

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

“ \_\_\_ ” \_\_\_\_\_ 2021 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
за другим (магістерським) рівнем вищої освіти  
на тему

**“Дослідження та програмна реалізація системи обробки  
навігаційних даних”**

Виконав здобувач вищої освіти  
II курсу, групи КІ-20М-1,4  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»

Гирба О.М.

« \_\_\_ » \_\_\_\_\_ 2021 р.

Керівник проекту  
доктор технічних наук, професор

Олексій СМІРНОВ

« \_\_\_ » \_\_\_\_\_ 2021 р.

Рецензент \_\_\_\_\_

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

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.т.н., проф.

Олексій СМІРНОВ  
« 6 » вересня 2021 року

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

Гирбі Олексію Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи обробки навігаційних даних

2. Керівник роботи Смірнов Олексій Анатолійович, докт. техн. наук, професор  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 42-13 від 02.08.2021 року

3. Строк подання студентом роботи до захисту 10.12.2021 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи обробки навігаційних даних

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

1. Призначення та область використання. 7. Економічна ефективність розробленої

2. Перегляд аналогічних існуючих систем. 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. Наукова новизна

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.                            | 05.10.2021     | 14.11.2021       |
| Охорона праці | Оришака О.В.                              | 06.10.2021     | 16.11.2021       |
|               |                                           |                |                  |

7. Дата видачі завдання « 6 » вересня 2021 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                                                                      | 10.10.2021 р.                                                                                         |          |
| 2.    | Постановка задачі, оформлення ТЗ                                                            | 15.10.2021 р.                                                                                         |          |
| 3.    | Розробка моделі компонента                                                                  | 20.10.2021 р.                                                                                         |          |
| 4.    | Розробка структур даних                                                                     | 25.10.2021 р.                                                                                         |          |
| 5.    | Розробка алгоритмів зв'язку та відображення                                                 | 30.10.2021 р.                                                                                         |          |
| 6.    | Програмування алгоритмів                                                                    | 10.11.2021 р.                                                                                         |          |
| 7.    | Розрахунок економічної ефективності                                                         | 13.11.2021 р.                                                                                         |          |
| 8.    | Розрахунки з охорони праці та техніки безпеки                                               | 15.11.2021 р.                                                                                         |          |
| 9.    | Оформлення ПЗ                                                                               | 17.11.2021 р.                                                                                         |          |
| 10.   | Попередній захист роботи                                                                    | 10.12.2021 р.                                                                                         |          |
|       |                                                                                             |                                                                                                       |          |

Дата видачі завдання  
« 6 » вересня 2021 р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
« 6 » вересня 2021 р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Гирба О.М. Дослідження та програмна реалізація системи обробки навігаційних даних. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи обробки навігаційних даних.

Метою розробки є дослідження та програмна реалізація системи обробки навігаційних даних.

Об'єктом дослідження є процес обробки навігаційних даних.

Предметом дослідження є методи обробки навігаційних даних.

Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи обробки навігаційних даних.

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 XP/Vista/7/8/10.

Програму розроблено в середовищі RAD Studio Delphi.

**Ключові слова:** комп'ютерна інженерія, навігаційні дані

## ABSTRACT

**Hyrba O.M. Research and software implementation of navigation data processing system. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021**

In this final qualification work on the second (master's) level of higher education the software which is intended for system of processing of navigation data is developed.

The purpose of development is research and software implementation of the navigation data processing system.

The object of research is the process of processing navigation data.

The subject of research is methods of navigation data processing.

Research methods are based on methods of information theory and coding, methods of mathematical statistics, methods of software development.

The result is a software implementation of the navigation data processing system.

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

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment of RAD Studio Delphi.

**Keywords:** computer engineering, navigation data

## ЗМІСТ

|                                                                                                                                                                              |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....                                                                                                                | 3  |
| ВСТУП.....                                                                                                                                                                   | 4  |
| 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....                                                                                                                                   | 6  |
| 1.1 Призначення системи.....                                                                                                                                                 | 6  |
| 1.2 Область застосування.....                                                                                                                                                | 8  |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....                                                                                                                                 | 13 |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти ..... | 13 |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....                                                                                             | 21 |
| 2.3 Розгорнута постановка завдання .....                                                                                                                                     | 27 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....                                                                                                                                | 29 |
| 3.1 Опис функціонування системи.....                                                                                                                                         | 29 |
| 3.2 Розробка структурної схеми .....                                                                                                                                         | 40 |
| 3.3 Розробка функціональної схеми.....                                                                                                                                       | 41 |
| 3.4 Розробка діаграми процесів .....                                                                                                                                         | 43 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...                                                        | 45 |
| 4.1 Розробка блок-схем та опис алгоритмів функціонування системи .....                                                                                                       | 45 |
| 4.2 Захист розробленого програмного забезпечення .....                                                                                                                       | 52 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....                                                                                                                        | 55 |
| 6 НАУКОВА НОВИЗНА .....                                                                                                                                                      | 59 |

**ВКРМ-123.21.0002.00.00.ПЗ**

| Вим.     | Арк. | № докум.     | Підп. | Дата |                                                                        |      |       |         |
|----------|------|--------------|-------|------|------------------------------------------------------------------------|------|-------|---------|
| Розроб.  |      | Гирба О.М.   |       |      | Дослідження та програмна реалізація системи обробки навігаційних даних | Лім. | Аркуш | Аркушів |
| Перев.   |      | Смірнов О.А. |       |      |                                                                        | М    | 1     | 97      |
| Н.контр. |      | Гермак В.С.  |       |      | ЦНТУ КІ-20М-1,4                                                        |      |       |         |
| Затв.    |      | Смірнов О.А. |       |      |                                                                        |      |       |         |

|                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------|----|
| 7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....                                                                            | 60 |
| 7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. .... | 60 |
| 7.2 Розрахунок трудомісткості розробки програмної продукції .....                                                              | 62 |
| 7.3 Визначення чисельності виконавців і планового фонду зарплати .....                                                         | 64 |
| 7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника .....                                           | 69 |
| 7.5 Визначення собівартості розробки та ціни програмної продукції. ....                                                        | 73 |
| 7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....                     | 76 |
| 7.7 Визначення експлуатаційних витрат.....                                                                                     | 77 |
| 7.8 Визначення економічної ефективності програмної продукції.....                                                              | 78 |
| 7.9 Висновок. ....                                                                                                             | 80 |
| 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....                                                                              | 81 |
| 8.1 Вступ.....                                                                                                                 | 81 |
| 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером .....                                                               | 82 |
| 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ..                                                   | 83 |
| 8.4 Розробка заходів з умов поліпшення охорони праці.....                                                                      | 86 |
| 8.5 Розрахункова частина .....                                                                                                 | 87 |
| 8.6 Висновки до розділу.....                                                                                                   | 88 |
| 9 ОСНОВНІ ВИСНОВКИ.....                                                                                                        | 89 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                                                                                               | 91 |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- КПК – кишеньковий персональний комп'ютер
- РМД – робоче місце диспетчера
- СЕР – Circular Error of Probability – окружність можливої помилки
- GPS – глобальна система місцевизначення.
- GSM – система цифрового стільникового телефонного зв'язку
- IOD – issue of data – часовий ідентифікатор
- LAAS – навігаційний комплекс
- LAT – широта
- LON – довгота
- MFC – Microsoft Foundation Class library – бібліотека класів
- OLE – технологія зв'язування й вбудовування об'єктів
- POI – point of interest – вбудована база «цікавих» точок
- SA – режим обмеженого доступу
- WAAS – система супутників і наземних станцій, що робить коректування сигналів GPS

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 3    |

## ВСТУП

**Актуальність теми.** Одним із сучасних досягнень науково-технічного прогресу є розробка системи GPS, за допомогою якої мандрівник може визначати свої координати, а пілот посадити літак у зоні з нульовою видимістю. У найближче десятиліття можливості глобальної системи позиціонування значно розширяться [1]. Можливості системи глобального позиціонування в найближчі 10 років стануть набагато ширше. Користувач зможе визначати свої координати з точністю до метра. Можливості системи GPS будуть розширюватися за рахунок модернізації, що припускає: введення додаткових каналів сигналу на супутнику, збільшення потужності сигналу й удосконалення системи його корекції, використання спрямованих антен, а також інтеграцію з телевізійними й телефонними стільниковими мережами [1-5]. За допомогою GPS літаки зможуть приземлятися в повній темряві. Система зможе відслідковувати місцезнаходження повітряних судів на всьому протязі польоту. Найближчим часом GPS допоможе контролювати рух автомобільного транспорту, забезпечуючи безпеку дорожнього руху, удосконалена система зможе бути застосована в електроенергетиці, у телекомунікаціях, при видобутку корисних копалин, картографії й навіть у сільському господарстві. Крім того, будь-який мандрівник зможе скористатися GPS на всій території земної кулі [2-5]. Якщо провести огляд сучасних систем GPS, то ми побачимо, що провідні світові держави мають свої супутникові системи, які забезпечують роботу приймача GPS. У США це система Navstar, у Європейському Союзі – Galileo [6-8]. В зв'язку з тим, що Україна поки не розвертає свою систему супутників для реалізації системи GPS, то актуальним буде розробка вітчизняного програмного забезпечення даної системи.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи обробки навігаційних даних.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 4    |

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

- Огляд існуючих систем обробки навігаційних даних.
- Дослідження системи обробки навігаційних даних.
- Програмна реалізація системи обробки навігаційних даних.

*Об'єктом дослідження є процес обробки навігаційних даних.*

*Предметом дослідження є методи обробки навігаційних даних.*

*Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод обробки навігаційних даних.
- Розроблено вітчизняний продукт обробки навігаційних даних, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі обробки навігаційних даних.

**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 LV Науково-технічна конференція здобувачів вищої освіти «Наука – виробництву», 2021, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №12.

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 5    |

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Розглянемо, як працює система GPS. Супутники передають сигнали двох видів. Один з них несе інформацію про місцезнаходження супутника й час передачі сигналу. Він приймається стаціонарними наземними станціями, обробляється й відправляється на супутник, що передає його всім користувачам системи. Другий сигнал – код, необхідний для визначення часу передачі сигналу. Творці системи називають його псевдовипадковим шумом. [2]

Щоб перебороти відстань в 20 тис. км, сигналу потрібен час. Якщо користувач зможе за допомогою свого приймача, у який закладений код, визначити час його відправлення, то нескладно буде зафіксувати час його проходження й, помноживши отримані дані на швидкість поширення, розрахувати відстань до супутника.

Якщо в GPS -приймачі установити годинники, то, одержавши видалення від трьох супутників, користувач зможе визначити широту, довготу й висоту свого місцезнаходження. Сигнал, що йде від супутників, нагадує три сфери, що перетинаються в різний час у різних точках. Для користувача, що перебуває на Землі, існує тільки один момент їхнього зіткнення в даний проміжок часу. Для більше злагодженої синхронізації сигналу на супутниках установлені атомний годинник, що забезпечують точність ходу до однієї мільярдної. У більшості GPS -приймачів вони можуть відставати на одну або більше секунд у день. Можна підрахувати, що помилка всього в одну секунду змінить відстань від супутника до користувача на 300 тис. км. Інженери називають процес виміру відстані між супутником і користувачем псевдовиміром. Справа в тому, що погрішність присутня й у сигналах від чотирьох супутників, у результаті чого ми одержуємо чотири рівняння із чотирма невідомими.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 6    |

Сучасні GPS-приймачі здатні враховувати доплерівський ефект у випадку, якщо виміри проводяться в русі. При переміщенні приймача в бік поширення хвилі її довжина стає більше, а при зустрічному ході – менше. З огляду на даний ефект, можна одержати швидкість руху GPS-приймача. Такий метод виміру швидкості дуже точний. Таким чином, GPS-приймачі визначають три координати й три вектори швидкості, а також роблять синхронізацію часу через мережу. При цьому самі приймачі не передають сигналів в ефір. Незабаром GPS будуть обладнані стільникові телефони, що приведе до подорожчання останніх усього на \$5 [3].

Супутники GPS-системи передають сигнал, що володіє класичною синусоїдальною формою, на звичайній радіочастоті. Зараз на мікрохвильовій частоті передаються два сигнали – L-1, L-2. Канал L-1 доступний для всіх. Вважається, що він призначений для цивільних користувачів, хоча й військові про нього не забувають. Канал L-2 призначений для військовослужбовців. Цивільні користувачі приймають на свої GPS-приймачі цей канал, але в силу того, що вони не мають доступу до PRN-коду, виникає помилка в позиціонуванні. Тільки дорогі приймачі дозволяють цивільним користувачам працювати в діапазоні L-2. Тому більшість із них приймає сигнал L-1, що дозволяє точно визначати координати від 5 до 10 м.

Складності при прийманні сигналу викликані головним чином тим, що радіохвилі на своєму шляху переборюють іоносферу Землі, що являє собою плазмену хмару, утворену сонячним вітром. Її границі простираються від 70 до 1300 км над поверхнею Землі, і при проходженні через іоносферу радіосигнали послабляються й спотворюються. У нічний час, коли іоносфера перебуває в стані спокою, затримка передачі сигналу становить 1 м, а вдень, коли активність плазми висока, – більше 10 м.

Для того щоб мінімізувати вплив іоносфери, використовують диференційований D-GPS. У такій схемі використовуються два приймачі: один мобільний, а другий перебуває в точці з відомими координатами. Дані, що

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 7    |

надходять із цих GPS, рівняються й обробляються, після чого в ідбувається коректування показань мобільного приймача. Чим ближче вони перебувають, тим точніше визначаються координати.

## 1.2 Область застосування

Починаючи з 2005 р. супутники передають додаткові сигнали, які допоможуть виключити перешкоди від іоносфери. По два сигнали додадуться до військових L-1 і L -2 і один – до цивільних L-1, а існуючі нині сигнали не перетерплять яких-небудь змін. Наступний етап удосконалювання системи почався в 2008 р. Супутники будуть передавати ще один цивільний сигнал L-5, що буде в 5 разів могутнішим, чим зараз. Здвоєний сигнал дозволить мінімізувати вплив іоносфери. GPS-приймачі майбутнього зможуть порівнювати перекручування двох сигналів, вносячи необхідні корективи в розрахунки.

Оператори, що використовують D-GPS-приймачі, також виявляться у вигазі. Нагадаємо, що точність роботи D-GPS-системи знижується в міру того, як збільшується відстань між фіксованим приймачем і мобільним GPS. Це пов'язане з тим що на приймачі попадають сигнали від супутників, що пройшли через різні шари іоносфери. При роботі із двома сигналами мобільний GPS здатний оцінити вплив іоносфери, а дані від фіксованого приймача допоможуть звести до мінімуму інші погрішності, які можуть становити від 30 до 50 див.

Щоб одержати точність позиціонування в межах сантиметрів або навіть міліметрів, користувачі можуть скористатися D-GPS-приймачами. Їхні сучасні моделі, маючи зв'язок зі стаціонарною станцією по радіоканалу, передають відомості про своє місцезнаходження й одержують відкоректовані дані. Довжина хвилі, на якій ведеться передача сигналу із супутника, становить 19 см. Приймач може виміряти час одержання сигналу з точністю до 1%. В абсолютному вираженні ця величина складе кілька міліметрів.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 8    |



пілот може вести літак до висоти 100 м. У зоні аеропорту екіпаж переходить на режим пілотування з використанням наземного навігаційного встаткування.

Згодом навігаційні комплекси LAAS, що працюють у короткохвильовому діапазоні, зможуть забезпечити приземлення при нульовій видимості з використанням каналу L-5. Військово-морські сили США розробляють для авіаносців систему точного наведення й посадки літака JPALS, в основі якої лежить принцип D-GPS-системи, що працює з каналами L-1 і L-2. При заході на посадку й приземленні льотчик морської авіації повинен контролювати відстань до палуби авіаносця з точністю до 1 м, щоб спеціальний гак на корпусі літака зміг зачепити гальмовий канат.

Вчені й інженери вже трудяться над створенням GPS-системи третього покоління. Запуск нових супутників відбудеться не раніше 2012 р. За рахунок використання супутникового зв'язку й установки на них могутніших обчислювальних комплексів істотно розширяться можливості системи.

Ця галузь стрімко розвивається й GPS-приймачами обладнані як легкові автомобілі, так і мобільні телефони й КПК.

Придбавши GPS-приймач вартістю в \$100, людина може розраховувати на відхилення в 5-10 м. Армійські прилади дозволяють визначати місцезнаходження з точністю до 5 м. Якщо ж GPS -приймач одержує сигнал від наземної станції й проводить відповідну корекцію даних, його точність зростає до 0,5 м.

Щоб успішно використовувати навігаційну апаратуру, що працює на технології GPS потрібно розуміти деякі особливості цієї системи.

Система GPS Navstar базується на 24-х супутниках, що досить швидко звертаються навколо землі.

GPS є пасивною системою навігації, і жодним чином не є системою зв'язку. Це означає, що Ви приймаєте сигнал супутників системи, але Ваша апаратура нічого не передає.

Сигнал супутників GPS має частоти 1.227 і 1.575 Ггц. Що це означає для користувача? Для електромагнітної хвилі такої частоти будуть непрозорими

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 10   |

металеві поверхні, деякі пластмаси, дерево, бетон. При цьому, скло пропускає сигнал, листя дерев пропускає, але послабляє. Приблизно оцінити проходження сигналу можна, провівши аналогію зі світлом.

Важливо пам'ятати це, вибираючи місце на транспортному засобі, куди буде встановлюватися антена GPS приймача. Оскільки супутники системи можуть перебувати в будь-якому місці небозводу, ідеальної є така установка антени, коли з її положення небозвід видний цілком. Антена, що лежить на приладовій дошці автомобіля, "побачить" у найкращому разі 2/3 можливих супутників. Так як сигнал GPS буде також екрануватися навколишніми будинками й деревами, забезпечення найкращого огляду для антени є необхідним.

Навіть якщо приймач видає координати із установленної під склом автомобіля антеною, це не означає, що все добре. Якість визначення координат може бути кращою або гіршою, залежно від того набору супутників, з якими працює прилад.

Справа в тому, що для одержання даних про місце розташування (місцевизначення) приймач повинен побачити 4 супутника (у найгіршому разі – 3, але помилка при цьому може бути значної). Як правило, на небозводі їх присутньо набагато більше. Якщо приймач має можливість вибрати з великої кількості прийнятих сигналів кращі, це позитивно позначиться на якості визначення координат. Якщо ж вибору ні, то точність роботи буде труднопередбачуваною.

Після включення GPS приймача навігаційна система почне працювати не відразу. Існує два періоди виходу приймача на режим.

У першому – "гарячий старт" – приймач запускається досить швидко, але тільки за умови, що він не діяв менш 30-ти минут.

Якщо часу пройшло більше – то запуск приймача займе істотно більше часу, до декількох минут. Протягом цього часу можуть бути отримані дані про час і швидкість, але не координати. Цей режим називається "холодним стартом".

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 11   |

Слово "бездіяв" відноситься не тільки до вимикання живлення, а також до втрати GPS сигналу. Наприклад, годинна стоянка із включеним приймачем у залізному ангарі викличе необхідність в "холодному старті".

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

Кафедра КБПЗ – 2021 рік

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 12   |

## 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**

### **Формати подання електронних карт**

#### **Растрове зображення**

Растрове зображення не є картою, утім розумінні, що прийнято в геодезії. По суті, подання растрового зображення на екрані комп'ютера є аналогом форми в якій растр зберігається у файл на диску комп'ютера. Однією з форм цього типу є бітовий образ (бітмап) представляємий файлом з розширенням .bmp. Файли .bmp швидко відображаються на екрані, але займають більше кількості пам'яті на диску. Одним з найбільш популярних методів стиску графічних файлів є .gif формат. Даний формат найбільше ефективно стискає файли, особливо карти. Іншим методом є .jpg. Цей формат менш вдалий для карт, тому що в процесі компресії губиться чіткість ліній. Для рішення цієї проблеми, був розроблений .png формат. Існують ще методи .tif і .tiff. У принципі, метод .tiff не набагато ефективніший .bmp формату, але розроблені версії стислого .tiff, що дозволяє значно зменшити розмір файлів. Варто врахувати, що стислий файл повинен бути декодований перед відображенням на екрані комп'ютера, і щоб цей процес не «нервував» користувача, потрібна відповідна потужність процесора. Іншим відомим форматом є .drg. По суті, цей той же самий .tiff формат, але доповнений деякою каліброваною інформацією.

Основною перевагою растра є можливість використовувати карти будь-яких регіонів і будь-яких масштабів, знявши електронну копію (сканування) з відповідного паперового оригіналу. Користувач може самостійно, за допомогою графічного редактора, внести зміни в зображення карти, додавши дорогу або

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 13   |

вносячи коментарі. Зміна масштабу растрової карти має на увазі наближення або збільшення зображення без зміни інформації. Збільшене зображення представляє із себе наближену картинку. Є програми, які при зміні масштабу, замінюють поточну карту іншою картою, більш детальною.

При досягненні границь карти, програма повинна самостійно або із вручну, за допомогою користувача, перемикатися на сусідній аркуш. Можна об'єднати аркуші сусідніх регіонів єдине зображення й використовувати для переміщення по карті смуги прокручування (scrollbars), але робота з об'єктами на карті вже буде менш зручною

Для використання з GPS приймачами, одного зображення карти недостатньо, необхідно його калібрувати. Це пов'язане з тим, що по суті, плоска карта є проекцією вигнутої земної поверхні. «Зігнутість» тим більше, чим більше площа території охопленої на карті. Додаткові зрушення можуть виникнути в процесі сканування, або оригінал карти зрушає на деякий кут щодо півночі.

Мінімальне калібрування карти можна зробити за допомогою двох точок розташованих на протилежних краях зображення. Передбачається що зображення карти лінійно й не містить перекручувань. Калібрування дозволяє інтерпретувати положення на карті виражене в пікселях у значення широти й довготи, й навпаки

З найбільш популярних навігаційних програмних продуктів, що використовуються для роботи з растровими картами, варто виділити OziExplorer і його КПК-шну версію OziCE. Популярність цієї програми серед користувачів настільки велика, що інші аналогічні програми просто не розглядаються. У мережі Інтернет є велика кількість ресурсів, де ентузіасти викладають уже відскановані й прив'язані карти для цієї програми. Карти покриваються фактично все територію України й представлені в різних масштабах.

### **Векторні карти**

Векторні карти самі по собі не є зображенням і не зберігають картинку місцевості. Фактично, при виклику векторної карти, вона генерується «на льоту» використовуючи інформацію й бази даних. Векторна карта – це база даних, у якій

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 14   |

зберігається інформація про точки, лінії з'єднуючі ці точки й полігонах, які є замкнутою послідовністю ліній. Кожний об'єкт має додаткові атрибути характеризуючи властивість, колір, підписи. Після того, як GPS координати приймача обчислені, його місце розташування відображається на карті відносно положення інших географічних об'єктів з бази дані векторної карти. При цьому на екран приймача (або комп'ютера) виводяться тільки об'єкти, розташовані в безпосередній близькості від поточного місця розташування, з урахуванням обраного масштабу. Переміщення приймача, супроводжується зміною «картинки», об'єкти, які виходять за межі, ховаються, і відображаються нові ділянки карти. Помітимо, що все це відбувається «на льоту». Навігаційна програма, що використовує векторні карти, аналізує поточні координати, обраний масштаб, налаштування приймача, і створює нове відображення. При цьому, залежно від масштабу, той самий об'єкт може бути представлений у різному виді – схематично, докладно, або взагалі не відображатися. Це зроблено для прискорення роботи з картою. Кожне переміщення вимагає нового перемальовування карти й всіх її видимих об'єктів. Чим більше об'єктів і чим детальніше вони представлені, тим більше часу займе цей процес. Відповідно, на маленькому масштабі (віддаленому) нема рації докладно вимальовувати кожний поворот дороги або вигин ріки, досить представити їх у загальному виді. На маленьких масштабах міста, не має змісту відображати кожний будинок, тому що всі вони зіллються в одна єдина чорна пляма. При збільшенні масштабу (наближенні) карти, детальність карти, а саме, складових її об'єктів, повинні збільшуватися. Відображати або не відображати об'єкти на різних масштабах, вирішує навігаційна програма використовуючи при цьому атрибути об'єктів і налаштування виду, установлені користувачем.

Підписи до об'єктів, так само зберігаються в базі даних векторної карти й можуть динамічно довантажуватися для відображення на карті. Деякі програми дозволяють змінювати налаштування пов'язані з поданням написів на карті. Залежно від поточного масштабу, написи можуть бути сховані або відображатися

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 15   |

вибірково. У міру наближення карти, написи проявляються, намагаючись не перекривати при цьому один одного.

На відміну від растрові зображення, що мають єдиний стандартизований формат (.jpg, .bmp, .gif і т.п), векторні карти поширюються в різних форматах. Деякі можуть бути представлені в єдиних картографічних форматах і використовуватися з більшістю популярних програм, інші тільки у власних, здатних працювати тільки з однією фірмовою програмою. «Закритість» формату обумовлена декількома причинами. По-перше, виробники навігаційного встаткування, разом з картами змушують купувати користувачів приймачі й програми тільки своєї фірми. По-друге, картографічні дані є інтелектуальною власністю, у виробництво якої були вкладені дійсно великі гроші, і компанії не зацікавлені в поширенні цих даних і використанні їх в інших навігаційних продуктах.

З іншого боку, всі ці «закриті» формати при бажанні можуть бути «зламани».

Але залишаються й інші програми: «iGo», «TomTom» «Destinator», «Автосупутник», які використовують картографічну базу відомих постачальників «TeleAtlas» і «NavTech», і не поспішають відкривати свій формат, змушуючи сумлінних користувачів здобувати карти за окремі не маленькі гроші, а несумлінних – використовувати піратські копії.

### **Виставлення шляхових точок, маршрутів, треків**

#### **Шляхові точки ( waypoints)**

Шляхові точки служать для ідентифікації розташування різного роду місць, цікавих для користувача. Це можуть бути й фізичні об'єкти, і адреси, і просто точки на карті. Сучасні приймачі можуть зберігати до 500 і більше шляхових точок. В Інтернеті є сайти, що містять велику кількість точок, відсортованих по категоріях, доступних користувачеві. Більше того, більшість навігаторів містять вбудовану базу «цікавих» точок (POI – point of interest). Кількість точок у базі може досягати кілька тисяч, існують фірмові програми для

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 16   |

їхнього відновлення, але вони недоступні користувачеві для редагування. Зберігаються вони в окремій пам'яті навігатора.

Введення шляхової точки в навігаторі може здійснюватися декількома способами:

1. *Введенням точних значень координат.* У цьому випадку, користувачеві необхідно знати не тільки точні координати, але й назва датума в якому ці координати представлені.

2. *Введення щодо відомої точки.* Якщо точні координати точки невідомі, але відомо її напрямок і відстань щодо якоїсь опорної точки, то може використовувати даний метод. При цьому опорна точка повинна бути збережена в навігаторі раніше.

3. *Введення точки на карті.* Метод застосується в приймачах з картографічною підтримкою. За допомогою курсору на екрані карти вибирається місце планованої точки й здійснюється введення.

4. *Введення поточного місця розташування.* Шляхова точка зберігається з координатами поточного місця розташування навігатора. З огляду на, що визначення позиції відбувається з деякою помилкою, величина якої може досягати декількох метрів, координати точки будуть не точні. Для підвищення точності координати точки, використовується спеціальний режим усереднення, які обчислює середнє значення координат за певний інтервал спостереження. Інтервал встановлюється користувачем і може досягати декількох хвилин. Весь цей час користувач повинен перебувати в одній точці.

5. *Завантаження з комп'ютера.* Як джерела, можуть використовуватися бібліотеки точок в Інтернеті, обмін з іншими користувачами, або раніше збережені на комп'ютері точки з минулих поїздок.

Кожна шляхова точка в момент створення автоматично одержує ім'я. У більшості випадків це звичайний тризначний номер, що послідовно збільшується при створенні чергової точки (001- 999). Користувач може змінити це поле, давши точці більше зрозуміле й зручне для подальшого використання ім'я.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 17   |

Залежно від моделі навігатора, це поле може містити від 6 до 10 символів. Для більше повного опису точки, існує окреме поле, у яке можна ввести від 50 символів. Але все-таки, найбільш важливу роль при описі властивості шляхової точки, приділяється символу, за допомогою якої вона відображається на карті. Навігатори містять більші бібліотеки символів, за допомогою яких можна найбільше точно охарактеризувати точку. Існують символи із зображенням будинків, машин, яхт, коробок, заправлень, черепів, схованок і т.п. Сучасні моделі приймачів дозволяють обновляти й редагувати ці бібліотеки

### **Маршрути**

Маршрути є послідовними наборами шляхових точок, що характеризують «ключові» ділянки шляхи. Звичайно, у сучасних приймача, максимальне кількість точок у маршруті не перевищує 50-і, а загальна кількість маршрутів – 20. Є приймачі, які підтримують тільки 1 маршрут. Як точки маршруту, можуть виступати також точки POI

Супровід по маршруту аналогічно функції «goto», що використовується при роботі зі звичайними шляховими точками. Тільки в цьому випадку, навігаційний приймач сам, автоматично, перемикає навігацію до чергової точки маршруту при досягненні попередньої. Фактично, ніщо не заважає користувачеві відмовитися від маршрутних функцій, відразу визначити кінцеву точку й рухатися до неї у відповідності зі стрілкою навігатора. Це зручно, якщо на шляху немає ніяких перешкод і можна зберігати прямолінійність руху до точки. Насправді, реальна навігація набагато складніше. Користувач, маючи інформацію про кінцеву точку, самостійно вибирає шлях. Якщо рух відбувається на автомобілі, то в облік беруться дорожня мережа, правила руху, знаки, розмітка й можливі утруднення, пов'язані з аваріями, пробками й дорожніми роботами. Тому рухатися по стрілці, що чітко вказує на стіну найближчого будинку просто неможливо. І вся статистична інформація про шлях, що залишився, до точки, теж буде невірною. Маршрут, що складається з набору ділянок шляхи між шляховими точками, надає більше достовірну інформацію. Точність маршруту

|      |      |          |        |      |  |                                  |      |
|------|------|----------|--------|------|--|----------------------------------|------|
|      |      |          |        |      |  | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |  |                                  | 18   |

багато в чому визначається кількістю шляхових точок з яких він створений. Є кілька способів створення маршрутів.

1. *Автоматична генерація.* Найбільш зручний і ефективний спосіб, реалізований у більшості сучасних моделей навігаторів. Досить визначити кінцеву точку, і навігатори самостійно прорахує маршрут з урахуванням дорожніх знаків, розмітки й правил руху. Для використання цього способу недостатньо використовувати навігатори підтримуючу функцію автороутингу, необхідні також карти, які містять всю необхідну дорожню інформацію. Користувач сам може вибрати кращий для генерації тип маршруту – найкоротший, економічний, або найшвидший.

2. *Використання збереженого треку.* Простий спосіб, у якому для генерації маршруту використовується раніше записаний і збережений у приймачі трек.

3. *Ручне введення точок маршруту на карті.* Додавання точок у маршрут здійснюється простим вибором місця на карті. При виборі чергового місця, автоматично створюється шляхова точка й додається в маршрут.

4. *Послідовність шляхових точок.* При створенні маршруту використовуються створені раніше шляхові точки.

5. *Завантаження з комп'ютера.* З огляду на, що маршрут фактично є послідовною сукупністю шляхових точок, те його також можна зберегти на комп'ютері.

Рух по маршруту здійснюється вибором потрібного маршруту зі списку, його інвертуванням (при необхідності) і активізацією. Після цього, вся навігаційна інформація відображується на екрані, буде відноситися до початкової точки маршруту. Коли точка буде досягнута, навігатор автоматично «перемкнеться» на наступну точку зі списку маршруту.

Налаштування приймачів, дозволяють користувачеві встановлювати «попереджуючі» сигнали пов'язані з наближенням до чергової точки маршруту. Це можуть бути звукові або голосові повідомлення які видає приймач за деяку

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 19   |

відстань або за деякий оцінний час, що залишився до точки. Час оцінюється з обліком поточної середньої швидкості. У сучасних навігаторах, з автоматичною прокладкою маршрутів, масштаб карти може автоматично збільшитися, щоб відбити всі особливості дорожнього руху й полегшити маневри водієві. Після того, як точка буде пройдена, масштаб карти відновлюється.

### Треки

Треки є послідовністю точок, що повністю відбиває пройдений шлях. При цьому не варто плутати точки треку й шляхові точки. Це різні об'єкти, що мають різне призначення й різні атрибути. Точки треку містять інформацію про поточні координати й час. Деякі моделі також включають в опис точки значення висоти. Швидкість підраховується по даними координат сусідніх точок і різниці в часі запису. Записані треки можуть бути збережені на комп'ютері й точно відобразити пройдений шлях, або за допомогою функції «trackback» допомогти користувачеві повернутися в початок свого шляху, майже що «слід у слід».

Чим більше точок може бути збережене в треку, тим точніше буде представлений пройдений шлях. Сучасні приймачі містять до 5000 точок у треку. Трек може бути збережений, але при цьому його деталізація істотно погіршується. Мало того, що для збережений трек не може містити більше 256 точок, атрибути точок містять тільки інформацію про координати. Час і висота не зберігаються.

Треки є безперервної послідовності точок шляху. І якщо приймач був на якийсь час виключений і потім включений в абсолютно іншому місці, то трек «чесно» з'єднає дві сусідні точки в єдиний шлях, навіть якщо відстань між ними становить сотні кілометрів. Це є особливістю запису треків, і якщо користувач планує використовувати надалі дані треків, то їх треба вчасно перетворювати в «збережені» треки.

У простих моделях навігаторів, запис треків здійснюється автоматично й не доступна для конфігурації користувачеві. Трек пишеться завжди. При

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 20   |

заповненні пам'яті, «нові» дані циклічно записуються поверх «старих». Більше сучасні й функціональні моделі надають користувачам кілька режимів запису:

- OFF – дані треку не пишуться. Цей режим автоматично включається, коли в приймач завантажується трек з комп'ютера.

- WRAP – постійний запис треку. При заповненні пам'яті «нові» дані затирають «старі»

- FILL – зупинка запису при заповненні пам'яті. Коли в пам'яті не залишилося вільних місць, на екрані відображається відповідне попереджуваче повідомлення

Додатково, у деяких моделях користувачі мають також можливість управляти частотою запису треку.

- Автоматичний. Інтелектуальний режим, у якому запис сусідніх точок здійснюється тільки у випадку зміні прямолінійного руху на ділянці більше 25 метрів (для деяких моделей 50 м), або при значній зміні швидкості.

- *По відстані.* Точки треку записуються кожні N метрів шляху.

- *За часом.* Запис здійснюється із установленим часовим кроком.

У кожному разі, на екрані навігатора відображається індикатор пам'яті треку, процентне заповнення якого відповідає розміру вільної пам'яті, що залишилася. Якщо планується надалі використовувати записаний трек, то рекомендується спочатку повністю звільнити пам'ять від попередніх записів, установити режим, що найбільше відповідає характеру маршруту й стилю водіння, і по прибуттю в кінцеву точку шляху, зберегти дані на комп'ютер.

## **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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 21   |

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.

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

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

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

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

|      |      |          |        |      |  |                                  |      |
|------|------|----------|--------|------|--|----------------------------------|------|
|      |      |          |        |      |  | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |  |                                  | 22   |

- Підтримка Metal Driver GPU для macOS і iOS.
- Вбудований Fmxlinux.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

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

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

- Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

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

- 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++,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 23   |

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.

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

### **Єдине керування пам'яттю**

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 24   |

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

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

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

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

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

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

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

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

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 25   |



- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

### **Оновлений менеджер пакетів Getit**

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

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

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

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

## **2.3 Розгорнута постановка завдання**

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи обробки навігаційних даних.

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 27   |

технологічного обладнання на виробництві в автоматизованому режимі.

Розробити функціональну та структурну схеми системи;

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 28   |

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Сполучення мобільних пристроїв з GPS

На даний момент існує, як мінімум, три найбільш затребувані способи спільного використання GPS і КПК.

По-перше, до наладоника можна докупити опціональний модуль. Останній може помітно розрізнятися по способу сполучення, але в основному найбільш популярні зараз карткова модифікація й Bluetooth-варіант. Перший не дуже зручний через те, що займає один або єдиний слот розширення, зате відносно дешевий, ну а другий цілком ергономічний – не прийдеться піклуватися про точне позиціонування, от тільки ціна його, як правило, досить висока.

Другим способом є збагатити наладонник GPS-функціями при їхній відсутності в штаті, це є з'єднання звичайного навігатора й КПК за допомогою спеціального кабелю. Подібний варіант гарний своєю роздільністю й повноцінністю GPS-частини: при відсутності потреби в КПК можна відправлятися в дорогу лише з навігатором, не обтяжуючи себе додатковою ношею.

Ну й по-третє, існує можливість сполучити обоє пристрої в одному корпусі, одержавши КПК із інтегрованим модулем GPS. Пристрій значно мобільніший багатьох персональних GPS-навігаторів. При цьому кишенькові комп'ютери мають відмінний кольоровий екран, про яке спеціалізовані приймачі можуть тільки мріяти. Дисплеям наладоників властива чутливість до натискань, більш по мірках навігаторів розміри, порівняно високий дозвіл, багата палітра кольорів і т.д. Загалом, звичайні приймачі програють по всіх статтях, будучи

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 29   |

оснащені несенсорним, часто монохромним екраном зі скромними характеристиками й досить високою інертністю.

Зовнішністю КПК, як правило, наділені більш привабливою, ніж навігатори. Правда, у подорожі це навряд чи придасться, а от відсутність у корпусів наладоників водонепроникності, а також їхні слабкі характеристики до фізичних навантажень досить недоречні.

Функціональність GPS-приймача – одна з основних якостей подібних пристроїв, оскільки в подорожі дизайнерські ізиски ніхто не оцінить, та й ергономіка відійде на другий план. Тому спеціалізовані пристрої настільки популярні: вони націлені виробником на навігацію, а тому мають весь базис необхідних функцій. Окремо завантажуються карти, але останнім часом багато продавців стали комплектувати навігатори подібними бонусами штатно. Як же обстоять справи із цим у КПК. Майже навпаки. Справа в тому, що наладоники – багатофункціональні пристрої, і “заточувати” їх під якесь певне завдання було б украй недоцільно. Ще більш мрячна перспектива вкласти в софтверну частину КПК весь його потенціал: навіть за умови використання карти пам'яті це навряд чи здійснено й майже ніколи не потрібно, оскільки кожний купує КПК для своїх власних потреб. Через це GPS-здатності КПК звичайно не забезпечені програмним комплектом. Як правило, користувачеві доводиться самому вишукувати потрібне.

Якщо розглядати апаратну половинку КПК із GPS і персонального навігатора, то базових розходжень тут мало – звичайно приймачі так само, як і КПК мають 12-ти канальність і підтримку WAAS, а також інші основні характеристиками. Зате опціональності в наладоників відсутні. Ще одна перевага КПК з інтегрованим навігаційним модулем перед звичайними GPS-приймачами – наявність слота розширення, а точніше, підтримка SDIO, тобто здатність працювати з периферією. Подібний дріб'язок додасть наладонику функціональності, оскільки дозволить взяти в подорож, скажемо, камеру, Bluetooth-адаптер або GSM-модуль, а також багато чого іншого – на

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 30   |

сьогоднішній день спектр доступного устаткування досить великий, і кожний зможе вибрати щось цікаве й корисне конкретно для себе. GPS -навігатори, як правило, позбавлені подібної можливості, а роздільні набори GPS і КПК не занадто зручні, оскільки далеко не всі наладоники оснащені двома слотами розширення.

### **Розрахунок часу старту GPS навігатора**

Час «старту» необхідний навігаційному приймачу на визначення позиції після включення, залежить від наявної в пам'яті початкової інформації. Виділяються наступні режими:

- «Холодний» старт («автопошук») – час, позиція, альманах і ефемериди невідомі
- «Теплий» старт – позиція й ефемериди невідомі, час і альманах відомі
- «Гарячий» старт («перезахват») – альманах, ефемериди відомі, час і позиція відомі з деякою помилкою

Навігаційні повідомлення передані із супутників містять два типи даних – ефемериди й альманах супутників. В альманасі передаються параметри орбіти, за допомогою яких можна обчислити зразкове місце розташування супутників з достатньо великим ступенем погрішності. Альманах, що зберігається в пам'яті приймача, постійно оновлюється, тому що кожний супутник передає дані альманасу для всіх супутників угруповання. Час «життя» альманасу становить 2-3 місяці. Далі, величина накопиченої помилки в розрахунках буде неприпустимою.

Дані ефемерид містять параметри, що дозволяють більш точно обчислити поточне місце розташування супутників. На відміну від альманасу, кожний із супутників передає, тільки свої власні ефемериди. Час «життя» ефемерид не перевищує 4-6 годин.

Інформація даних ефемерид і альманасу, передана із супутників, постійно коректується. Це відбувається один раз у добу. Мережа наземних станцій, одержує інформацію із супутників, за аналогією зі звичайними користувачами,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 31   |

аналізує виміри, порівнює їх з опорними, розраховує коригувальні виправлення й передає їх на головну станцію, з якої здійснюється передача даних на супутники.

«Холодний» старт приймача може бути зв'язаний не тільки з його тривалою бездіяльністю, але переміщенням на велику відстань у виключеному стані. Якщо перший випадок пов'язаний із застарілим альманахом і помилкою у визначенні поточного точного часу, то в другому випадку приймач, не знаючи про своє переміщення, буде намагатися знайти супутники, яких повинні бути видимі на «старому» місці. Користувач може «допомогти» приймачу й зменшити час «холодного» старту, указавши на базовій карті, зразкове «нове» місце розташування. Під час «холодного» старту приймач сканує весь діапазон можливих значень частот і тимчасових затримок навігаційних сигналів. При цьому, у багатоканальних приймачах, кілька каналів можуть використовуватися для пошуку одного супутника, щоб прискорити час його захвата. Після того, як сигнал хоча б від одного супутника буде отриманий і розібраний, приймач буде мати повну інформацію про альманах всього угруповання й, по суті, перейде до «теплого» старту

При «теплому» старті, приймач, включений після 6-ї годин бездіяльності, почне «пошук» сигналів супутників, використовуючи значення поточного часу й дані, що зберігаються в пам'яті, альманаху. Буде здійснюватися пошук тільки тих супутників, які, по теоретичних розрахунках перебувають у видимій півкулі й повинні бути доступні приймачу. Відповідно, відомий досить вузький діапазон частот і часових затримок, що потрібно просканувати у процесі пошуку сигналів. Ця інформація істотно прискорює час захвата супутників, у порівнянні з «холодним» стартом, коли пошук ведеться на широкому діапазоні всіх можливих значень затримок і частот

Варто відзначити, що в момент включення, багатоканального приймача починає пошук сигналів з декількох супутників одночасно. Інформація передана із супутників прив'язана до єдиної шкали часу, містить однакову структуру й досягає антени приймача, приблизно в один й той же час. Тому дані ефемерид,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 32   |

одночасно захоплених супутників, надійдуть у приймач майже що одночасно. Якщо кількість таких супутників більше або рівняється трьом, то це дозволяє приймачу відразу ж розрахувати позицію. У випадку, коли сигнали блокуються перешкодами, то може знадобитися досить тривалий час на визначення позиції.

Наявність повністю отриманих ефемерид, не гарантує використання цього супутника у підрахунку позиції. Інформація передана в ефемеридах може бути неправильною, помилковою, або пов'язаною з несправністю в роботі супутника. Це може бути зв'язано не тільки з несправністю супутника, але й діагностичними роботами проведеними на його борті, процесом уведення його в експлуатацію або тестуванням нових режимів.

«Гарячий» старт пов'язаний з короткочасним вимиканням приймача (до 6-і годин) не вимагає тривалого часу на визначення позиції. Це пояснюється тим, що отримані раніше ефемериди містять «свіжі» дані, використовувані для визначення точних координат супутників і можуть використовувати в обчисленні позиції. У випадку включення приладу після граничного часу, ефемериди розглядаються застарілими й починає діяти принцип «теплого» старту. Якщо на момент включення приймача видимими залишилися менш 3-х супутників з «свіжими» ефемеридами, то для визначення позиції буде потрібно якийсь час на збір даних ефемерид нового супутника.

Дані ефемерид передаються в складі трьох пакетів. Кожний з пакетів містить однаковий часовий ідентифікатор (IOD – issue of data) по якому можна об'єднати загальну інформацію. Інформація ефемерид передана із супутників кожні 30 секунд, змінюється раз в 2 години, і містить однаковий на цей час IOD. Якщо один з пакетів був пропущений, або отриманий з помилками, то можна виділити аналогічний пакет з наступного повідомлення, перевірити його ідентифікатор і не чекаючи наступних пакетів, використовувати його з раніше отриманими. Це дозволяє приймачу прискорити час «старту».

Існує мінімальний можливий час, необхідний приймачу на «старт», і це визначається структурою переданого сигналу із супутників. Виробники

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 33   |

навігаційної апаратури, використовуючи стандартні методи навігації, можуть наблизитися до цього часу, але зменшити його не зможуть. Одним з методів, призначених для рішення цієї проблеми, є Assisted-GPS (A-GPS). Його принцип полягає в обчислення точного місця розташування супутників без інформації ефемерид, на одержанні яких потрібен час. Обчислення здійснюється на використанні точних моделей орбіт супутників, доступних через спеціальні Інтернет – сервіси.

З іншого боку, максимальний час «старту» може значно перевищувати заявлене в технічній специфікації на навігатор час. Це пояснюється навколишніми умовами, у яких відбувається «захват» супутників і «старт» приймача. Якщо приймач перебуває в умовах сильних фізичних перешкод, то навігаційний сигнал піддається зовнішньому впливу, містить помилки й неправильно декодується. Більше, того геометричний фактор цих супутників, що є одним із критеріїв точно визначення позиції, сильно погіршується. Всі ці умови можуть значно збільшити час «старту» приймача.

### **Обчислення радіуса окружності помилки для оцінки точності GPS-вимірів**

Показник CEP – окружність можливої помилки (Circular Error of Probability) один з можливих шляхів оцінити точність вироблених GPS вимірів у даній точці тепер. Завдяки великій кількості факторів зовнішнього середовища впливаючих на виміри – в одній точці показання приладу будуть різними в різні моменти часу. До таких факторів відносяться вплив іоносфери, вплив нижніх шарів атмосфери, багатопроменевість, наявність перешкод на шляху сигналу. Показник CEP використовує опорну точку, або задаваємо користувачем, або що обчислюється як середнє геометричне між всіма вимірами, для того, щоб побудувати серію окружностей що показують відповідно 50, 90, 95, 99% можливої помилки.

Для того, щоб визначити CEP повинна бути взята серія вимірів зроблена в одній точці. Наприклад, включений і нерухливий GPS з інтервалом в 2-5 сік

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 34   |





точок ІНГЕО потрібно застосувати розтягання, а потім зсув, щоб одержати Пулково-1942.

Для обчислення параметрів розтягання й зсувів, необхідно мати хоча б дві точки на карті, для яких ми знаємо й старі й нові координати. Загалом кажучи, точок потрібно як мінімум дві, можна більше, але не обов'язково чим більше тим краще. Справа в тому, що задаємо дві точки, і обчислюючи по них параметри перетворення, ми домагаємося того, що ці дві точки будуть однозначно збігатися на місцевості й карті, – тобто в них прив'язка буде абсолютно точною. Погрішність прив'язки точок, що відстоять недалеко від двох точок прив'язки, буде невелика, але вона буде зростати при видаленні від них. Якщо прив'язувати карту по багатьом точкам, то неможливо буде знайти параметри перетворення, що абсолютно прив'язують всі ці точки, зате карта в цілому, можливо, буде прив'язана краще. Формули для обчислення параметрів прив'язки у випадку більше двох точок прив'язки можна виписати, наприклад, скориставшись будь-яким методом апроксимації лінійної функції – хоч методом найменших квадратів.

Отже, точка1 має координати на місцевості ( $n_{x1}=439395$ ,  $n_{y1}=6072078$ ), а в карті ( $x_{x1}=3208.34$ ,  $y_{y1}=5030.94$ ); а точка2 – ( $n_{x2}=433902$ ,  $n_{y2}=6068556$ ) на місцевості, і ( $x_{x2}=-397.74$ ,  $y_{y2}=-427.93$ ) у карті.

Помітимо із самого початку, що в Пулково перша координата задає зсув по горизонталі, а друга по вертикалі, а в ІНТЕГРО, як і в WGS84, навпаки. Тому щоб обчислити параметри перетворень коректно, необхідно поміняти місцями координати в карті. Тоді перша точка на карті буде мати координати ( $x_{x1}=5030.94$ ,  $y_{y1}=3208.34$ ), а друга ( $x_{x2}=-427.93$ ,  $y_{y2}=-397.74$ );

Підставивши координати двох точок в (3.1) одержуємо систему рівнянь, з яких перебувають параметри перетворень у такий спосіб:

$$s_x = (n_{x1} - n_{x2}) / (x_{x1} - x_{x2})$$

$$d_x = (n_{x2} * x_{x1} - n_{x1} * x_{x2}) / (x_{x1} - x_{x2})$$

$$s_y = (n_{y1} - n_{y2}) / (y_{y1} - y_{y2})$$

$$d_y = (n_{y2} * y_{y1} - n_{y1} * y_{y2}) / (y_{y1} - y_{y2}),$$

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 37   |

підставляючи куди наші координати одержуємо:  $sx = 1.0062522097064$ ,  
 $dx = 434332.6055081$ ,  $sy = 0.976683822876919$ ,  $dy = 6068944.46622371$

Тепер можна переходити до вивантаження об'єктів з ІНГЕО. Вибиремо там сервіс-експорт-MID/MIF. Вибираємо шари, які необхідно експортувати (наприклад, тільки вулиці), указуємо, куди експортувати, і на наступному вікні задаємо параметри трансформації координат. Задаємо перше перетворення: "збільшення-стик",  $1.0062522097064$  по X і  $0.976683822876919$  по Y. Потім задаємо два послідовних перетворення, що приводять до того, що X і Y поміняються місцями (відбиття відносно прямої  $y=x$ ): друге перетворення: "поворот" на 90 градусів, третє перетворення: "збільшення-стик" на -1 по X і 1 по Y. І останнє перетворення: "зсув",  $10434332.6055081$  по X і  $6068944.46622371$  по Y. Цифри 10 я приписав до споконвічного зсуву по X, рівному  $434332.6055081$ , для того, щоб формально привести використовувані координати до виду Пулково-1942, де, як відомо, до зсуву по X приписується ліворуч номер зони, що для України є 10 (центральний меридіан – 57). Натискаю ОК і в обраній директорії зберігаються файли MID і MIF для кожного із шарів. Дані вивантажені.

#### **Імпорт даних у відомий формат.**

Всі вивантажені шари лежать у файлах MID і MIF. Для початку перейменую їх у коротких, мовців назви, якщо це ще не так. Далі конвертуємо MIF-MID-файли у формат SHP. Найкраще створити bat-файл, що запускає перекачування в SHP і внести в нього рядка типу:

```
mifshape.exe point streets streets_pt
```

```
mifshape.exe text streets streets_t
```

```
mifshape.exe line streets streets_l
```

```
mifshape.exe poly streets streets_p
```

(за умови, що експортовані файли streets.mif і streets.mid лежать у тій же каталозі, що й конвертер mifshape). Програма створювала по трьох файлу для кожного типу даних, які будуть скачані з MIF -формату: для точок, для тексту, для ліній, і для замкнених полігонів.

|      |      |          |        |      |  |                                  |      |
|------|------|----------|--------|------|--|----------------------------------|------|
|      |      |          |        |      |  | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |  |                                  | 38   |



виправити ситуацію за допомогою простого зрушення, а якщо це не вийшло, тоді вже намагатися застосовувати афінне перетворення, попередньо зберігши карту в безпечне місце, через деяку непередбачуваність поведження об'єктів карти при такому складному перетворенні.

### **Завантаження карти в GPS.**

Після того як карта виходить прив'язаною із задовільною точністю, все інше виконується як з будь-якою іншою картою. А саме, настраюються властивості карти (її ID і NAME, обов'язково CODEPAGE – 1251, якщо ви хочете, щоб український текст відображався коректно), потім карта зберігається в форматі MP, і з тої ж програми викликається програма, що готує файл IMG, заливається безпосередньо в GPS за допомогою розробленого програмного забезпечення або його візуального розширення img2gps.

### **3.2 Розробка структурної схеми**

На рисунку 3.2 представлена структурна схема роботи системи. При розробці системного програмного забезпечення обробки навігаційних даних використовувалася технологія GPS (Global Positioning System) – "Система глобального позиціонування" система визначення місця розташування об'єктів, заснована на використанні штучних супутників Землі (більш докладно розглянуто вище).

Розроблена програма є універсальною системою, що може використовуватися у всіх мобільних пристроях, підтримуючих систему навігації GPS і маючих віконний інтерфейс на основі операційної системи Windows mobile.

Коли користувач набудовує й запускає розроблену програму, відбувається пошук штучних супутників Землі для визначення координат. Сучасні приймачі GPS вбудовані в мобільні пристрої одночасно можуть приймати дані з 12 супутників, що забезпечує високу точність позиціонування (50 – 100 метрів).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 40   |



приймачем через апаратну частину й операційну систему приймають цей сигнал у розроблену програму по запиту користувача.

Через розроблений інтерфейс програми, при правильному налаштуванню й запуску можна побачити наступні геопараметричні дані:

- геопараметрична широта;
- геопараметрична довгота;
- висота над рівнем моря;
- сторони світа, куди спрямований мобільний пристрій;
- поточний напрямок (сторона світу);
- година сходу сонця;
- час заходу сонця;
- прискорення при русі;
- горизонтальна швидкість при русі;
- вертикальна швидкість при русі;
- кут підйому;
- кут схилу;
- довжина маршруту;
- поточний час із виправленням;
- максимальна висота;
- швидкість розгону 0-100 km/h;
- швидкість розгону 80-120 km/h;
- загальний час руху.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 42   |

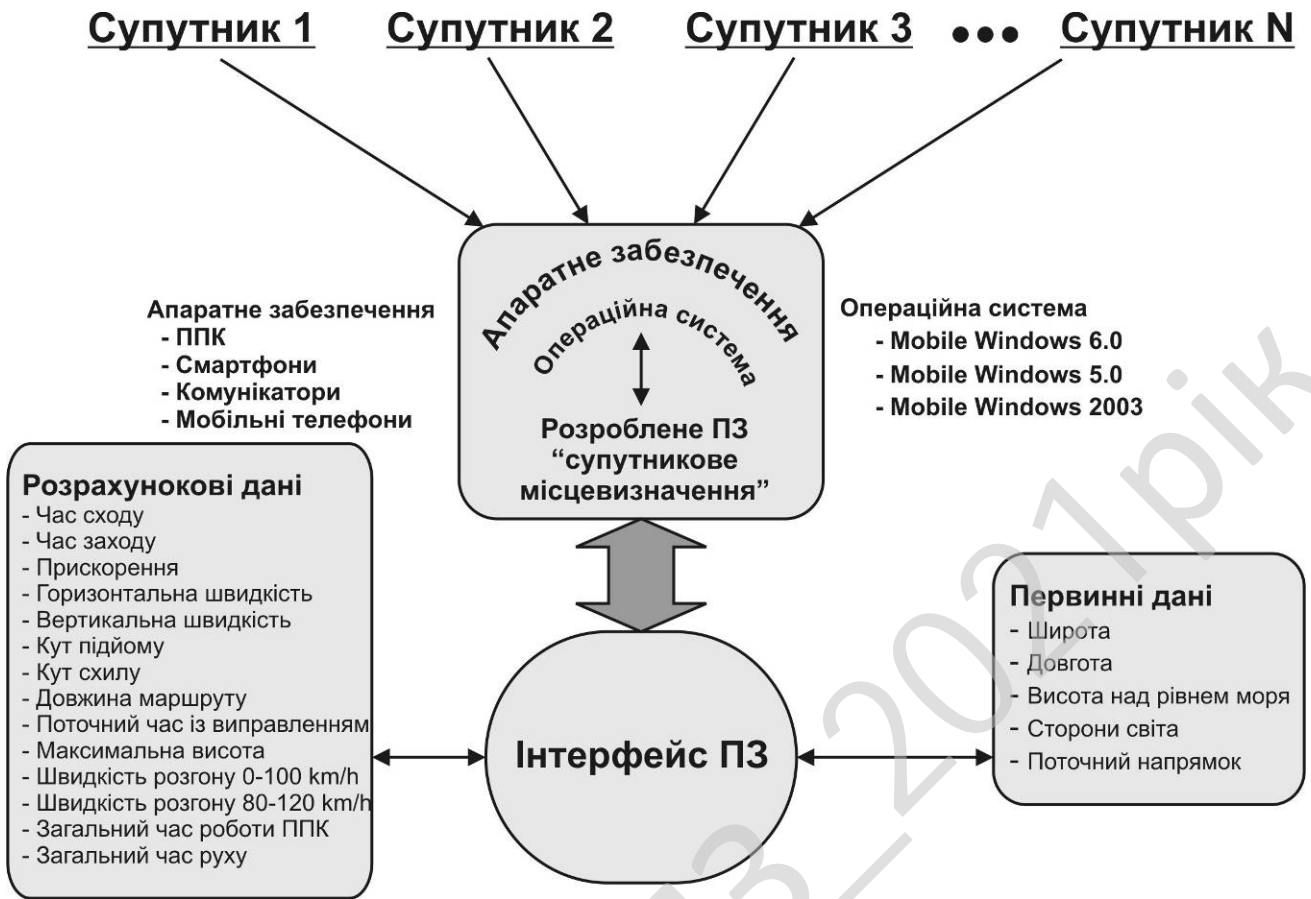


Рисунок 3.3 – Функціональна схема роботи системи

### 3.4 Розробка діаграми процесів

На рисунку 3.4 представлена розроблена діаграма процесів. Починається і закінчується розроблена програма у першому блоці, це є основною крапкою відрахунку діаграми. При переміщенні по стрілках можна побачити загальну схему взаємодії блоків і їх входження один в одного.

Через модуль захисту ПЗ та ОС та бібліотеки налагодження взаємодії з бездротовою лінією зв'язку (Wi-Fi) ми попадаємо до інтерфейсу мобільної операційної системи через яку виводиться інтерфейс розробленого програмного забезпечення.

Інтерфейс розробленого програмного забезпечення виконує тестування лінії зв'язку з модулем GPS, виконує виклик форми авторського права та

отримує первинну та оброблену розрахункову інформацію при задовільній перевірці лінії, зв'язку з модулем GPS.

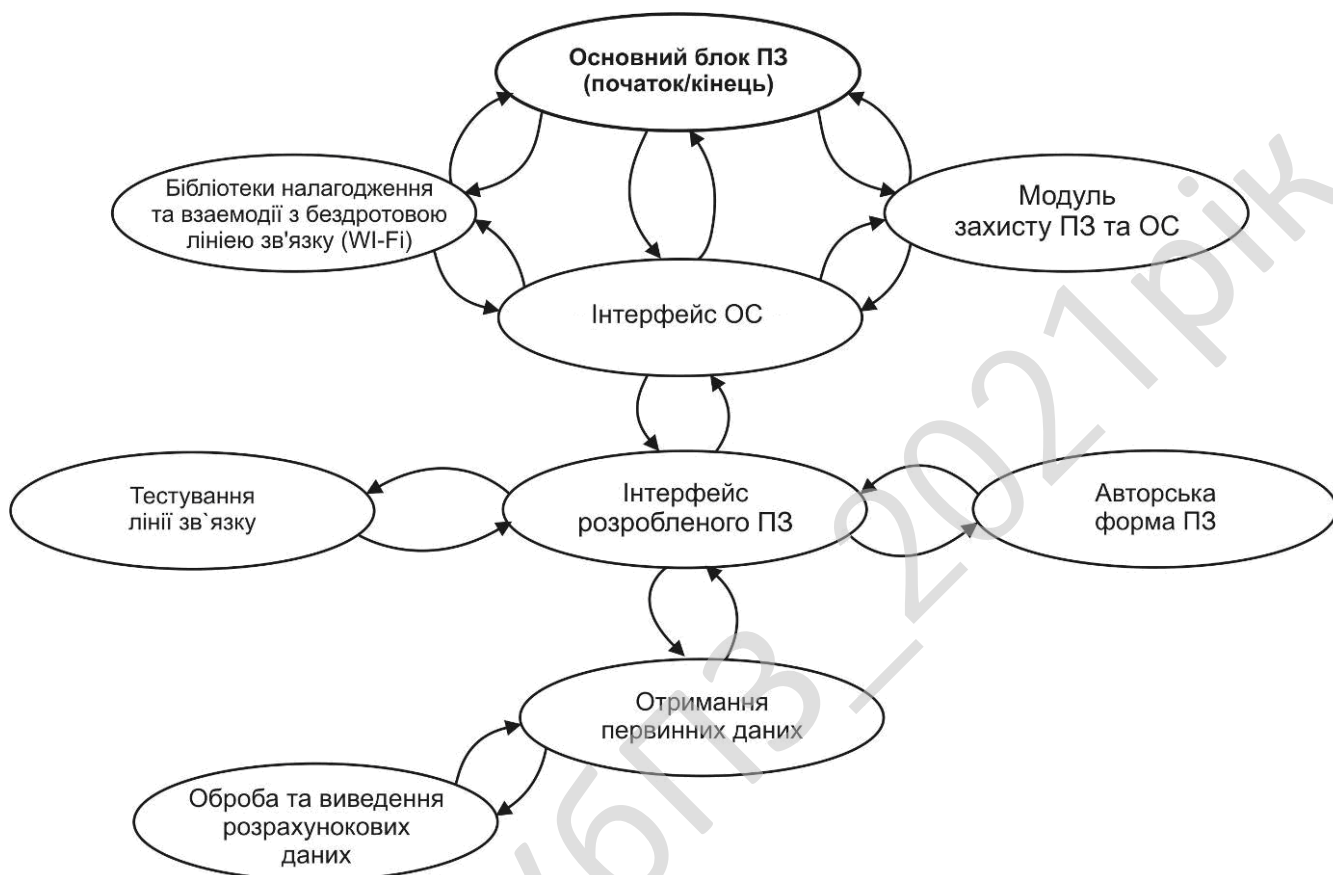


Рисунок 3.4 – Діаграма процесів

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

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

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

На рисунку 4.1 показана основна блок-схема програми, у якій реалізований алгоритм роботи розробленої системи визначення геопараметричних даних об'єкта з відображенням на мобільні пристрої.

При розробці програми застосовувалася тестова оболонка Microsoft Windows 6.0 Emulator, технічні дані Microsoft Compact Frame Work SDK і Compact Framework Build Helper с використанням середовища розробки Delphi.

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. Тому від точності й детальної блок-схеми залежить результат всієї програми.

Розроблена частина складається з декількох основних блоків.

Перший блок ініціалізація програми – виділення пам'яті програми, ініціалізація початкових значень і модуля GPS, підключення файлів, тест доступу й розпакування ресурсів. Розпакування ресурсів грає особливу першорядну роль, при розробці програм на мобільних платформах, облік дискового простору, виділюваної пам'яті дозволяє оптимізувати роботу програми. Забезпечити захист від несанкціонованого розповсюдження програми. Другий блок робота програми – після проведення початкових ініціалізацій і перевірок на екран мобільного пристрою виводиться основне вікно програми з переходом у режим очікування дій користувача. Користувач може зробити дії настройки параметрів, виводу авторських даних і запуску програми визначення поточного положення. Алгоритм визначення поточного положення винесений у підпрограму (рисунок 4.2). Третій блок завершення роботи програми – звільнення виділених ресурсів, вивід повідомлень при виникненні помилки, видалення тимчасових файлів, які з'являються, при розпакуванні.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 45   |

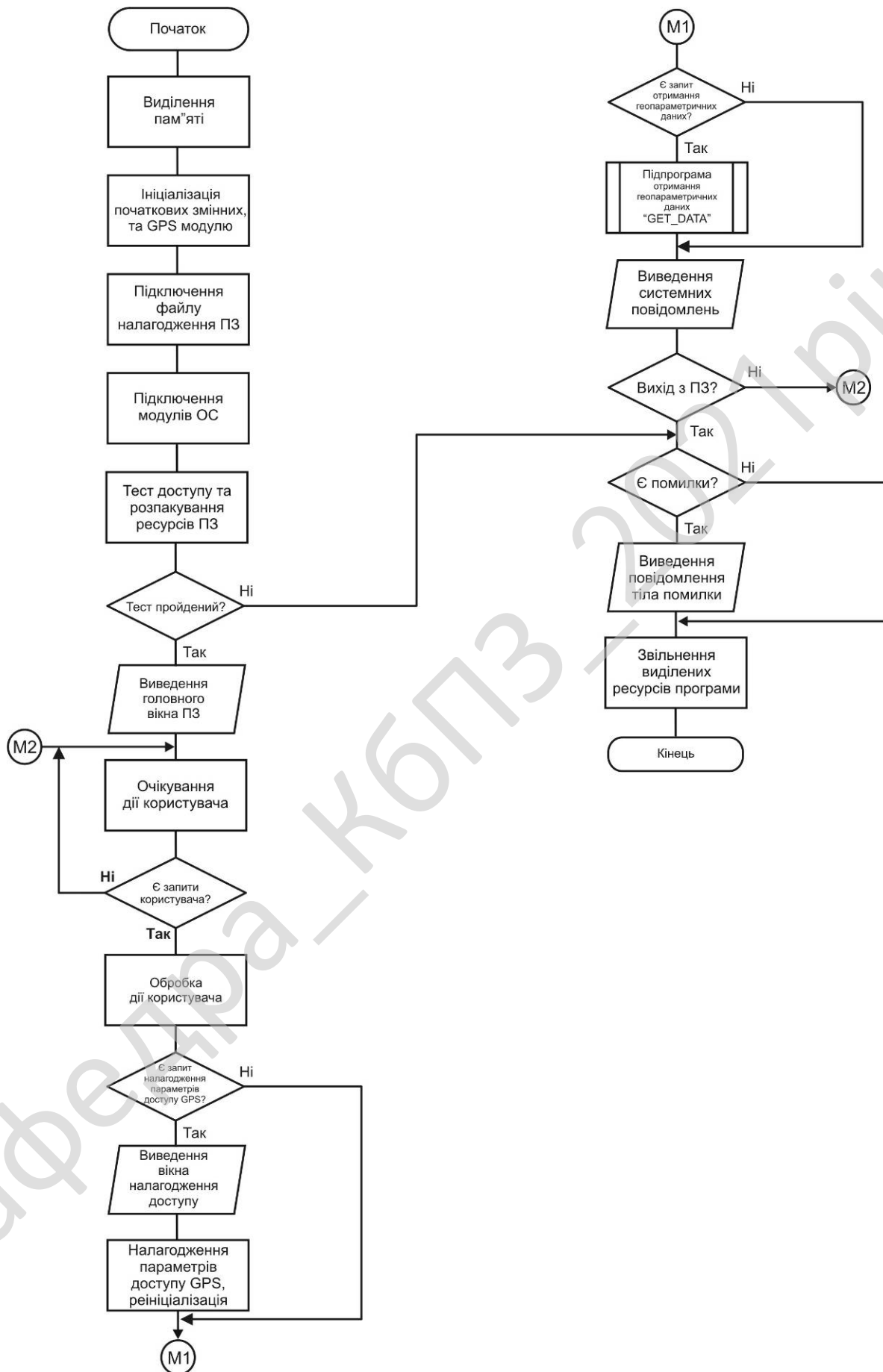


Рисунок 4.1 – Блок-схема програми

|      |      |          |        |      |
|------|------|----------|--------|------|
| Вим. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

ВКРМ-123.21.0002.00.00.ПЗ

Арк.

46

Підпрограма отримання геопараметричних даних "GET\_DATA"

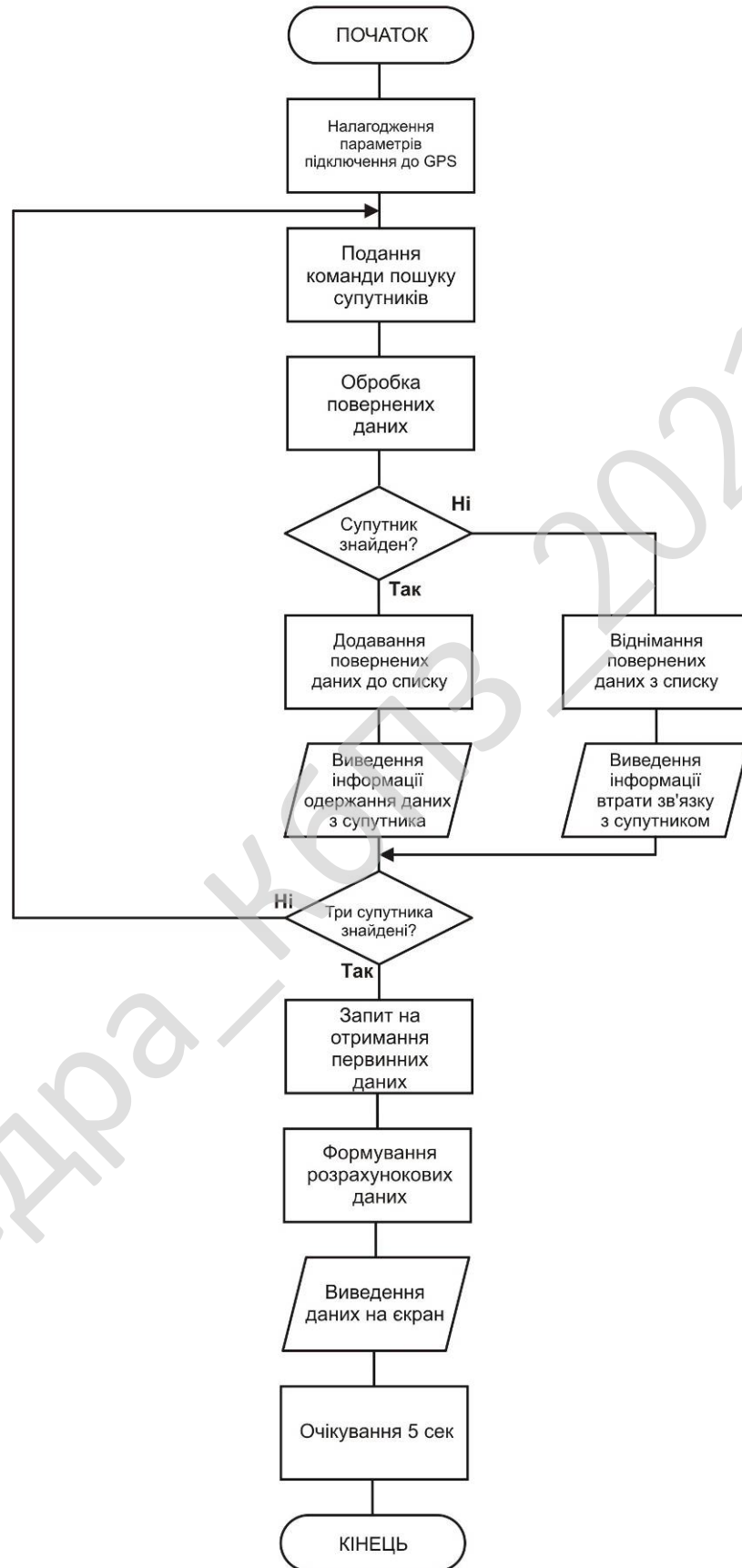


Рисунок 4.2 – Блок схема підпрограми

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Вим. | Арк. | № докум. | Підпис | Дата |

ВКРМ-123.21.0002.00.00.ПЗ

Арк.

47

## Оптимізація ресурсів розробленої програми

При розробці програми під мобільний пристрій автор стикнувся із проблемою недовліку місця для коректної роботи програми. Для подолання даних утруднень був розроблений модуль стиску використовуваних даних і розпакування необхідних даних.

Для того, щоб архівувати папку, необхідно спочатку склеювати всі файли, що перебувають у деякій папці. Для цього був розроблений наступний формат:

- 1-4 байт сигнатура, що позначає, що це файл необхідного формату;
- 5-8 байт кількість файлів у склеєному файлі;
- 8-N байт масив структур, описуючий всі файли в склеєному файлі, з

наступною структурою: 2 байти довжина ім'я файлу, 4 байти – розмір файлу. У даній структурі максимальний розмір одного файлу може бути максимум  $2^{32}$  байт.

Розглянемо функцію одержання списку всіх файлів у папці, що передає шлях до папки у форматі TStringList.

```
function GetAllFiles (Filter, Folder: string):TFilesList;
var
  sr: TSearchRec;
  sDirList, _FilesList, _LST: TStringList;
  i, j: Integer;
begin
  j:=0;
  _FilesList:= TStringList.create;
  _FilesList.Clear;
```

На початку створюється об'єкт результат, потім відбувається пошук файлів, які перебувають у папці, шлях до якої був переданий як другий параметр.

Перші два знайдених файли ігноруються тому, що це папки й при переході на які відбувається перехід на один рівень нагору.

```
if FindFirst (Folder + Filter, faAnyFile , sr) = 0 then
  repeat
    j:=j+1;
    if j 0;
    FindClose(sr);
```

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Вим. | Арк. | № докум. | Підпис | Дата |

ВКРМ-123.21.0002.00.00.ПЗ

Арк.

48

Створюємо об'єкт, у якому зберігатися список підпапок, і викликаємо для кожної підпапки саму себе. Після одержання списку файлів у кожній підпапки відбувається копіювання отриманого списку в результат.

```
sDirList := TStringList.Create;
  try
    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
    sDirList.Free;
  end;
  Result:= _FilesList;
end;
```

Процедура одержання списку розширеної інформації підпапок. Аркуш із результатом вертається як результат.

```
procedure GetSubDirs (Folder: string; sList: TStringList);
  var
    sr: TSearchRec;
  begin
    if FindFirst (Folder + '.*', faDirectory, sr) = 0 then
      try
        repeat
          if (sr.Attr and faDirectory) = faDirectory then
            sList.Add (sr.Name);
          until FindNext(sr) 0;
        finally
          FindClose(sr);
        end;
      end;
```

Розглянемо розроблену функцію, що буде передавати параметри функції склеювання файлів (функція зворотного виклику).

```
...
type
```

|      |      |          |        |      |                           |  |  |  |      |
|------|------|----------|--------|------|---------------------------|--|--|--|------|
|      |      |          |        |      |                           |  |  |  | Арк. |
|      |      |          |        |      |                           |  |  |  | 49   |
| Вим. | Арк. | № докум. | Підпис | Дата | ВКРМ-123.21.0002.00.00.ПЗ |  |  |  |      |

```

TActionFuntion
function (SourceFileName, DestFileName:string):boolean;
....
function DoFolderAction(FolderPath, ArchivePath:string;
    ActionFunction :TActionFuntion):boolean;
var
    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);
    {сигнатура}
    Value:=_Files.Count;
    WriteFile(ArchiveFile, value, 4, _writed, 0);
    { количество файлов}
    for i:=0 to value-1 do
        begin
            Виклик функції зворотного виклику, який передаємо файл джерело й файл
            результат, файл-результат це всього лише проміжний файл, потім ми його
            видалимо. Змінна bkpFile оголошена як константа.
            if not ActionFunction(_Files.Strings[i], bkpFile) then
                begin
                    { якщо функція зворотного виклику завершилася невдало, виходимо з функції }
                    CloseHandle(ArchiveFile);
                    exit;
                end;
            CurrFile := CreateFile(pchar(bkpFile), GENERIC_READ,
            FILE_SHARE_READ, 0, OPEN_EXISTING, 0, 0);
            CurrFileSize := GetFileSize(CurrFile, nil);
            { одержуємо "шлях файлу в папці"}

```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 50   |

```

path_in_archive_file :=copy(Files.Strings[i],
                                length(FolderPath),1000);
if path_in_archive_file[1] = '\' then
Delete(path_in_archive_file,1,1);
Write_LS(ArchiveFile,path_in_archive_file);
{имя файла}
{Функція Write_LS пише у файл стоку у форматі LS}
WriteFile(ArchiveFile,currFileSize,4,_writed,0);
{розмір файлу}
{місце під буфер}
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;
    {визволення буфера}
VirtualFree(pBuff,MEM_RELEASE,0);
CloseHandle(CurrFile);
DeleteFile(bkpFile);
end;
CloseHandle(ArchiveFile);
Result:=True;
end;

```

### Розглянемо функцію, розпакування файлів:

```

function De_DoFolderAction(FolderPath,ArchivePath:string;
    DeActionFunction :TActionFuntion):boolean;
var
...
begin
{перевірка формату}
ArchiveFile:=CreateFile(pchar(ArchivePath),GENERIC_READ+GEN
    ERIC_WRITE,FILE_SHARE_READ,0,OPEN_EXISTING,0,0);
ReadFile(ArchiveFile,value,4,_readed,0);
if Value ArchiveSignature then
begin
...
end;
{кількість файлів}
ReadFile(ArchiveFile,_Count,4,_readed,0);
for i:=1 to _Count do

```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 51   |

```

begin
  path_in_archive_file := Read_LS(ArchiveFile);
  ReadFile(ArchiveFile,CurrFileSize,4,_readed,0);
  ...
  for j:=1 to CurrFileSize div MainBufferSize do
  begin
  ReadFile(ArchiveFile,pBuff^,MainBufferSize,_readed,0);
  WriteFile(CurrFile,pBuff^,_readed,_writed,0);
  end;

  ReadFile(ArchiveFile,pBuff^,CurrFileSize mod MainBufferSize,_readed,0);
  WriteFile(CurrFile,pBuff^,_readed,_writed,0);
  ...
  if FolderPath[Length(FolderPath)]'\ ' then
    Delete(FolderPath,Length(FolderPath),1);
if path_in_archive_file[1]'\ ' then
  path_in_archive_file:= '\'+path_in_archive_file;
  path_in_archive_file:=FolderPath+path_in_archive_file;
  mkdir(ExtractFileDir(path_in_archive_file));
if not DeActionFunction(bkpfiler,path_in_archive_file) then
  begin
  ...
  end;
  ...
  end;
  ...
  end;

```

#### Використовуваний основний запуск склейки файлів:

```

function DODO(f11,f12:string):boolean;
begin
  Result:=CopyFile(pchar(f11),pchar(f12),false);
end;

```

#### 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ММВ, в основі якого лежить змішування операцій різних алгебраїчних груп. ММВ – ітеративний алгоритм, що складається з лінійних дій (XOR і використання ключа) і паралельного застосування чотирьох

|      |      |          |        |      |  |                                  |      |
|------|------|----------|--------|------|--|----------------------------------|------|
|      |      |          |        |      |  | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |  |                                  | 52   |

великих оборотних нелінійних підстановок. Ці підстановки визначаються за допомогою множення по модулю  $2^{32}-1$  з постійними множниками. У підсумку з'являється алгоритм, що використовує 128-бітовий ключ і 128-бітовий блок.

Алгоритм ММВ оперує 32-бітовими підблоками тексту  $(x_0, x_1, x_2, x_3)$  і 32-бітовими підблоками ключу  $(k_0, k_1, k_2, k_3)$ . Це спрощує реалізацію алгоритму на сучасних 64-бітових процесорах. Чергуючись із операцією XOR, шість разів використовується нелінійна функція  $f$ . Запишемо операції алгоритму (всі операції з індексами виконуються по модулю 4):

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

Функція  $f$  виконується в три кроки:

1.  $x_i = c_i * x_i$  для  $i = 0..3$  (Якщо на вході множення одні одиниці, то на виході – теж одні одиниці).

2. Якщо молодший значущий біт  $x_0 = 1$ , то  $x_0 = x_0 \oplus C$ . Якщо молодший значущий байт  $x_3 = 0$ , то  $x_3 = x_3 \oplus C$ .

3.  $x_i = x_{i-1} \oplus x_i \oplus x_{i+1}$  для  $i = 0..3$ .

Всі операції з індексами виконуються по модулю 4. Операція множення на кроці 1 виконується по модулі  $2^{32}-1$ . Спеціальний випадок для даного алгоритму:

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 53   |

якщо другий операнд дорівнює  $2^{32}-1$ , результат теж дорівнює  $2^{32}-1$ . В алгоритмі використовуються наступні константи:

$$C = 2\text{aaaaaaaa}, c_0 = 025f1cdb, c_1 = 2 * c_0, c_2 = 2^3 * c_0, c_3 = 2^7 * c_0.$$

Константа  $C$  – «найпростіша» константа без кругової симетрії, високою трійковою вагою й нульовим молодшим значущим бітом. У константи  $c_0$  є інші особливі характеристики. Константи  $c_1$ ,  $c_2$  і  $c_3$  – зрушені версії  $c_0$ , і служать для запобігання атак, заснованих на симетрії.

Розшифрування виконується у зворотному порядку, Етапи 2 і 3 інверсні їм самим. На етапі 1 замість  $c_i$  використовується  $c_i^{-1}$ . Значення  $c_0^{-1} = 0dad4694$ .

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 54   |

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Після написання програмного продукту його потрібно впровадити у експлуатацію. При цьому виправляються помилки, які були помічені, система настраюється на відповідний режим роботи.

Для налаштування системи на оптимальні умови роботи необхідно виконати наступні кроки:

- перевірити стан оперативної пам'яті в мобільної ОС;
- перевірити кількість вільного дискового простору для правильної роботи програми;
- перевірити стан GSM приймача;
- провести інсталяцію розробленого програмного забезпечення на мобільний пристрій;

Після запуску програми на екран мобільної платформи виводиться основне вікно програми (рисунок 5.1).

У початковому положенні програма не виводить геопараметричних даних. З основного вікна програми можна перейти в додаткове вікно налаштування програми, перехід на форму авторських даних, вимикання програми й включення визначення геопараметричних даних.

Після натискання на кнопку визначення положення й при правильному налаштуванні мобільної платформи відбувається запит даних з GPS приймача мобільної платформи. Для визначення координат необхідно як мінімум зв'язок із трьома супутниками, при меншій кількості супутників визначити геопараметричне місце розташування неможливо. На рисунку 5.2 показаний процес пошуку супутників.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 55   |



Рисунок 5.1 – Головне вікно програми, стан очікування дії користувача

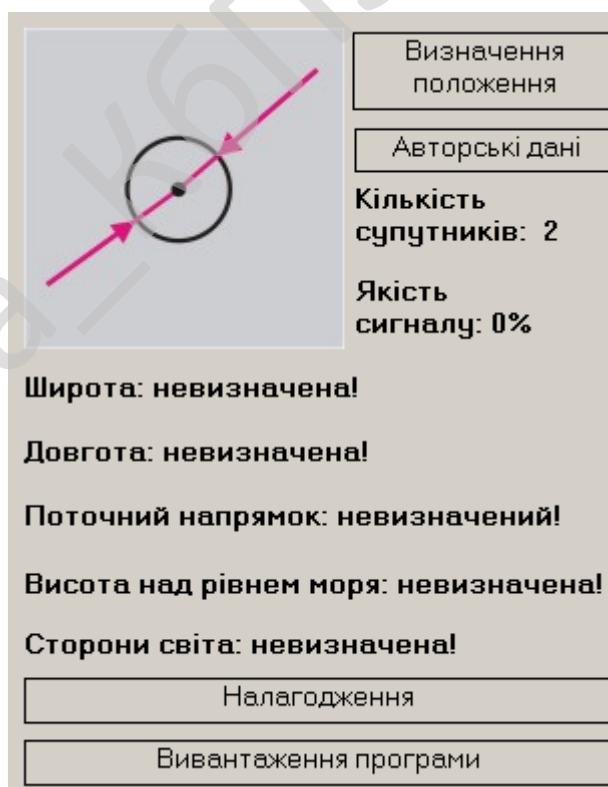


Рисунок 5.2 – Головне вікно програми, стан пошук супутників

При знаходженні більше трьох штучних супутників землі, програма формує й виводить на екран поточну первинну інформацію про геопараметричне положення користувача з наступними параметрами:

- поточна широта користувача із вказівкою геопараметричних мінут і секунд;
- поточна довгота користувача із вказівкою геопараметричних мінут і секунд;
- поточний напрямок руху користувача, виводиться винятково при русі;
- висота над рівнем моря (у метрах) величина визначальна на відносне положення;
- поточна сторона світла, грає роль електронного компаса. Показує сторону світла або комбінацію сторін світла – південь, південний захід і.т.ін.

Приклад вказівки первинних даних показаний на рисунку 5.3.



Рисунок 5.3 – Головне вікно програми, стан виведення геопараметричних даних користувача

Для дотримання авторських прав у розробленій програмі створена форма авторського права зображена на рисунку 5.4. На ній вказані дані про вищий навчальний заклад, факультет, тему, автора розробки й керівника дипломного проектування.

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

Факультет механіко-технологічний

**Дослідження та програмна  
реалізація системи обробки  
навігаційних даних**

**Розробив - Гирба О.М.**  
КІ-20М-1,4

**Керівник - Олексій СМІРНОВ**

Приховати

Рисунок 5.4 – Форма авторського права

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 58   |

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи обробки навігаційних даних.

*Метою розробки є дослідження та програмна реалізація системи обробки навігаційних даних.*

*Об'єктом дослідження є процес обробки навігаційних даних.*

*Предметом дослідження є методи обробки навігаційних даних.*

*Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод обробки навігаційних даних.
- Розроблено вітчизняний продукт обробки навігаційних даних, який має більш широкі можливості, на відміну від існуючих аналогів.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 59   |

## 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 були проведені дослідження та виконана програмна реалізація системи обробки навігаційних даних.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

| Показники                                    | Позначення | Характеристика або величина               |
|----------------------------------------------|------------|-------------------------------------------|
| 1                                            | 2          | 3                                         |
| 1. Кількість розроблених програм період, шт. | N          | 1                                         |
| 2. Кількість екземплярів програм, шт.        | Ne         | 20 (2 ост. цифри № зал* 10 <sup>1</sup> ) |
| 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)                                          | – | 1 |
| 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 |

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Вим. | Арк. | № докум. | Підпис | Дата |

ВКРМ-123.21.0002.00.00.ПЗ

Арк.

61

Продовження таблиці 7.1

| 1                                                                   | 2   | 3                                          |
|---------------------------------------------------------------------|-----|--------------------------------------------|
| 28. Досвід роботи з мовою і інструментами середовища розробки (1-6) | –   | 2                                          |
| 29. Досвід роботи з програмними інструментами розробки (1-6)        | –   | 3                                          |
| 30. Розробка ПЗ для декількох серверів одночасно (1-6)              | –   | 2                                          |
| 31. Вимоги до дотримання встановленого графіка робіт (1-6)          | –   | 2                                          |
| 32. Вартість ПЗ у розробника (НМА), грн.                            | –   | 20000 (2 ост. цифр № зал*10 <sup>4</sup> ) |
| 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 як одна з найбільш тривалих і трудомістких робіт значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 62   |

де:  $A$  – коефіцієнт Боєма,  $A = 2,45$ ;  $Size$  – 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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026.$$

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

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

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

де:  $\prod V_j$  – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

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

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкість 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:

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

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)%.

$$T_{РП} = 0,3 \cdot 3,23 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 82 = 168 \text{ люд/день.}$$

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 63   |

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

| Стадії розробки   | Трудомісткість за типовими нормами та розрахунками |           |
|-------------------|----------------------------------------------------|-----------|
|                   | Величина, люд/дні                                  | Підстава  |
| Технічне завдання | 9                                                  | Д5        |
| Ескізний проект   | 10                                                 | Д6        |
| Технічний проект  | 9                                                  | Д7        |
| Робочий проект    | 168                                                | Ф 7.1-7.4 |
| Впровадження      | 13                                                 | Д13       |
| Всього            | 209                                                | –         |

### 7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

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

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

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

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

| Найменування обладнання              | Профілактичне обслуговування |                      |                    |                     |
|--------------------------------------|------------------------------|----------------------|--------------------|---------------------|
|                                      | Кількість хв. на один. обл.  | Кількість обладнання | Затрати часу в хв. | Затрати часу в год. |
| Системний блок ПК                    | 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 <sub>ч</sub>     | 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:

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

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

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 (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi | 2   | 0,5                  |
|                                          | Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)                                                                                                                                                 | 0,5 |                      |
|                                          | Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ                                                                                                                                     | 0,5 |                      |
|                                          | Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет                                                                                                                                                        | 1   |                      |
| Всього                                   |                                                                                                                                                                                                                            | 4   |                      |

Продовження таблиці 7.4

| Посада              | Вид роботи                                                                                                          | Час  | К-ть штатних одиниць |
|---------------------|---------------------------------------------------------------------------------------------------------------------|------|----------------------|
| Продакт-менеджер    | Презентації нової продукції, пошук каналів збуту                                                                    | 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    |                      |

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Вим. | Арк. | № докум. | Підпис | Дата |

ВКРМ-123.21.0002.00.00.ПЗ

Арк.

67

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                | 8000                         | 24000                           |
| Продакт-менеджер          | 0,25             | 6000                         | 4500                            |
| Інженер-програміст        | 3,8              | 6000                         | 68400                           |
| Інженер-електронщик       | 1,2              | 6000                         | 21600                           |
| Інженер-системотехнік     | 0,25             | 6000                         | 4500                            |
| Адміністратор мережі      | 0,5              | 6000                         | 9000                            |
| Системний програміст      | 0,25             | 7000                         | 5250                            |
| Дизайнер WEB              | 0,25             | 6000                         | 4500                            |
| Інженер-верстальник       | 0,25             | 5000                         | 3750                            |
| Бухгалтер-економіст       | 0,5              | 7000                         | 10500                           |
| Всього за період розробки | $R_{cn} = 8,25$  | -                            | $\Phi_{роб} = 156000$           |

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.

$$z_{cd} = \frac{156000}{8,25 \cdot 60} = 315 \text{ грн.}$$

#### 7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 68   |

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

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

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно 8  $m^2$ . З урахуванням цього:

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

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

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет-магазину Компбест за 09.11.21 – джерело <https://compbest.com.ua>.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 69   |

Таблиця 7.6 – Специфікація

| Найменування комплектуючої або обладнання | Тип                                                                                                                                                                                                                                                   | Оптова ціна |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| Персональний комп'ютер                    |                                                                                                                                                                                                                                                       | 10947       |
| Системний блок                            | Hyundai iTMC Pentino Tower                                                                                                                                                                                                                            | 7347        |
| Процесор                                  | Intel Core i5-2400 (4 ядра по 3.1 - 3.4 GHz)<br>Cache Memory 6 MB                                                                                                                                                                                     | -           |
| Системна плата                            | Материнська плата MSI B75MA-P45 (s1155)<br>Intel B75, PCI-Ex16, 2x USB 3.0, 6x USB 2.0<br>5x Audio Ports, LAN (RJ -45), 2x PS/2; порт<br>дискретної графіки - DVI, DisplayPort                                                                        | -           |
| Відеокарта                                | ATI Radeon HD 8490, 1 GB GDDR3, 64-bit<br>Integrated Intel HD Graphics 2000                                                                                                                                                                           | -           |
| Жорсткий диск                             | 120 GB SSD, 500 GB HDD                                                                                                                                                                                                                                | -           |
| Оперативна пам'ять                        | 16 GB DDR3                                                                                                                                                                                                                                            | -           |
| DVD-привод                                | DVD -RW/+RW , LG SATA SuperMulti Bul<br>22x, SecurDisc, black                                                                                                                                                                                         | -           |
| Корпус                                    | ATX Pentino Tower, 3GTLA-489, PSU<br>350W(FSP Brand: ATX-350PNR, 12cm)<br>black, (front bezel – black+light silver; bod<br>material – 0.6mm), 80mm fan (rear<br>2xUSB2.0/AUDIO/MIC, Air Duct, Tool-les<br>chassis design,Thermally Advantaged Chassis | -           |
| Кардрідер внутрішній                      | USB 2.0 Card reader STORM CR-35U1A4-E<br>int. 3.5", 1*USB2.0+AUDIO+1394, multi: All<br>Type Cards, black                                                                                                                                              | -           |

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>БКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 70   |

Продовження таблиці 7.6

| Найменування комплектуючої або обладнання | Тип                                                         | Оптова ціна |
|-------------------------------------------|-------------------------------------------------------------|-------------|
| інше                                      | Клавіатура, мишка, система охолодження процесора            | Подарунк    |
| Монітор                                   | 22" TFT, ASUS VW223D ( 5ms, 300/3000: 170/160, D-SUB, Wide) | 3600        |
| Принтер лазерний                          | Canon i-SENSYS LBP6030W                                     | 2700        |
| Принтер струминний                        | Epson Stylus Photo P50 (C11CA45341) + USB cable             | 5500        |
| Копіювальний апарат                       | Canon i-SENSYS MF217W with Wi-Fi                            | 5965        |

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

| Найменування обчислювальної техніки | Кількість, шт. | Ціна за одиницю, грн. | Витрати на транспортування, монтаж та випробування. | Загальна вартість, грн. |
|-------------------------------------|----------------|-----------------------|-----------------------------------------------------|-------------------------|
| Персональні комп'ютери              | 12             | 10947                 | 13136,4                                             | 144500,4                |
| Принтер лаз.                        | 2              | 2700                  | 540                                                 | 5940                    |
| Принтер струм.                      | 1              | 5500                  | 550                                                 | 6050                    |
| Копіюв. апарат                      | 1              | 5965                  | 596,5                                               | 6561,5                  |
| Всього                              | —              | —                     | —                                                   | 163051,9                |

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      | 163051,9                 | -           | -                  |
| Всього по групі               | 163051,9                 | 50          | 81525,95           |
| Група 5, 6                    |                          |             |                    |
| 4. Вимірювальні пристрої      | 9031                     | 25          | 2257,75            |
| 5. Транспортні засоби         | 143000                   | 20          | 28600              |
| 6. Господарський інвентар     | 28000                    | 25          | 7000               |
| Всього по групі 5,6           | 180031                   | -           | 37857,75           |
| Нематеріальні активи          |                          |             |                    |
| 7. Нематеріальні активи       | 20000                    | 10          | 2000               |
| Разом                         | $K_p = 1771082,9$        |             | $A_p = 189783,7$   |

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



Згідно виданих викладачем норм приймаємо одну пачку паперу на три місяці розробки. Тоді, враховуючи, що вартість пачки паперу складає  $C_n = 105$  грн., визначаємо вартість паперу за період розробки  $N_m = 3$  міс:

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

$$Z_{M1} = 105 \cdot 1 = 105 \text{ грн.}$$

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

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

де:  $C_d$  – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 2 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 3 грн./шт.

$$Z_{M2} = 20 \cdot 2 + 3 = 43 \text{ грн.}$$

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

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

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

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

$$Z_M = (105 + 43 + 1702) / 20 = 93 \text{ грн.}$$

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

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

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

$$O_n = 3292 \cdot 15 \cdot 0,01 = 494 \text{ грн.}$$

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>БКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 74   |

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

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

$$A_m = 189783,7 \cdot 3 / (20 \cdot 12) = 2372 \text{ грн.}$$

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

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

$$C_n = 3292 + 392 + 797 + 494 + 93 + 494 + 2372 = 7934 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де:  $P_n$  – рівень рентабельності, %.

$$\Pi_p = 0,01 \cdot 50 \cdot 7934 = 3967 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

| Найменування статей витрат                              | Позначення     | Величина, грн |
|---------------------------------------------------------|----------------|---------------|
| 1                                                       | 2              | 3             |
| 1. Основна зарплата виконавців                          | $Z_o$          | 3292          |
| 2. Додаткова зарплата виконавців                        | $Z_d$          | 392           |
| 3. Відрахування на соціальні потреби                    | $C_{oc}$       | 797           |
| 4. Загальногосподарські витрати                         | $\Gamma_{ocn}$ | 494           |
| 5. Витрати на матеріали                                 | $Z_m$          | 93            |
| 6. Освоєння нових операційних систем, мов програмування | $O_n$          | 494           |

Продовження таблиці 7.9

| 1                                                                     | 2     | 3       |
|-----------------------------------------------------------------------|-------|---------|
| 7. Амортизація основних фондів                                        | $A_m$ | 2372    |
| 8. Повна собівартість програмного забезпечення                        | $C_n$ | 7934    |
| 9. Плановий прибуток                                                  | $P_p$ | 3967    |
| 10. Ціна підприємства $C_n = C_n + P_p$                               | $C_n$ | 11901   |
| 11. Податок на додану вартість<br>$ПДВ = 0.01 \cdot H_{об} \cdot C_n$ | $ПДВ$ | 2380,2  |
| 12. Відпускна ціна програмної продукції $C = C_n + ПДВ$               | $C$   | 14281,2 |

### 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     | –                        | 14281 |
| Всього капітальних витрат         | –                        | 14281 |



$$Z_{ел баз} = 0,45 \cdot 2400 \cdot 2,1 = 2268 \text{ грн.}$$

$$Z_{ел нов} = 0,40 \cdot 2400 \cdot 2,1 = 2016 \text{ грн.}$$

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                  | –                                       | 14281 | –                                   | 3570,25 |
| Всього відрахувань    | -                   | –                                       | 14281 | –                                   | 3570,25 |

### 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.

$$E_e = (11901 - 7934) \cdot 20 - (0,05 \cdot 1408000 + 0,5 \cdot 163051,9 + 0,2 \cdot 143000 + 0,25 \cdot 37031 + 0,1 \cdot 20000) \cdot 3/12 = 31394 \text{ грн.}$$

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 без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_o = \frac{363083}{(11901 - 7934) \cdot 20 \cdot 12 / 3} = 1,1 \text{ року.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

| Найменування показників                                                                 | Одиниця виміру | Величина |
|-----------------------------------------------------------------------------------------|----------------|----------|
| 1. Кількість екземплярів програми                                                       | Прим.          | 20       |
| 2. Повна собівартість розробленої програми                                              | Грн.           | 7934     |
| 3. Ціна розробленої програми                                                            | Грн.           | 11901    |
| 4. Плановий прибуток від реалізації розробленої програми                                | Грн.           | 3967     |
| 5. Рентабельність програмної продукції                                                  | %              | 50       |
| 6. Об'єм додаткових капітальних вкладень у виробника програмної продукції               | Грн.           | 1771083  |
| 7. Загальний прибуток від реалізації програмної продукції                               | Грн.           | 79340    |
| 8. Величина економічного ефекту при виготовлені програмної продукції                    | Грн.           | 31394    |
| 9. Період окупності додаткових капітальних вкладень у виробника програмної продукції    | Років          | 1,1      |
| 10. Об'єм додаткових капітальних вкладень у споживача програмної продукції              | Грн.           | 14281    |
| 11. Величина економічного ефекту у користувача програмної продукції                     | Грн.           | 535,8    |
| 12. Період окупності додаткових капітальних вкладень у користувача програмної продукції | Років          | 4        |

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

$$E_{cn} = (15688 - 12296) - 0,2 \cdot 14281 = 535,8 \text{ грн.}$$

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

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

$$T_{cn} = \frac{14281}{15688 - 12296} = 4 \text{ роки.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

## 7.9 Висновки

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

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 80   |

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя [3]. Охорона праці є складовою частиною безпеки життя.

Законом України “Про охорону праці” [2] 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» [4], 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 [1].

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

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 81   |

терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [1], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

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

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

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

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

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці;

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 82   |

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

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

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

Таблиця 8.1 – Розміри приміщення

| Найменування | Значення, м |
|--------------|-------------|
| Ширина       | 5           |
| Довжина      | 6           |
| Висота       | 2,7         |

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

| Геометрична характеристика | Одиниця виміру | Нормативне значення* | Фактичне значення |
|----------------------------|----------------|----------------------|-------------------|
| Площа, S                   | м <sup>2</sup> | не менше 6.0         | 7,5               |
| Обсяг, V                   | м <sup>3</sup> | не менше 20.0        | 20,25             |

\* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин) [1].

У зазначеному приміщенні працює 4 особи. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста 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», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [1] та НПАОП 0.00 -1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

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

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

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 84   |



освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

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

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

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

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

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: наочне знайомство персоналу з шляхами 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'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 86   |



$h$  – розрахункова висота підвісу,  $h = 2,7$  м (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

$A$  – ширина приміщення,  $A = 5$  м;

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

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$i=1,01.$$

Знаючи індекс приміщення ( $i$ ), за знаходимо  $n = 0,44$  (з табличних даних коефіцієнтів використання світлового потоку ( $n$ ) світильників відповідного типу [6]). Підставимо всі значення у формулу, визначимо світловий потік:  $F=33750$  Лм.

Будемо використовувати світильники “LED Евросвет ОПАЛ-40 6400К 3000Лм 36Вт.”, світловий потік яких  $F_n = 3000$  Лм.

Число світильників визначається по формулі:

$$N=F/F_n$$

де:

$F$  – світловий потік,

$F_n$  – світловий потік одного світильника.

Підставимо всі значення у формулу та визначимо індекса приміщення:  $N=$

$$33750/3000=11,25 \text{ шт.}$$

Приймаємо необхідну кількість світильників 12 шт.

## 8.6 Висновки до розділу

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

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

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 88   |

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи обробки навігаційних даних.

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 обробки навігаційних даних.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем обробки навігаційних даних.
- Досліджена система обробки навігаційних даних.
- На основі отриманих результатів досліджень створена програмна реалізація системи обробки навігаційних даних.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання обробки навігаційних даних.

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

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

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 89   |

Програма реалізована на мові високого рівня RAD Studio Delphi. 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 XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ММВ.

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 535,8 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 4 роки.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 90   |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гирба О.М. Дослідження та програмна реалізація системи обробки навігаційних даних // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.
2. Пер Энге GPS добрался до метра// В МИРЕ НАУКИ № 6 2007 с35-43
3. GPS: Все, что Вы хотели знать, но боялись спросить. Неофициальное пособие по глобальной системе местоопределения, М.: «Бук Пресс» -2006. 352 с.
4. <http://offline.computerra.ru/2004/569/36864/>
5. <ftp://igscb.jpl.nasa.gov/pub/center>
6. <ftp://igscb.jpl.nasa.gov/pub/general>
7. Коваленко А.С. Разработка структуры базы данных интегрированной информационной системы / А.С. Коваленко, А.В. Коваленко // Информационные технологии и защита информации в информационно-коммуникационных системах: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – С. 54-64.
8. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.
9. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.
10. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 91   |

11. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

12. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

13. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

14. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смирнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

15. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смирнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.

16. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смирнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.

17. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смирнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 92   |



24. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

25. Коваленко А.С. Задачи распознавания ситуаций в системах организационной стратегии интеграции производства и операций / А.С. Коваленко, А.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVI міжнар. наук.-практ. сем., 11-12 квіт. 2014 р., м. Кіровоград: зб. тез. – Кіровоград: КНТУ, 2014. – С. 53-55.

26. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.

27. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28 -31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.

28. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.

29. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 94   |

30. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

31. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у промисловості і освіті: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

32. Коваленко А.С. Обґрунтування необхідності створення розподіленої бази даних для забезпечення захисту рухомих повітряних об'єктів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Перспективні напрями захисту інформації: I всеукр. наук.-практ. конф., 07 вер. 2015 р., м. Одеса: зб. тез доп. – Одеса: ОНАЗ, 2015. – С. 35-39.

33. Коваленко А.С. Розробка інформаційної моделі автоматизованої оцінки технічного стану інтегральної інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформаційні технології та взаємодії (ІТ & І): II між нар. наук.-практ. конф., 3-5 лист. 2015 р., м. Київ: тези доп. – Київ: КНУ ім. Т. Шевченка, 2015. – С. 41-42.

34. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Информационные и телекоммуникационные технологии: образование, наука, практика: II междунар. научн.-практ. конф., 3-4 дек. 2015 г., г. Алматы, Казахстан: сб. труд. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – Т.2. – С. 423-427.

35. Королюк Н.А. Оценка временных интервалов работы лица, принимающего решение, на автоматизированном командном пункте / Н.А. Королюк, А.И. Тимочко // Системи обробки інформації. – Х.: ХУПС, 2005. – Вип. 8 (48). – С. 51-54.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-123.21.0002.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 95   |

36. Костерев В.В. Надёжность технических систем и управление риском: учебн. пособ. / В.В. Костерев. – М.: МИФИ, 2008. – 280 с.
37. Костюков А.В. Підвищення операційної ефективності підприємств на основі моніторингу в реальному часі. / А.В. Костюков, В.М. Костюков. – М.: Машинобудування, 2009. – 192 с.
38. Лазарев А.А. Выбор показателя затрат для анализа сравнительной экономической эффективности техники конечного потребления / А.А. Лазарев, М.В. Бейлин // Сборник научных трудов ХГПУ.– Х.: ХГПУ, 1999. – Вып. 74. – С. 27-29.
39. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 1. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 308 с.
40. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 2. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 208 с.
41. Лапсарь А.П. Метод оценки состояния сложных технических объектов для синтеза быстродействующих прогнозирующих систем / А.П. Лапсарь // Измерительная техника. – 2004. – № 2. – С. 7-10.
42. Линейные задачи оптимизации: Учеб. пособие / С.В. Лутманов. – Пермь: ЛИТЕР-А, 2004. – Ч.1. – Линейное программирование. – 128 с.
43. Литвак Б.Г. Экспертные технологии в управлении. Учебное пособие / Б.Г. Литвак. – М.: Дело, 2014. – 318 с.
44. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: Посібн. / В.М. Локазюк, Ю.Г. Савченко. – К.: Видавничий центр «Академія», 2004. – 376 с.
45. Лопатников Л. И. Экономико-математический словарь: Словарь современной экономической науки / Л.И. Лопатников. – М.: Дело, 2003. – 520 с.
46. Манухина С.Ю. Инженерная психология и эргономика: хрестоматия / С.Ю Манухина. – М.: Изд. центр ЕАОИ, 2009. –224 с.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-123.21.0002.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 96   |

47. Мартыненко М.В. Человекомашинные процедуры поддержки организационно–управленческих решений: учеб. пособие СПбГЭТУ / М.В. Мартыненко, О.И. Шеховцов. – СПб, 2012. – 250 с.

48. Мунипов О.В. Эргономика: человекоориентированное проектирование техники, программных средств и среды: Учебник / О.В. Мунипов, В.П. Зинченко. – М.: Логос, 2001. – 356 с.

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

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

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

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

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

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

|           |              |             |        |      |                                  |       |         |
|-----------|--------------|-------------|--------|------|----------------------------------|-------|---------|
|           |              |             |        |      | <b>ВКРМ-123.21.0002.00.00.ТЗ</b> |       |         |
| Вим.      | Арк.         | № документа | Підпис | Дата |                                  |       |         |
| Розробив  | Гирба О.М.   |             |        |      | Літ.                             | Аркуш | Аркушів |
| Перевірів | Смірнов О.А. |             |        | М    |                                  |       |         |
| Н. Контр. | Гермак В.С.  |             |        |      | ЦНТУ КІ-20М-1,4                  |       |         |
| Затв.     | Смірнов О.А. |             |        |      |                                  |       |         |

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи обробки навігаційних даних.

## 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. № 42-13 від 02.08.2021 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи обробки навігаційних даних.

## 4 Джерела розробки

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

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

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

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-123.21.0002.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 2    |

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

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи обробки навігаційних даних;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

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

## 5.4 Вимоги до архітектури

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

## 5.5 Вимоги до надійності

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

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРМ-123.21.0002.00.00.ТЗ</b> | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                                  | 3    |

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 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 XP/Vista/7/8/10.

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище RAD Studio Delphi.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-123.21.0002.00.00.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 2021 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута розробка заходів з умов поліпшення охорони праці.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-123.21.0002.00.00.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 – 97 аркушів.

## 10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2021 р.

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРМ-123.21.0002.00.00.ТЗ</b> | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                                  | 6    |

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

ЗАТВЕРДЖУЮ

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

\_\_\_\_\_ Смірнов О.А.

*Дослідження та програмна реалізація  
системи обробки навігаційних даних*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 42

Літера: РП

Кропивницький – 2021 року

## Головний файл програми

```

program GPS_NAVIGATION;
{Hyrba O.M.KI-20M1,4}
uses
  Forms,
  GPS_About in 'GPS_About.pas' {F2},
  GPS_MainForm in 'GPS_MainForm.pas' {F1},
  GPS in 'GPS.pas';
{$R *.RES}

procedure GetSubDirs (Folder: string; sList: TStringList);
var
  sr: TSearchRec;
begin
  if FindFirst (Folder + '.*', faDirectory, sr) = 0 then
  try
    repeat
      if (sr.Attr and faDirectory) = faDirectory then
        sList.Add (sr.Name);
      until FindNext(sr) 0;
    finally
      FindClose(sr);
    end;
  end;
end;

function GetAllFiles (Filter, Folder: string):TFilesList;
var
  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
      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
  sDirList.Free;
end;
Result:= _FilesList;
end;

function DoFolderAction(FolderPath, ArchivePath:string;
  ActionFunction :TActionFuntion):boolean;
var
  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);
    pBuffer:=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;

begin
    Application.Initialize;
    Application.CreateForm(TF1, F1);
    Application.Run;
end.

```

**Особливі ключі компілятора DATA.dof**

```
[Compiler]
A=1 B=0 C=1 D=1
E=0 F=0 G=1 H=1
I=1 J=1 K=0 L=1
M=0 N=1 O=1 P=1
Q=0 R=0 S=0 T=0
U=0 V=1 W=0 X=1
Y=0 Z=1
ShowHints=1
ShowWarnings=1
UnitAliases=WinTypes=Windows;WinProcs=Windows;DbiTypes=BDE;DbiProcs=BDE;DbiEr
rs=BDE;
[Linker]
MapFile=0
OutputObjs=0
ConsoleApp=1
DebugInfo=0
MinStackSize=16384
MaxStackSize=1048576
ImageBase=4194304
ExeDescription=
[Directories]
OutputDir=
UnitOutputDir=
SearchPath=
Packages=VCLX30;VCL30;VCLDB30;VCLDBX30;INETDB30;INET30;VCLSM30;QRPT30;TEEUI3
0;TEEDB30;
Conditionals=
DebugSourceDirs=
UsePackages=0

[Parameters]
RunParams=
HostApplication=

[Version Info]
IncludeVerInfo=0
AutoIncBuild=0
MajorVer=1
MinorVer=0
Release=0
Build=0
Debug=0
PreRelease=0
Special=0
Private=0
DLL=0
Locale=3082
CodePage=1252

[Version Info Keys]
CompanyName=
FileDescription=
FileVersion=1.0.0.0
InternalName=
LegalCopyright=
LegalTrademarks=
OriginalFilename=
ProductName=
ProductVersion=1.0.0.0
Comments=
```

## Налагодження параметрів GPS.pas

```

unit GPS;

  {Hyrba O.M.KI-20M1,4}

interface
uses
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, ExtCtrls, buttons, grids, stdctrls, clipbrd, Printers;

type
TMouseEvent = procedure (Sender: TObject; Shift: TShiftState; Points: trect)
of object;
TGPSdataArray=array[0..16]of double;

TGPS = class(TPanel)
private
MAXELNUM:integer;
sel:TSELARRAY;
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;
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;
FOnMouseBox:TMouseEvent;
procedure time(sender:tobject);
procedure replaceglis;
Procedure CalcChartRatios;
Procedure SetPBgColor(k:integer;c:tcolor);
function GetPBgColor(k:integer):tcolor;
Procedure SetPFgColor(k:integer;c:tcolor);
function GetPFgColor(k:integer):tcolor;
Procedure SetPPattern(k:integer;c:byte);
function GetPPattern(k:integer):byte;
Procedure SetChartLegend(k:word;s:string);
function GetChartLegend(k:word):string;
Procedure SetBitMap(fname:string);
Procedure DrawBitMap(h:hdc);
Procedure CleanHDC(h:hdc;col:tcolor;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 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 WMRButtonDown (var msg:TWMRButtonDown); message WM_RBUTTONDOWN;
Procedure WMLButtonUp (var msg:TWMLButtonUp); message WM_LBUTTONUP;
Procedure WMRButtonUp (var msg:TWMRButtonUp); message WM_RBUTTONUP;
Procedure WMouseMove (var msg:TWMMouseMove); message WM_MOUSEMOVE;
Procedure Loaded; Override;

public
drw:tdraws;
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);
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 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 PBgColor[i:integer]:tcolor read GetPBgColor write SetPBgColor;
property PFgColor[i:integer]:tcolor read GetPFgColor write SetPFgColor;

```

implementation

```

constructor TGPS.create(sender:tcomponent);

```

```

begin

```

```

  inherited create(sender);

```

```

  ready:=false;

```

```

  fgrid:=8;

```

```

  MaxElNum:=TGPS_MAXELEMENTS;

```

```

  MyVersion:=TGPS_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;

```

```

Procedure TGPS.Loaded;

```

```

begin

```

```

  Inherited Loaded;

```

```

  CreatePanels;

```

```

end;

```

```

Procedure TGPS.CreatePanels;

```

```

procedure createmasks(bm:tbitmap);

```

```

var t: tcolor;
    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:= 'GPS 2.0';    infopanel.visible:=true;
InfoPanel.parent:=self;
if bitmap<>nil then SetBitmap(bmpfname);
if polfname<>' ' then Fromfile(polfname);
end;

```

```

destructor TGPS.destroy;
var 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;

```

```

Procedure TGPS.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
fillcolor,bordercolor:longint;
numpuntos:word;

```

```

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

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

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

new(buffer);
assignfile(f,fname);
reset(f,1);

blockread(f,ax,100);
if ax<>MyVersion then begin
closefile(f);
dispose(buffer);
exit;
end;
screen.cursor:=crHourGlass;
for i:=1 to npolypoints do np[i].free;
npolypoints:=0;
ready:=false;
c:=tsapoint.create;
try
if nbloqs> MaxElNum then nbloqs:=maxelnum;
npolypoints:=nbloqs;
for i:=1 to npolypoints do begin
blockread(f,bloque,sizeof(bloque));
np[i]:=TSPolyPol.create(bloque.nummarcos,bloque.numpolis,bloque.nombre);
for j:=1 to bloque.nummarcos do begin
blockread(f,elemento,sizeof(elemento));
blockread(f,buffer^,sizeof(tSmallPoint)*elemento.numpuntos);
np[i].SetMValsSmall(elemento.fillcolor,elemento.bordercolor,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));
blockread(f,buffer^,sizeof(tSmallPoint)*elemento.numpuntos);
np[i].SetPValsSmall(elemento.fillcolor,elemento.bordercolor,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));
np[i].npoints:=nupu;
for ipu:=1 to nupu do begin
blockread(f,c.c,sizeof(c.c));
np[i].points[ipu]:=c;
end;
blockread(f,np[i].eo,sizeof(np[i].eo));
blockread(f,np[i].link,sizeof(np[i].link));
blockread(f,np[i].values,sizeof(np[i].values));
np[i].CalcCaja;
np[i].drawer:=drw;
end;
finally

```

```

closefile(f);
dispose(buffer);
c.free;
drw.setfonts;
arr.free;
arr:=TSArrange.create(2,npolypoints,'Normal');
if bmpfname<>' ' then setbitmap(bmpfname);
if arrfname<>' ' then ArrFromFile(arrfname);
CalcCaja;
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;
Procedure TGPS.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
fillcolor,bordercolor:longint;
numpuntos:word;
layer:word;
other:longint;
nombre:str50;
end;
buffer:^SPltSmallPoint;
nupu,ipu:word;
c:tsapoint;
begin
new(buffer);
c:=tsapoint.create;
assignfile(f,fname);
rewrite(f,1);
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.fillcolor:=np[i].m[j].bkcol;
elemento.bordercolor:=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.fillcolor:=np[i].p[j].bkcol;
elemento.bordercolor:=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);
dispose(buffer);
c.free;
end;

Procedure TGPS.ReadData(fname:string);
var f:file of TGPSdataArray; TGPSdataArray=array[0..16]of double;
ad:TGPSdataArray;
i, j:integer;
begin
assignfile(f, fname);
reset(f);
i:=1;
try
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
closefile(f);
end;
end;

Procedure TGPS.SaveData(fname:string);
var f:file of TGPSdataArray; TGPSdataArray=array[0..16]of double;
ad:TGPSdataArray;
i, j:integer;
begin
assignfile(f, fname);
rewrite(f);
try
for i:=1 to elementcount do begin
for j:=0 to 16 do ad[j]:=np[i].values[j];
write(f, ad);
end;
finally
closefile(f);
end;
end;

```

```

Function TGPS.GetFont(i:integer):tfont;
begin
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;

Procedure TGPS.ArrCalcBoxes;
var i,j:integer;
begin
for i:=1 to arr.n do begin
prepect(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;

Procedure TGPS.ArrFromFile(fname:string);
begin
arrfname:=fname;
if not fileexists(fname) then exit;
arr.fromfile(fname);
arrcalcboxes;
end;

Function TGPS.getElmName(i:word):string;
begin
if ((i>0) and (i<=npolypoints)) then GetElmName:=np[i].nombre
else GetElmName:='';
end;

Procedure TGPS.SetElmName(i:word;s:string);
begin
if ((i>0) and (i<=npolypoints)) then np[i].nombre:=s;
end;

Procedure TGPS.SetSel(lb:tlistbox);
var i:integer;
begin
for i:=1 to lb.items.count do sel[i]:=(lb.selected[i-1]);
end;

Function TGPS.IsArrSelected(father:word):boolean;
var i:integer;
begin
IsArrSelected:=true;
for i:=1 to npolypoints do if sel[i] then
if arr.Esta(father,i) then exit;
IsArrSelected:=false;
end;

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.color:=drw.co.boxbackcolor;
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:='Total: '+ax;
strpcopy(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.linecolor,drw.co.colors[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.colors[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.LineColor,drw.co.Colors[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.colors[1],drw.co.linecolor,drw.co.Colors[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.colors[1],drw.co.Colors[i],drw.co.linecolor);
  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.ndecimals=0 then ax:=inttostr(trunc(ds[i]))
    else str(ds[i]:3:drw.co.ndecimals,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;

end;

procedure SelObjects(c:tcolor;f:tfont);
begin
  oldf:=selectobject(h,f.handle);
  settextcolor(h,f.color);
  drw.getbrocha(h,0,f.color,c);
  rectangle(h,rto.left,rto.top,rto.right,rto.bottom);
  drw.releasebrocha;
end;

var 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.color,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;

Procedure ChartLegend(h:hdc;r:trect);
var i:integer;
wy,wb,wm:integer;
a:array[0..255] of char;
begin
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;
settextcolor(h,drw.bo.linecolor);
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);
strcpy(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];
    strcpy(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); strcpy(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);
    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_TOP);

    for i:=1 to drw.bo.custom.nranges do begin
        strcpy(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;
    settextcolor(h,drw.bo.linecolor);
    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;
    f.free;
end;
end;
end;

Procedure TGPS.CalcChartRatios;
var i,j:integer;
    mx,mx0,mx1:real;
begin
    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;

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

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

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

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

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

Procedure TGPS.ArrSetName(name:string;father:word);
begin
if not ready then exit;

```

```

if father>arr.n then exit;
arr.titulos[father]^:=name;
end;

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

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

Procedure TGPS.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;

Function TGPS.GetElSelected (i:word):boolean;
begin
GetElSelected:=false;
if i>0 then if i<=npolypoints then GetElSelected:=Sel[i];
end;

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

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

Procedure TGPS.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;

Procedure TGPS.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
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 TGPS.CleanHDC(h:hdc;col:tcolor;box:trect);
var b,oldb:hbrush;
p,oldp:hpen;
i,j,k,wy:integer;
gr:TGrArray;
w,hh,x,y:integer;
bm:tbitmap;
begin
CalcColGradients(drw.bko.mincolor,drw.bko.maxcolor,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 TGPS.Meridians (h:hdc; box:trect);
var i, j, w, hh, x, y:integer;
p, oldp:hpen;
pix:tcolor;
bm:tbitmap;
begin
if drw.bko.lines then begin
if not tometa then box:=drw._back;
w:=drw.bko.pcth; 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.linecolor);
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;
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 TGPS.SetColors(t:byte;f,b:tcolor;ix:word);
begin
np[ix].setcolors(t,f,b);
end;

Procedure TGPS.SetPBGColor(k:integer;c:tcolor);
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 TGPS.GetPBGColor(k:integer):tcolor;
begin
if np[k].npolis>0 then
GetPBGColor:=np[k].p[1].bkcol
else if np[k].nmarcos>0 then
GetPBGColor:=np[k].m[1].bkcol
else GetPBGColor:=clwhite;
end;

Procedure TGPS.SetPFgColor(k:integer;c:tcolor);
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 TGPS.GetPFgColor(k:integer):tcolor;
begin
if np[k].npolis>0 then
GetPFgColor:=np[k].p[1].FRcol
else if np[k].nmarcos>0 then
GetPFgColor:=np[k].m[1].FRcol
else GetPFgColor:=clBLACK;
end;

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

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

Procedure TGPS.SetShadows(a,b:tcolor);
begin
drw.calcshadows(a,b);
end;

Procedure TGPS.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
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;
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].SetColors(0,0,drw.sot.colsf[j]);
break;
end;
begin
calcchartrratios;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
case drw.sot.kind of
0:begin
drw.calcshadows(drw.so.mincolor,drw.so.maxcolor);

```

```

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].SetColor(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].SetColor(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].SetColor(0,0,drw.sot.cols[j]);
break;
end;
end;

Procedure TGPS.RectToPolyLine(r2:trect;var cuadro:array of tpoint);
begin
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 TGPS.adaptlegrect(r,r1:trect;var r2:trect);
begin
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 TGPS.ClipWMF;
var
hdcmeta:hdc;
hmeta:hmetafile;
r:trect;
twmf:tmetafile;
begin
tometa:=true;
hdcmeta:=createmetabase(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:=closemetabase(hdcmeta);
twmf:=tmetafile.create;
twmf.width:=width;
twmf.height:=height;
with TMetafileCanvas.Create(twmf, 0) do

```

```

try
PlayMetaFile (handle,hmeta);
finally
free;
end;
twmf.enhanced:=false;
clipboard.assign (twmf);
twmf.free;
tometa:=false;
end;
Procedure TGPS.WMF_clip2file (wnd:HWND;fname:string);
var
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);
copymetafile (lpmfp^.hmf,a);
GlobalUnlock (hGMem);
end;
CloseClipboard;
end;

procedure TGPS.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);
reset (f,1);
blockread (f,tm,sizeof (tm));
n:=0;
while not eof (f) do begin
blockread (f,rs,4);
blockread (f,cf,2);
if cf=$0324 then inc (n);
for j:=1 to rs-3 do blockread (f,cf,2);
end;
closefile (f);
if n=0 then exit;

assignfile (f,fname);
reset (f,1);

assignfile (ft,textfname);
rewrite (ft);

blockread (f,tm,sizeof (tm));
I:=1;

while not eof (f) do begin
blockread (f,rs,4);
blockread (f,cf,2);
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,'+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);
closefile(ft);
dispose(ltp);
end;

Function TGPS.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
new(ltp);
assignfile(f,fname);
reset(f,1);
blockread(f,tm,sizeof(tm));
n:=0;
re.left:=32000;re.top:=32000;re.bottom:=-32000;re.right:=-32000;

while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
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);
if n=0 then exit;

assignfile(f,fname);
reset(f,1);
blockread(f,tm,sizeof(tm));
I:=1;

while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
if ((cf=$0324) and (i<maxelnum)) then begin
blockread(f,np,2);
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);
dispose(ltp);
ClpWMFRect:=re;
end;

Procedure TGPS.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);
reset(f,1);
blockread(f,tm,sizeof(tm));
n:=0;
while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
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);
if n=0 then exit;

assignfile(f,fname);
reset(f,1);

assignfile(ft,textfname);
rewrite(ft);

blockread(f,tm,sizeof(tm));
I:=1;
puntos:=0;
while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
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);
blockread(f,ltp^,np*sizeof(TSmallPoint));
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);
blockread(f,a,lt); a[lt+1]:=#0; ax:=strpas(a);
writeln(ft,'/T'+inttostr(puntos)+' #3');
blockread(f,lt,2);
write(ft,inttostr(lt)+ ' , ');
blockread(f,lt,2);
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);
closefile(ft);
dispose(ltp);
end;

Procedure TGPS.FromWMF(fname:string);
var tmf:tmetafile;
a:array[0..255] of char;
ax,bx:string;
begin
GetWindowsDirectory(a,200);
ax:=strpas(a)+'\temp.clp';
bx:=strpas(a)+'\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;

Procedure TGPS.FromClipWMF;
var a:array[0..255] of char;
ax,bx:string;
begin
if not clipboard.hasformat(CF_metafilepict) then exit;
GetWindowsDirectory(a,200);
ax:=strpas(a)+'\temp.clp';
bx:=strpas(a)+'\temp.txt';

```

```

WMF_clip2file (parent.handle, ax);
ScaleClpWMF2Text (ax,bx,true);
FromTextFile (bx);
end;

```

```

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

```

```

Procedure TGPS.ClipBmp;
var
r:trect;
bm01:tbitmap;
begin
if not ready then exit;
bm01:=tbitmap.create;
try
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
bm01.free;
end;
end;

```

```

Procedure TGPS.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
TGPS.Snapshot (destinationcanvas:tcanvas;destrect:trect;maprect:trect);
var r:trect;
begin
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 TGPS.SetChartValue(element,item:word;value:real);
begin
if element>npolypoints then exit;
if element<1 then exit;
if item>drw.co.nvalues then exit;
np[element].values[item]:=value;
end;

```

```

Function TGPS.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;

```

```

Procedure TGPS.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 TGPS.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;

```

```

Procedure TGPS.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  TGPS.GetPatternBits(i:word):string;
begin
if i>8 then exit;
if i=0 then GetPatternBits:='000000'
else GetPatternBits:=trama[i];
end;

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

Procedure TGPS.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.linecolor,drw.co.colors[pat]);
rectangle(h,x1,y1,x2,y2);
drw.ReleaseBrocha;
end;

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

Function  TGPS.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 TGPS.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 TGPS.ShowPolNames(b:boolean);
begin
drw.showpols:=b;
end;

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

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

Function TGPS.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;

Function TGPS.Nearest(p:tpoint):integer;
var i,j,k:integer;
mybox:trect;
begin
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 TGPS.NearestElement(p:tpoint):integer;
var i,j,k:integer;
mybox:trect;
begin
k:=0;
NearestEement:=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:^SPltpoint;
maxcaja:trect;
c:char;
thelayer:integer;
lm:longint;
begin
reset(f);
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].CalcCaja;
comprect(np[i].caja,maxcaja);
np[i].drawer:=drw;
end;
closefile(f);
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;

Function  TGPS.AddPoint(elem:integer;c:tcitypoint):integer;
var a:tsapoint;
begin
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 TGPS.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  TGPS.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  TGPS.GetPointCount(elem:integer):integer;
begin
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 TGPS.ClearPoints(elem:integer);
begin
if elem>npolypoints then exit; if elem<1 then exit;
np[elem].npoints:=0;
end;

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

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

Procedure TGPS.MyMouseUp(Sender: TObject; Button: TMouseButton;Shift:
TShiftState; X, Y: Integer);

```

```

var p:tpoint;
i,k:integer;
pu:tpoint;
pr:trect;
begin
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
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;
end;

Procedure TGPS.WMLButtonDown(var msg:TWMLButtonDown);
begin
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 TGPS.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 TGPS.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 TGPS.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 TGPS.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 TGPS.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[npolypoints]:=auxp;
selected[npolypoints]:=auxs;
end;

```

```

Procedure TGPS.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 TGPS.ReMap(var m:array of integer);
var sp:^TSPOLARRAY;
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 TGPS.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;

Procedure TGPS.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 TGPS.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;

Procedure TGPS.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;
end;
dec(nbacks);
end;
end;
end;

```

```

Procedure TGPS.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 TGPS.CalcCaja;
var i:integer;
begin;
prepect(MapLimits);
for i:=1 to npolypoints do comprect(np[i].caja,MapLimits);
end;

Procedure TGPS.Merge(fname:string;layeroffset:integer; x,y:integer);
var agps:TGPS;
i,j,k:integer;
begin
if not fileexists(fname) then exit;
ready:=false;
agps:=TGPS.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;
CalcCaja;
ready:=true;
zoombox:=MapLimits;
end;

Procedure TGPS.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;

Procedure TGPS.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';
CalcCaja;
drawer:=drw;
end;
end;
calccaja;
selected[npolypoints]:=true;
end;

Procedure TGPS.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 TGPS.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;

procedure TGPS.posterxN(Nx,Ny:integer);
VAR
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
with p do begin
HorzPages:=nx;
VertPages:=ny;
HorzMarge:=100;
VertMarge:=100;
Image:='temp.wmf';
LabelPages:=true;

end;
SaveWMF('temp.wmf');

```

```

p.doprint;
finally
p.free;
screen.cursor:=crdefault;
end;
end;

```

```

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

```

```

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

```

```

Procedure TGPS.LoadMosaicBitmap(fname:string);
begin
drw.mosaicbmp.loadfromfile(fname);
end;

```

```

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

```

```

procedure TGPS.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;

```

```

function TGPS.GlyphCount:integer;
begin
GlyphCount:=gli.count;
end;

```

```

Procedure
TGPS.GlyphSettings(ixglyph:integer;imageix:integer;animate:boolean;enable:boolea
n);
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;

```

```

Procedure TGPS.LoadGlyphsBitmap(fname:string);
begin

```

```
gli.Loadfromfile(fname);  
end;  
  
end.
```

Кафедра \_ КБПЗ \_ 2021 рік

**Файл авторського права About.pas**

```
unit About;

{Hyrba O.M.KI-20M1,4}
interface

uses WinTypes, WinProcs, Classes, Graphics, Forms, Controls, StdCtrls,
    Buttons, ExtCtrls;

type
    TAboutBox = class(TForm)
        Panell: 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
    AboutBox: TAboutBox;

implementation

{$R *.DFM}

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

end.
```