

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи терміналів
мережі платіжних автоматів”

Виконав здобувач вищої освіти
II курсу, групи КІ-21М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Большов В.С.
« ____ » _____ 2022 р.

Керівник проекту
доктор технічних наук, доцент
_____ Коваленко А.С.
« ____ » _____ 2022 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2022 року

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

Большову Владиславу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи терміналів мережі платіжних автоматів*

2. Керівник роботи *Коваленко Анна Степанівна, канд. техн. наук, доцент*

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

затверджені наказом вищого навчального закладу № 19-13 від 17.08.2022 року

3. Строк подання студентом роботи до захисту *10.12.2022 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи терміналів мережі платіжних автоматів*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника

Коваленко А.С.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача

Большов В.С.
(прізвище та ініціали)

АНОТАЦІЯ

Большов В.С. Дослідження та програмна реалізація системи терміналів мережі платіжних автоматів. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи терміналів мережі платіжних автоматів.

Метою розробки є дослідження та програмна реалізація системи терміналів мережі платіжних автоматів.

Об'єктом дослідження є процес терміналів мережі платіжних автоматів.

Предметом дослідження є методи терміналів мережі платіжних автоматів.

Методи дослідження базуються на методах теорії комп'ютерних систем та мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи терміналів мережі платіжних автоматів.

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

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

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

Програму розроблено в середовищі Embarcadero Delphi.

Ключові слова: комп'ютерна інженерія, платіжні автомати

ABSTRACT

Bolshov V.S. Research and software implementation of the terminal system of the network of payment machines. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this final qualification work for the second (master's) level of higher education, software is developed, which is intended for the terminal system of the network of payment machines.

The goal of the development is the research and software implementation of the terminal system of the network of payment machines.

The object of the research is the process of the terminals of the network of payment machines.

The subject of research is the methods of terminals of the network of payment machines.

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

The result of the work is the software implementation of the terminal system of the network of payment machines.

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

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

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

The program was developed in the Embarcadero Delphi environment.

Keywords: computer engineering, payment machines

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	21
2.3 Розгорнута постановка завдання	26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	28
3.1 Опис функціонування системи	28
3.2 Розробка структурної схеми.....	41
3.3 Розробка функціональної схеми	45
3.4 Розробка діаграми процесів.....	46
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	48
4.2 Захист розробленого програмного забезпечення.....	57
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	61
6 НАУКОВА НОВИЗНА	67

ВКРМ-123.22.0002.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Большов В.С.			Дослідження та програмна реалізація системи терміналів мережі платіжних автоматів	Лім.	Аркуш	Аркушів
Перев.		Коваленко А.С.				М	1	107
Н.контр.		Гермак В.С.			ЦНТУ КІ-21М-1,4			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	68
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	68
7.2 Розрахунок трудомісткості розробки програмної продукції.....	70
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	72
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	75
7.5 Визначення собівартості розробки та ціни програмної продукції.....	80
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	83
7.7 Визначення експлуатаційних витрат.....	84
7.8 Визначення економічної ефективності програмної продукції.....	85
7.9 Висновок.....	87
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	88
8.1 Вступ.....	88
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	89
8.3 Аналіз умов праці програміста	90
8.4 Розробка заходів з умов поліпшення охорони праці.....	92
8.5 Розрахункова частина	94
8.6 Висновки до розділу.....	96
9 ОСНОВНІ ВИСНОВКИ.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

ВСТУП

Актуальність теми. Сучасний світ неможливо уявити без мобільного зв'язку. Він проник в усі ніши нашого життя. Причому сучасний мобільний зв'язок виконує не тільки роль телефонного зв'язку, але є й могутнім центром телекомунікаційних засобів передачі інформації. GSM, CDMA, GPRS, EDGE, 3G, 4G, 5G – це далеко не повний перелік стандартів стільникового зв'язку та технологій передачі різного виду даних, починаючи від суто телефонної розмови і відеозв'язку й закінчуючи мобільним з'єднанням з мережею INTERNET [1-3]. Але як відомо за усе в житті приходиться платити. І тут виникає наступна задача, що потребує вирішення: як можна спростити процедуру оплати мобільного зв'язку. Звісно можна зайти до магазину й поповнити рахунок у продавця. Та на даному етапі розвитку техніки, існує й альтернативний варіант оплати витрат з мобільного зв'язку – це застосування автоматів самообслуговування. Термінал мережі платіжних автоматів (ТПМА) дозволяє повністю автоматизувати різні елементи процесів торгівлі й обслуговування залежно від розглянутої галузі, представляючи надійне й функціональне рішення, що дозволяє приймати наявні платежі [1-5]. Термінал мережі платіжних автоматів поставляється у вандалостійкому виконанні, що припускає здатність витримувати агресивні впливи з боку зовнішнього середовища зі збереженням повної працездатності [6]. Термінал мережі платіжних автоматів ідеально підходить для установки на частково охоронюваних територіях адміністративних будинків, торгових центрів і т.д.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи терміналів мережі платіжних автоматів.

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

- Огляд існуючих систем терміналів мережі платіжних автоматів.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

- Дослідження системи терміналів мережі платіжних автоматів.
- Програмна реалізація системи терміналів мережі платіжних автоматів.

Об'єктом дослідження є процес терміналів мережі платіжних автоматів.

Предметом дослідження є методи терміналів мережі платіжних автоматів.

Методи дослідження базуються на методах теорії комп'ютерних систем та мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод терміналів мережі платіжних автоматів.
- Розроблено вітчизняний продукт терміналів мережі платіжних автоматів, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі терміналів мережі платіжних автоматів.

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

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

В магістерському проєкті, необхідно розробити програмне забезпечення терміналів мережі платіжних автоматів. Для цього у цьому розділі ми розглянемо призначення та область використання цих автоматів. Вони призначені для задоволення потреби у швидкій оплаті послуг оператора мобільного зв'язку, у будь який зручний для користувача час. Це на відміну від магазинів та центрів обслуговування, де оплату можна здійснити тільки у час, коли вони працюють, та, як правило тільки за послуги відповідного оператора, дозволяє спростити оплату послуг користувачу мобільного телефону. Це є тим паче актуальним, в зв'язку з тим, що багато громадян одночасно користується послугами двох, або трьох операторів мобільного зв'язку.

На даний час існує ціла лінійка автоматів самообслуговування, які можливо використовувати у ролі автоматів з оплати послуг мобільного зв'язку:

- Автомат самообслуговування ОСМП-2.
- Автомат самообслуговування ОСМП 2 із другим монітором.
- Автомат самообслуговування ОСМП-МІНІ.
- Автомат самообслуговування ОСМП-ВУЛИЦЯ.
- Автомат самообслуговування ОСМП 2 з лайт-боксом.

Вони відрізняються, як правило тільки зовнішнім виглядом та відповідно ціною. На даний час ціна на них коливається від 14000 грн. до 24000 грн. [1-5]. Але для усіх цих автоматів функціональні блоки є однаковими.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1.2 Область застосування

Розглянемо які експлуатаційні вимоги висуваються до цих автоматів. При експлуатації автомата повинні дотримуватися наступні вимоги:

1. У приміщеннях, призначених для експлуатації автомата, повинні бути відсутніми агресивні середовища, масова концентрація пилу в повітрі повинна бути не більше $0,75 \text{ мг/м}^3$, електрична складова електромагнітного поля перешкод не повинна перевищувати $0,3 \text{ в/м}$ у діапазоні частот від $0,15$ до $300,00 \text{ МГц}$.

2. Напруга живлення мережі повинне бути $220 \text{ В} \pm 10 \%$. 50 Гц .

3. Вимоги по забезпеченню пожежної електробезпеки (заземлення) у приміщеннях повинні бути виконані відповідно до ДСТ Р 50276.0.

4. Кліматичні фактори приміщення для експлуатації виробів повинні бути за ДСТ Р 50444 для годиться кліматичного виконання УХЛ категорії 4.2.

Автомат зберігає працездатність при впливі наступних кліматичних факторів:

1. Температура навколишнього повітря від 10 до 40 градусів Цельсія.

2. Відносна вологість повітря від 40 до 80% при температурі 25 градусів Цельсія, без конденсації.

Розглянувши експлуатаційні характеристики перейдемо до розгляду типового складу системи. Стандартна комплектація автомата містить у собі наступні компоненти [1-5].

ІВМ РС сумісний комп'ютер

1. Процесор Intel Celeron 310 (2130MHz , 533 MHz FSB) $256\text{Kb BOX w/cooler}$ (Socket 478).

2. Материнська плата ECS P4M 800-M (VIA P4M800) SVGA, Sound, LAN, 3 PCI, 2 DDR 3200, mATX (Retail).

3. Модуль пам'яті DDR SDRAM 256Mb PC-3200 .

4. Контролер ST-Lab I-152 PCI 2 port fast serial + 1 port EPP combo (Retail).

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

5. Сторожовий таймер версія 1.0 (таймер версії 2.0 може бути встановлений за додаткову плату).

6. Блок живлення InWin 300W ATX (P4).

7. Вентилятор GlacialTech GT1225 EBDL-1/чорний 120x120x25 для корпусу, підшипник ковзання.

8. Накопичувач HDD 40Gb 7200rpm 2Mb.

9. Кабель мережний монітор-комп'ютер 1.8 м.

10. RAM-512 Mb DDR.

Сенсорний монітор

1. Вандалостійкий сенсорний монітор TFT 17".

2. Інтерфейсний шнур монітора.

3. Шнур живлення монітора.

4. Вандалостійкий сенсорний екран 17 дюймів.

5. Контролер сенсорного екрана.

Світлова панель

1. Світлова панель розміром 71x50 см.

2. Патрон G5.

3. Вбудований електронний баласт.

4. Компактлюмінісцентний світильник.

5. Лампа денного світла FERON 2 шт.

Вандалостійкий корпус

1. Основа – корпус терміналу самообслуговування Eсо.

2. Блок розеток.

3. Блок живлення (250 Вт).

4. Джерело безперебійного живлення.

5. Додаткова вентиляція.

6. Активні колонки.

7. Замок високого ступеня захисту Abloy 2 шт. із ключами.

8. Замок для зовнішнього вмикання/вимикання живлення 1 шт.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

9. Електричний запор.
10. Коннектор для електричного запору.
11. Концевик двері.
12. Сирена сигналізації.

Пристрій для прийому грошей:

1. Купюроприймач CashCode SM (стеккер на 1500 купюр).
2. Сейф.
3. Замок на сейф купюроприймача 1 шт.
4. Інтерфейсний шнур.
5. Змінний ящик для зберігання купюр – стеккер.
6. Блок живлення

Пристрій для печаті бланків:

1. Вбудований пристрій для печаті інформації на термобумазі – термопринтер:
 - Рри 700.
 - EPSON 422.
 - CUSTOM VKP80.
2. Інтерфейсний шнур.
3. Блок живлення.

Приймально-передавальний пристрій:

1. Бездротовий GPRS/GSM-модем (робочий стандарт GSM 900/1800).
2. Антена GSM.
3. Інтерфейсний шнур.
4. Шнур живлення.

Зовнішній вигляд типового автомату представлено на рисунку 1.1.

На рисунку 1.1 цифрами відмічені наступні блоки автомату по оплаті послуг мобільного зв'язку:

1. Світлова панель.
2. Моніторна збірка.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

3. Принтер.
4. Купюроприймач.
5. Комп'ютерний відсік.

Крім цих блоків у складі блоків автомату по оплаті послуг мобільного зв'язку знаходиться електромеханічний замок вмикання/вимикання живлення.

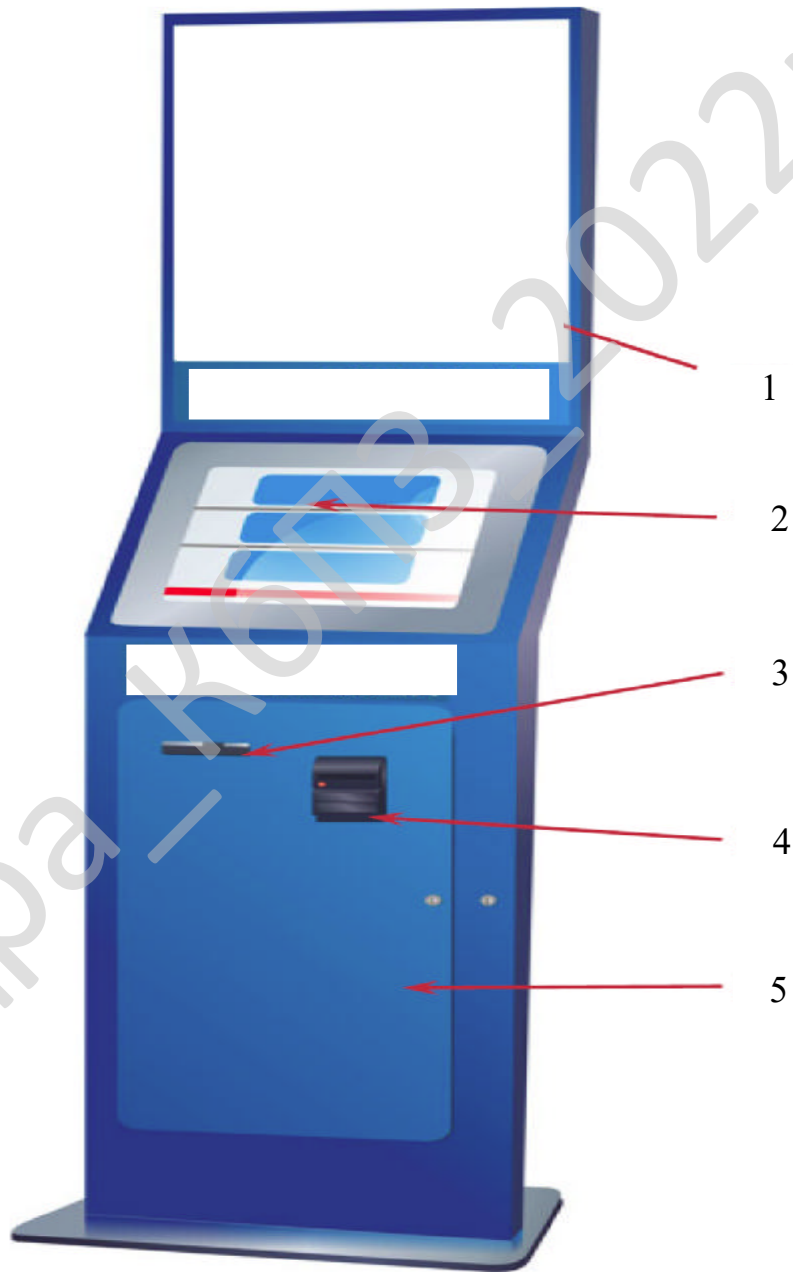


Рисунок 1.1 – Типовий термінал мережі платіжних автоматів

Усі ці компоненти зв'язані один з одним та повинні, якимось керуватись. Як бачимо з опису складових автомату по оплаті послуг центральне місце займає IBM-сумісний комп'ютер. Він призначений для керування цим автоматом. Відповідно потрібно розробити програмне забезпечення, яке буде реалізовувати функції та задачі, що стоять перед цим автоматом. Тобто воно відповідає за розпізнання типу купюр, відслідковує вибір оператора мобільного зв'язку, чиї послуги оплачуються, робить запит через мережу стільникового зв'язку GSM/GPRS до відповідної бази даних потрібного оператора, та роздруковує код поповнення на суму, яка була внесена для цього оператора мобільного зв'язку. Розробці цього програмного забезпечення й будуть призначені наступні розділи.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи терміналів мережі платіжних автоматів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

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

Опис функціональних частин автомата оплати

Моніторна збірка

Є обов'язковим елементом всіх автоматів самообслуговування й призначена для відображення інформації кінцевому користувачеві й організації функціонального інтерфейсу.

Складається зі спеціального монітора, що вбудовується, і спеціального вандалостійкого сенсорного екрана.

Налаштування й ремонт здійснюють інженери служби підтримки постачальника рішення.

Комп'ютерний відсік

Являє собою комп'ютерну частину автомата й вузол об'єднання всіх пристроїв у єдину систему.

Включає у свій склад IBM PC сумісний комп'ютер, розташований на спеціальному шасі.

Налаштування й ремонт здійснюють інженери служби підтримки постачальника рішення.

Відсік має окремий замок із ключем.

Сторожовий таймер

У комп'ютерному відсіку розташований сторожовий таймер, що виконує наступні функції:

- Перезавантаження комп'ютера у випадку тривалої відсутності сигналу від комп'ютера;

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- Перезавантаження модему у випадку тривалої відсутності сигналу від модему;
- Вмикання сигналізації при відкритті дверей;
- Вмикання сигналізації при ударі;
- Контроль лінії мережі 220В;
- Контроль лінії мережі 12В;
- Режим вмикання/вимикання автомата за розкладом;
- Авторизація електронних ключів;
- Контроль температури зовнішнього середовища.

Налаштування сторожового таймера здійснюється з адміністративного інтерфейсу програмного забезпечення автомата.

Купюроприймач

Призначений для прийому готівки при оплаті послуг мобільного зв'язку. При прийманні купюр купюроприймач визначає номінал прийнятої купюри й перевіряє її на факт підробки. Купюроприймач призначений для прийому українських гривень номіналом 10, 25, 50, 100 і 300 гривень. Зберігання прийнятих купюр здійснюється в спеціальному ящику – грошовому стеккері, що знімається при інкасації автомата. Відкриття купюроприймача здійснюється за допомогою спеціального ключа.

Налаштування й ремонт здійснюють інженери служби підтримки постачальника рішення.

Підключення купюроприймача здійснюється через комп'ютерний відсік.

Принтер

Призначений для печаті інформації на бланках термобумаги. Видача інформаційних бланків здійснюється в спеціальний лоток, розташований у передній частині автомата. Термобумага для печаті розміщується в спеціальному рулоні, що заправляється через комп'ютерний відсік.

Налаштування й ремонт здійснюють інженери служби підтримки постачальника. Підключення принтера здійснюється через комп'ютерний відсік.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

GPRS/GSM модем з антеною

Призначений для організації обміну інформацією між автоматом і віддаленим комп'ютером (сервером) за технологією бездротового зв'язку GPRS або GSM. Для роботи модему необхідна установка в нього спеціальної SIM-карти, що відповідає оператора стільникового зв'язку, що надає послуги з передачі даних у місці установки автомата. SIM-карта може не входити в базовий комплект поставки автоматного рішення. Установка SIM-карти й налаштування модему можуть здійснюватися постачальником або замовником рішення за домовленістю на основі відповідного керівництва. Ремонт здійснюють інженери служби підтримки постачальника рішення. Підключення GPRS/GSM модему здійснюється через комп'ютерний відсік.

Електромеханічний замок вмикання/вимикання живлення

Служить для зовнішнього вмикання/вимикання автомата. Може застосовуватися для швидкого виконання операції вмикання/вимикання, а також служити додатковим інструментом організації системи безпеки.

Налаштування й ремонт здійснюють інженери служби підтримки постачальника рішення. Підключення електромеханічного замка здійснюється через комп'ютерний відсік. Замок має окремий ключ.

Електричний запор

Служить для відкриття/закриття автомата, а також для авторизації персони для доступу до адміністративних функцій автомата.

Налаштування й ремонт коннектора здійснюють інженери служби підтримки постачальника рішення. Прописування в системі ключів відбувається через адміністративний інтерфейс автомата.

Світлова панель

Служить для оформлення точки прийому платежів, розміщення реклами.

Обслуговування здійснюють інженери служби підтримки постачальника рішення. Електротехнічні характеристики: 20 W, 6400 K, 220-240 V, 50/60 Hz, MAX 0.6 A (100 W)

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Опис інструкції з установки й експлуатації автомату оплати

У даному розділі наведені інструкції з установки й експлуатації автомату по оплаті послуг операторів мобільного зв'язку, а також регламентується покрокове виконання операцій, пов'язаних з монтажем і підключенням комплексу.

З погляду обслуговуючого персоналу комплексу можна виділити наступні операції, виконання яких необхідно в процесі установки й підключення апаратного комплексу:

1. Операція монтажу.
2. Операція підключення живлення.
3. Операція підключення мережного кабелю.

1. Операція монтажу.

Виконання даної операції передбачає фіксацію апаратного комплексу до напольної поверхні, на яку відбувається його установка. Незафіксований до напольної поверхні апаратний комплекс зберігає стійке положення зі збереженням повної функціональності, тому монтаж є необов'язковою операцією, що виконується за рішенням замовника.

2. Операція підключення живлення.

Підключення живлення виконується підключенням кабелю живлення за допомогою спеціального роз'єму на задній стінці корпусу автомата. Підключення живлення здійснюється до заземленої розетки – 220V.

3. Операція підключення мережного кабелю.

Підключення мережного кабелю виконується з боку задньої стінки автомата. Всередині автомата кабель підключається до мережної карти. Уникайте використання ушкоджених і оголених проводів. Роз'єм мережного проводу, що підключається всередині комплексу, повинен відповідати типу RJ-45.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- Переконалися в тому, що світлові індикатори роботи комп'ютера й HDD горять.

- Закрити комп'ютерний відсік.

Варіант 2 (швидке вмикання):

- Переконалися в тому, що вилка живлення автомата уткнута в розетку.

- Переконалися в тому, що пристрій безперебійного живлення UPS вмикано (у випадку наявності UPS).

- Повернути ключ в електромеханічному замку за годинниковою стрілкою й повернути його в початкове положення.

- Переконалися по зображенню, що з'явилося на екрані, в тому, що автомат увімкнувся

Етапи операції вимикання

Варіант 1:

- Відкрити комп'ютерний відсік.

- Нажати й тримати кнопку вмикання/вимикання комп'ютера до повного вимикання.

- Переконалися в тому, що світлові індикатори роботи комп'ютера й HDD не горять.

- Виключити пристрій безперебійного живлення UPS (у випадку наявності UPS).

- Закрити комп'ютерний відсік.

- Вийняти вилку живлення автомата з розетки.

Варіант 2 (швидке вимикання):

- Повернути ключ в електромеханічному замку за годинниковою стрілкою й повернути його в початкове положення.

- Переконалися по зниклому на екрані зображенню в тому, що автомат виключився.

Етапи операції перезавантаження.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Уникайте операції перезавантаження! Перезавантаження автомата допускається тільки в тому випадку, якщо більше ніякі методи не допомагають.

При перезавантаженні можлива повна втрата даних

- Відкрити комп'ютерний відсік.
- Нажати кнопку перезавантаження комп'ютера.
- Переконатися в тому, що світлові індикатори роботи комп'ютера й HDD згасли й знову зайнялися.
- Закрити комп'ютерний відсік.

Особливо уникайте влучення води на головну панель, тому що підводима напруга становить 220V. Зробіть перевірку розташування землі й фази в силовій розетці, до якої збираєтеся підключати апаратний комплекс

2. Операція інкасації купюро приймача.

Виконання даної операції здійснюється через комп'ютерний відсік.

- Відкрити комп'ютерний відсік автомата.
- Відкрити сейф купюроприймача за допомогою спеціального ключа.
- Зняти грошову касету із кріплення купюроприймача.
- Відкрити грошову касету й витягти її вміст в інкасаторський мішок.
- Закрити грошову касету й заклацнути її в кріплення купюроприймача.
- Закрити сейф купюроприймача.
- Закрити комп'ютерний відділ автомата.

На практиці можливий випадок переповнення грошової касети, ємність якої становить не більше 400, 1000 або 1500 купюр, і наступної за цим зупинки роботи автомата. Для запобігання цієї проблеми необхідно стежити за заповненням касети, і робити її інкасацію до повного заповнення. У випадку використання запасної грошової касети процедура інкасації полягає в почерговій заміні заповненої касети на порожню. При цьому добавлення грошей може бути зроблено в безпечному місці.

3. Операція заміни термобумаги для принтера.

Citizen PPU-700. Заміна термобумаги.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Варіант 1:

- За допомогою важеля відкрийте кришку принтера.
- Вставте папір і закрийте кришку.
- При цьому папір буде автоматично завантажена й почнеться процес ініціалізації.
- Після завершення процесу ініціалізації принтер готовий до роботи.

Варіант 2:

- Не відкриваючи кришку, вставте край паперу в отвір.
- Дочекайтеся автоматичного завантаження паперу в принтер.
- Почнеться процес ініціалізації.
- Після завершення процесу ініціалізації принтер готовий до роботи.

Для успішного завантаження паперу в принтер, край паперу повинен бути акуратно відрізаний. Не допускається відривання й загинання краю бумаги, що вставляється.

Epson 422. Заміна термобумаги.

- Відкрити комп'ютерний відсік автомата.
- Зняти рулон термобумаги з папером, що залишився, і поставити на його місце новий.
- Заправити початок термобумаги в друкуючий механізм (на задній стінці принтера) і дочекатися автоматичного завантаження паперу.

Custom VKP80. Заміна термобумаги.

- Відкрити комп'ютерний відсік автомата.
- Край паперу просунути у вхідний отвір і дочекатися автоматичного завантаження паперу.

Налаштування параметра Paper Retracting.

Після того, як папір заправлений у принтер, виконаєте наступну послідовність дій:

- Виключіть принтер.
- Натисніть кнопку Line Feed (верхня кнопка).

- Включіть принтер.
- Дочекайтесь, поки принтер роздрукує довідкову інформацію, і швидко натисніть кнопку Form Feed (нижня кнопка).

- Натискайте послідовно кнопку Line Feed доти, поки не буде роздрукований рядок:

Paper Retracting . . .:Disabled

- Натисніть Form Feed для зміни значення. На чеку буде роздрукований рядок:

Paper Retracting . . .:Enabled

- Натисніть Line Feed.
- Виключіть й знову включіть принтер.

Установка прошивання.

- Запустите програму UpgCePrn.
- Натисніть кнопку Select (*.psw).
- У вікні, що відкрилося, виберіть файл нового прошивання.
- Після цього натисніть кнопку Begin Upgrade для початку процесу установки нового прошивання.

- Вам запропоновано вибрати порт, до якого підключений принтер.
- Виберіть порт і натисніть ОК. При цьому буде видане попередження, що містить список дій, які необхідно виконати перед початком установки.

- Виконайте всі вимоги, зазначені в попередженні й натисніть ОК. При цьому буде початий процес установки відновлення.

- При завершенні процесу установки буде видане відповідне повідомлення.

Режим автоматичного вмикання автомата.

Для того щоб настроїти автоматичне вмикання автомата після збоїв електрики в мережі, вам необхідно включити відповідну опцію в BIOS автомата.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.
 - Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.
 - Відладник Win 64 (на LLDB) і збирач для C++.
 - Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.
 - Підтримка Metal Driver GPU для macOS і iOS.
 - Вбудований Fmxlinux.
 - Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
 - Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).
 - Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.
 - Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services
 - У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey
- RAD Studio 10.4 Короткий огляд:
- Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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 – використовуючи класичну реалізацію керування пам'яттю об'єктів.

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

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

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладоочними 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.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

забезпечення, яке призначено для системи терміналів мережі платіжних автоматів.

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

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

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

Розробити функціональну та структурну схеми системи;

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Опис GSM/GPRS

Як було відмічено вище, для організації обміну інформацією між автоматом і віддаленим комп'ютером (сервером), використовується технологія бездротового зв'язку GPRS або GSM. Розглянемо ці технології.

Опис стандарту GSM

Стандарт GSM відноситься до другого покоління стандартів для стільникового зв'язку, заснованому на цифрових технологіях. Реалізоване в системах GSM повношвидкісне кодування мови дозволяє зробити її якість порівнянною з якістю стаціонарних телефонних мереж. Радіотелефон стандарту GSM можна умовно розділити на дві частини: абонентський модуль SIM (SIM-карта) і безпосередньо сам телефон, що містить апаратне й програмне забезпечення. SIM-карта служить підтвердженням дійсності абонента й містить у своїй пам'яті всі необхідні дані, пов'язані з повноваженнями конкретного абонента. Щоб викрадач не зміг нею скористатися, у неї вводять спеціальний ідентифікаційний номер (PIN-код). Використання SIM-карти також зручно тим, що при зміні апарата абонентові не потрібно міняти свій мобільний номер, він просто переставляє карту, і всі збережені на ній дані, включаючи записну книжку, стають доступними в новому апараті. Коли SIM-карти немає в апараті, доступ до абсолютної більшості послуг закритий, за винятком екстрених викликів (якщо дозволяє мережа). Виготовити дублікат SIM-карти дуже складно й у сукупності з функціями захисту, вона дає високий рівень захисту користувачів і мереж від несанкціонованого доступу.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Можливості GSM

У стандарті GSM уведено кілька функцій захисту. У першу чергу це шифрація радіоканалу, що виключає прослуховування третьою стороною, а також захист номера абонента (для запобігання розкриття його місцезнаходження). Крім стандартних можливостей, надаваних операторами стільникового зв'язку – місцевий, міжміський й міжнародний зв'язок, переадресація виклику й інших, телефони стандарту GSM дають своїм власникам ряд додаткових функцій: збереження мовних повідомлень, що надійшли в період, коли абонент був недоступний (голосова пошта), прийом повідомлення про факс, що прийшов (факс-пошта), визначення номера що дзвонить. Передбачено можливість передачі коротких повідомлень "із точки в точку" (пейджингу), тобто абоненти при бажанні можуть обмінюватися простими короткими (кілька десятків символів) повідомленнями (тарифи на цю послугу нижче, ніж на звичайні переговори). Функція мобільного модему/факсу поряд з повсюдним поширенням портативних комп'ютерів дає можливість доступу до Інтернету й електронної пошти через мережу GSM. Ці послуги значно збільшують привабливість використання телефонів GSM для користувачів. Так, приміром, факс-пошта може бути дуже корисна діловій людині, тому що дозволяє не пропустити інформацію про факс у будь-який час, незалежно від місцезнаходження абонента. Мобільний телефон сповістить свого власника, а той може одержати факс коли завгодно й де завгодно, тому що факс автоматично надходить у його електронну поштову скриньку.

Технологія GPRS

Нові можливості GPRS дозволили ввести принципово нові послуги, які раніше не були доступні. Насамперед це мобільний доступ до ресурсів Internet із задовольняючого споживача швидкістю й з дуже вигідною системою тарифікації. Приміром, за допомогою системи GPRS абонент може переглядати WEB-сторінки в Internet стільки, скільки йому необхідно, оскільки плата стягується тільки за обсяг прийнятої інформації, а не за час знаходження в мережі Internet.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

При введенні погодинної оплати на фіксованих телефонних лініях, тарифи на доступ в Internet з мобільного GPRS-телефону будуть ще більш конкурентоспроможні. Технологія GPRS дозволить швидко передавати й одержувати більші обсяги даних, відеозображення, музичні файли стандарту MP3 і іншу мультимедійну інформацію. Для корпоративних користувачів система GPRS може послужити відмінним інструментом для забезпечення безпечного й швидкого доступу співробітників до корпоративних мереж підприємств, до поштових, інформаційних серверів, віддаленим базам даних. Технології GPRS може застосовуватися в системах телеметрії: пристрій може бути увесь час підключений, не займаючи при цьому окремих канал. Така послуга може бути затребувана службами охорони, банками для підключення банкоматів і в інших областях, у тому числі й промислових.

Мережі GPRS-GSM

Щоб підтримувати GPRS, мережі GSM повинні бути доповнені двома основними функціональними елементами: Serving GPRS Support Node (SGSN) – вузол, яким грає в мережі GPRS ту ж роль, що й комутатор MSC і гостьовий регістр VLR у мережі GSM: і Gateway GPRS Support Node (GGSN) – еквівалент блоку IWF (interworking unit) у мережі GSM:

- SGSN – відповідає за доступ абонентів у мережу GPRS (автентифікацію), а також моніторинг абонентів і трафіка. Він служить для автентифікації встаткування й фіксує всі «дії» мобільного абонента в мережі.
- GGSN служить гейтом, тобто, з'єднує мережу GPRS і зовнішні мережі загальнодоступні або частні.

Втім, говорити про два основні елементи можна лише з деякою натяжкою, оскільки кожен контролер у мережі GSM повинен бути дообладнаний пристроєм PCU (блок керування пакетним зв'язком).

Додамо, що в мережі GSM-GPRS з'являється кілька нових інтерфейсів, кожний із префіксом «G».

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

- Gp – це протокол Frame Relay, однак, перехід до технології UMTS дозволить впровадити замість Frame Relay протокол АТМ (Asynchronous Transfer Mode) – це, до речі, викличе перехід до комутаторів АТМ від сьогоднішніх комутаторів MSC.

- Gr – протокол з комутацією з'єднань типу С7.

- Gn – протокол пакетної передачі типу TCP/IP (Transmission Control Protocol і Internet Protocol).

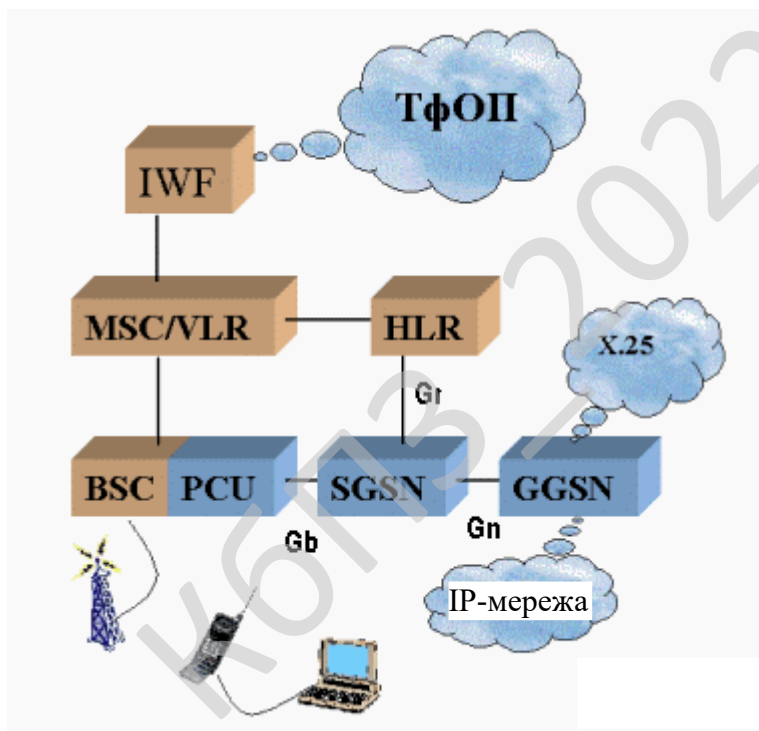


Рисунок 3.1 – Структура мережі GSM-GPRS

Підключення по GPRS

Набір додатків по оцінках фахівців може включати:

- e-mail;
- доступ до Internet;
- телеметрію;
- підтримка протоколу бездротових додатків (WAP);
- підтримка протоколу передачі файлів (FTP).

Електронна пошта – це послуга, що забезпечує абонентів стільникової мережі (клієнтів) можливістю одержувати й відсилати повідомлення по мережі

Internet без необхідності доступу до комп'ютера. Кожний, хто має адресу в Internet, зможе відіслати повідомлення будь-кому, чия електронна адреса вам відомий. Джерелом адреси для вас може бути ISP (Internet Service Provider – провайдер) або оператор стільникової мережі, якщо він візьме на себе ці функції. У технічному плані – це комп'ютер, підключений до Internet, що буде зберігати отримані на ваше ім'я повідомлення поки ви їх не заберете (наприклад, до пам'яті телефону або вашого домашнього комп'ютера).

Доступ до Internet. Web – це розподілена інформаційна система, заснована на технології сервер-клієнт. Клієнтська програма, це ваш браузер (Internet Explorer, Netscape, Linux або який-небудь інший). Ви можете підключитися браузером до сайту, де зберігаються сторінки, що цікавлять вас, включаючи текст, графіку, а також мультимедіа (відеофрагменти, музика).

Телеметрія. Віддалений моніторинг сайтів в Internet або якого-небудь устаткування – створені нібито спеціально для GPRS, оскільки «підривний» характер передачі даних надзвичайно вдало вписується в технологію пакетного зв'язку GPRS.

WAP. Протокол Бездротових Додатків – WAP зараз рекламується як якийсь прорив, що дозволить забезпечити тотальний доступ до інтернет-ресурсам з мобільних пристроїв, насамперед телефонів. Число WAP-сумісних ресурсів в Internet зараз росте швидкими темпами, з'явилися й перші українські ресурси.

FTP. Це сервіс, призначений для копіювання файлів з одного комп'ютера на інший. Може використовуватися в торговельних й бізнес-додатках.

Стандарти GPRS-терміналів

Клас А. Термінал дозволяє одночасно здійснювати передачу мови й даних у режимі GPRS

Клас В. Термінал підтримує й голосове з'єднання, і передачу даних у пакетному режимі (GPRS), але або режими використовуються не одночасно (під час передачі даних через GPRS абонент не може робити й приймати голосові дзвінки й навпаки).

						ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			32

майже в 12 разів швидше роботи передачі даних у звичайних мережах GSM (9,6 Кбіт/с). Уже сьогодні доступна швидкість до 33 Кбіт/с.

EDGE

EDGE (Enhanced Data rates for Global Evolution) – у перекладі з англійського – це вдосконалена технологія передачі даних для глобального розвитку, що забезпечує високошвидкісну передачу великих обсягів інформації, характерну для мереж третього покоління мобільного зв'язку.

Технологія EDGE надається абонентам на базі послуги GPRS. Технологія EDGE забезпечує швидкість передачі даних, у середньому в три рази перевищуючу можливості GPRS, а також – більш ефективне використання частотних ресурсів і поліпшення покриття мережі в порівнянні зі звичайною мережею стандарту GSM. Технічні особливості EDGE дозволять абонентам більш оперативно працювати з Інтернет-ресурсами (web, wap, e-mail, ICQ) і користуватися найширшим спектром неголосових послуг, що вимагають високих швидкостей передачі даних. Виявившись у зоні дії базової станції з підтримкою EDGE, телефон абонента буде автоматично використовувати EDGE замість традиційного GPRS. Швидкість з'єднання залежить від кількості одночасних користувачів, класу телефону з підтримкою EDGE, умов радіоприйома й співвідношення сигнал/шум. Середня швидкість у мережі 80-120 Кбіт/с.

Опис програмного забезпечення

Програмне забезпечення повинне забезпечувати наступні функції:

1. Первісне налаштування.
2. Сервісне обслуговування автоматів.
3. Установка й налаштування таймера.
4. Налаштування додаткових параметрів через командний рядок.
5. Моніторинг функціонування терміналів.
6. Робочий режим.

Для коректної роботи ПЗ необхідно, щоб на автоматі самообслуговування було попередньо встановлене наступне програмне забезпечення:

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

- Операційна система Windows 10/11.
- MS Internet Explorer версії 5 або вище.

Крім того, перед початком роботи треба переконатися, що в операційній системі автомата настроєне з'єднання з Інтернет. У якості Інтернет-з'єднання може використовуватися як канал GPRS, так і будь-який інший доступний спосіб з'єднання. Для доступу в Інтернет ПЗ використовує налаштування Internet Explorer, тому у випадку використання проху-сервера, при підключенні автомата в локальну мережу, необхідно додати проху у налаштування «Internet Explorer».

Розглянемо перераховані вище функції більш докладно.

1. Первісне налаштування.

При першому запуску системи вам необхідно настроїти дані для авторизації автомата в системі оплати послуг мобільних операторів. Спершу треба перейти в параметри авторизації, тобто ввести значення параметрів:

- Номер терміналу – ідентифікатор терміналу в системі оплати послуг оператора мобільного зв'язку (ТПМА). У модулі адміністратора системи ТПМА повинен бути зареєстрований термінал, під яким будуть проводитися платежі. Терміналу повинен бути заданий тип Автомат ТПМА. Неприпустиме використання того самого номера терміналу декількома пристроями. Для кожного автомата, реєструйте в системі новий термінал:

- Ім'я користувача – логін користувача, від імені якого будуть відбуватися платежі.

- Пароль – пароль користувача, від імені якого будуть відбуватися платежі.

- Підтвердження пароля – у дане поле необхідно повторно ввести пароль користувача для підтвердження його коректності.

У якості логіна й пароля повинні використовуватися дані персони, зареєстрованої в системі ТПМА із правами «Продавець». Припустимо використання однакових даних користувача в різних автоматах.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

автомата й розблокувати сервісний режим за допомогою імені користувача й пароля.

Сервісне меню. Містить такі дані як:

- Версія інтерфейсу.
- Кількість купюр у купюроприймачі й суму.
- Статус купюроприймача.
- Статус принтера.
- Статус з'єднання.
- Статус сторожового таймера.
- Статус платежів.

За допомогою кнопок у вікні сервісного меню можна виконувати також наступні дії:

- Змінити номер терміналу, логін і пароль ТПМА.
- Змінити параметри входу в сервісний режим.
- Змінити параметри Інтернет і опції автомата.
- Налаштування інтерфейсу.
- Налаштування e-mail оповіщень.
- Налаштування Multi SIM підключення.
- Видалити файл конфігурації й запустити знову програму.
- Переглянути лог.
- Запустити відновлення.
- Видалити файл конфігурації й запустити знову програму.
- Вийти й запустити знову програму.
- Настроїти параметри безпеки при роботі з додатком.

Крім цього, дозволяє виконувати й контролювати наступні дії:

- Проведення інкасації.
- Налаштування параметрів принтера.
- Рівень сигналу GSM мережі.
- Налаштування одержання балансу.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- Налаштування оповіщень.
- Налаштування параметрів сторожового таймера.
- Налаштування інтерфейсу.
- Режим підтримки двох SIM-карт.
- Безпека.
- Режим блокування системних комбінацій клавіатури.

Зміна провайдеру GPRS зв'язку. Призначено для введення параметрів передачі даних по мережі GSM, операторів, які не прописані в автоматі ТПМА.

Мультиканальний режим роботи модему. Даний режим дозволяє здійснювати одночасно наступні дії:

- передача даних по Інтернет;
- відправлення SMS;
- одержання рівня сигналу й інших даних про стан GSM мережі;
- відправлення USSD запитів балансу SIM карти.

Таким чином, використання даного режиму дозволяє, не розриваючи модемне з'єднання з Інтернет, одержувати безупинно дані про баланс SIM карти, коливання рівня сигналу й дані про стан GSM мережі. Підключення мультиканального режиму відбувається в кілька етапів:

- Установка драйвера для мультиканального режиму.
- Установка стандартного драйвера модему.
- Створення модемного з'єднання.
- Налаштування ПЗ автомата.

3. Установка й налаштування таймера.

Таймер дозволяє здійснювати дві основні функції:

- Перезавантаження модему, при відсутності відгуку від модему протягом деякого часу. Даний режим роботи включений за замовчуванням.
- Перезавантаження комп'ютера при відсутності сигналу від комп'ютера протягом 30 минут.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Таким чином, за допомогою плати відслідковується «зависання» комп'ютера або модему й коректуються дані проблеми шляхом перезавантаження модему або комп'ютера. Період в 30 мінут для перезавантаження комп'ютера обраний з метою виключення ситуацій помилкового спрацьовування: при відновленні даних із сервера й т.п.

4. Налаштування додаткових параметрів через командний рядок.

Зо допомогою командного рядка ви можете вказати параметри, відмінні від параметрів за замовчуванням для принтера й купюроприймача. Вони підрозділяються на:

- Параметри принтера й купюроприймача.
- Додаткові параметри.

5. Моніторинг функціонування терміналів.

Для зручності відстеження роботи ваших автоматів самообслуговування, у системі ТПМА передбачена можливість віддаленого моніторингу в реальному часі. Дана функціональність доступна користувачам із правами головного менеджера, і містить у собі:

- Моніторинг через Інтернет.
- Моніторинг за допомогою мобільного телефону.

6. Робочий режим.

Його робота полягає у виконанні трьох основних функцій:

- Прийом платежів. Інтерфейс для прийому платежів розроблений таким чином, щоб максимально спростити процес звернення клієнта до автомату і зробити прийом платежів за допомогою автомата інтуїтивно зрозумілим без додаткових інструкцій.
- Відновлення ПЗ. Після виходу нової версії ПЗ, автомат автоматично завантажує відновлення, не припиняючи роботу. Також автоматично відбувається відновлення зовнішнього вигляду інтерфейсу автомата: додавання можливості оплати послуг нових провайдерів і т.п.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

- Режим автоматичного включення автомата. Призначений для того, щоб настроїти автоматичне включення автомата після збоїв електрики в мережі.

3.2 Розробка структурної схеми

Створене програмне забезпечення призначене для забезпечення користувачів послугою поповнення рахунку мобільного телефону. Воно універсальне, може бути використана терміналах фірми «Об'єднана система швидких платежів» серії: ОСМП-2; ОСМП-МІНІ; ОСМП-ВУЛИЦЯ; ОСМП 2 з лайт-боксом; ОСМП 2 з двома моніторами. На рисунку 3.2 показана структурна схема програмного забезпечення, розглянемо її зверху долилиць.



Рисунок 3.2 – Структурна схема програмної частини

Коли на термінал надходить запит обслуговування, невідомо користувач є адміністратором чи ні. Для проходження автентифікації крім магнітного ключа адміністраторові необхідно знати системний 9 значний номер телефону перекладу терміналу в інтерфейс адміністратора.

Коли магнітний ключ приєднаний до терміналу, й користувач ввів системний номер телефону, на екрані з'являється форма автентифікації, що запитує ім'я користувача й пароль, при правильному введенні даних користувачеві привласнюється статус адміністратора й автомат переходить у режим налаштування, у протилежному випадку користувачеві дається ще дві спроби після відбування, яких відбувається включення тривоги, передача сигналу на базовий ПК і повне блокування терміналу.

Адміністратор повністю управляє комп'ютером, він може управляти всіма налаштуваннями й опціями терміналу: статус з'єднання; встановлення рядка ініціалізації модему; встановлення архівації та кодування даних; кількість купюр у купюроприймачі й суму; статус купюроприймача; статус принтера; статус сторожового таймера; статус платежів; налаштування інтерфейсу; налаштування e-mail оповіщень; налаштування Multi SIM підключення; видалення файлу конфігурації й перезапуск терміналу.

Також за допомогою кнопок у вікні сервісного меню можна управляти наступними опціями: зупиняти авт. при помилках купюроприймача; зупиняти автомат при помилках принтера; зупиняти автомат при відсутності засобів на рахунку користувача; зупиняти автомат при відсутності зв'язку; змінювати номер терміналу, логін і пароль; змінити параметри входу в сервісний режим; переглядати лог. файл.

Також адміністратор може зробити інкасацію коштів. Виконання даної операції здійснюється через комп'ютерний відсік шляхом зняття грошової касети із кріплення купюроприймача, попередньо запустивши процес інкасації в програмному забезпеченні терміналу.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Якщо системний номер адміністратора не був введений, термінал переходить в інтерфейс користувача, перемикається на режим обслуговування клієнтів, і користувач, натискаючи на сенсорній панелі кнопки, вибирає необхідні розділи, після чого відбувається видача чека з поповненням рахунку мобільного телефону.

На рисунку 3.3 можна побачити складові частини терміналу, а саме: TFT 17" вандалостійкий сенсорний монітор; Вандалостійкий корпус; IBM PC сумісний комп'ютер; Пристрій для печатки бланків; Пристрій для прийому грошей.

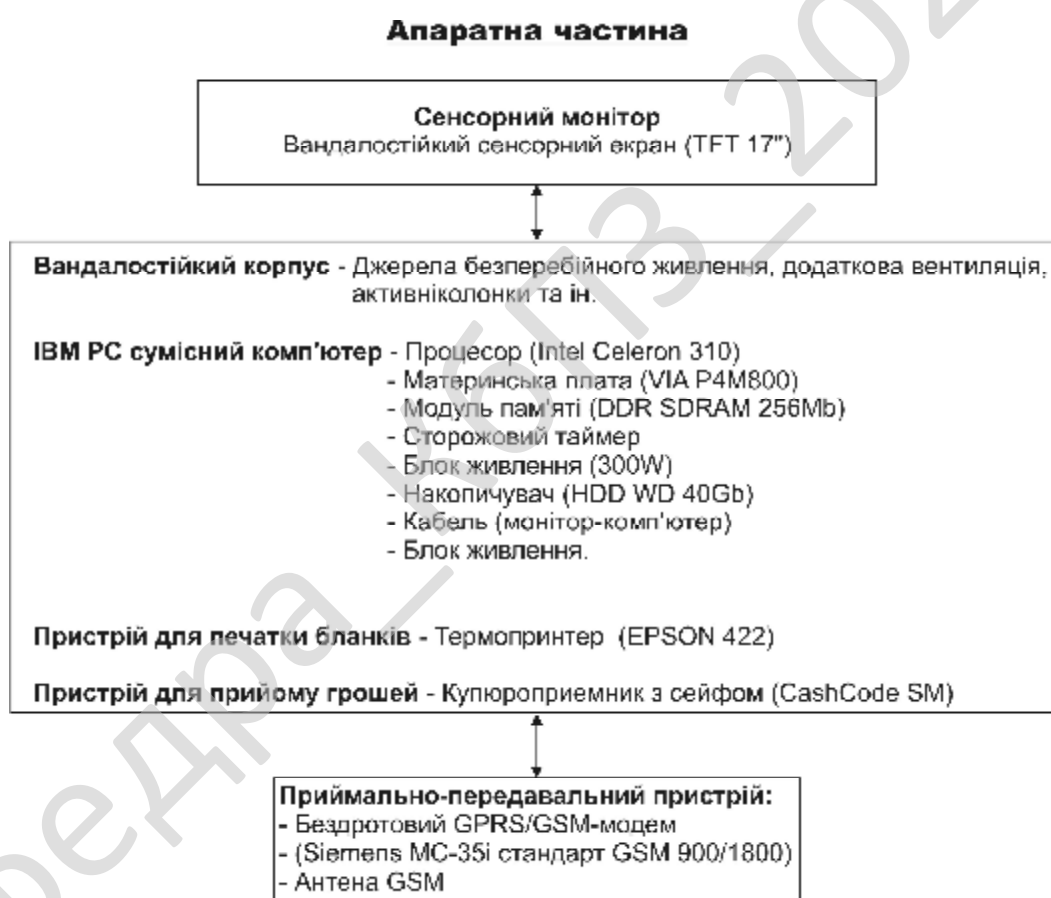


Рисунок 3.3 – Структурна схема апаратної частини

Розглянемо докладніше основні елементи терміналу. Серце терміналу комп'ютерний відсік, що знаходиться у вандалостійкому корпусі. Являє собою

комп'ютерну частину терміналу й вузол об'єднання всіх пристроїв у єдину систему. Включає у свій состав IBM PC сумісний комп'ютер, розташований у верхній частині терміналу (рисунок 3.4), що складається з наступних складових елементів:

- Процесор (Intel Celeron 310);
- Материнська плата (VIA P4M800);
- Модуль пам'яті (DDR SDRAM 256Mb);
- Сторожовий таймер;
- Блок живлення (300W);
- Накопичувач (HDD WD 40Gb);
- Кабель (монітор-комп'ютер)
- Блок живлення.

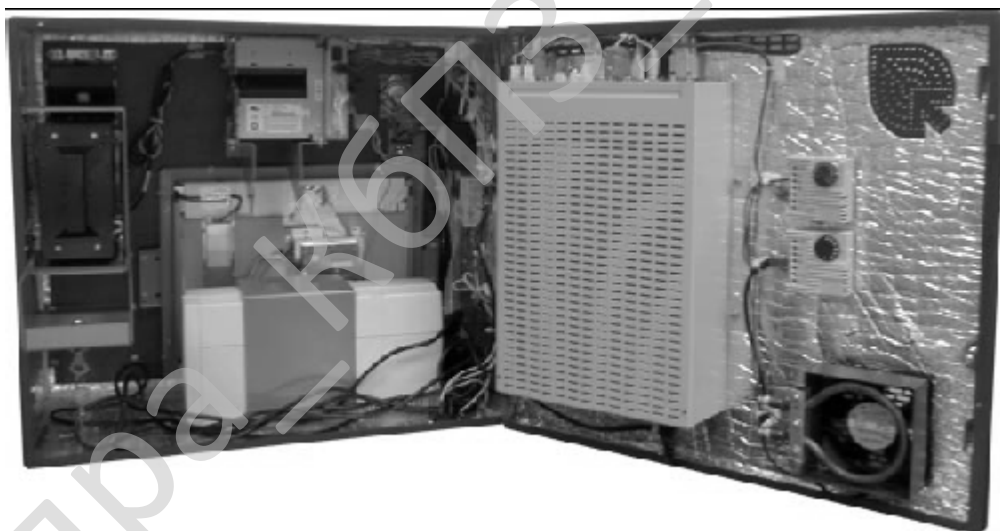


Рисунок 3.4 – Комп'ютерний відсік

Як видно з основних характеристик комп'ютерного відсіку комп'ютер повністю сполучимий з настільним ПК, у ньому встановлена операційна система Windows 10/11 (залежно від конфігурації складових елементів) з відповідними перевагами й недоліками. Всі вузли терміналу (сенсорний монітор, принтер, GPRS/GSM модем, і т.д.) підключаються через стандартні роз'єми ПК. Основна

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

відмінність від настільного ПК є форма корпусу, який поміщено у вандалостійкий корпус (рисунок 3.4).

Для програмування такого терміналу, по розрахункам фірми виробника (апаратної частини) ідеальним є мова програмування високого рівня Borland Delphi. Так як він дозволяє гнучко й швидко змінювати інтерфейс терміналу й по багатьом інших причинах. Усі компоненти для взаємодії із сенсорною панеллю розроблені для даної мови.

3.3 Розробка функціональної схеми

На рисунку 3.5 показана функціональна схема роботи системи.



Рисунок 3.5 – Функціональна схема роботи системи

У ній можна чітко простежити основні компоненти терміналу. Як видно з рисунка користувач сприймає термінал як 3 основні компоненти: сенсорний екран; купюроприймач; монетоприймач (є не у всіх версіях). Адміністраторові доступні всі налаштування й дії стосовно автомата при наявності спеціального ключа й знання спеціального системного номера, що вводить апарат у режим налаштування.

Кінцева дія апарату видача чека поповнення мобільного рахунку є одним з найважливіших компонентів терміналу. Чек є юридичним папером і при виникненні проблем з ним можна звертатися в різні інстанції.

3.4 Розробка діаграми процесів

Розглянемо діаграму процесів (рисунок 3.6), на діаграмі процесів можна точно зрозуміти, як працює ПЗ у цілому. Одним з важливих критеріїв при розробці будь-якого програмного забезпечення це грамотна розробка структури роботи системи потоків і процесів.

Починається й закінчується програма в першому блоці що є основною точкою звіту в діаграмі, при переміщенні по стрілках можна побачити загальну схему взаємодії блоків і їхнього входження друг у друга.

Основний блок терміналу, так як він працює із грошима, є блок захисту терміналу, через який проходять всі дії. Він взаємодіє із блоком перевірки GPRS/GSM – модему. У Кропивницькому можна побачити багато терміналів поповнення мобільних послуг на яких написано – “вибачите термінал тимчасово не працює”. Це означає тільки одне пропав зв'язок з основним ПК. Саме в модулі терміналу як було описано вище відбувається перевірка прав доступу адміністратора або вивід стандартного інтерфейсу користувача.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46



Рисунок 3.6 – 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

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована, проект, що розробляється, вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулям:

- обробки запитів користувача;
- обробки запитів адміністратора;
- обробки запитів з базової станції за допомогою GPRS/GSM – модему;
- Перевірки доступу до модему.

Створення повноцінного коду програмного продукту з розробленою ціновою політикою і системою знижок покупцям програмної продукції.

На рисунку 4.1 зображена основна блок-схема підпрограми. При детальному розгляді програма розбита на дещо важливих блоків.

Розглянемо складові елементи блок-схеми та їхній опис:

Початок (термінатор)

Початковий блок, вхід із зовнішнього середовища операційної системи. Програма забезпечує роботу термінальної консолі. Яка являє собою материнську плату комп'ютера розмішену у вандалостійкому корпусі з сенсорним монітором.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

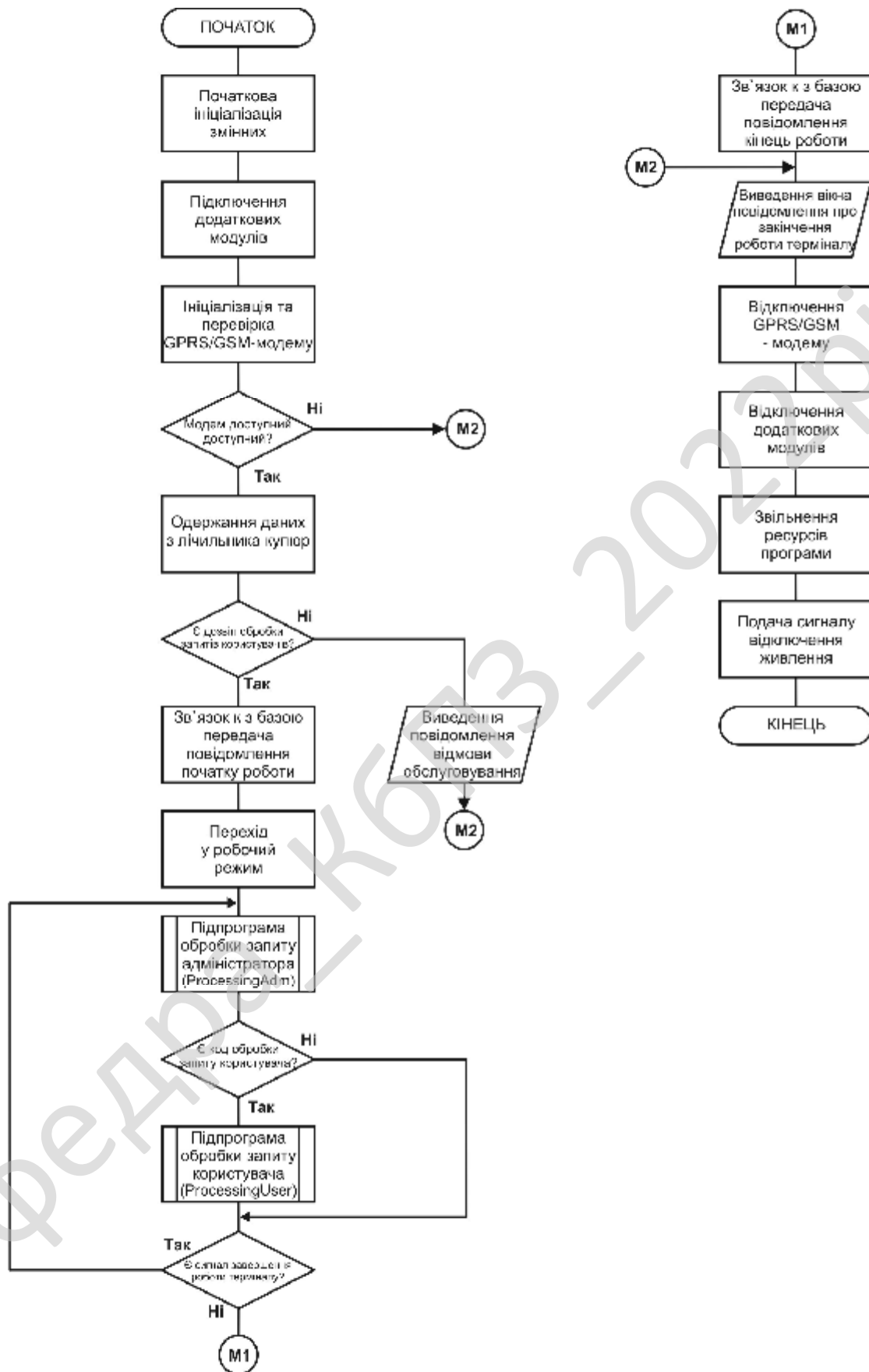


Рисунок 4.1 – Блок схема основної програми

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.22.0002.00.00.ПЗ

Арк.

49

Початкова ініціалізація змінних (процес)

Первісний блок у будь-якій блок-схемі, що вказує на початкову стадію запуску програми – виділення пам'яті програмі операційною системою.

Підключення додаткових модулів (процес)

Підключення додаткових модулів забезпечуючих правильну роботу програмного забезпечення.

Ініціалізація та перевірка GPRS/GSM-модему (процес)

Встановлення рядка ініціалізації модему, спроба з'єднання з головним ПК.

Модем доступний?(рішення)

Входи

Вхід – Результати спроба з'єднання з головним ПК.

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід по мітці M2 до завершення роботи терміналу.

Одержання даних з лічильника купюр (процес)

За допомогою COM1 з'єднаємося з лічильником купюр та обновляємо перемінні.

Є дозвіл обробки запитів користувачів?(рішення)

Входи

Вхід – Результати з лічильника купюр, є дозвіл обробки запитів?

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – Виведення повідомлення відмови обслуговування (введення/виведення даних), перехід по мітці M2 до завершення роботи терміналу.

Зв'язок к з базою передача повідомлення початку роботи (процес)

Зв'язок к з базою та сповіщення про вдалий запуск і початок обробки запитів. Повернення поточної дати та часу.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Перехід у робочий режим (процес)

Встановлення інформації поточної дати та часу та перехід у робочий режим.

Підпрограма обробки запиту адміністратора ProcessingAdm (Визначений процес, перехід на другу блок-схему, рисунок 4.7)

Перехід на другу блок-схему, обробка та аутентифікація адміністратора.

Отримання потокової інформації терміналу (процес)

Отримання даних з вандалостійкого сенсорного екрану.

Очікування дій користувача (процес)

Очікування дії з боку користувача по проведенню операції введення номеру телефону.

Введений секретний номер телефону поповнення?(рішення)

Входи

Вхід – Перевірка введення 8 знакового системного номеру адміністратора.

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід у кінець підпрограми, повернення до основної блок-схеми програми.

Виведення інтерфейсу адміністратора, запит аутентифікації (введення/виведення даних)

При вдалому введенні системного номеру проходить запит логіну та паролю адміністратора.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Підпрограма обробки запиту адміністратора
(ProcessingAdministrator)

Підпрограма обробки запиту користувача
(ProcessingUser)

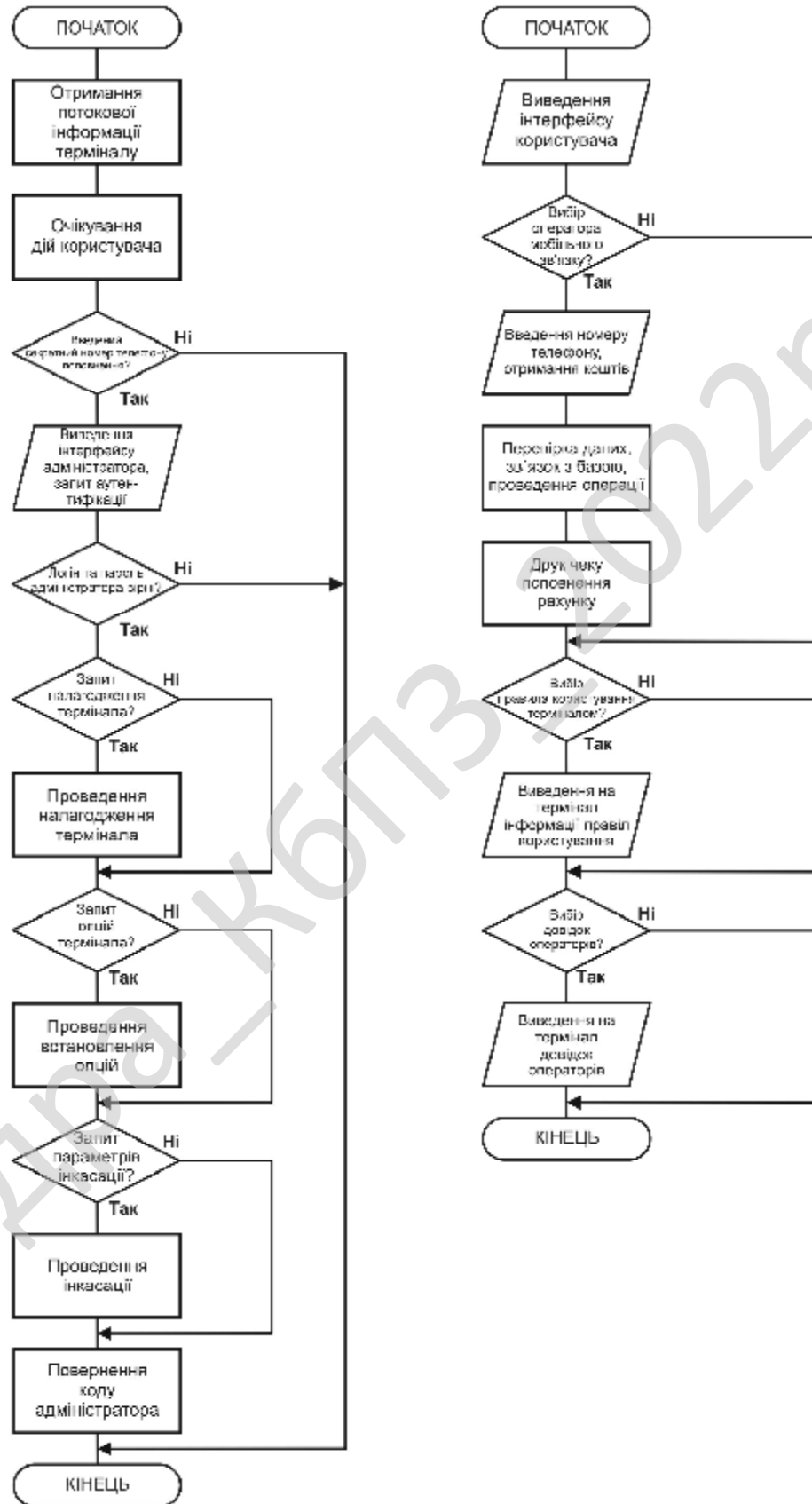


Рисунок 4.2 – Блок схема підпрограм

Логін та пароль адміністратора вірні?(рішення)

Входи

Вхід – Перевірка введених даних, аутентифікація адміністратора. Логін та пароль адміністратора вірні?.

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід у кінець підпрограми, повернення до основної блок-схеми програми.

Запит налагодження термінала? (рішення)

Входи

Вхід – Перевірка запиту налагодження терміналу (дія адміністратора). Є запит налагодження термінала?

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід у блок слідуєчого запиту (рішення).

Проведення налагодження термінала (процес)

Налагодження параметрів термінала: умови з'єднання з Інтернет; встановлення рядка ініціалізації модему; встановлення умов архівації та кодування даних; та інші параметри.

Запит опцій термінала?(рішення)

Входи

Вхід – Перевірка запиту встановлення опцій термінала. Є запит опцій термінала?

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід у блок слідуєчого запиту.

Проведення встановлення опцій (процес)

Встановлення опцій термінала: -зупиняти автомат при помилках купюроприймача зупиняти автомат при помилках принтера; зупиняти автомат

							ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата				53

при відсутності засобів на рахунку користувача; зупиняти автомат при відсутності зв'язку; та інші параметри.

Запит параметрів інкасації?(рішення)

Входи

Вхід – Запит параметрів інкасації. Банківської операції збору у касах підприємств, організацій, установ готівки й цінностей і доставка їх у каси кредитних установ. Є запит параметрів інкасації?

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід у кінець підпрограми до блоку повернення коду адміністратора.

Проведення інкасації (процес)

Проведення налагодження та процесу інкасації (знімання коштів).

Повернення коду адміністратора (процес, перехід на першу блок-схему)

Встановлення коду доступу адміністратора. Нехтування інтерфейсом користувача.

Є код обробки запиту користувача? (рішення, перша блок-схема)

Перевірка коду обробки, обробка дій виключно користувача.

Підпрограма обробки запиту користувача ProcessingUser (Визначений процес перехід на другу блок-схему, рисунок 4.2)

Перехід на другу блок-схему, обробка дій користувача.

Виведення інтерфейсу користувача (введення/виведення даних)

Виведення на сенсорному моніторі меню користувача: вибір оператора мобільного зв'язку; вибір правила користування терміналом; вибір довідок операторів.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Вибір оператора мобільного зв'язку?(рішення)

Входи

Вхід – Вибрав користувач на сенсорному екрані підменю вибору оператора мобільного зв'язку. Є запит вибору оператора мобільного зв'язку?

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід у блок слідуєчого запиту.

Введення номеру телефону, отримання коштів (введення/виведення даних)

Введення користувачем стільникового номеру якому необхідно перевести гроші.

Перевірка даних, зв'язок з базою, проведення операції (процес)

Зв'язок з головним ПК та перевірка введеного стільникового номеру телефону. Отримання грошей користувача. Формування чеку з кодом поповнення.

Друк чеку поповнення рахунку (процес)

Друк форми звітності з кодом поповнення, датою та часом проведення операції поповнення рахунку.

Вибір правила користування терміналом?(рішення)

Входи

Вхід – Вибрав користувач на сенсорному екрані підменю вибору правила користування терміналом. Є запит вибору правила користування терміналом?

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід у блок слідуєчого запиту (рішення).

Виведення на термінал інформації правіл користування (введення/виведення даних)

Виведення на термінал даних по використанню терміналу та дій при виникненні непередбачених ситуацій.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Вибір довідок операторів?(рішення)

Входи

Вхід – Вибрав користувач на сенсорному екрані підменю довідки операторів стільникового зв'язку (МТС, Jeans, Kievstar, і.т.і.). Є запит вибору довідок операторів?

Виходи

Вихід так – перехід у наступний блок.

Вихід ні – перехід на кінець підпрограми.

Виведення на термінал довідок операторів (введення/виведення даних)

Виведення інформації по поповненню рахунків стільникового зв'язку різних операторів.

Є сигнал завершення роботи терміналу?(рішення)

Перевірка встановлення адміністратором сигналу завершення роботи, сигналу з купюроприймача та модему.

Зв'язок з базою передача повідомлення кінець роботи (процес)

Передача повідомлення завершення роботи терміналу, передача дати та часу завершення.

Виведення вікна повідомлення про закінчення роботи терміналу (введення/виведення даних)

Сповіднення користувача чи адміністратора.

Відключення GPRS/GSM – модему (процес)

Проведення дій по відключенню модема.

Відключення додаткових модулів (процес)

Відключення інших засобів.

Звільнення ресурсів програми (процес)

Коректне завершення роботи програми. Запуск Vat файлу.

Подача сигналу відключення живлення (процес)

Виконання Vat файлу завершення роботи операційної системи.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Кінець (термінатор)

Кінцевий блок. Завершення роботи програмного забезпечення.

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7624:2014 («Калина»). При розробці національного стандарту ДСТУ 7624:2014 (блоковий шифр «Калина» і режими його роботи) було ухвалене рішення забезпечити прозорість проектування й використовувати консервативний підхід із застосуванням добре досліджених конструкцій, що забезпечують запас стійкості для безпечного застосування алгоритму в умовах істотного прогресу криптоаналітичних технік і засобів обробки даних.

Національний стандарт підтримує розмір блоку й довжину ключа шифрування 128, 256 і 512 біт (довжина ключа дорівнює розміру блоку або у два рази перевищує його), забезпечуючи нормальний, високий і надвисокий рівень стійкості (зараз це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі). Різні варіанти забезпечують гнучкість вибору параметрів для розроблювачів систем криптографічного захисту, що дозволяє одержати як найвищий рівень швидкодії, так і найбільший запас стійкості перетворення.

Високорівнева конструкція використовує добре досліджену Square-подібну SPN-структуру, застосовувану в алгоритмах AES/Rijndael, Whirlpool, «Стрибог», «Коник» і багатьох інших. Циклове перетворення побудоване на базі таблиць підстановки (S-блоків) і множення на МДР-матрицю над кінцевим полем, забезпечуючи необхідні криптографічні властивості. Застосування саме такої конструкції дозволяє забезпечити доказову стійкість до диференціального, лінійному й ін. видам криптоаналізу, одночасно забезпечуючи ефективну реалізацію на широкому спектрі програмних і програмно-апаратних платформ. При виборі розміру МДР-матриці був прийнятий в увагу розмір кешу L1

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

сучасних і перспективних процесорів, що дозволило оптимізувати швидкодія програмної реалізації шифру.

Альтернативний варіант, який розглядався при розробці циклової функції, – ARX перетворення (Addition-rotation-xor). Цей підхід реалізований у шифрі SPECK (розроблений Агентством національної безпеки США й переданий у відкритий доступ), у блокових алгоритмах, на основі яких побудовано сімейство геш-функцій SHA-0,1,2 і ін. Перевагою походу є компактність і швидкодія перетворення. Але, у той же час, є й істотний недолік, пов'язаний з відсутністю методів, що дозволяють виконати строге аналітичне обґрунтування криптографічної стійкості таких розв'язків. Навіть із наймогутнішими у світі можливостями для аналізу, США кілька раз були змушено модифікувати свої стандарти гешування через знайдені уразливості: з 1993 по 1995 рр. діяв SHA-0, з 1995 по 2001 рр. застосовувався SHA-1, з тих пор використовується SHA-2. Через питання, що піднімаються, до стійкості й цієї версії, у США з 2008 по 2012 рр. був проведений міжнародний конкурс SHA-3. До теперішнього часу розроблений проект стандарту FIPS 202, що описує нову криптографічну геш-функцію.

Таким чином, консервативний і прозорий підхід до проектування нового національного стандарту України обумовив вибір добре перевіреної конструкції на базі S-блоків і лінійного перетворення, для якої можливо забезпечити доказову стійкість до різних видів криптоаналізу.

При порівнянні характеристик S-блоків і інших симетричних перетворень, у т.ч. нових білоруських і російських стандартів, можна відзначити, що саме національний стандарт України забезпечує найбільшу нелінійність булевих функцій, що дає додатковий запас стійкості до лінійного криптоаналізу. Ще більше значення нелінійності для взаємооднозначної підстановки можна забезпечити, застосовуючи, наприклад, афінно-тквівалентні статечні функції в кінцевому полі, але такі перетворення, використані в AES, Camellia і ін. алгоритмах, ставлять шифр під погрозу реалізації алгебраїчної атаки (цей метод

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

блоковими й потоковими шифрами забезпечують найбільшу нелінійність булевих функцій (104), що дає додатковий запас стійкості перетворення;

– збільшений розмір МДР-перетворення, що поліпшує криптографічні властивості й дозволяє оптимізувати швидкодія на сучасних 64-бітових платформах;

– нову односпрямовану схему формування циклових ключів, що забезпечує захист від атак, ефективність програмної й програмно-апаратної реалізації, разом з додатковою стійкістю до методів аналізу спеціального виду.

Оцінка криптографічної стійкості до диференціального, лінійному, алгебраїчному, інтегральному й іншим методам аналізу (практичний критерій) показала, що шифр є стійким при 6 циклах для 128-бітового блоку, 7 циклах для 256-бітового й 9 циклах для 512-бітового (кожний додатковий цикл забезпечує експонентний ріст складності криптоаналізу). Таким чином, шифр, що містить 10, 14 і 18 циклів для блоку 128, 256 і 512 біт відповідно, забезпечує захист від розглянутих видів аналізу й має істотний запас стійкості.

Крім блокового шифру, ДСТУ 7624:2014 визначає режими роботи, що відповідають як ISO 10116:2006, так і додаткові, призначені для сучасних систем криптографічного захисту IP-трафіка, прозорого шифрування носіїв інформації й ін. У стандарті визначені обсяги повідомлень, після обробки яких потрібна обов'язкова зміна ключа. Крім того, приводяться рекомендації розроблювачам, що обертають увагу на необхідність запобігання атак з використанням особливостей реалізації засобів шифрування. Зокрема, враховані особливості, що дозволяли організацію атак BEAST і CRIME/BREACH у протоколах SSL/TLS, повторне приймання повідомлення, відновлення конфіденційних параметрів на основі залежності часу шифрування від оброблюваних даних (через промахи кешу при табличній реалізації) і ін.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

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

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

- перевірити роботу терміналу під наглядом декілька місяців (тестова робота.)
- перевірити проведення інкасації з терміналу (зняття всіх коштів)
- при встановленні на місце перевірити зону покриття для доступу до головного ПК.

В умовах магістерської роботи не повністю було проведено тестування терміналу на це потрібно багато часу.



Рисунок 5.1 – Загальний вигляд терміналу.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

На рисунку 5.1 зображено загальний вигляд термінала. Розглянемо роботу програми терміналу у тестовому режимі.

На рисунку 5.2 зображене головне вікно яке є головним вікном користувача з неї користувач може здійснити перехід на дочірні вікна, адміністратор здійснити введення системного номеру переходу до інтерфейсу адміністратора.

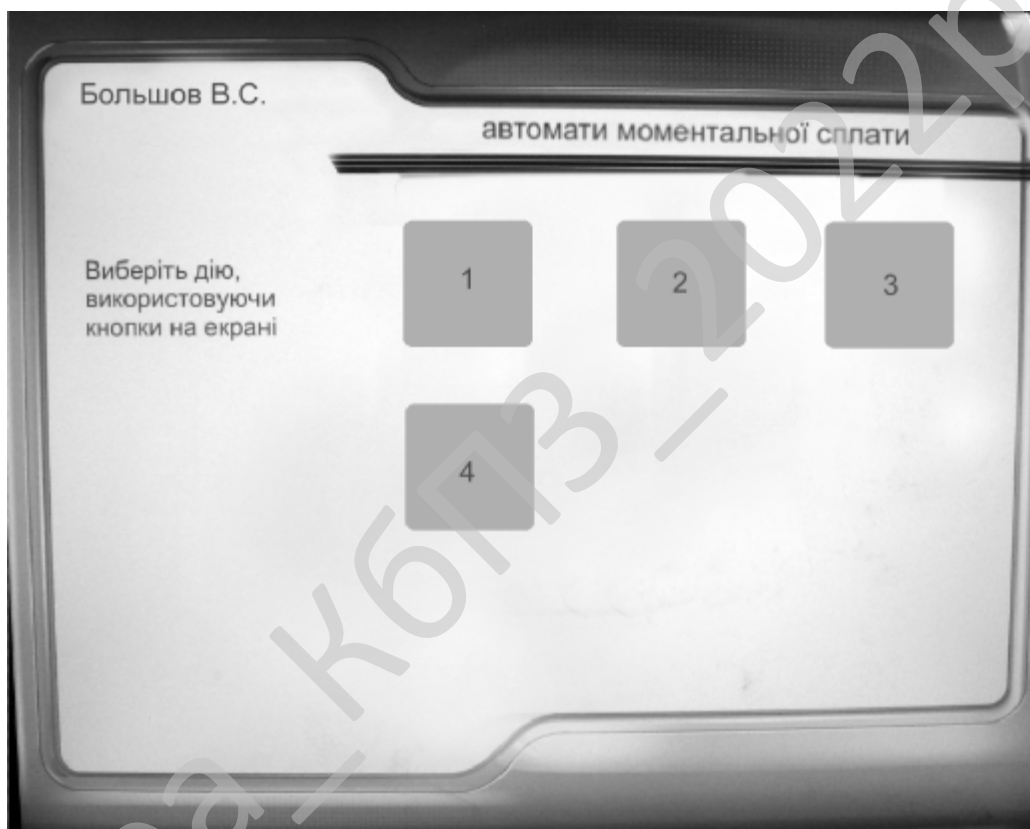


Рисунок 5.2 – Головне вікно терміналу

При натисканні користувачем на сенсорний екран програма прораховує координати натискання й виконує відповідні дії.

На рисунку 5.1. наведені тестові об'єкти сплати – об'єкт 1 – об'єкт 7.

Якщо користувач нажав у довільному місці екрана, програма не виконує ніяких дій.

При натисканні користувачем у прямокутній області певного оператора відбувається перехід у підменю відповідного оператора.

На рисунку 5.3 зображено тестове підменю – 10, 20, 100, 200, 500.

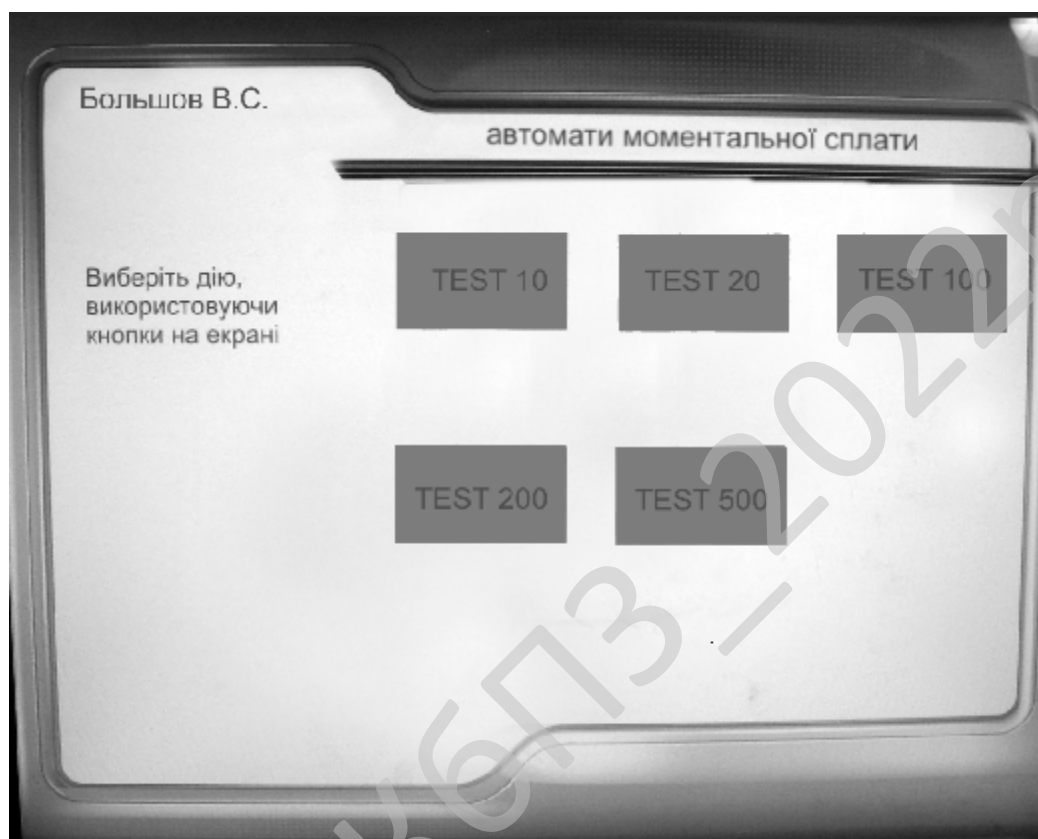


Рисунок 5.3 – Тестове меню

Якщо користувач вибрав і нажав на певний номінал суми поповнення, відбувається вивід повідомлення (рисунок 5.4) про необхідність внесення повної суми оплати чека поповнення не порваними й м'якими купюрами, а також про те, що термінал не видає здачу (це пов'язано з тестовим режимом роботи автомата).

Після чого на сенсорний екран користувачеві виводиться меню прийому грошей з користувача (рисунок 5.5). При внесенні користувачем купюри (рисунок 5.5,а) купюроприймач перевіряє її й програма виводить на меню число уведеної суми.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

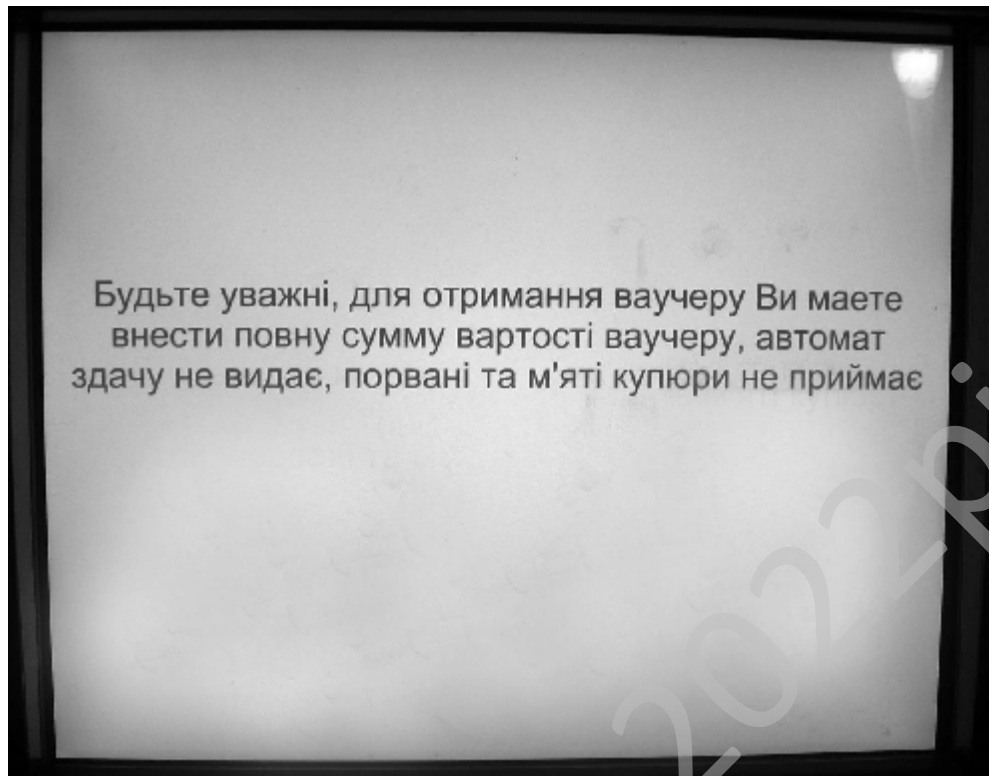


Рисунок 5.3 – Повідомлення про тестову роботу терміналу (неможливості видачі здачі)

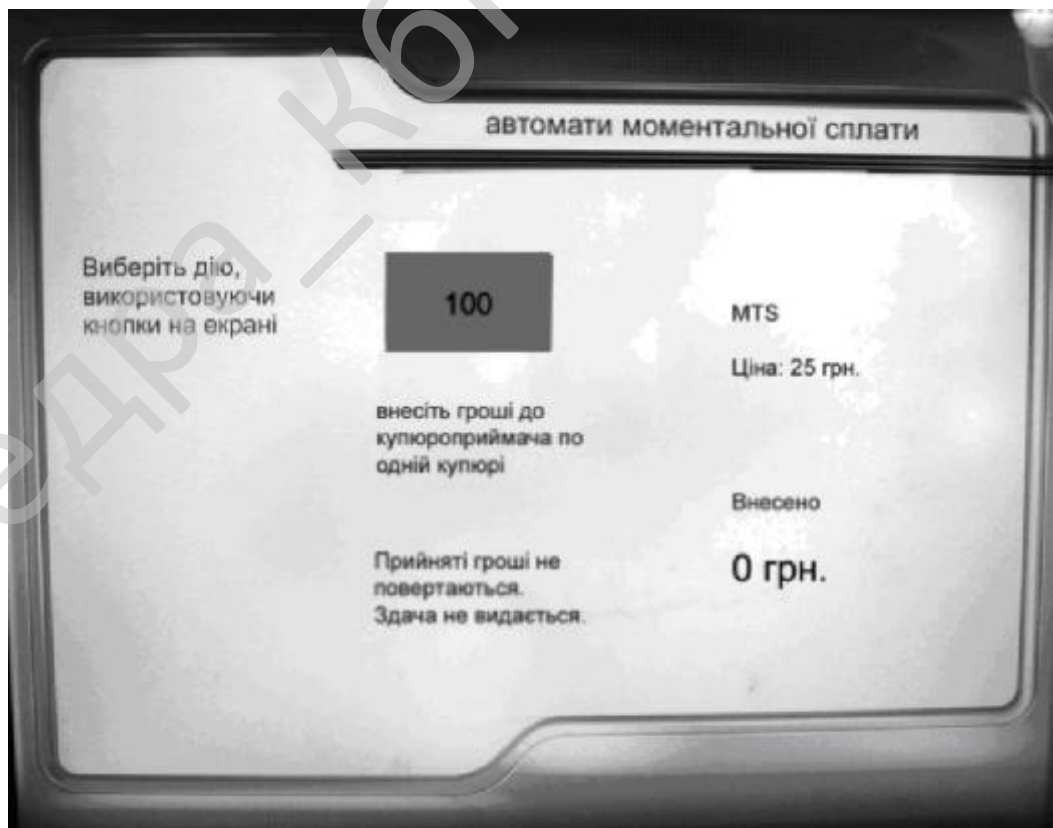


Рисунок 5.4 – Меню приймання грошей з клієнта

При повному внесенні суми ваучера, програма посилає на термопринтер сигнал про видачу чека поповнення (рисунок 5.5,б)

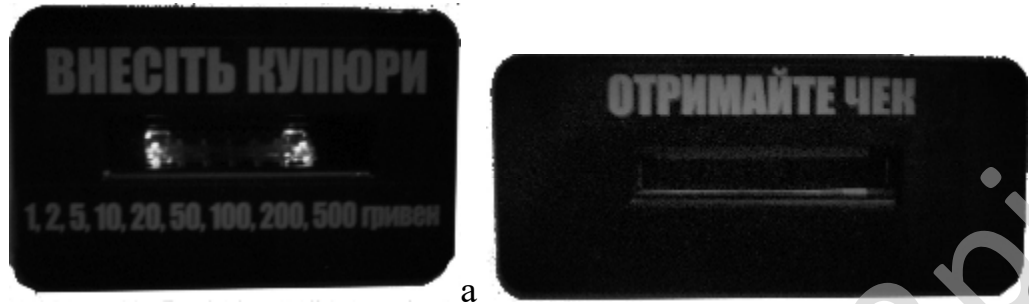


Рисунок 5.5 – Купюро приймач и відсік термопринтера для видачі чеку

Таким чином, у цьому розділі у повній мірі наведена методика впровадження системи в промислову експлуатацію.

Згідно цієї методики приведено алгоритм, дій користувача та адміністратора системи терміналів швидкого продажу карток поповнення мобільних операторів.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи терміналів мережі платіжних автоматів.

Метою розробки є дослідження та програмна реалізація системи терміналів мережі платіжних автоматів.

Об'єктом дослідження є процес терміналів мережі платіжних автоматів.

Предметом дослідження є методи терміналів мережі платіжних автоматів.

Методи дослідження базуються на методах теорії комп'ютерних систем та мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод терміналів мережі платіжних автоматів.
- Розроблено вітчизняний продукт терміналів мережі платіжних автоматів, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

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. Проаналізувавши ринок програм аналогів було визначено ймовірну кількість програмних рішень. Також було підраховано необхідний час для розробки та впровадження програми і визначено орієнтовну вартість програмного забезпечення розробника. Час на розробку склав 60 днів (три місяці). В магістерській роботі були проведені дослідження та виконана програмна реалізація системи терміналів мережі платіжних автоматів. Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір та невеликі системні потреби;
- б) незалежність від встановлених на комп'ютері баз даних, 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
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

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
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП 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.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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.:

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

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

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

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

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

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

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

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

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

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

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

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

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

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

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

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	

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	7875	23625
Продакт-менеджер	0,25	6500	4875
Інженер-програміст	3,8	6500	74100
Інженер-електронщик	1,2	6500	23400
Адміністратор мережі	0,5	6500	9750
Дизайнер WEB	0,25	6500	4875
Інженер-верстальник	0,25	6500	4875
Бухгалтер-економіст	0,5	7000	10500
Всього за період розробки	$R_{cn} = 7,75$	-	$\Phi_{роб} = 156000$

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:

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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

$C_{пл}$ – 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 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

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

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

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

$$I_{нв} = R_{сн}^1 \cdot C_{м}, \quad (7.10)$$

де: $C_{м}$ – 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 09.11.22 – джерело <https://compbest.com.ua>.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок	Hyundai iTMC Pentino Tower	7347

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Процесор	Intel Core i5-2400 (4 ядра по 3.1 – 3.4 GHz) Cache Memory 6 MB	-
Системна плата	Материнська плата MSI B75MA-P45 (s1155) Intel B75, PCI-Ex16, 2x USB 3.0, 6x USB 2.0 5x Audio Ports, LAN (RJ-45), 2x PS/2; порт дискретної графіки – DVI, DisplayPort	-
Відеокарта	ATI Radeon HD 8490, 1 GB GDDR3, 64-bit 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 Blu-ray 22x, SecurDisc, black	-
Корпус	ATX Pentino Tower, 3GTLA-489, PSU 350W(FSP Brand: ATX-350PNR, 12cm) black, (front bezel – black+light silver; bod material – 0.6mm), 80mm fan (rear) 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: A Type Cards, black	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка, система охолодження процесора	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1170/160, 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

Кількість паперу визначаємо за потребою. Приймаємо одну пачку паперу (250 арк.) на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 105$ грн., визначаємо вартість паперу:

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

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

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

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

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 23 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 43 грн./шт.

$$Z_{M2} = 43 \text{ грн.}$$

Згідно методики розрахунку, 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.):

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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

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%.

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

де: P_n – рівень рентабельності, %.

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

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	Γ_{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 $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	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.

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

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.

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

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

де: I_{δ} , I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно; K_{δ} , K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (29108 - 19006) - 0,2 \cdot 14281 = 7246 \text{ грн.}$$

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

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

$$T_{cn} = \frac{14281}{29108 - 19006} = 1,4 \text{ роки.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	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

Продовження таблиці 7.13

Найменування показників	Одиниця виміру	Величина
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.	7246
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	1,4

7.9 Висновки

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

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Персональні електронно-обчислювальні машини (ПЕОМ) відіграють важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ПЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ПЕОМ у професійних цілях, оскільки завдяки ПЕОМ з'явилося багато нових професій.

В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей впливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці.

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

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

8.3. Аналіз умов праці програміста

Умови праці в приміщенні, в якому знаходиться робоче місце програміста є сприятливими. Приміщення обладнане автономною системою газового опалення, основною перевагою якого є програмування режиму роботи в залежності від погодних умов, оскільки клімат є нестійким. Використовується система природної та штучної вентиляції, що забезпечує ефективну циркуляцію повітря. В кабінеті знаходиться кондиціонер AKAİ AK-AC7010-OF.

Засоби копіювальної техніки знаходяться на достатньо далекій відстані від робочих місць, оскільки приміщення складає 20 м², а у відділі налічується два працівники, тобто концентрація озону та оксиду азоту в повітрі є невисокою. Таким чином на кожного програміста приходиться 10,0 м² що відповідає нормам Державним санітарним правилам і нормам ДСанПіН 3.3.2.007-98 [1]. Висота стелі приміщення складає 2,9 метри, що також не порушує нормативні вимоги. Прибиральники підтримують порядок в службових приміщеннях, дотримуються санітарно-гігієнічних норм по прибиранню приміщень, витирають пил, підмітають підлогу наприкінці кожного робочого дня.

В цілому потрібно відмітити застарілість офісної техніки та відсутність клавіатур з ергономічною розкладкою та рідкокристалічних моніторів, які здійснюють менш негативний вплив на стан здоров'я працівників відділу.

Оформлення інтер'єру приміщення є відповідне вимогам з ергономіки та стимулює працівників до підвищення працездатності та зниження втоми. Стеля білого кольору створює оптичний ефект збільшення висоти приміщення, підлога пофарбована коричневим кольором, а стіни – у жовтий. Перевагами даного кольору є створення відчуття теплоти, здатність привертати увагу без додаткової втоми.

Висота столу складає 71,5 см, до того ж його можна регулювати відповідно до власних потреб. Стіл має достатній внутрішній об'єм, завдяки ширині у 73 см та висоті простору під столом – у 63 см, є достатньо важким для забезпечення

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

стійкості. Крісла забезпечують фізіологічно раціональну позу, мають підлокітники, здатні обертатися та регулятор висоти, кута нахилу спинки й відстані спинки від краю сидіння.

В кабінеті створено оптимальні умови праці відносно температури, вологості приміщення та вентиляції.

Наприкінці аналізу небезпечних факторів праці побудуємо підсумкову таблицю 8.1.

Таблиця 8.1 – Підсумкова таблиця значень параметрів небезпечних факторів праці

Найменування параметра	Значення параметра		Нормативний документ
	Фактичне	Нормоване	
1	2	3	4
Освітленість штучна, лк	300	300	ДБН.В 2.5-28:2018 [2]
Значення КПО, %	1,0	1,1	ДБН.В 2.5-28:2018 [2]
Повітрообмін, м /год			
взимку	76	80	ДСН 3.3.6.042-99[3]
влітку	36	80	ДСН 3.3.6.042-99 [3]
Температура повітря. °С			
взимку	22	21-25	ДСанПіН 3.3.2-007-98
влітку	24	27-28	ДСанПіН 3.3.2-007-98
Відносна вологість, %			
взимку	60	<75	ДСанПіН 3.3.2-007-98
влітку	55	<60	ДСанПіН 3.3.2-007-98
Швидкість переміщення повітря, м/с			
взимку	0,16	<0,2	ДСанПіН 3.3.2-007-98
влітку	0,10	<0,2	ДСанПіН 3.3.2-007-98

Щодо вимог електробезпеки, то приміщення за небезпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

Для запобігання поразки електричним струмом в приміщенні відділу використовується ряд організаційно-технічних заходів: розташування проводів живлення поза зоною пересування людей; допуск до роботи електроприладів тільки тих робітників, що знайомі із технікою безпеки; використання мережних продовжувачів з вбудованими запобіжниками на 0,1 А; при ремонті обладнання персонал попереджується.

Устаткування, що працює в приміщенні живиться від мережі 220В та частотою 50Гц. Споживачами цієї напруги є також джерела штучного освітлення. Вони розташовуються на висоті 3 м, що задовольняє нормі, відповідно до якого джерела освітлення повинні розташовуватися на висоті 2,5 м від підлоги.

Проводка схована. У якості розеток для підключення устаткування застосовуються розетки з заземленим кожухом, захищеного від випадкового доторку до струмоведучих частин. Електроустаткування, що знаходиться в приміщенні відділу відноситься до установок напругою до 1000В.

На робочому місці програміста з всього устаткування металевим є лише корпус системного блоку комп'ютера, але тут використовуються системні блоки, що відповідають стандартів фірми ІВМ, у яких крім робочої ізоляції передбачений елемент для заземлення і провід з жилою, що заземлює, для приєднання до джерела живлення.

Основні причини ураження людини електричним струмом на робочому місці:

- дотик до металевих неструмоведучих частин (корпусу, периферії комп'ютера), що можуть виявитися під напругою в результаті ушкодження ізоляції;
- нерегламентоване використання електричних приладів;

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

– відсутність інструктажу співробітників з правил електробезпеки.

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5-10 см від екрана напруженість електростатичного поля складає 60-280 кВ/м, тобто в 10 разів перевищує норму 20 кВ/м.

Отже за результатами проведеного аналізу можна зробити висновки, що всі показники знаходяться у межах запропонованих значень

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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

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

8.5 Розрахункова частина

Для захисного штучного заземлення будемо застосовувати вертикальні електроди з сталевого прокату круглого перерізу діаметром 35 мм., довжиною $L=1,5$ м., та горизонтальний електрод – металева полоса з перетином 35·4 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру (прямокутником).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,5$ м. Відстань між вертикальними заземлювачами (електродами) $A=2$ м. Глибина закладення горизонтального контура заземлення $t=0,7$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,7+1,5/2=1,45 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м.}$$

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багат шаровому ґрунті [11]; $\rho_2 = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунта (глина) [11].

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи терміналів мережі платіжних автоматів.

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 терміналів мережі платіжних автоматів.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем терміналів мережі платіжних автоматів.
- Досліджена система терміналів мережі платіжних автоматів.
- На основі отриманих результатів досліджень створена програмна реалізація системи терміналів мережі платіжних автоматів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання терміналів мережі платіжних автоматів.

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

Програма реалізована на мові високого рівня Embarcadero 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 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7624:2014.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 7246 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,4 роки.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

9. Kovalenko Oleksandr, The mathematical model of the testing technology for DOM XSS vulnerabilities / O. Kovalenko, O. Smirnov, A.Kovalenko, S. Smirnov, V. Vialkova // Scientific & practical cyber security journal (SPCSJ) Volume 2 Issue 1, P. 22-28. Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667. URL: <https://journal.scsa.ge/wp-content/uploads/2018/12/04-3-o.kovalenko-a.kovalenko-o.smirnov-s.smirnov-v.vialkova.pdf>

10. Коваленко А.В. Технология тестирования DOM XSS уязвимости / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Scientific & practical cyber security journal (SPCSJ) Volume 1. Issue 1. P. 38-45 Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667. URL: <https://journal.scsa.ge/wp-content/uploads/2018/12/8-dom-xss-testing-technology-vulnerabilities.pdf>

11. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: монографія / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 305 с.

12. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

15. Коваленко А.В. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 4(120). – Х.: ХУПС – 2014. – С. 161-164.

16. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 5(142). – Х.: ХУПС – 2016. – С. 153-157.

17. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць "Системи обробки інформації". – Випуск 3(140). – Х.: ХУПС – 2016. – С. 40-42.

18. Коваленко А.В. Метод качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.

19. Коваленко А.В. Метод количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 128-133.

20. Коваленко А.В. Использование псевдобулевых методов бивалентного программирования для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.

21. Коваленко А.В. Метод управления рисками разработки программного обеспечения / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 2 (38). – Полтава: ПолтНТУ. – 2016. – С. 93-100.

22. Коваленко А.В. Технология тестирования уязвимости к SQL инъекциям / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

5 (45). – Полтава: ПолтНТУ. – 2017. – С. 66-71.

23. Коваленко А.В. Масштабирование имитационной модели технологии тестирования безопасности / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 6 (46). – Полтава: ПолтНТУ. – 2017. – С. 181-184.

24. Коваленко А.В. Имитационная модель технологии тестирования безопасности Web-приложений / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 1 (47). – Полтава: ПолтНТУ. – 2018. – С. 114-123.

25. Коваленко О.В. Методи якісного аналізу та кількісної оцінки ризиків розроблення програмного забезпечення/ О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 3 (49). – Полтава: ПолтНТУ. – 2018. – С. 116-125.

26. Коваленко О.В. Управління ризиками розроблення програмного забезпечення за умови обмеженості коштів виділених на усунення помилок безпеки/ О.В. Коваленко // Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Випуск 31. – Кропивницький: ЦНТУ. – 2018. – С. 128-140.

27. Коваленко О.В. GERT-мережевий синтез технології тестування на вразливість WEB-додатків/ О.В. Коваленко // Захист інформації. – Випуск 20(2) – К.: НАУ. – 2018. – С. 89-94.

28. Коваленко О.В. Імітаційна модель технології тестування безпеки на основі положень теорії масштабування / О.В. Коваленко // Безпека інформації. – Випуск 24 (2). – К.: НАУ. – 2018. – С. 110-117.

29. Коваленко О.В. Оцінка ефективності технології тестування безпеки / О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 2, 2018. – С. 137-141

30. Коваленко О.В. Методи та засоби управління безпекою додатків / О.В. Коваленко // Інформаційно-керуючі системи на залізничному транспорті. №4, 2018. – С. 41-44.

31. Коваленко О.В. Розробка інформаційної технології передтестової

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

компіляції та розподілу доступу / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 4 (50). – Полтава: ПолтНТУ. – 2018. – С. 115-119.

32. Коваленко О.В. Аналіз та дослідження інформаційних технологій розробки програмного забезпечення/ О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 5, 2018. – С. 131-137.

33. Коваленко О.В. Удосконалений метод управління ризиками розроблення програмного забезпечення на основі напівмарковської моделі прийняття рішень/ О.В. Коваленко // Сучасні інформаційні системи. – Випуск 2(3). – Харків. – 2018. – С. 41-48.

34. Коваленко О.В. Математичні моделі технології тестування DOM XSS вразливості та вразливості до SQL ін'єкцій / О.В. Коваленко // Вісник Черкаського державного технологічного університету. Серія : Технічні науки №4, 2018. – С. 29-36.

35. Коваленко О.В. Математична модель технології тестування вразливості до SQL ін'єкцій / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 6 (58). – Полтава: ПолтНТУ. – 2019. – С. 43-47.

36. Коваленко О.В. Математична модель технології тестування комплексу DOM XSS вразливостей для аналітичної оцінки часових витрат / О.В. Коваленко // Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 173-180, 2019.

37. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць II міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 24-27 лютого 2016 р. – Київ: Європейський університет. – 2016. – С. 138-139.

38. Коваленко А.В. Анализ и оценка рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез «Securitea internationala 2015-2016». Conferenta internationala (editia a XII-a). Chisinau.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

Moldova. 3 martie 2016. – Chisinau: ADSEM. – 2016. – P. 96-102.

39. Коваленко А.В. Исследование источников и причин риска разработки программного обеспечения, этапов и работ, при выполнении которых возникает риск / А.В. Коваленко, А.А. Смирнов // Збірник тез VII всеукраїнської науково-практичної конференції "Інформатика та системні науки (ІСН-2016)". м. Полтава. 10-12 березня 2016 р. – Полтава.: ПУЕТ – 2016. – С. 264-266.

40. Коваленко А.В. Оценка показателя чистой приведенной стоимости для количественной оценки рисков проекта разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез науково-практичної конференції "Проблеми кібербезпеки інформаційно-телекомунікаційних систем". м. Київ. 10-11 березня 2016 р. – Київ: КНУ ім. Тараса Шевченко – 2016. – С. 81-82.

41. Коваленко А.В. Методика структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 24-25 березня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 71-72.

42. Коваленко А.В. Методы качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016). м. Харків. 30 березня – 1 квітня 2016 р. – Харків: НТУ «ХПІ». – 2016. – С. 6-7.

43. Коваленко А.В. Структурная идентификация рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 15-16 квітня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 175-182.

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

44. Коваленко А.В. Исследование разработанной методики структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез VIII міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 28-29 квітня 2016 р. – Харків: ХНЕУ. – 2016. – С. 49.

45. Коваленко А.В. Исследование дерева рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез III міжнародної науково-практичної конференції «Інформаційна та економічна безпека» (INFECO-2016)». м. Харків. 28-30 квітня 2016 р. – Харків: ХННІ ДВНЗ «УБС». – 2016. – С. 174-178.

46. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Сборник тезисов XII международной конференции "Стратегия качества в промышленности и образовании". г. Варна. Болгария. 30 мая – 02 июня 2016 г – Варна. ТУВ. – 2016. – С. 585-589.

47. Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Матеріали Всеукраїнської науково-практичної конференції «Кібербезпека в Україні: правові та організаційні питання». м. Одеса, 21 жовтня 2016 р. – Одеса : ОДУВС, 2016. – С.146-148.

48. Коваленко А.В. Метод управления рисками разработки программного обеспечения с использованием псевдобулевых методов бивалентного программирования / А.В. Коваленко, А.А. Смирнов // Матеріали Всеукраїнської науково-практичної конференції «Актуальні задачі та досягнення у галузі кібербезпеки». м. Кропивницький, 23-25 листопада 2016 року – Кропивницький: ЦНТУ, 2016. – С. 162.

49. Коваленко А.В. Псевдобулевые методы бивалентного программирования для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко,

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

С.А. Смирнов // Збірник наукових праць III міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 22-25 лютого 2017 р. – Київ: Європейський університет. – 2017. – С. 158-162.

50. Коваленко А.В. Метод управління ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов // Збірник тез II науково-практичної конференції “Проблеми кібербезпеки інформаційно-телекомунікаційних систем”. м. Київ. 23-24 березня 2017 р. – Київ: КНУ ім. Тараса Шевченка – 2017. – С. 203-205.

51. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>

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

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

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

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

56. 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>

57. Постанова № 42 від 01.12.1999 Головного державного санітарного

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>

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

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

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

					ВКРМ-123.22.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.22.0002.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Большов В.С.				<i>Дослідження та програмна реалізація системи терміналів мережі платіжних автоматів</i>	Літ.	Аркуш	Аркушів
Перевірів	Коваленко А.С.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-21М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи терміналів мережі платіжних автоматів.

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. № 19-13 від 17.08.2022 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи терміналів мережі платіжних автоматів.

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-123.22.0002.00.00.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 терміналів мережі платіжних автоматів;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.22.0002.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Embarcadero Delphi.

					ВКРМ-123.22.0002.00.00.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 2022 року.

8 Вимоги щодо охорони праці

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

					ВКРМ-123.22.0002.00.00.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 – 107 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

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

					ВКРМ-123.22.0002.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Коваленко А.С.

*Дослідження та програмна реалізація
системи терміналів мережі платіжних автоматів*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 33

Літера: РП

Кропивницький – 2022 року

Файл проекту терміналу

```
program Bondarenko_Terminal;

uses
  Forms,
  Unit1 in 'Ter_Unit1.pas' {Form1},
  Unit2 in ' Ter_Unit2.pas' {Form2},
  Unit3 in ' Ter_Unit3.pas' {Form3};
  Unit4 in ' Ter_Unit4.pas' {Form4};
  Unit5 in ' Ter_Unit5.pas' {Form5};
  Unit5 in ' Ter_Unit6.pas' {Form6};

{$R *.RES}

begin
  Form6:=TForm5.Create(Application);
  Form6.Show;
  Form6.Update;
  Try
    Application.HintPause:=200;
    Application.HintHidePause:=7000;
    Application.HintShortPause:=25;
    Application.Initialize;
    Application.Title := 'BonDarenkoTerminal';
    Application.CreateForm(TForm1, Form1);
    Application.CreateForm(TForm2, Form2);
    Application.CreateForm(TForm3, Form3);
    Application.CreateForm(TForm4, Form4);
    Application.CreateForm(TForm5, Form5);
  finally
    Form6.Free;
  end;
  Application.Run;
end.
```

Файл проекту Ter_Unit1.pas - організація захищеного стеку термінала

```

//*****//
// Testuvalos`a u: Embarcadero Delphi//
// OS: WINDOWS 10/11 //
// Testova versiya teminalu //
// coded by: Bolshov V.S. //
//*****//

unit Unit1;
//Terminal_Traffic;

interface

uses SysUtils, Windows, IPHelper, IPHLPAPI;

type
TTerminalTraffic = Class;

TNewInstanceEvent = procedure(Sender :TTerminalTraffic) of object;
TFreezeEvent = procedure(Sender :TTerminalTraffic) of object;

TTerminalTraffic = Class
private
FIP: string;
FMac: string;
FInPerSec: Dword;
FInTotal: Dword;
FPeakInPerSec: Dword;
FInterfaceIndex: DWord;
FActiveCountIn: Dword;
FSecondsActive: Cardinal;
FPrevCountIn: DWord;
FDescription: string;
FOutTotal: Dword;
FPeakOutPerSec: Dword;
FOutPerSec: Dword;
FPrevCountOut: DWord;
FActiveCountOut: Dword;
FAverageInPerSec: Dword;
FAverageOutPerSec: Dword;
FStartedAt: TDateTime;
FRunning: boolean;
FOnFreeze: TFreezeEvent;
FOnUnFreeze: TFreezeEvent;
FConnected: boolean;

```

FFound: boolean;

FSpeed: DWord;

```

function GetIPFromIFIndex(InterfaceIndex: Cardinal): string;
public
property Found : boolean read FFound write FFound;
property Connected : boolean read FConnected;
property Running : boolean read FRunning;
property InterfaceIndex : DWord read FInterfaceIndex;
property IP : string read FIP;
property Mac : string read FMac;
property Description : string read FDescription;
property StartedAt : TDateTime read FStartedAt;
property SecondsActive : Cardinal read FSecondsActive;
property Speed : DWord read FSpeed;
property ActiveCountIn : Dword read FActiveCountIn;
property PrevCountIn : DWord read FPrevCountIn;
property InPerSec : Dword read FInPerSec; property AverageInPerSec : Dword read
FAverageInPerSec;
property InTotal : Dword read FInTotal;
property PeakInPerSec : Dword read FPeakInPerSec;
property ActiveCountOut : Dword read FActiveCountOut;
property PrevCountOut : DWord read FPrevCountOut;
property OutPerSec : Dword read FOutPerSec; property AverageOutPerSec : Dword
read FAverageOutPerSec;
property OutTotal : Dword read FOutTotal;
property PeakOutPerSec : Dword read FPeakOutPerSec;
procedure NewCycle(const InOctets, OutOctets, TrafficSpeed : Dword);
procedure Reset;
procedure Freeze;
procedure UnFreeze;
procedure MarkDisconnected;
function GetStatus : string;
function FriendlyRunningTime:string;
constructor Create(const AMibIfRow : TMibIfRow; OnNewInstance :
TNewInstanceEvent);
published
property OnFreeze :TFreezeEvent read FOnFreeze write FOnFreeze;
property OnUnFreeze :TFreezeEvent read FOnUnFreeze write FOnUnFreeze;
end;

function BytesToFriendlyString(Value : DWord) : string;
function BitsToFriendlyString(Value : DWord) : string;

```

implementation

```

function BytesToFriendlyString(Value : DWord) : string;
const
OneKB = 1024;
OneMB = OneKB * 1024;
OneGB = OneMB * 1024;
begin
if Value < OneKB then
Result := FormatFloat('#,##0.00 B',Value)
else
if Value < OneMB then
Result := FormatFloat('#,##0.00 KB', Value / OneKB)
else
if Value < OneGB then
Result := FormatFloat('#,##0.00 MB', Value / OneMB)
end; (*BytesToFriendlyString*)

```

```

function BitsToFriendlyString(Value : DWord) : string;
const
OneKB = 1000;
OneMB = OneKB * 1000;
OneGB = OneMB * 1000;
begin
if Value < OneKB then
Result := FormatFloat('#,##0.00 bps',Value)
else
if Value < OneMB then
Result := FormatFloat('#,##0.00 Kbps', Value / OneKB)
else
if Value < OneGB then
Result := FormatFloat('#,##0.00 Mbps', Value / OneMB)
end; (*BytesToFriendlyString*)

```

```

{ TTerminalTraffic }
constructor TTerminalTraffic.Create(const AMibIfRow: TMibIfRow; OnNewInstance :
TNewInstanceEvent);
var
Descr : string;
begin
inherited Create;

FRunning := True;
FConnected := True;

self.FInterfaceIndex := AMibIfRow.dwIndex;
self.FIP := GetIPFromIFIndex(self.InterfaceIndex);

```

```

self.FMac := MacAddr2Str(TMacAddress(AMibIfRow.bPhysAddr),
AMibIfRow.dwPhysAddrLen);

SetLength(Descr, Pred(AMibIfRow.dwDescrLen));
Move(AMibIfRow.bDescr, Descr[1], pred(AMibIfRow.dwDescrLen));
self.FDescription := Trim(Descr);

self.FPrevCountIn := AMibIfRow.dwInOctets;
self.FPrevCountOut := AMibIfRow.dwOutOctets;

self.FStartedAt := Now;

self.FSpeed := AMibIfRow.dwSpeed;

FActiveCountIn := 0;
FActiveCountOut:= 0;
FInTotal := 0;
FOutTotal:= 0;
FInPerSec:= 0;
FOutPerSec:= 0;
FPeakInPerSec := 0;
FPeakOutPerSec:=0;

if Assigned(OnNewInstance) then OnNewInstance(self);
end; (*Create*)

procedure TTerminalTraffic.NewCycle(const InOctets, OutOctets, TrafficSpeed:
Dword);
begin
Inc(self.FSecondsActive);

If not Running then Exit;

FSpeed := TrafficSpeed;

//IN
self.FInPerSec := InOctets - self.PrevCountIn;
Inc(self.FInTotal, self.InPerSec);
if InPerSec > 0 then Inc(FActiveCountIn);
if InPerSec > PeakInPerSec then FPeakInPerSec := InPerSec;
try
self.FAverageInPerSec := InTotal div ActiveCountIn
except
self.FAverageInPerSec := 0
end;

```

```
FPrevCountIn := InOctets;
```

```
//OUT
```

```
self.FOutPerSec := OutOctets - self.PrevCountOut;
```

```
Inc(self.FOutTotal, self.OutPerSec);
```

```
if OutPerSec > 0 then Inc(FActiveCountOut);
```

```
if OutPerSec > PeakOutPerSec then FPeakOutPerSec := OutPerSec;
```

```
try
```

```
self.FAverageOutPerSec := OutTotal div ActiveCountOut
```

```
except
```

```
self.FAverageOutPerSec := 0
```

```
end;
```

```
FPrevCountOut := OutOctets;
```

```
end; (*NewCycle*)
```

```
function TTerminalTraffic.GetIPFromIFIndex(InterfaceIndex: Cardinal): string;
```

```
var
```

```
i: integer;
```

```
IPArr : TMIBIPAddrArray;
```

```
begin
```

```
Result := '!not_found!';
```

```
Get_IPAddrTableMIB( IPArr );
```

```
if Length(IPArr) > 0 then
```

```
for i := low(IPArr) to High(IPArr) do
```

```
if IPArr[i].dwIndex = InterfaceIndex then
```

```
begin
```

```
Result := IPAddr2Str(IPArr[i].dwAddr);
```

```
BREAK;
```

```
end;
```

```
end; (*GetIPFromIFIndex*)
```

```
procedure TTerminalTraffic.Reset;
```

```
begin
```

```
self.FPrevCountIn := InPerSec;
```

```
self.FPrevCountOut := OutPerSec;
```

```
self.FStartedAt := Now;
```

```
FSecondsActive := 0;
```

```
FActiveCountIn := 0;
```

```
FActiveCountOut:= 0;
```

```
FInTotal := 0;
```

```
FOutTotal := 0;
```

```
FInPerSec := 0;
```

```
FOutPerSec := 0;
```

```
FPeakInPerSec := 0;
```

```

FPeakOutPerSec := 0;
end; (*Reset*)

procedure TTerminalTraffic.Freeze;
begin
FRunning := False;
if Assigned(FOnFreeze) then OnFreeze(Self);
end; (*Freeze*)

procedure TTerminalTraffic.UnFreeze;
begin
FRunning := True;
if Assigned(FOnUnFreeze) then OnUnFreeze(Self);
end; (*UnFreeze*)

procedure TTerminalTraffic.MarkDisconnected;
begin
self.FConnected := False;
self.FRunning := False;
end; (*MarkDisconnected*)

function TTerminalTraffic.GetStatus: string;
begin
if self.Connected then
Result := 'Connected'
else
Result := 'NOT connected';

if self.Running then
Result := Result + ', 1'
else
Result := Result + ', 2';
end; (*GetStatus*)

function TTerminalTraffic.FriendlyRunningTime: string;
var
H, M, S: string;
ZH, ZM, ZS: Integer;
begin
ZH := SecondsActive div 3600;
ZM := Integer(SecondsActive) div (60 - ZH * 60);
ZS := Integer(SecondsActive) - (ZH * 3600 + ZM * 60);
H := Format('%.2d', [ZH]);
M := Format('%.2d', [ZM]);
S := Format('%.2d', [ZS]);

```

```
Result := H + ':' + M + ':' + S;  
end; (*FriendlyRunningTime*)
```

```
end.
```

Кафедра _ КБПЗ _ 2022 рік

Файл проекту Ter_Unit2.pas - перевірка апаратної частини

```

//*****//
// Testuvalos`a u: Embarcadero Delphi//
// OS: WINDOWS 10/11 //
// Testova versiya teminalu //
// coded by: Bolshov V.S. //
//*****//

unit Unit2;
//TerminalSysInfoU;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, StdCtrls, Buttons, ImgList, Commctrl, ShellAPI, Menus;

type
TForm1 = class(TForm)
PageControl1: TPageControl;
TabSheet2: TTabSheet;
TabSheet1: TTabSheet;
ListView1: TListView;
StatusBar1: TStatusBar;
ListView2: TListView;
ImageList1: TImageList;
PopupMenu: TPopupMenu;
Details1: TMenuItem;
KillProcess1: TMenuItem;
RefreshList1: TMenuItem;
PopupMenu2: TPopupMenu;
CloseWindow1: TMenuItem;
RefreshList2: TMenuItem;
SpeedButton1: TSpeedButton;
procedure PageControl1Change(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ListView1Click(Sender: TObject);
procedure ListView1Db1Click(Sender: TObject);
procedure ListView1KeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure Details1Click(Sender: TObject);
procedure KillProcess1Click(Sender: TObject);
procedure RefreshList1Click(Sender: TObject);
procedure CloseWindow1Click(Sender: TObject);
procedure RefreshList2Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
private
{ Private declarations }

```

```

public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{$R *.DFM}
uses
TlHelp32, AdditU;
procedure ListProcesses;
var
hSnapshot: THandle;
lppe: TProcessEntry32;
hIcon: THandle;
Count: Integer;
procedure _FillList;
begin
with Form1.ListView1.Items.Add, lppe do
begin
hIcon:= ExtractIcon(hInstance, lppe.szExeFile, 0);
if hIcon = 0 then
hIcon:= LoadImage(0, IDI_WINLOGO, IMAGE_ICON, LR_DEFAULTSIZE,
LR_DEFAULTSIZE, LR_DEFAULTSIZE or LR_DEFAULTCOLOR or LR_SHARED);
ImageIndex:= ImageList_AddIcon(Form1.ImageList1.Handle, hIcon);
Caption:= ExtractFileName(szExeFile);
SubItems.Add(Format('$%x', [Th32ProcessID]));
SubItems.Add(Format('$%x', [Th32ParentProcessID]));
case pcPriClassBase of
4: SubItems.Add(Format('%d (Idle)', [pcPriClassBase]));
8: SubItems.Add(Format('%d (Norm)', [pcPriClassBase]));
13: SubItems.Add(Format('%d (High)', [pcPriClassBase]));
24: SubItems.Add(Format('%d (Real)', [pcPriClassBase]));
else SubItems.Add(Format('%d', [pcPriClassBase]));
end;
SubItems.Add(Format('%d', [cntThreads]));
SubItems.Add(Format('%d', [cntUsage]));
SubItems.Add(szExeFile);
end;
Inc(Count);
end;
begin
hSnapshot:= CreateToolHelp32Snapshot(TH32CS_SNAPPROCESS, 0);
if hSnapshot <> INVALID_HANDLE_VALUE then
begin
Form1.ListView1.Items.Clear;
lppe.dwSize:= SizeOf(lppe);

```

```

Count:= 0;
if Process32First(hSnapShot, lppe) then _FillList;
while Process32Next(hSnapShot, lppe) do _FillList;
Form1.StatusBar1.Panels[1].Text:= 'Total: ' + IntToStr(Count);
CloseHandle(hSnapShot);
end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
end;
procedure ListWindows;
function EnumWindowsProc(hWnd: THandle; lParam: Integer): Boolean; stdcall;
var
Text, PName: string;
hIcon: THandle;
lpdwPID: PDWORD;
i: Integer;
begin
New(lpdwPID);
GetWindowThreadProcessID(hWnd, lpdwPID);
SetLength(Text, 255);
if GetWindowText(hWnd, PChar(Text), 255) <> 0 then
with Form1.ListView2.Items.Add, Form1.ListView1 do
begin
hIcon:= GetClassLong(hWnd, GCL_HICON);
ImageIndex:= ImageList_AddIcon(Form1.ImageList1.Handle, hIcon);
for i:= 0 to Items.Count - 1 do
if Cardinal(StrToInt(Items[i].SubItems[0])) = lpdwPID^ then
PName:= Items[i].Caption;
Caption:= Text;
SubItems.Add(Format('%d', [hWnd]));
SubItems.Add(Format('$%x', [lpdwPID^]));
SubItems.Add(PName);
Dispose(lpdwPID);
end;
Result:= true;
end;
begin
Form1.ListView2.Items.Clear;
EnumWindows(@EnumWindowsProc, 0);
end;

procedure ListModules(OwnerID: Cardinal);
var
hSnapShot: THandle;
lpme: TModuleEntry32;

```

```

procedure _FillList;
begin
with Form2.ListView1.Items.Add, lpme do
begin
Caption:= ExtractFileName(szModule);
SubItems.Add(Format('%d', [modBaseSize]));
SubItems.Add(Format('$%p', [modBaseAddr]));
SubItems.Add(Format('%d', [ProccntUsage]));
SubItems.Add(Format('%d', [GlblcntUsage]));
SubItems.Add(szExePath);
end;
end;

begin
hSnapShot:= CreateToolHelp32Snapshot(TH32CS_SNAPMODULE, OwnerID);
if hSnapShot <> INVALID_HANDLE_VALUE then
begin
Form2.ListView1.Items.Clear;
lpme.dwSize:= SizeOf(lpme);
if Module32First(hSnapShot, lpme) then _FillList;
while Module32Next(hSnapShot, lpme) do _FillList;
CloseHandle(hSnapShot);
end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
end;

procedure ListThreads(OwnerID: Cardinal);
var
hSnapShot: THandle;
lpte: TThreadEntry32;

procedure _FillList;
begin
if lpte.Th32OwnerProcessID =
Cardinal(StrToInt(Form1.ListView1.Selected.SubItems[0])) then
with Form2.ListView2.Items.Add, lpte do
begin
Caption:= Format('$%x', [Th32ThreadID]);
case TpBasePri of
4: SubItems.Add(Format('%d (Idle)', [TpBasePri]));
8: SubItems.Add(Format('%d (Norm)', [TpBasePri]));
13: SubItems.Add(Format('%d (High)', [TpBasePri]));
24: SubItems.Add(Format('%d (Real)', [TpBasePri]));
else SubItems.Add(Format('%d', [TpBasePri]));
end;
end;

```

```

case TpDeltaPri of
-15: SubItems.Add(Format('%d (Idle)', [TpDeltaPri]));
-2: SubItems.Add(Format('%d (Lowest)', [TpDeltaPri]));
-1: SubItems.Add(Format('%d (Low)', [TpDeltaPri]));
0: SubItems.Add(Format('%d (Normal)', [TpDeltaPri]));
1: SubItems.Add(Format('%d (High)', [TpDeltaPri]));
2: SubItems.Add(Format('%d (Highest)', [TpDeltaPri]));
15: SubItems.Add(Format('%d (Time Critical)', [TpDeltaPri]));
else SubItems.Add(Format('%d', [TpDeltaPri]));
end;
SubItems.Add(Format('%d', [cntUsage]));
end;
end;

begin
hSnapShot:= CreateToolhelp32Snapshot(TH32CS_SNAPTHREAD, OwnerID);
if hSnapShot <> INVALID_HANDLE_VALUE then
begin
Form2.ListView2.Items.Clear;
lpte.dwSize:= SizeOf(lpte);
if Thread32First(hSnapShot, lpte) then _FillList;
while Thread32Next(hSnapShot, lpte) do _FillList;
CloseHandle(hSnapShot);
end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
end;

procedure ListHeaps(OwnerID: Cardinal);
var
hSnapShot: THandle;
lph1: THeapList32;
lphe: THeapEntry32;

procedure _FillList;
begin
if Heap32First(lphe, lph1.Th32ProcessID, lph1.Th32HeapID) then
repeat
with Form2.ListView3.Items.Add, lphe do
begin
Caption:= Format('$%x', [Th32HeapID]);
SubItems.Add(Format('%d', [dwBlockSize]));
SubItems.Add(Format('$%x', [dwAddress]));
case dwFlags of
LF32_FIXED: SubItems.Add('Fixed');
LF32_FREE: SubItems.Add('Free');

```

```

LF32_MOVEABLE: SubItems.Add('Moveable')
else SubItems.Add('Unknown');
end;
end
until not Heap32Next(lphe);
end;

begin
try
Form1.Caption:= 'SysInfo: PROCESSING... PLEASE WAIT';
hSnapShot:= CreateToolhelp32Snapshot(TH32CS_SNAPHEAPLIST, OwnerID);
if hSnapShot <> INVALID_HANDLE_VALUE then
begin
Form2.ListView3.Items.Clear;
lph1.dwSize:= SizeOf(lph1);
lphe.dwSize:= SizeOf(lphe);
if Heap32ListFirst(hSnapShot, lph1) then _FillList;
while Heap32ListNext(hSnapShot, lph1) do _FillList;
CloseHandle(hSnapShot);
end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
finally
Form1.Caption:= 'SysInfo';
end;
end;

procedure ListThreadWindows;
var
i: Integer;
function EnumThreadWindowsProc(hWnd: THandle; lParam: Integer): Boolean;
stdcall;
var
Text: string;
hIcon: THandle;
begin
SetLength(Text, 255);
if GetWindowText(hWnd, PChar(Text), 255) <> 0 then
with Form2.ListView4.Items.Add do
begin
hIcon:= GetClassLong(hWnd, GCL_HICON);
ImageIndex:= ImageList_AddIcon(Form1.ImageList1.Handle, hIcon);
Caption:= Text;
SubItems.Add(Format('%d', [hWnd]));
end;
Result:= true;

```

```

end;

begin
Form2.ListView4.Items.Clear;
for i:= 0 to Form2.ListView2.Items.Count - 1 do
EnumThreadWindows(Cardinal(StrToInt(Form2.ListView2.Items[i].Caption)),@EnumThreadWindowsProc, 0);
end;

{-----}
procedure TForm1.PageControl1Change(Sender: TObject);
begin
ListProcesses;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
ListProcesses;
ListWindows;
end;

procedure TForm1.ListView1Click(Sender: TObject);
begin
with (Sender as TListView) do
if Selected <> nil then
StatusBar1.Panels[0].Text:= Selected.SubItems[5] else
StatusBar1.Panels[0].Text:= '';
end;

procedure TForm1.ListView1Db1Click(Sender: TObject);
begin
Details1Click(self);
end;

procedure TForm1.ListView1KeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if flag = true then flag:= false
else
begin
if Key = 13 then ListView1Db1Click(ListView1);
if Key <> 0 then ListView1Click(ListView1);
end;
end;

procedure TForm1.Details1Click(Sender: TObject);

```

```
begin
with ListView1, Form2 do
if Selected <> nil then
begin
ListModules(StrToInt(Selected.SubItems[0]));
ListThreads(StrToInt(Selected.SubItems[0]));
ListHeaps(StrToInt(Selected.SubItems[0]));
ListThreadWindows;
Caption:= 'Details for ' + Selected.Caption;
Show;
end
'Information', MB_OK or MB_ICONINFORMATION);
end;

procedure TForm1.KillProcess1Click(Sender: TObject);
var
hProcess: THandle;
S: string;
ID: Cardinal;
begin
if ListView1.Selected <> nil then
begin
S:= ListView1.Selected.Caption;
ID:= StrToInt(ListView1.Selected.SubItems[0]);
hProcess:= OpenProcess(PROCESS_ALL_ACCESS, false, ID);
if hProcess <> INVALID_HANDLE_VALUE then
begin
if not TerminateProcess(hProcess, 0) then
MessageBox(0, PChar('Unable to kill process: ' + S),
'Error', MB_ICONWARNING or MB_OK);
CloseHandle(hProcess);
Sleep(500);
ListProcesses;
end;
end
MB_OK or MB_ICONINFORMATION);
end;

procedure TForm1.RefreshList1Click(Sender: TObject);
begin
ListProcesses;
end;

procedure TForm1.CloseWindow1Click(Sender: TObject);
begin
if ListView2.Selected <> nil then
```

```
begin
PostMessage(StrToInt(ListView2.Selected.SubItems[0]), WM_CLOSE, 0, 0);
Sleep(500);
ListWindows;
end
end;

procedure TForm1.RefreshList2Click(Sender: TObject);
begin
ListWindows;
end;

end.
```

Кафедра _ КБПЗ _ 2022 рік

Файл проекту Ter_Unit3.pas - Організація часових проміжків, системні функції

```

//*****//
// Testuvalos`a u: Embarcadero Delphi//
// OS: WINDOWS 10/11 //
// Testova versiya teminalu //
// coded by: Bolshov V.S. //
//*****//

unit Unit3;
//FormUnit;
interface
uses
Windows, Graphics, ExtCtrls, Controls, StdCtrls, Buttons, Tabs,
ComCtrls, Classes, SysUtils, Forms, dialogs,
TerminalUnit;
type
TMainForm = class(TForm)
pnlMain: TPanel;
pnlBottom: TPanel;
pc: TPageControl;
tsAbout: TTabSheet;
tsTraffic: TTabSheet;
ExitButton: TButton;
TrafficTabs: TTabSet;
GroupBox: TGroupBox;
ledAdapterDescription: TLabeledEdit;
UnFreezeButton: TBitBtn;
FreezeButton: TBitBtn;
ClearCountersButton: TBitBtn;
ledMACAddress: TLabeledEdit;
gbIN: TGroupBox;
ledOctInSec: TLabeledEdit;
ledAvgINSec: TLabeledEdit;
ledPeakINSec: TLabeledEdit;
ledTotalIN: TLabeledEdit;
gbOUT: TGroupBox;
ledOctOUTSec: TLabeledEdit;
ledAvgOUTSec: TLabeledEdit;
ledPeakOUTSec: TLabeledEdit;
ledTotalOUT: TLabeledEdit;
Timer: TTimer;
gbTime: TGroupBox;
ledStartedAt: TLabeledEdit;
ledActiveFor: TLabeledEdit;
RemoveInactiveButton: TBitBtn;
StatusText: TStaticText;

```

```

cbOnTop: TCheckBox;
Panel3: TPanel;
ProductName: TLabel;
lblURL: TLabel;
Label3: TLabel;
ProgramIcon: TImage;
StaticText1: TStaticText;
ledSpeed: TLabeledEdit;
procedure TimerTimer(Sender: TObject);
procedure ClearCountersButtonClick(Sender: TObject);
procedure cbOnTopClick(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure TrafficTabsChange(Sender: TObject; NewTab: Integer;
var AllowChange: Boolean);
procedure ExitButtonClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FreezeButtonClick(Sender: TObject);
procedure UnFreezeButtonClick(Sender: TObject);
procedure RemoveInactiveButtonClick(Sender: TObject);
procedure lblURLClick(Sender: TObject);
procedure StaticText1Click(Sender: TObject);
procedure pcChange(Sender: TObject);
procedure ledAdapterDescriptionChange(Sender: TObject);
private
procedure HandleNewAdapter(ATraffic : TTerminalTraffic);
procedure HandleFreeze(ATraffic : TTerminalTraffic);
procedure HandleUnFreeze(ATraffic : TTerminalTraffic);
function LocateTraffic(AdapterIndex : DWord) : TTerminalTraffic;
procedure ProcessMIBData;
procedure ClearDisplay;
procedure RefreshDisplay;
public
{ Public declarations }
end;
var
MainForm: TMainForm;
ActiveTraffic : TTerminalTraffic;

implementation
{$R *.dfm}
procedure TMainForm.ClearDisplay;
var
j:integer;
begin
TrafficTabs.Tabs.Clear;
StatusText.Caption:='';

```

```

for j:= 0 to GroupBox.ControlCount-1 do
begin
if GroupBox.Controls[j] is TCustomEdit then
TCustomEdit(GroupBox.Controls[j]).Text := '';
end;
end;

procedure TMainForm.TimerTimer(Sender: TObject);
begin
Timer.Enabled := False;
ProcessMIBData;
Timer.Enabled := True;
end; (*TimerTimer*)

procedure TMainForm.ClearCountersButtonClick(Sender: TObject);
begin
ActiveTraffic.Reset;
RefreshDisplay;
end;

procedure TMainForm.cbOnTopClick(Sender: TObject);
begin
if cbOnTop.Checked = true then
FormStyle := fsSTAYONTOP
else
FormStyle := fsNORMAL;
end;

procedure TMainForm.FormDestroy(Sender: TObject);
var
i : integer;
begin
Timer.OnTimer := nil;
ActiveTraffic := nil;
for i:= 0 to -1 + TrafficTabs.Tabs.Count do
TrafficTabs.Tabs.Objects[i].Free;
end;

procedure TMainForm.TrafficTabsChange(Sender: TObject; NewTab: Integer; var
AllowChange: Boolean);
begin
if NewTab = -1 then
ActiveTraffic := nil
else
ActiveTraffic := TTerminalTraffic(TrafficTabs.Tabs.Objects[NewTab]);
RefreshDisplay;
end;

```

```
procedure TMainForm.ExitButtonClick(Sender: TObject);
begin
Close;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
Timer.Interval := 1000;
ClearDisplay;
ActiveTraffic := nil;
pcChange(Sender);
Timer.Enabled := True;
end;

procedure TMainForm.RefreshDisplay;
begin
if not Assigned(ActiveTraffic) then
begin
ClearDisplay;
Exit;
end;

with ActiveTraffic do
begin

FreezeButton.Visible := Connected;
UnFreezeButton.Visible := Connected;
ClearCountersButton.Visible := Connected;
RemoveInactiveButton.Visible := not Connected;

FreezeButton.Enabled := Running;
UnFreezeButton.Enabled := not Running;

ledAdapterDescription.Text := Description;
ledMACAddress.Text := MAC;

ledSpeed.Text := BitsToFriendlyString(Speed);

ledOctInSec.Text := BytesToFriendlyString(InPerSec);
ledPeakInSec.Text := BytesToFriendlyString(PeakInPerSec);
ledAvgINSec.Text := BytesToFriendlyString(AverageInPerSec);
ledTotalIN.Text := BytesToFriendlyString(InTotal);

ledOctOUTSec.Text := BytesToFriendlyString(OutPerSec);
```

```

ledPeakOUTSec.Text := BytesToFriendlyString(PeakOutPerSec);
ledAvgOUTSec.Text := BytesToFriendlyString(AverageOutPerSec);
ledTotalOUT.Text := BytesToFriendlyString(OutTotal);

self.ledStartedAt.Text := DateTimeToStr(StartedAt);
self.ledActiveFor.Text := FriendlyRunningTime;

StatusText.Caption := GetStatus;
end;
end; (*RefreshDisplay*)

procedure TMainForm.ProcessMIBData;
var
MibArr : IpHlpAPI.TMIBIfArray;
i : integer;
ATraffic : TTerminalTraffic;
begin
Get_IfTableMIB(MibArr);

for i:= 0 to -1 + TrafficTabs.Tabs.Count do
begin
ATraffic := TTerminalTraffic(TrafficTabs.Tabs.Objects[i]);
if ATraffic.Connected then ATraffic.Found := False;
end;
ATraffic := nil;

//process
if Length(MibArr) > 0 then
begin
for i := Low(MIBArr) to High(MIBArr) do
begin
ATraffic := LocateTraffic(MIBArr[i].dwIndex);
if Assigned(ATraffic) then
begin
ATraffic.NewCycle(MIBArr[i].dwInOctets, MIBArr[i].dwOutOctets,
MIBArr[i].dwSpeed);
end
else
begin
ATraffic := TTerminalTraffic.Create(MIBArr[i], HandleNewAdapter);
ATraffic.Found := True;
ATraffic.OnFreeze := HandleFreeze;
ATraffic.OnUnFreeze := HandleUnFreeze;
end;
end;
end;
end;
end;

```

```
for i:= 0 to -1 + TrafficTabs.Tabs.Count do
if NOT TTerminalTraffic(TrafficTabs.Tabs.Objects[i]).Found then
TTerminalTraffic(TrafficTabs.Tabs.Objects[i]).MarkDisconnected;

RefreshDisplay;
end; (*ProcessMIBData*)

function TMainForm.LocateTraffic(AdapterIndex : DWord): TTerminalTraffic;
var
j : cardinal;
ATraffic : TTerminalTraffic;
begin
Result := nil;
if TrafficTabs.Tabs.Count = 0 then Exit;

for j:= 0 to -1 + TrafficTabs.Tabs.Count do
begin
ATraffic := TTerminalTraffic(TrafficTabs.Tabs.Objects[j]);
if ATraffic.InterfaceIndex = AdapterIndex then
begin
Result := ATraffic;
Result.Found := True;
Break;
end;
end;
end;

procedure TMainForm.HandleNewAdapter(ATraffic: TTerminalTraffic);
begin
TrafficTabs.Tabs.AddObject(ATraffic.IP, ATraffic);
TrafficTabs.TabIndex := -1 + TrafficTabs.Tabs.Count;
end; (*HandleNewAdapter*)

procedure TMainForm.FreezeButtonClick(Sender: TObject);
begin
ActiveTraffic.Freeze;
end;

procedure TMainForm.UnFreezeButtonClick(Sender: TObject);
begin
ActiveTraffic.UnFreeze;
end;

procedure TMainForm.HandleFreeze(ATraffic: TTerminalTraffic);
begin
```

```
self.FreezeButton.Enabled := ATraffic.Running;
self.UnFreezeButton.Enabled := not ATraffic.Running;
end;

procedure TMainForm.HandleUnFreeze(ATraffic: TTerminalTraffic);
begin
self.FreezeButton.Enabled := ATraffic.Running;
self.UnFreezeButton.Enabled := not ATraffic.Running;
end;

procedure TMainForm.RemoveInactiveButtonClick(Sender: TObject);
begin
If not ActiveTraffic.Connected then
begin
ActiveTraffic.Free;
ActiveTraffic := nil;
TrafficTabs.Tabs.Delete(TrafficTabs.TabIndex);
TrafficTabs.SelectNext(False);
end;

RefreshDisplay;
end;

procedure TMainForm.pcChange(Sender: TObject);
begin
pnlBottom.Visible := pc.ActivePage = tsTraffic;
end;

procedure TMainForm.ledAdapterDescriptionChange(Sender: TObject);
begin
ledAdapterDescription.Hint := ledAdapterDescription.Text;
ledAdapterDescription.ShowHint := Canvas.TextWidth(ledAdapterDescription.Text)
> ledAdapterDescription.ClientWidth;
end;
end.
```

Файл проекту Ter_Unit4.pas - заголовки внутренние переменны библиотеки IPHLPAPI

```
//*****//
// Testuvalos`a u:  Embarcadero Delphi//
// OS: WINDOWS 10/11 //
// Testova versiya teminalu //
// coded by: Bolshov V.S. //
//*****//

unit Unit4
//IPHLPAPI;

interface
uses
  Windows, winsock;
const
  VERSION      = '1.3';
const
  ANY_SIZE      = 1;
  MAX_ADAPTER_DESCRIPTION_LENGTH = 128; // arb.
  MAX_ADAPTER_NAME_LENGTH = 256; // arb.
  MAX_ADAPTER_ADDRESS_LENGTH = 8; // arb.
  DEFAULT_MINIMUM_ENTITIES = 32; // arb.
  MAX_HOSTNAME_LEN = 128; // arb.
  MAX_DOMAIN_NAME_LEN = 128; // arb.
  MAX_SCOPE_ID_LEN = 256; // arb.

  BROADCAST_NODETYPE = 1;
  PEER_TO_PEER_NODETYPE = 2;
  MIXED_NODETYPE = 4;
  HYBRID_NODETYPE = 8;
ATypes : array[0..8] of string[20] =
( 'UNKNOWN', 'BROADCAST', 'PEER_TO_PEER', '', 'MIXED', '', '', '', 'HYBRID');
  IF_OTHER_ADAPTERTYPE = 0;
  IF_ETHERNET_ADAPTERTYPE = 1;
  IF_TOKEN_RING_ADAPTERTYPE = 2;
  IF_FDDI_ADAPTERTYPE = 3;
  IF_PPP_ADAPTERTYPE = 4;
  IF_LOOPBACK_ADAPTERTYPE = 5;
  IF_SLIP_ADAPTERTYPE = 6;
//
  AdaptTypes : array[0..6] of string[10] =
( 'other', 'ethernet', 'tokenring', 'FDDI', 'PPP', 'loopback', 'SLIP' );
  MAX_INTERFACE_NAME_LEN = 256; { mrapi.h }
  MAXLEN_PHYSADDR = 8; { iprtmib.h }
  MAXLEN_IFDESCR = 256; { --"--- }
//-----
type
```

```
TMacAddress = array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte;
```

```
PTIP_ADDRESS_STRING = ^TIP_ADDRESS_STRING;
```

```
TIP_ADDRESS_STRING = array[0..15] of char;
```

```
PTIP_ADDR_STRING = ^TIP_ADDR_STRING;
```

```
TIP_ADDR_STRING = packed record // for use in linked lists
```

```
    Next: PTIP_ADDR_STRING;
```

```
    IPAddress: TIP_ADDRESS_STRING;
```

```
    IpMask: TIP_ADDRESS_STRING;
```

```
    Context: DWORD;
```

```
end;
```

```
PTFixedInfo = ^TFixedInfo;
```

```
TFixedInfo = packed record
```

```
    HostName: array[0..MAX_HOSTNAME_LEN + 4] of char;
```

```
    DomainName: array[0..MAX_DOMAIN_NAME_LEN + 4] of char;
```

```
    CurrentDNSServer: PTIP_ADDR_STRING;
```

```
    DNSServerList: TIP_ADDR_STRING;
```

```
    NodeType: UINT;
```

```
    ScopeID: array[0..MAX_SCOPE_ID_LEN + 4] of char;
```

```
    EnableRouting: UINT;
```

```
    EnableProxy: UINT;
```

```
    EnableDNS: UINT;
```

```
end;
```

```
PTMibIfRow = ^TMibIfRow;
```

```
TMibIfRow = packed record
```

```
    wszName: array[1..MAX_INTERFACE_NAME_LEN] of WCHAR;
```

```
    dwIndex: DWORD;
```

```
    dwType: DWORD;
```

```
    dwMTU: DWORD;
```

```
    dwSpeed: DWORD;
```

```
    dwPhysAddrLen: DWORD;
```

```
    bPhysAddr: array[1..MAXLEN_PHYSADDR] of byte;
```

```
    dwAdminStatus: DWORD;
```

```
    dwOperStatus: DWORD;
```

```
    dwLastChange: DWORD;
```

```
    dwInOctets: DWORD;
```

```
    dwInUcastPkts: DWORD;
```

```
    dwInNUCastPkts: DWORD;
```

```
    dwInDiscards: DWORD;
```

```
    dwInErrors: DWORD;
```

```
    dwInUnknownProtos: DWORD;
```

```
    dwOutOctets: DWORD;
```

```
    dwOutUCastPkts: DWORD;
```

```
    dwOutNUCastPkts: DWORD;
```

```

dwOutDiscards: DWORD;
dwOutErrors: DWORD;
dwOutQLen: DWORD;
dwDescrLen: DWORD;
bDescr: array[1..MAXLEN_IFDESCR] of char; //byte;
end;

TMIBIfArray = array of TMIBIFRow;

PTMibIfTable = ^TMIBIfTable;
TMibIfTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIfRow;
end;

TTIME_T = array[1..325] of byte;

PTIP_ADAPTER_INFO = ^TIP_ADAPTER_INFO;
TIP_ADAPTER_INFO = packed record
    Next: PTIP_ADAPTER_INFO;
    ComboIndex: DWORD;
    AdapterName: array[1..MAX_ADAPTER_NAME_LENGTH + 4] of char;
    Description: array[1..MAX_ADAPTER_DESCRIPTION_LENGTH + 4] of char;
    AddressLength: UINT;
    Address: array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte;
    Index: DWORD;
    aType: UINT;
    DHCPEnabled: UINT;
    CurrentIPAddress: PTIP_ADDR_STRING;
    IPAddressList: TIP_ADDR_STRING;
    GatewayList: TIP_ADDR_STRING;
    DHCPServer: TIP_ADDR_STRING;
    HaveWINS: BOOL;
    PrimaryWINSServer: TIP_ADDR_STRING;
    SecondaryWINSServer: TIP_ADDR_STRING;
    LeaseObtained: TTIME_T; //??
    LeaseExpires: TTIME_T; //??
end;

PTMibTCPRow = ^TMibTCPRow;
TMibTCPRow = packed record
    dwState: DWORD;
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
    dwRemoteAddr: DWORD;
    dwRemotePort: DWORD;

```

```

end;
//
PTMibTCPTable = ^TMibTCPTable;
TMibTCPTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..0] of TMibTCPRow;
end;
//
PTMibTCPStats = ^TMibTCPStats;
TMibTCPStats = packed record
    dwRTOAlgorithm: DWORD;
    dwRTOMin: DWORD;
    dwRTOMax: DWORD;
    dwMaxConn: DWORD;
    dwActiveOpens: DWORD;
    dwPassiveOpens: DWORD;
    dwAttemptFails: DWORD;
    dwEstabResets: DWORD;
    dwCurrEstab: DWORD;
    dwInSegs: DWORD;
    dwOutSegs: DWORD;
    dwRetransSegs: DWORD;
    dwInErrs: DWORD;
    dwOutRsts: DWORD;
    dwNumConns: DWORD;
end;

PTMibUDPRow = ^TMibUDPRow;
TMibUDPRow = packed record
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
end;
//
PTMibUDPTable = ^TMIBUDPTable;
TMIBUDPTable = packed record
    dwNumEntries: DWORD;
    UDPTable: array[0..ANY_SIZE - 1] of TMibUDPRow;
end;
//
PTMibUdpStats = ^TMIBUdpStats;
TMIBUdpStats = packed record
    dwInDatagrams: DWORD;
    dwNoPorts: DWORD;
    dwInErrors: DWORD;
    dwOutDatagrams: DWORD;
    dwNumAddrs: DWORD;

```

```
end;

PTMibIPNetRow = ^TMibIPNetRow;
TMibIPNetRow = packed record
    dwIndex: DWord;
    dwPhysAddrLen: DWord;
    bPhysAddr: TMacAddress;
    dwAddr: DWord;
    dwType: DWord;
end;

PTMibIPNetTable = ^TMibIPNetTable;
TMibIPNetTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIPNetRow;
end;

PTMibIPStats = ^TMibIPStats;
TMibIPStats = packed record
    dwForwarding: DWORD;
    dwDefaultTTL: DWORD;
    dwInReceives: DWORD;
    dwInHdrErrors: DWORD;
    dwInAddrErrors: DWORD;
    dwForwDatagrams: DWORD;
    dwInUnknownProtos: DWORD;
    dwInDiscards: DWORD;
    dwInDelivers: DWORD;
    dwOutRequests: DWORD;
    dwRoutingDiscards: DWORD;
    dwOutDiscards: DWORD;
    dwOutNoRoutes: DWORD;
    dwReasmTimeOut: DWORD;
    dwReasmReqds: DWORD;
    dwReasmOKs: DWORD;
    dwReasmFails: DWORD;
    dwFragOKs: DWORD;
    dwFragFails: DWORD;
    dwFragCreates: DWORD;
    dwNumIf: DWORD;
    dwNumAddr: DWORD;
    dwNumRoutes: DWORD;
end;

//
PTMibIPAddrRow = ^TMibIPAddrRow;
TMibIPAddrRow = packed record
```

```

dwAddr: DWORD;
dwIndex: DWORD;
dwMask: DWORD;
dwBCastAddr: DWORD;
dwReasmSize: DWORD;
Unused1,
    Unused2: WORD;
end;

TMibIPAddrArray = array of TMIBIPAddrRow;

PTMibIPAddrTable = ^TMibIPAddrTable;
TMibIPAddrTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIPAddrRow;
end;

PTMibIPForwardRow = ^TMibIPForwardRow;
TMibIPForwardRow = packed record
    dwForwardDest: DWORD;
    dwForwardMask: DWORD;
    dwForwardPolicy: DWORD;
    dwForwardNextHop: DWORD;
    dwForwardIFIndex: DWORD;
    dwForwardType: DWORD;
    dwForwardProto: DWORD;
    dwForwardAge: DWORD;
    dwForwardNextHopAS: DWORD;
    dwForwardMetric1: DWORD;
    dwForwardMetric2: DWORD;
    dwForwardMetric3: DWORD;
    dwForwardMetric4: DWORD;
    dwForwardMetric5: DWORD;
end;
//
PTMibIPForwardTable = ^TMibIPForwardTable;
TMibIPForwardTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIPForwardRow;
end;

PTMibICMPStats = ^TMibICMPStats;
TMibICMPStats = packed record
    dwMsgs: DWORD;
    dwErrors: DWORD;

```

```

dwDestUnreachs: DWORD;
dwTimeEcxcds: DWORD;
dwParmProbs: DWORD;
dwSrcQuenchs: DWORD;
dwRedirects: DWORD;
dwEchos: DWORD;
dwEchoReps: DWORD;
dwTimeStamps: DWORD;
dwTimeStampReps: DWORD;
dwAddrMasks: DWORD;
dwAddrReps: DWORD;
end;

PTMibICMPInfo = ^TMibICMPInfo;
TMibICMPInfo = packed record
    InStats: TMibICMPStats;
    OutStats: TMibICMPStats;
end;
//-----+-----
function GetAdaptersInfo( pAdapterInfo: PTIP_ADAPTER_INFO;
    pOutBufLen: PULONG ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetNetworkParams( FixedInfo: PTFixedInfo; pOutPutLen: PULONG ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetTcpTable( pTCPTable: PTMibTCPTable; pDwSize: PDWORD;
    bOrder: BOOL ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetTcpStatistics( pStats: PTMibTCPStats ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetUdpTable( pUdpTable: PTMibUDPTable; pDwSize: PDWORD;
    bOrder: BOOL ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetUdpStatistics( pStats: PTMibUdpStats ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetIpStatistics( pStats: PTMibIPStats ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetIpNetTable( pIpNetTable: PTMibIPNetTable;
    pdwSize: PULONG;
    bOrder: BOOL ): DWORD;

```

```
stdcall; external 'IPHLPAPI.DLL';
```

```
function GetIpAddrTable( pIpAddrTable: PTMibIPAddrTable;  
    pdwSize: PULONG;  
    bOrder: BOOL ): DWORD;  
stdcall; external 'IPHLPAPI.DLL';
```

```
function GetIpForwardTable( pIPForwardTable: PTMibIPForwardTable;  
    pdwSize: PULONG;  
    bOrder: BOOL ): DWORD;  
stdCall; external 'IPHLPAPI.DLL';
```

```
function GetIcmpStatistics( pStats: PTMibICMPInfo ): DWORD;  
stdCall; external 'IPHLPAPI.DLL';
```

```
function GetRTTAndHopCount( DestIPAddress: DWORD; HopCount: PULONG;  
    MaxHops: ULONG; RTT: PULONG ): BOOL;  
stdCall; external 'IPHLPAPI.DLL';
```

```
function GetIfTable( pIfTable: PTMibIfTable; pdwSize: PULONG;  
    bOrder: boolean ): DWORD;  
stdCall; external 'IPHLPAPI.DLL';
```

```
function GetIfEntry( pIfRow: PTMibIfRow ): DWORD;  
stdCall; external 'IPHLPAPI.DLL';
```

```
implementation
```

```
end.
```