) do
try // спроба з можливим викликом виключення
PlayMetaFile(handle,hmeta);
finally // виконати у будь якому випадку навідь при появі виключення
free;
end;
twmf.enhanced:=false;
clipboard.assign(twmf);
twmf.free;
tometa:=false;
end;
Procedure TMAPS_DATA.WMF_clip2file(wnd:HWND;fname:string);
Var //розділ об'яви змінних
hGMem:longint;
lpmfp:^tMETAFILEPICT ;
f:file;
a:array[0..250] of char;
begin
OpenClipboard(Wnd);
hGMem:= GetClipboardData(CF_METAFILEPICT) ;
if (hGMem<>0) then begin
lpmfp:= GlobalLock(hGMem);
strcpy(a,fname);
copymetasfile(lpmfp^.hmf,a);
GlobalUnlock(hGMem);
end;
CloseClipboard;
end;

procedure TMAPS_DATA.ClpWMF2Text(fname:string;textfname:string);
var tm:tmetaheader;
f:file;
ft:textfile;
rs:longint;
cf:word;
i,r,j:word;
s:longint;
tlf:tlogfont;
n,np:word;
ltp:^SPltSmallPoint;
begin
new(ltp);
assignfile(f,fname); // підключення файлу
reset(f,1); // встановлення типу роботи з файлом

blockread(f,tm,sizeof(tm)); // читання даних з фалу
n:=0;
while not eof(f) do begin
blockread(f,rs,4); // читання даних з фалу
blockread(f,cf,2); // читання даних з фалу
if cf=$0324 then inc(n);
for j:=1 to rs-3 do blockread(f,cf,2);
end;
closefile(f); // закриття файлу
if n=0 then exit;

assignfile(f,fname); // підключення файлу

```

```

reset(f,1); // встановлення типу роботи з файлом

assignfile(ft,textfname); // підключення файлу
rewrite(ft); // встановлення типу роботи з файлом

blockread(f,tm,sizeof(tm)); // читання даних з фалу
I:=1;

while not eof(f) do begin
blockread(f,rs,4); // читання даних з фалу
blockread(f,cf,2); // читання даних з фалу
if ((cf=$0324) and (i<maxelnum)) then begin
blockread(f,np,2); // читання даних з фалу
blockread(f,ltp^,np*sizeof(TSmallPoint)); // читання даних з фалу
writeln(ft,':Element '+inttostr(i));
writeln(ft,'+P1');
for j:=1 to np do
writeln(ft,inttostr(ltp^[j].x)+' , '+inttostr(ltp^[j].y));
writeln(ft);
inc(i);
end
else begin
for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
end;
closefile(f); // закриття файлу
closefile(ft); // закриття файлу
dispose(ltp);
end;

Function TMAPS_DATA.ClpWMFRect(fname:string):trect;
var tm:tmetaheader;
f:file;
rs:longint;
cf:word;
i,r,j:word;
s:longint;
tlf:tlogfont;
n,np:word;
ltp:^SPltSmallPoint;
re:trect;
begin //початок
new(ltp);
assignfile(f,fname); // підключення файлу
reset(f,1); // встановлення типу роботи з файлом
blockread(f,tm,sizeof(tm)); // читання даних з фалу
n:=0;
re.left:=32000;re.top:=32000;re.bottom:=-32000;re.right:=-32000;

while not eof(f) do begin
blockread(f,rs,4); // читання даних з фалу
blockread(f,cf,2); // читання даних з фалу
if cf=$0324 then inc(n);
for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
closefile(f); // закриття файлу
if n=0 then exit;

assignfile(f,fname); // підключення файлу
reset(f,1); // встановлення типу роботи з файлом
blockread(f,tm,sizeof(tm)); // читання даних з фалу
I:=1;

while not eof(f) do begin
blockread(f,rs,4); // читання даних з фалу
blockread(f,cf,2); // читання даних з фалу
if ((cf=$0324) and (i<maxelnum)) then begin
blockread(f,np,2); // читання даних з фалу
blockread(f,ltp^,np*sizeof(TSmallPoint));

```

```

for j:=1 to np do begin
if ltp^[j].x<re.left then re.left:=ltp^[j].x;
if ltp^[j].y<re.top then re.top:=ltp^[j].y;
if ltp^[j].x>re.right then re.right:=ltp^[j].x;
if ltp^[j].y>re.bottom then re.bottom:=ltp^[j].y;
end;
inc(i);
end
else begin
    for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
end;
closefile(f); // закриття файлу
dispose(ltp);
ClpWMFRect:=re;
end;
// реалізація внутрішньої процедури класу TMAPS_DATA ScaleClpWMF2Text
Procedure TMAPS_DATA.ScaleClpWMF2Text (fname,textfname:string;iso:boolean);
var tm:tmetaheader;
f:file;
ft:textfile;
rs:longint;
cf:word;
i,r,j:word;
s:longint;
tlf:tlogfont;
n,np:word;
ltp:^SPltSmallPoint;
e, re:trect;
sx,sy:real;
lt:integer;
puntos:integer;
a:array[0..255] of char;
ax:string;
begin
e:=rect(0,0,10000,10000);
re:=ClpWMFRect(fname);

if (re.right-re.left)>(re.bottom-re.top) then begin
sx:=(e.right-e.left)/(re.right-re.left);
sy:=(e.bottom-e.top)/(re.right-re.left);
end
else begin
sx:=(e.right-e.left)/(re.bottom-re.top);
sy:=(e.bottom-e.top)/(re.bottom-re.top);
end;

new(ltp);
assignfile(f,fname); // підключення файлу
reset(f,1); // встановлення типу роботи з файлом
blockread(f,tm,sizeof(tm)); // читання даних з фалу
n:=0;
while not eof(f) do begin
blockread(f,rs,4); // читання даних з фалу
blockread(f,cf,2); // читання даних з фалу
if cf=$0324 then inc(n);
if cf=$0325 then inc(n);
for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
closefile(f); // закриття файлу
if n=0 then exit;

assignfile(f,fname); // підключення файлу
reset(f,1); // встановлення типу роботи з файлом

assignfile(ft,textfname); // підключення файлу
rewrite(ft);

blockread(f,tm,sizeof(tm)); // читання даних з фалу

```

```

I:=1;
puntos:=0;
while not eof(f) do begin
blockread(f,rs,4); // читання даних з фалу
blockread(f,cf,2); // читання даних з фалу
if ((cf=$0324) and (i<maxelnum)) then begin
blockread(f,np,2);
blockread(f,ltp^,np*sizeof(TSmallPoint));
writeln(ft,':Element '+inttostr(i));
writeln(ft,'+P'+inttostr(i)+' #2');
for j:=1 to np do begin
ltp^[j].x:=trunc((ltp^[j].x - re.left)*sx + e.left);
ltp^[j].y:=trunc((ltp^[j].y - re.top)*sy + e.top);
writeln(ft,inttostr(ltp^[j].x)+', '+inttostr(ltp^[j].y));
end;
writeln(ft);
inc(i);
puntos:=1;
end
else if ((cf=$0325) and (i<maxelnum)) then begin
blockread(f,np,2); // читання даних з фалу
blockread(f,ltp^,np*sizeof(TSmallPoint)); // читання даних з фалу
writeln(ft,':Element '+inttostr(i));
writeln(ft,'_L'+inttostr(i)+' #2');
for j:=1 to np do begin
ltp^[j].x:=trunc((ltp^[j].x - re.left)*sx + e.left);
ltp^[j].y:=trunc((ltp^[j].y - re.top)*sy + e.top);
writeln(ft,inttostr(ltp^[j].x)+', '+inttostr(ltp^[j].y));
end;
writeln(ft);
inc(i);
end
else if ((cf=$0521) and (i<100)) then begin
blockread(f,lt,2); // читання даних з фалу
blockread(f,a,lt); a[lt+1]:=#0; ax:=strupas(a);
writeln(ft,'/T'+inttostr(puntos)+' #3');
blockread(f,lt,2);
write(ft,inttostr(lt)+' , ');
blockread(f,lt,2); // читання даних з фалу
writeln(ft,inttostr(lt)+' , '+ax);
inc(puntos);
end
else begin
for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
end;
closefile(f); // закриття файлу
closefile(ft); // закриття файлу
dispose(ltp);
end;

Procedure TMAPS_DATA.FromWMF(fname:string);
var tmf:tmetafile;
a:array[0..255] of char;
ax,bx:string;
begin //початок
GetWindowsDirectory(a,200);
ax:=strupas(a)+'\gps_temp.clp';
bx:=strupas(a)+'\gps_temp.txt';
tmf:=tmetafile.create;
tmf.loadfromfile(fname);
clipboard.assign(tmf);
WMF_clip2file(parent.handle,ax);
tmf.destroy;
ScaleClpWMF2Text(ax,bx,true);
FromTextFile(bx);
end;

```

```

// реалізація внутрішньої процедури класу TMAPS_DATA FromClipWMF
Procedure TMAPS_DATA.FromClipWMF;
var a:array[0..255] of char;
ax,bx:string;
begin //початок
if not clipboard.hasformat(CF_metafilepict) then exit;
GetWindowsDirectory(a,200);
ax:=strpas(a)+'\gps_temp.clp';
bx:=strpas(a)+'\gps_temp.txt';
WMF_clip2file(parent.handle,ax);
ScaleClpWMF2Text(ax,bx,true);
FromTextFile(bx);
end;

Procedure TMAPS_DATA.SaveWMF(fname:string);
var tmf:tmetafile;
begin
ClipWMF;
tmf:=tmetafile.create;
tmf.assign(clipboard);
tmf.savetofile(fname);
tmf.destroy;
end;

procedure TMAPS_DATA.ClipBmp;
Var //розділ об'яви змінних
r:trect;
bm01:tbitmap;
begin
if not ready then exit;
bm01:=tbitmap.create;
try // спроба з можливим викликом виключення
r:=rect(0,0,width,height);
bm01.width:=(r.right-r.left)+1;
bm01.height:=(r.bottom-r.top)+1;
bm01.canvas.copyrect(r,canvas,r);
clipboard.assign(bm01);
finally // виконати у будь якому випадку навідь при появі виключення
bm01.free;
end;
end;

Procedure TMAPS_DATA.SaveBmp(fname:string);
var bm:tbitmap;
r:trect;
begin
if not ready then exit;
bm:=tbitmap.create;
r:=rect(0,0,width,height);
bm.width:=(r.right-r.left)+1;
bm.height:=(r.bottom-r.top)+1;
bm.canvas.copyrect(r,canvas,r);
bm.SaveToFile(fname);
bm.free;
end;

Procedure TMAPS_DATA.SnapShot(destinationcanvas:tcanvas;
                             destrect:trect;maprect:trect);
var r:trect;
begin //початок
if not ready then exit;
drw.convertarect(maprect,r);
StretchBlt(destinationcanvas.handle,
           destrect.left,
           destrect.top,
           destrect.right-destrect.left+1,
           destrect.bottom-destrect.top+1,
           canvas.handle,

```

```

    r.left,
    r.top,
    r.right-r.left+1,
    r.bottom-r.top+1,
    SRCCOPY);

end;

Procedure TMAPS_DATA.SetChartValue(element,item:word;value:real);
Begin //початок
if element>npolypoints then exit;
if element<1 then exit;
if item>drw.co.nvalues then exit;
np[element].values[item]:=value;
end;

Function TMAPS_DATA.GetChartValue(element,item:word):real;
begin
GetChartValue:=0;
if element>npolypoints then exit;
if element<1 then exit;
if item>drw.co.nvalues then exit;
GetChartValue:=np[element].values[item];
end;

// реалізація внутрішньої процедури класу TMAPS_DATA ValuesToGrid
Procedure TMAPS_DATA.ValuesToGrid(grid:tstringgrid);
var i,j:integer;
ax:string;
begin
with grid do begin
colcount:=drw.co.nvalues+2;
rowcount:=npolypoints+1;
fixedcols:=0;
fixedrows:=0;
cells[0,0]:='';
cells[1,0]:='Bullet';
for i:=1 to drw.co.nvalues do cells[1+i,0]:=drw.co.LegTexts[i];
for i:=1 to npolypoints do cells[0,i]:=np[i].nombre;
for i:=1 to npolypoints do
cells[1,i]:=fcadd(np[i].values[0],3,drw.co.ndecimals);
for i:=1 to drw.co.nvalues do
for j:=1 to npolypoints do
cells[i+1,j]:=fcadd(np[j].values[i],3,drw.co.ndecimals);
end;
end;

Procedure TMAPS_DATA.ValuesFromGrid(grid:tstringgrid);
var i,j:integer;
ax:string;
begin
with grid do begin
drw.co.nvalues:=colcount-2;
for i:=0 to drw.co.nvalues do drw.co.LegTexts[i]:=cells[1+i,0];
for i:=1 to rowcount-1 do np[i].nombre:=cells[0,i];
for i:=1 to rowcount-1 do np[i].values[0]:=fnum(cells[1,i]);
for i:=1 to drw.co.nvalues do
for j:=1 to rowcount-1 do
np[j].values[i]:=fnum(cells[i+1,j]);
end;
end;
CalcChartRatios;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA ImportDataFromGrid
Procedure TMAPS_DATA.ImportDataFromGrid(grid:tstringgrid);
var i,j,k,code:integer;
r:real;
ax:string;
begin

```

```

if grid.colcount>18 then grid.colcount:=18;
drw.co.nvalues:=grid.colcount-2;
for i:=1 to grid.rowcount-1 do begin
ax:=uppercase(grid.cells[0,i]);
for j:=1 to ElementCount do
if uppercase(ElementName[j])=ax then
for k:=1 to grid.colcount-1 do begin
val(grid.cells[k,i],r,code);
chartvalue[j,k-1]:=r;
end;
end;
for i:=1 to grid.colcount-1 do chartlegend[i-1]:=grid.cells[i,0];
end;

Function TMAPS_DATA.GetPatternBits(i:word):string;
Begin //початок
if i>8 then exit;
if i=0 then GetPatternBits:='000000'
else GetPatternBits:=trama[i];
end;

Procedure TMAPS_DATA.SetPatternBits(i:word;s:string);
begin
if ((i<1) or (i>8)) then exit;
trama[i]:=copy(s,1,64);
end;

Procedure TMAPS_DATA.Bar(h:hdc;x1,y1,x2,y2:integer;pat:word);
begin
if pat>16 then exit;if pat<1 then exit;
drw.GetBrocha(h,drw.co.patt[pat],drw.co.lineGPS,drw.co.GPSs[pat]);
rectangle(h,x1,y1,x2,y2);
drw.ReleaseBrocha;
end;

Function TMAPS_DATA.GetPolLayer(element,pol:word):byte;
Begin //початок
GetPolLayer:=1;
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
GetPolLayer:=np[element].p[pol].layer;
end;

Procedure TMAPS_DATA.SetPolLayer(element,pol:word;l:byte);
begin
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
np[element].p[pol].layer:=l;
end;

Function TMAPS_DATA.GetPolCount(element:word):byte;
begin
GetPolCount:=0;
if element=0 then getpolcount:=npolypoints;
if (element>npolypoints) then exit;
GetPolCount:=np[element].npolis;
end;

Function TMAPS_DATA.GetPolName(element,pol:word):string;
begin
GetPolName:='';
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
GetPolName:=np[element].p[pol].nombre;
end;

```

```

Procedure TMAPS_DATA.SetPolName(element,pol:word;s:string);
begin
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
np[element].p[pol].nombre:=s;
end;

Procedure TMAPS_DATA.ShowPolNames(b:boolean);
begin
drw.showpols:=b;
end;

Function TMAPS_DATA.GetDistance(p1,p2:tpoint):double;
begin
GetDistance:=distancia(p1.x,p1.y,p2.x,p2.y)*drw.mo.scale;
end;

Procedure TMAPS_DATA.Setnumbox(b:boolean);
begin
drw.numbox:=b;
end;
Function TMAPS_DATA.GetNumBox:boolean;
begin
GetNumBox:=drw.numbox;
end;

Function TMAPS_DATA.ElementbyName(s:string):integer;
var i,k:integer;
begin
k:=1;
for i:=1 to elementcount do
if pos(uppercase(s),uppercase(elementname[i]))>0 then begin
k:=i;
break;
end;
ElementbyName:=k;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA Nearest
Function TMAPS_DATA.Nearest(p:tpoint):integer;
var i,j,k:integer;
mybox:trect;
begin //початок
k:=0;
Nearest:=0;
for i:=npolypoints downto 1 do
if np[i].npolis>0 then begin
if ((p.x>np[i].caja.left) and (p.x<np[i].caja.right)
and (p.y>np[i].caja.top) and (p.y<np[i].caja.bottom))
then begin Nearest:=i;exit;end;
end;
Nearest:=k;
end;

Function TMAPS_DATA.NearestElement(p:tpoint):integer;
var i,j,k:integer;
mybox:trect;
begin
k:=0;
NearestElement:=0;
for i:=npolypoints downto 1 do if sel[i] then
if np[i].npolis>0 then begin
if ((p.x>np[i].caja.left) and (p.x<np[i].caja.right)
and (p.y>np[i].caja.top) and (p.y<np[i].caja.bottom))
then begin NearestElement:=i;exit;end;
end;
NearestElement:=k;

```

```

end;

procedure muerte;
begin
showmessage('Fatal error reading '+fname);
halt;
end;
var i,j,k,im,ip,ipu:integer;
ax,bx,nom:string;
fin:boolean;
p:tpoint;
f:textfile;
nsp:^tnsp;
n:word;
buffer:^SPtpoint;
maxcaja:trect;
c:char;
thelayer:integer;
lm:longint;
begin
reset(f); // встановлення типу роботи з файлом
ax:='';
for i:=1 to npolypoints do begin
while ((ax[1]<>':') and (not eof(f))) do begin
readln(f,ax);
ax:=ltrim(ax);
end;
if eof(f) then muerte;
infopanel.caption:=inttostr(i)+' of '+ inttostr(npolypoints);
infopanel.repaint;
ax:=rltrim(derc(ax,':'));
np[i]:=TSPolyPol.create(nsp^[i].m,nsp^[i].p,ax);
np[i].npoints:=nsp^[i].pu;
im:=0;ip:=0; ipu:=0;
for j:=1 to nsp^[i].m + nsp^[i].p + nsp^[i].pu do begin
repeat
readln(f,ax);
ax:=ltrim(ax);
until ((ax[1] in ['_', '+', '/']) or eof(F));
if eof(f) then muerte;
c:=ax[1];
nom:=rltrim(izqc(derc(ax,c), '#'));
ax:=rltrim(derc(ax, '#'));
if ax='' then ax:='1';
thelayer:=num(ax);
if c='_' then begin
inc(im);
np[i].SetMVals(clwhite,clblack,im,k,buffer^);
np[i].m[im].layer:=thelayer;
np[i].m[im].drawer:=drw;
np[i].m[im].nombre:=nom;
end
else
if c='+' then begin
inc(ip);
np[i].SetPVals(clwhite,clblack,ip,k,buffer^);
np[i].p[ip].layer:=thelayer;
np[i].p[ip].drawer:=drw;
np[i].p[ip].nombre:=nom;
end;
end;
np[i].CalcGPS;
comprect(np[i].caja,maxcaja);
np[i].drawer:=drw;
end;
closefile(f); // закриття файлу
dispose(nsp);
dispose(buffer);
for i:=1 to npolypoints do sel[i]:=true;

```

```

if arr<>nil then arr.destroy;
arr:=TSArrange.create(2,npolypoints,'Normal');
ready:=true;
Titulo:=fname;
Nbloqs:=npolypoints;
MapLimits:=MaxCaja;
end;

// додавання точки на карті
Function TMAPS_DATA.AddPoint(elem:integer;c:tcitypoint):integer;
var a:tsapoint;
begin //початок
AddPoint:=0;
if elem>npolypoints then exit; if elem<1 then exit;
if np[elem].npoints>15 then exit;
a:=tsapoint.create;
a.c:=c;
np[elem].npoints:=np[elem].npoints+1;
np[elem].points[np[elem].npoints]:=a;
AddPoint:=np[elem].npoints;
a.free;
end;

Procedure TMAPS_DATA.GetPoint(elem,ix:integer;var c:tcitypoint);
begin
if elem>npolypoints then exit; if elem<1 then exit;
if ix<1 then exit;if ix> np[elem].npoints then exit;
c:=np[elem].points[ix].c;
end;

Function TMAPS_DATA.RemovePoint(elem,ix:integer):boolean;
var i:integer;
begin
removepoint:=false;
if elem>npolypoints then exit; if elem<1 then exit;
if np[elem].npoints<1 then exit;
if ix<1 then exit;if ix> np[elem].npoints then exit;
for i:=ix to np[elem].npoints-1 do np[elem].points[i]:=np[elem].points[i+1];
np[elem].npoints:=np[elem].npoints-1;
Removepoint:=true;
end;

Function TMAPS_DATA.GetPointCount(elem:integer):integer;
Begin //початок
GetPointCount:=0;
if elem>npolypoints then exit; if elem<1 then exit;
if np[elem].npoints<1 then exit;
GetPointCount:=np[elem].npoints;
end;

Procedure TMAPS_DATA.ClearPoints(elem:integer);
begin
if elem>npolypoints then exit; if elem<1 then exit;
np[elem].npoints:=0;
end;

Procedure TMAPS_DATA.SetSymbol(f:tfont;c:char);
begin
drw.fontsymbol.assign(f);
drw.asymbol:=c;
end;

procedure TMAPS_DATA.time(sender:tobject);
begin
replaceglis;
gli.paint(canvas);
end;

```

```

Procedure TMAPS_DATA.MyMouseUp(Sender: TObject; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
var p:tpoint;
i,k:integer;
pu:tpoint;
pr:trect;
begin //початок
case drw.mo.mousemode of
8:begin
    p.x:=x;p.y:=y;
    p:=PointInMap(p);
    if button=mbright then begin
        p.x:=(gpsbox.right+gpsbox.left) div 2;
        p.y:=(gpsbox.top+gpsbox.bottom) div 2;
    end;
    k:=(zoombox.right-zoombox.left) div 2;
    i:=(zoombox.bottom-zoombox.top) div 2;
    fzoombox.left:=p.x-k; fzoombox.right:=p.x+k;
    fzoombox.top:=p.y-i; fzoombox.bottom:=p.y+i;
    repaint;
end;

10:if bb_drawing then begin
    bb_drawing:=false;
    DrawFocusRect(canvas.handle,rect(bb_anchorx,bb_anchory,bb_curx,bb_cury));
    p:=PointInMap(point(bb_anchorx,bb_anchory));
    if button=mbright then SelFromBox(gpsbox,true)
    else
    if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
    begin
        pr.left:=p.x;pr.top:=p.y;
        p:=PointInMap(point(bb_curx,bb_cury));
        pr.right:=p.x;pr.bottom:=p.y;
        if ssCtrl in Shift then SelFromBox(pr,false)
        else SelFromBox(pr,true);
    end
    else begin
        k:=Nearest(p);
        if k>0 then begin
            if ssCtrl in Shift then selected[k]:=true
            else selected[k]:= not selected[k];
        end;
    end;
    invalidate;
end;

1,9,12,15:if bb_drawing then begin
    bb_drawing:=false;
    DrawFocusRect(canvas.handle,rect(bb_anchorx,bb_anchory,bb_curx,bb_cury));

    if drw.mo.mousemode =9 then begin
        pr.left:=trunc((bb_anchorx-left)/(drw._back.right-drw._back.left)*100);
        pr.right:=trunc((bb_curx-left)/(drw._back.right-drw._back.left)*100);
        pr.top:=trunc((bb_anchory-top)/(drw._back.bottom-drw._back.top)*100);
        pr.bottom:=trunc((bb_cury-top)/(drw._back.bottom-drw._back.top)*100);
    end;

    case drw.mo.mousemode of
    9:
        if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
        begin
            viewpct:=pr;
            infopanel.caption:=inttostr(viewpct.left)+' ','+
            inttostr(viewpct.top)+' - '+'
            inttostr(viewpct.right)+' ','+
            inttostr(viewpct.bottom);

            end
            else viewpct:=rect(1,1,99,99);

```

```

1:
  if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
  begin
    p:=PointInMap(point(bb_anchorx,bb_anchory));
    fzoombox.left:=p.x;fzoombox.top:=p.y;
    p:=PointInMap(point(bb_curx,bb_cury));
    fzoombox.right:=p.x;fzoombox.bottom:=p.y;
  end
  else

  begin
    //початок
    p.x:=x;p.y:=y;
    p:=PointInMap(p);
    k:=(zoombox.right-zoombox.left) div 2;
    i:=(zoombox.bottom-zoombox.top) div 2;
    fzoombox.left:=p.x-k; fzoombox.right:=p.x+k;
    fzoombox.top:=p.y-i; fzoombox.bottom:=p.y+i;
  end;

12:if Assigned(FOnMouseBox) then begin
  if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
  begin
    p:=PointInMap(point(bb_anchorx,bb_anchory));
    pr.left:=p.x;pr.top:=p.y;
    p:=PointInMap(point(bb_curx,bb_cury));
    pr.right:=p.x;pr.bottom:=p.y;
  end
  else pr:=gpsbox;
  OnMouseBox(self,shift,pr);
end;
15:begin
  if ((bb_curx-bb_anchorx>20) and (bb_cury-bb_anchory>20)) then
  begin
    p:=PointInMap(point(bb_anchorx,bb_anchory));
    pr.left:=p.x;pr.top:=p.y;
    p:=PointInMap(point(bb_curx,bb_cury));
    pr.right:=p.x;pr.bottom:=p.y;
  end
  else pr:=gpsbox;
  scaleto(pr);
end;
end;
invalidate;
end;
3:begin
  bb_drawing:=false;
  invalidate;
end;
4,11:if bb_drawing then begin
  bb_drawing:=false;
  SetRop2(canvas.handle,r2_not);
  movetoex(canvas.handle,drw.cvx(bb_anchorx),drw.cvy(bb_anchory),nil);
  lineto(canvas.handle,drw.cvx(bb_curx),drw.cvy(bb_cury));
  if drw.mo.mousemode=11 then if assigned(FOnMouseLine)
  then OnMouseLine(self,shift,rect(bb_anchorx,bb_anchory,bb_curx,bb_cury));
end;
5:if bb_drawing then begin
  bb_drawing:=false;
  SetRop2(canvas.handle,r2_not);
  p.x:=abs(bb_curx-bb_anchorx);
  p.y:=p.x;

  k:=abs(drw.cvx(bb_curx)-drw.cvx(bb_anchorx));
  pu.x:=drw.cvx(bb_anchorx);
  pu.y:=drw.cvy(bb_anchory);
  Arc(canvas.handle,pu.x-k,pu.y-k,
    pu.x+k,pu.y+k,

```

```

        pu.x,pu.y,pu.x,pu.y);
end;
16:  if not userline.drawing then begin
        if button=mbright then invalidate;
    end;
end;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA WMLButtonDown
Procedure TMAPS_DATA.WMLButtonDown(var msg:TWMLButtonDown);
Begin //початок
if not ready then exit;
MyMouseDown(self,mbleft,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
If Assigned(FOnMouseDown) then
OnMouseDown(self,mbleft,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
end;

Procedure TMAPS_DATA.WMRButtonDown(var msg:TWMRButtonDown);
begin
if ready then begin
MyMouseDown(self,mbright,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
If Assigned(FOnMouseDown) then
OnMouseDown(self,mbright,KeysToShiftState(msg.keys),msg.xpos,msg.ypos)
else if drw.mo.mousemode=0 then parent.dispatch(msg);
end
else inherited;
end;

Procedure TMAPS_DATA.DeleteElements;
var i,j,k:integer;
ts:tspolypol;
begin
k:=npolypoints;
for j:=k downto 1 do if selected[j] then begin
if j<npolypoints then begin
ts:=np[j];
np[j]:=np[npolypoints];
selected[j]:=false;
ts.free;
end
else begin
np[npolypoints].free;
end;
npolypoints:=npolypoints -1;
end;
end;

Procedure TMAPS_DATA.JoinElements(orig,dest:integer);
var i,j,k:integer;
ts:tspolypol;
begin
if orig<npolypoints then begin
ts:=np[orig];
np[orig]:=np[npolypoints];
ts.free;
end
else begin
np[npolypoints].free;
end;
npolypoints:=npolypoints -1;

end;

// реалізація внутрішньої процедури класу TMAPS_DATA posterxN
procedure TMAPS_DATA.posterxN(Nx,Ny:integer);
VAR //розділ об'яви змінних
P: TPosterPrint;
begin
if not ready then exit;

```

```

if (( nx<1) or (ny<1) or (nx>10) or (ny>10)) then exit;
screen.cursor:=crhourglass;
p:=TPosterPrint.create(self);
try // спроба з можливим викликом виключення
with p do begin
    HorzPages:=nx;
    VertPages:=ny;
    HorzMarge:=100;
    VertMarge:=100;
    Image:='gps_temp.wmf';
    LabelPages:=true;
end;
SaveWMF('gps_temp.wmf');
p.doprint;
finally
// виконати у будь якого випадку навідь при появи виключення
p.free;
screen.cursor:=crdefault;
end;
end;

Function TMAPS_DATA.GetArea(elem:integer):longint;
begin
GetArea:=trunc(np[elem].GetArea(drw.mo.scale,drw.mo.scale));
end;

Procedure TMAPS_DATA.LoadPaletteFromFile(filename:string);
begin
deleteobject(paleta);
paleta:=CreatePaletteFromFile(filename);
end;

Procedure TMAPS_DATA.LoadMosaicBitmap(fname:string);
begin
drw.mosaicbmp.loadfromfile(fname);
end;
// реалізація внутрішньої процедури класу TMAPS_DATA placegli
procedure TMAPS_DATA.placegli(ix:integer;p:tpoint);
var pp:tpoint;
begin
if ((ix>0) and (ix<=gli.count)) then begin
gli.glyph[ix].mappos:=p;
drw.convertapoint(p,pp);
gli.glyph[ix].pos:=pp;
gli.glyph[ix].paint(canvas);
end;
end;

procedure TMAPS_DATA.addgli(p:tpoint);
begin
gli.add(0);
placegli(gli.count,p);
end;

procedure TMAPS_DATA.replaceglis;
var i:integer;
p,pp:tpoint;
begin
for i:=gli.count downto 1 do begin
drw.convertapoint(gli.glyph[i].mappos,p);
gli.glyph[i].pos:=p;
end;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA GlyphCount
function TMAPS_DATA.GlyphCount:integer;
begin
GlyphCount:=gli.count;
end;

```

```

// реалізація внутрішньої процедури класу TMAPS_DATA GlyphSettings
Procedure TMAPS_DATA.GlyphSettings (ixglyph:integer;imageix:integer;
                                animate:boolean;enable:boolean);
begin
if gli.glyph[ixglyph]<>nil then with gli.glyph[ixglyph] do begin
if imageix<gli.gn then index:=imageix;
animated:=animate;
enabled:=enable;
end;
end;
// реалізація внутрішньої процедури класу TMAPS_DATA LoadGlyphsBitmap
Procedure TMAPS_DATA.LoadGlyphsBitmap (fname:string);
begin
gli.Loadfromfile (fname);
end;

Procedure TMAPS_DATA.WMLButtonUp (var msg:TWMLButtonUp);
begin
if not ready then exit;
MyMouseUp (self,mbleft,KeysToShiftState (msg.keys),msg.xpos,msg.ypos);
if ready then If Assigned (fOnMouseUp) then
OnMouseUp (self,mbleft,KeysToShiftState (msg.keys),msg.xpos,msg.ypos);
inherited;
end;

Procedure TMAPS_DATA.WMRButtonUp (var msg:TWMRButtonUp);
begin
if ready then begin
MyMouseUp (self,mbright,KeysToShiftState (msg.keys),msg.xpos,msg.ypos);
if ready then If Assigned (fOnMouseUp) then
OnMouseUp (self,mbright,KeysToShiftState (msg.keys),msg.xpos,msg.ypos);
end;
end;

Procedure TMAPS_DATA.WMMouseMove (var msg:TWMMouseMove);
begin
if not ready then exit;
MyMouseMove (self,KeysToShiftState (msg.keys),msg.xpos,msg.ypos);
if ready then if Assigned (fOnMouseMove) then
OnMouseMove (self,KeysToShiftState (msg.keys),msg.xpos,msg.ypos);
end;

Procedure TMAPS_DATA.ToFront (elem:integer);
var auxp:TSPolyPol;
i:integer;
auxs:boolean;
begin
if ((elem<1) or (elem>npolypoints)) then exit;
auxp:=np[elem];
auxs:=selected[elem];
for i:=elem to npolypoints-1 do begin
np[i]:=np[i+1];
selected[i]:=selected[i+1];
end;
np[mpolypoints]:=auxp;
selected[mpolypoints]:=auxs;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA ToBack
Procedure TMAPS_DATA.ToBack (elem:integer);
var auxp:TSPolyPol;
i:integer;
auxs:boolean;
begin
if ((elem<1) or (elem>npolypoints)) then exit;
auxp:=np[elem];
auxs:=selected[elem];
for i:=elem downto 2 do begin
np[i]:=np[i-1];

```

```

selected[i]:=selected[i-1];
end;
np[1]:=auxp;
selected[1]:=auxs;
end;

Procedure TMAPS_DATA.ReMap(var m:array of integer);
var sp:^TSPOLARARRAY;
i:integer;
begin
new(sp);
for i:=1 to npolypoints do sp^[i]:=np[m[i]];
for i:=1 to npolypoints do np[i]:=sp^[i];
dispose(sp);
end;

Function TMAPS_DATA.IsInBox(ix:integer;r:trect):boolean;
var rr:trect;
begin
IntersectRect(rr,r,np[ix].caja);
IsInBox:=((rr.left<>0) or (rr.right<>0) or (rr.bottom<>0) or (rr.top<>0));
end;

// реалізація внутрішньої процедури класу TMAPS_DATA SelFromBox
Procedure TMAPS_DATA.SelFromBox(r:trect;cleanup:boolean);
var i:integer;
begin
if cleanup then begin
for i:=1 to elementcount do selected[i]:= IsInBox(i,r);
end
else
for i:=1 to elementcount do if IsInBox(i,r) then selected[i]:= true;
end;

Procedure TMAPS_DATA.ForwardLink(ix:integer);
var ax:string;
begin
if ix>0 then
if np[ix].link<>' ' then
if fileexists(np[ix].link) then begin
if nbacks<5 then inc(nbacks);
BackLink[nbacks]:=polfname;
ax:=np[ix].link;
FileName:=ax;
drw.mo.mousemode:=14;
end;
end;

// реалізація внутрішньої процедури класу TMAPS_DATA BackwardLink
Procedure TMAPS_DATA.BackwardLink;
var ax:string;
begin
if nbacks>0 then begin
if fileexists(BackLink[nbacks]) then begin
ax:=BackLink[nbacks];
FileName:=ax;
drw.mo.mousemode:=14;
end;
dec(nbacks);
end;
end;

Procedure TMAPS_DATA.SetBackgroundOptions(var op:tbkop);
begin
drw.bko:=op;
if drw.bko.tilefname<>' ' then if fileexists(drw.bko.tilefname)
then backbitmap.loadfromfile(drw.bko.tilefname);
if drw.bko.logofname<>' ' then if fileexists(drw.bko.logofname)
then logobitmap.loadfromfile(drw.bko.logofname);

```

```
end;
```

```
Procedure TMAPS_DATA.CalcGPS;
var i:integer;
begin;
prepect (MapLimits);
for i:=1 to npolypoints do compect (np[i].caja,MapLimits);
end;

// реалізація внутрішньої процедури класу TMAPS_DATA Merge
Procedure TMAPS_DATA.Merge (fname:string;layeroffset:integer; x,y:integer);
var agps:TMAPS;
i,j,k:integer;
begin
if not fileexists(fname) then exit;
ready:=false;
agps:=TMAPS_DATA.create(self);
agps.filename:=fname;
for i:=1 to agps.elementcount do
np[elementcount+i]:=agps.np[i];
npolypoints:=npolypoints+agps.elementcount;
agps.npolypoints:=0;
agps.free;
CalcGPS;
ready:=true;
zoombox:=MapLimits;
end;
```

```
Procedure TMAPS_DATA.ScaleTo (r:trect);
var sb,aux:trect;
kx,ky:double;
x0,y0,x1,y1:integer;
i,j:integer;
function newx(x:integer):integer;
begin
newx:=trunc (x1+(x-x0)*kx);
end;
function newy(y:integer):integer;
begin
newy:=trunc (y1+(y-y0)*ky);
end;

function newrect(t:trect):trect;
var tt:trect;
begin
tt.left:=newx(t.left);
tt.right:=newx(t.right);
tt.top:=newy(t.top);
tt.bottom:=newy(t.bottom);
newrect:=tt;
end;
begin
for i:=1 to elementcount do if selected[i] then
np[i].scaleto(newrect(np[i].caja));
end;
```

```
// реалізація внутрішньої процедури класу TMAPS_DATA AddPoly
Procedure TMAPS_DATA.AddPoly (closed:boolean);
var i,j:integer;
var slt:^SPltpoint;
begin
if userline.np<2 then exit;
npolypoints:=npolypoints+1;
np[npolypoints]:=TSPolyPol.create(1,0,'New');
slt:=addr(userline.p);
with np[npolypoints] do begin
SetMVals(clred,clblue,1,userline.np,slt^);
m[1].drawer:=drw;
```

```
m[1].layer:=2;  
m[1].nombre:='Line 1';  
CalcGPS;  
drawer:=drw;  
end;  
end;  
CalcGPS;  
selected[mpolypoints]:=true;  
end;  
  
end.
```

КБПЗ_2023

ABOUT.PAS - ФАЙЛ АВТОРСЬКОГО ПРАВА

```
unit About; // Форма авторського права
{
Дослідження та програмна реалізація системи оновлення цифрових
топографічних карт геоінформаційної хмари
Виконав: студент 6 курсу,
групи КН-22М-2
напряму підготовки (спеціальності) 122 Комп'ютерні науки
Шепель Артем Сергійович
}
Interface // Інтерфейсна частина - об'ява даних

// об'ява бібліотек
WinTypes, WinProcs, Classes, Graphics, Forms, Controls, StdCtrls,
  Buttons, ExtCtrls;

Type
// об'ява власних типів даних
TAboutBox = class(TForm) // розроблений клас
  Panel1: TPanel;
  OKButton: TBitBtn;
  ProgramIcon: TImage;
  ProductName: TLabel;
  Version: TLabel;
  Copyright: TLabel;
  Label1: TLabel;
  procedure OKButtonClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

Var //розділ об'яви змінних
  AboutBox: TAboutBox;

Implementation
// Реалізація об'явлених даних у інтерфейсній частині

{$R *.DFM}

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
  close;
end;

end.
```