Міністерство освіти і науки України

Коледж інформаційних систем і технологій

ДВНЗ «Київський національний економічний

університет імені Вадима Гетьмана»

**Курсовий проект**

по предмету: «Алгоритмізація та програмування»

за темою: «Розробка системи проїзду в метро»

Керівник: Кисіль Т. М. Виконав: Артеменко В.О.

Допущено до захисту: ст.\_\_\_\_179101\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (№зал. книжка)

гр. 291 \_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 р. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з оцінкою (підпис виконавця)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оцінка)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 р.

Члени комісії:

Грицак Н.Ю. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Котлярова Ю.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кисіль Т.М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КІСІТ КНЕУ 5.05010101.01 ПЗ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КОЛЕДЖ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ

ДВНЗ «КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ

УНІВЕРСИТЕТ ІМЕНІ ВАДИМА ГЕТЬМАНА»

Спеціальність:­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­\_5.05010101.01\_Обслуговування програмних системи і комплексів

Дисципліна:\_\_\_\_\_\_\_\_\_ «Алгоритмізація та програмування»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Курс \_2\_\_ Група \_291\_ Семестр:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЗАВДАННЯ

на курсовий проект студента

\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Артеменка Владислава Олександровича*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім’я, по батькові)

1. Тема проекту:\_\_\_\_\_\_Розробка системи проїзду в метро\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Термін захисту розробленого проекту: «\_01\_»\_\_\_червня\_\_\_\_\_\_\_\_2019 р.
2. Вимоги до проекту:\_\_розробка програмного забезпечення системи оплати проїзду в метрополітені;\_ створення інтерфейсу користувача з виведенням та збереженням\_\_\_ результатів у масиви; створення інформаційної довідки та інструкції\_\_\_\_\_\_\_\_\_\_\_\_\_\_ користувача, оператора, адміністратора\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Зміст пояснювальної записки:

1. \_\_\_\_Титульний аркуш, лист завдання на курсовий проект, календарний план, доповідь, зміст, перелік скорочень та умовних позначень; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_ Вступ: актуальність вибраної теми\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_Теоретична частина: характеристика та аналіз предметної області, аналіз\_\_\_\_\_\_ існуючих рішень, функціональні можливості системи, вибір і обгрунтування\_\_\_\_\_\_\_\_\_

програмних рішень, інформаційна модель системи, вхідні та вихідні дані, опис\_\_\_\_\_\_\_ математичної моделі, функціонування моделі системи, алгоритм роботи системи; \_\_\_\_

4. \_\_\_Практична частина: проектування інтерфейсу системи за його основними\_\_\_\_\_\_ пунктами, забезпечення ідентифікаторів при неконкретному введенні/виведенні даних, тестування системи, опис створеного програмного забезпечення\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (інтерфейсу користувача, оператора, адміністратора), інструкції по\_встановленню\_\_\_\_ розробленого\_програмного забезпечення. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. \_\_\_Висновки до теоретичної та практичної частини. Перелік використаних джерел, додатки (таблиці, діаграми, лістинги програм)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

**проектування курсового проекту**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **п/п** | **Етапи проектування курсового проекту** | **Термін виконання** | **Примітка** |
|  | Вибір теми. Опрацювання літературних джерел. Оформлення листа завдання. | 1 тиждень |  |
|  | Структура змісту курсового проекту. Розроблення вимог до системи та узгодження з керівником. | 2 тиждень |  |
|  | Визначення актуальності вибраної теми. | 3 тиждень |  |
|  | Структура теоретичної частини курсового проекту. | 4 тиждень |  |
|  | Постановка задачі. Визначення вхідних та вихідних даних поставленої задачі. | 5 тиждень |  |
|  | Проектування функціональної моделі системи. | 6 тиждень |  |
|  | Розроблення алгоритму роботи системи. | 7 тиждень |  |
|  | Вибір алгоритмів реалізації поставленої задачі. | 8 тиждень |  |
|  | Висновки до теоретичної частини. Перше узгодження з керівником. | 9 тиждень |  |
|  | Структура розрахункової частини курсового проекту. | 10 тиждень |  |
|  | Вибір і обґрунтування інструментальних засобів для розроблення системи. | 11 тиждень |  |
|  | Розробка та проектування алгоритму вирішення поставленої задачі. | 12 тиждень |  |
|  | Визначення ідентифікаторів алгоритму роботи системи. Опис розробленого алгоритму. | 13 тиждень |  |
|  | Висновки до розрахункової частини. Друге узгодження з керівником. | 14 тиждень |  |
|  | Реалізація інтерфейсу проектованої системи. Опис інтерфейсу користувача, адміністратора. | 15 тиждень |  |
|  | Тестування розробленого програмного забезпечення та перевірка коректності введених даних. | 16 тиждень |  |
|  | Встановлення та обслуговування програмного забезпечення. | 17 тиждень |  |
|  | Висновки до практичної частини. Третє узгодження з керівником. | 18 тиждень |  |
|  | Нормативний контроль, підготовка доповіді, презентації, роздаткового матеріалу, архівація та здача курсового проекту керівнику. | 19 тиждень |  |
|  | Захист та демонстрація курсового проекту. | 20 тиждень |  |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Артеменко В. О.\_\_\_\_\_\_\_\_

(підпис) (прізвище, ім’я, по батькові )

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Кисіль Т. М.\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище, ім’я, по батькові )

**ЗМІСТ**

[Вступ 5](#_Toc11098362)

[1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 6](#_Toc11098363)

[1.1 Характеристика предметної області 6](#_Toc11098364)

[1.2 Аналіз існуючих рішень 6](#_Toc11098365)

[1.3 Функціональні можливості системи 11](#_Toc11098366)

[2 РОЗРОБЛЕННЯ ЗАГАЛЬНОЇ КОНЦЕПЦІЇ СИСТЕМИ 13](#_Toc11098367)

[2.1 Вибір і обґрунтування програмних рішень 13](#_Toc11098368)

[2.2 Інформаційна модель системи 16](#_Toc11098369)

[2.2.1 Вхідні дані системи 16](#_Toc11098370)

[2.2.2 Вихідні дані системи 16](#_Toc11098371)

[2.3 Функціональна модель системи 16](#_Toc11098372)

[2.4 Алгоритм роботи системи 22](#_Toc11098373)

[3 ПРОЕКТУВАННЯ СИСТЕМИ 47](#_Toc11098374)

[3.1 Проектування інтерфейсу системи 47](#_Toc11098375)

[3.2 Тестування системи 48](#_Toc11098376)

[3.3 Вимоги щодо встановленого програмного забезпечення 54](#_Toc11098377)

[ВИСНОВКИ 55](#_Toc11098378)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ПОСИЛАНЬ 56](#_Toc11098379)

[ДОДАТКИ 57](#_Toc11098380)

[Додаток А. Лістинг розробленого програмного забезпечення 57](#_Toc11098381)

**Вступ**

На даний момент актуальною проблемою є доцільність використання існуючої системи оплати проїзду в мережі комунального підприємства «Київський метрополітен», адже на даний момент існуюча система оплати включає в себе лише загальну оплату, не залежно від часу перебування і довжини поїздок пасажирів.

Метою курсового проекту – є розробка програмного забезпечення мовою програмування С++, що дасть можливість використовувати систему оплати в мережі комунального підприємства «Київський метрополітен» більш економно, ефективно та для поліпшення фінансового становища підприємства, а отже і покращення якості обслуговування пасажирів.

Пошуки шляхів досягнення цієї мети обумовили необхідність визначення наступних *завдань*:

* доступ до загальної бази даних користувачів;
* можливість виводити інформацію користувачу про стан його картки чи ваучера та про кількість можливих поїздок;
* можливість вибору напрямку руху та кількості станцій, що планує проїхати пасажир;
* можливість безконтактного поповнення кількості поїздок через картку або смарт-девайс

**Практичне значення одержаних результатів** полягає в тому, що пасажири зможуть більш доцільно та рівноправно розпоряджатися своїми грошима, а також вигідно, перш за все, для себе взаємодіяти із системою оплати проїзду в метро.

**Структура роботи** зумовлена метою і завданням та складається із вступу, трьох розділів, висновків, додатків та переліку джерел посилань.

1. **ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**
   1. **Характеристика предметної області**

Об’єктом предметної області, для якої проектується система, є існуюча платіжна система, що допомагає користувачам даної послуги(пасажирам) здійснювати оплату безготівковим методом.

Метою створення даного програмного забезпечення є поліпшення стану обслуговування пасажирів на комунальному підприємстві «Київський метрополітен» шляхом запровадження більш точної, ніж існуюча, системи оплати проїзду.

Призначенням даного програмного забезпечення є полегшення методу замовлення або поповнення існуючих проїзних квитків.

Майбутніми споживачами даного програмного забезпечення є усі пасажири комунального підприємства «Київський метрополітен».

Застосовувати дане програмне забезпечення можна також і в інших сферах комунального або приватного громадського транспорту, наприклад тролейбусах чи маршрутних таксі.

Дане програмне забезпечення вирішить низку проблем: по-перше, незручність існуючої системи оплати, по-друге, недофінансування з міського бюджету комунального підприємства «Київський метрополітен, по-третє, несправедливу систему оцінки проїзду.

* 1. **Аналіз існуючих рішень**

На даний момент проїзд у метрополітені оплачується за допомогою:

* Жетонів, які можна придбати в касах станцій метрополітену або в автоматах видачі жетонів, що знаходяться у касових залах.
* Проїзних квитків на термін дії на основі картонних електронних безконтактних карток для проїзду в метрополітені та наземному міському пасажирському транспорті, які можна придбати в касах станцій.
* Проїзних квитків на основі пластикових електронних безконтактних карток з продовженням терміну дії або на кількість поїздок, які можна придбати в касах станцій або універсальних автоматах, а поповнити їх ресурс – в касах станцій, автоматах дозапису безконтактних карток чи універсальних автоматах, встановлених у касових залах.
* Банківськими картками та платіжними пристроями з технологією безконтактної оплати PayPass

Проїзні квитки на основі безконтактних карток реалізуються з 22 числа поточного місяця – на наступний календарний місяць, з 15 числа – на другу половину поточного місяця за 50% вартості. Реалізація та продовження терміну дії проїзних квитків на другу половину місяця здійснюється з відкриття станцій.

Також у світі існують безліч зразкових способів системи оплати проїзду.

Узяти до уваги хоча б місто Лондон. У ньому існують такі види сплати:

Travelcard - проїзний квиток, який використовується для оплати проїзду на різних видах лондонського громадського транспорту, визначеного терміну дії (від одного дня до року)

1. Карта Oyster. На відміну від квитків, карта Oyster призначена для багаторазового використання. На неї можна покласти певну суму грошей (в тому числі поповнити баланс рахунку «оплата при вході» або проїзного Travelcard) в касах або за допомогою комп'ютеризованих квиткових автоматів. У разі оплати за системою «оплата при вході», вартість кожної поїздки буде зніматися з рахунку. З жовтня 2005 року проїзні Travelcard терміном на тиждень, місяць і рік, випущені Лондонським метрополітеном або службою «Транспорт для Лондона», доступні тільки на картах Oyster. Далі на табл. 1.1 наведено переваги і недоліки у порівнянні зі старою системою:

Таблиця 1.1 – Недоліки та переваги у порівнянні зі старою системою

|  |  |  |
| --- | --- | --- |
| Назва існуючого способу оплати | **Переваги** в порівнянні з запропонованою системою | **Недоліки** в порівнянні з запропонованою системою |
| Жетони | - | 1. Незручність у повсякденному користуванні 2. Складний спосіб покупки 3. Черги при покупці та біля автоматів пропуску |
| Проїзні квитки метрополітену | - | 1. Складний спосіб покупки 2. Мала кількість автоматів для використання |
| Банківські картки | 1. Можливість поповнення через майже всі системи прийому платежів | 1. Невелика кількість пристроїв зчитування карток 2. Можливість швидкого розмагнічування |
| PayPass | 1. Можливість оплати через смартфони з підтримкою NFC 2. Можливість поповнення через майже всі системи прийому платежів | 1. Ускладнена процедура верифікації на стороні клієнта 2. Підтримка малої кількості пристроїв |

Однак обидва ці способи мають значний недолік – відсутність дійсно вигідної для пасажира програми лояльності. Запропонована ж система має водночас усі переваги перевіреної часом Лондонської системи, а також гнучку систему знижок.

У наступній таблиці проаналізовано усі переваги й недоліки існуючих методів оплати проїзду в порівнянні з запропонованою мною новітньою системою керування кількістю поїздок з можливістю поповнення

На даний момент в існуючій системі оплати проїзду в метрополітені існує така система знижок:

Таблиця 1.2 – Існуюча система знижок КП «Київський метрополітен»

|  |  |  |
| --- | --- | --- |
| Вид послуг | Кількість поїздок | Тариф на 1 поїздку, грн |
| Перевезення пасажирів: | | |
| Метрополітеном(у разі використання пластикової безконтактної картки) | 1-9 | 8,00 |
| 10-19 | 7,70 |
| 20-29 | 7,40 |
| 30-39 | 7,10 |
| 40-49 | 6,80 |
| 50+ | 6,50 |

При запровадженні оновленої системи оплати проїзду PassWay буде існувати більш лояльна до пасажирів система знижок при придбанні багаторазового ваучера або поповненні рахунку в застосунку. Досягти цього можна через зменшення складності механізмів рівня передачі оплати від кінцевого користувача о банку в якому проходить оплата. Якщо раніше кошти повинні були пройти через фінансову систему посередника – КП «Київпастранс», то нині все буде спрямовано безпосередньо до банку. На таблиці 1.3 приведено приклад можливих знижок у разі запровадження новітньої системи

Таблиця 1.3 – Система знижок при використані системи PassWay

|  |  |  |  |
| --- | --- | --- | --- |
| Вид послуг | Кількість поїздок | Тариф на 1 поїздку, грн | Економія, % |
| Перевезення пасажирів: | | | |
| Метрополітеном(у разі використання пластикової безконтактної картки) | 1-9 | 8,00 | 0 |
| 10-19 | 7,50 | 5 |
| 20-29 | 7,20 | 10 |
| 30-39 | 7,00 | 15 |
| 40-49 | 6,50 | 20 |
| 50+ | 6,20 | 25 |

* 1. **Функціональні можливості системи**

До комп’ютерної системи оплати проїзду сформовано вимоги, що представлені на рис. 1

Рисунок 1.1 – Структурна схема комп’ютерної системи

Для роботи з комп’ютерною системою оплати проїзду , що проектується можна виділити такі інтерфейси:

* Користувача
* Оператора
* Адміністратора

Їх взаємодія показана на схемі (рис. 1.2).

Головним завданням курсового проекту – є розробка оптимізованого та функціонального програмного забезпечення для оплати проїзду в комунальному підприємстві «Київський метрополітен» в терміналах типу PassWay, мовою програмування С++ реалізованого в середовищі програмування Microsoft Visual Studio 2019



Рисунок 1.2 – Функціональна схема системи оплати проїзду

1. **РОЗРОБЛЕННЯ ЗАГАЛЬНОЇ КОНЦЕПЦІЇ СИСТЕМИ**
   1. **Вибір і обґрунтування програмних рішень**

При проектуванні комп’ютерної системи оплати проїзду в метрополітені в якості мови програмування було обрано С++.

Мова програмування С++ ідеально підходить для розробки системи даного типу, адже вона дуже швидко працює з консольними додатками для виконання операцій типу банківських або оплати проїзду різних видів.

Перевагами використання саме С++ в якості головної мови проекту є:

1. Малий розмір кінцевого додатку користувача, що дозволить встановити його навіть на системах старого типу
2. Сумісність з усіма наявними методами зв’язку апаратної моделі системи з програмним забезпеченням
3. Підвищена швидкодія в порівнянні з існуючими системами, а також утричі швидший відгук на безконтактну оплату зі смартфону

Для розробки комп’ютерної системи оплати проїзду в метрополітені було обрано середовище програмування Microsoft Visual Studio 2019.

Дане середовище найкраще підходить для роботи з мовою програмування С++, адже воно було розроблено саме під цей тип мови і постійно отримує перевірені оновлення безпеки та швидкодії у консольних додатках.

Перевагами середовища програмування Microsoft Visual Studio 2019 є:

1. Сумісність з усіма операційними системами, що будуть задіяні в розробці і запуску проекту
2. Можливість швидкого редагування коду при непередбаченому випадку зміни правил користування або оплати
3. Спосіб розповсюдження даного програмного забезпечення не вимагає додаткових витрат з боку розробників системи оплати проїзду в метро, а також з боку тестувальників та іншого обслуговуючого персоналу.

В процесі реалізації комп’ютерної системи необхідно використання наступних бібліотек мови С++, які наведено в таблиці 2.1

Таблиця 2.1 – Бібліотеки С++ та їх призначення

|  |  |
| --- | --- |
| Бібліотека | Призначення |
| iostream | Стандартна бібліотека вводу-виводу |
| сtime | Бібліотека, що працює з датами та часом |
| cmath | Бібліотека виконання обчислень |
| iomanip | Бібліотека, що працює з маніпуляторами |
| fstream | Бібліотека, що надає змогу виконувати зчитування, обробку та зміну даних у текстових файлах |
| windows.h | Бібліотека, яка дозволяє працювати з редагуванням вигляду командного рядка |
| array | Бібліотека, що дозволяє працювати з різноманітними скінченими масивами |
| vector | Бібліотека, що дозволяє працювати з різноманітними необмеженими масивами |

При розробці комп’ютерної системи необхідне використання наступних ідентифікаторів та змінних, які наведено в наступній таблиці

Таблиця 2.2 – Змінні та ідентифікатори та їх переваги

|  |  |
| --- | --- |
| Ідентифікатор / змінна | **Спосіб використання** |
| menuanswer | Визначення, введеного користувачем, пункту меню |
| var | Відповідність введених даних допустимим |
| c\_o\_t | Загальний лічильник кількості квитків |
| sum | Загальна сума грошей, що були витрачені на покупку квитків |
| timeinfo | Збереження поточної дати та часу |
| num\_t | Порядковий номер квитка |
| new\_price\_t | Нова(змінена) ціна квитка |
| add\_nm | Нове ім’я користувача |
| add\_bal | Сума введених користувачем коштів |
| cot1 | Кількість квитків при виборі першого типу |
| сot2 | Кількість квитків при виборі другого типу |
| сot3 | Кількість квитків при виборі третього типу |
| Var(local) | Загальна вартість придбаних квитків одного типу |
| acc\_idd | Введений ідентифікатор користувача |
| usr\_id | Ідентифікатор покупця |
| operator\_password | Введений пароль оператора |
| opw | Вірний пароль оператора |
| opwp | Логічна змінна, для перевірки паролю оператора |
| admin\_password | Введений пароль адміністратора |
| apw | Вірний пароль адміністратора |
| apwp | Логічна змінна, для перевірки паролю адміністратора |
| id\_user | Масив користувачів |
| balance | Масив коштів на балансі користувачів |
| seconds | Змінна, що дозволяє програмі доступ до системного годинника |
| rate | Масив вартості квитків |

* 1. **Інформаційна модель системи**
     1. **Вхідні дані системи**

Вхідними даними в проектованій системі для здійснення оплати проїзду є:

* Номер картки;
* Сума поповнення;
* Кількість зупинок, які планує проїхати пасажир
  + 1. **Вихідні дані системи**

В результаті роботи системи вихідною інформацією являються наступні дані:

* Час початку поїздки
* Вартість поїздки на задану кількість станцій з урахуванням знижок
* Формування унікального коду для валідації
  1. **Функціональна модель системи**

На рис. 2.1 показано функціональну взаємодію компонентів програми, а також інтерфейсів оператора, користувача та адміністратора.

Входячи до меню користувача пасажир має вибір між реєстрацією нового аккаунту та входом до існуючого, а також виходом в головне меню. При вході у існуючий обліковий запис користувач отримує повідомлення з його даними по картці на даний момент, а також вибір між інтерфейсом нового замовлення та виходу на попередню сторінку.

Під час використання облікового запису оператора персонал, що обслуговує даний термінал зможе подивитися загальну інформацію стосовно кількості продаж ваучерів та можливих збоїв системи для подальшого повідомлення про них адміністратору.

Входячи до особистого облікового запису адміністратор має можливість дізнатися конкретні дані щодо безперебійності роботи системи, а також усю інформацію щодо якості поточного зв’язку конкретно цього терміналу з головним сервером.

Функціональна модель системи представляє собою три робочі інтерфейси: інтерфейс користувача, інтерфейс оператора та інтерфейс адміністратора. Усі інтерфейси знаходяться у головному меню. Головне меню дає вибір для подальших дій. Тобто ви можете вибрати інтерфейс з яким надалі працюватиме. Головне меню, яке можна побачити на рис. 2.2 представляє вхід до трьох інтерфейсів: користувача, оператора та адміністратор



Рисунок 2.1 - Функціональні схеми роботи інтерфейсів користувача, оператора та адміністратора

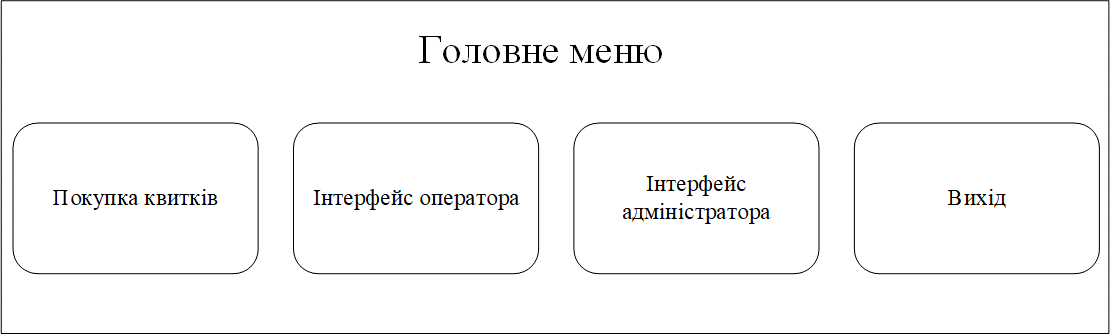


Рисунок 2.2 – Функціональна модель головного меню

Меню покупки квитків є безпосередньо інтерфейсом користувача, націленим на швидке та зрозуміле виконання поставленої їм мети. У цьому меню, рис. 2.3, користувач може вибрати між придбанням квитків, інформацією про власний баланс, а також створенням нового акаунту:



Рисунок 2.3 - Функціональна модель покупки квитків

Меню вибору поїздок (рис 2.4) просте і зрозуміле – обрання однієї з категорій, уточнення та можливість друку чеку за бажанням:



Рисунок 2.4 – Функціональна модель вибору типу квитка

Інтерфейс оператора, що наведений на рис. 2.5 націлений на майже миттєвий доступ вповноваженого персоналу станції до основної інформації про готівку та квитки.



Рисунок 2.5 - Функціональна модель інтерфейсу оператора

Працюючи з інтерфейсом адміністратора (рис. 2.6) , людина, що обслуговує пристрій, може завжди дізнатися про можливі збої в системі чи мережі, а також без особливих труднощів вивести їх у вигляді текстового файлу:



Рисунок 2.6 - Функціональна модель інтерфейсу адміністратора

* 1. **Алгоритм роботи системи**

У цьому розділі ми представимо алгоритм роботи системи, а саме алгоритм роботи інтерфейсів.

Спочатку ми розглянемо інтерфейс головного меню. Дане меню представляє собою вибір трьох інтерфейсів, з одним із яких ви надалі будете працювати. Коротко розповімо про роботу даного меню.

У головному меню ви можете вибрати тільки один інтерфейс, після чого відбувається перехід до вибраного інтерфейсу. Наприклад, якщо ви натиснете за інструкцією номер інтерфейсу, відбувається очищення даних головного меню і з’являється меню обраного інтерфейсу.

Для зручності, ми додали можливість повернення у головне меню, і ви за бажанням можете або продовжити отримання потрібної інформації або повернутись у головне меню для слідування іншим інтерфейсом чи закінчити роботу із системою зовсім.

У далі поданих схемах я показую типові алгоритми роботи різноманітних меню в створеній мною системі оплати:



Рисунок 2.7 – Блок-схема реалізації інтерфейсу головного меню

Опис блок-схеми 2.7 покроково-словесною мовою:

**Початок**

**Початок циклу**

Введіть номер інтерфейсу SWCH:

Якщо Введено «1»

Тоді Запустити інтерфейс користувача

Інакше

Якщо Введено «2»

Тоді Запустити інтерфейс оператора

Інакше

Якщо Введено «3»

Тоді Запустити інтерфейс адміністратора

Інакше повтор циклу

**Кінець циклу**

**Кінець**



Рисунок 2.8 – Блок-схема реалізації інтерфейсу користувача

Опис блок-схеми 2.8 покроково-словесною мовою:

**Початок**

**Початок циклу**

Введіть номер інтерфейсу SWCH:

Якщо Введено «1»

Тоді Придбати квиток

Інакше

Якщо Введено «2»

Тоді Дізнатись баланс

Інакше

Якщо Введено «3»

Тоді Додати нового користувача

Інакше

Якщо Введено «4»

Тоді Вийти до головного меню

Інакше повтор циклу

**Кінець циклу**

**Кінець**

Розробляючи дану систему для поліпшення роботи та оптимізації самої системи потрібно розробити певні функції, які виконуватимуть певні дії при виклику функцій у потрібний момент. Для цього я приведу у приклад функції розроблені для оптимізації та полегшення розуміння лістингу програми.

1. **main()** – функція у якій користувач має можливість зробити вибір інтерфейсу. Текст функції представлений нижче:

int main()

{

SetConsoleDisplayMode(GetStdHandle(STD\_OUTPUT\_HANDLE), CONSOLE\_FULLSCREEN\_MODE, 0);

system("cls");

setlocale(0, "ukr");

main\_menu\_interface();

main\_menu\_switch();

return 0;

}

1. **main\_menu\_interface()** – інтерфейс головного меню. Текст функції представлений нижче:

system("cls");

system("color 06");

cout << " ------------------------------------------------------------------ " << endl << endl;

cout << " | Головне меню системи PassWay - | " << endl << endl;

cout << " | система надшвидкої оплати проїзду в КП 'Київський метрополiтен' | " << endl << endl;

cout << " | тут ви зможете придбати проїзний чи поповнити iснуючий | " << endl << endl;

cout << " ------------------------------------------------------------------ " << endl << endl << endl;

cout << " ------------------------ --------------------------- " << endl << endl;

cout << " | 1) - Придбати квиток | | 2) - Увiйти як оператор | | 3) - Увiйти як адмiнiстратор | | 4) - Вийти з програми |" << endl << endl;

cout << " ------------------------ --------------------------- -------------------------------- ------------------------ " << endl << endl << endl;

cout << " Щоб здiйснити вибiр мiж пунктами меню введiть цифру та натиснiть Enter " << endl << endl;

cout << " -----------------------------------------------" << endl;

cout << " | Кiлькiсть поїздок | Цiна за одну поїздку, грн |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 1-9 | 8 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 10-19 | 7.5 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 20-29 | 7.2 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 30-39 | 7 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 40-49 | 6.5 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 50+ | 6.2 |" << endl;

cout << " -----------------------------------------------" << endl;

1. **admin\_interface()** – інтерфейс адміністратора та перевірка пароля. Текст функції представлений нижче:

string admin\_password;

string apw = "admin";

bool apwp = true;

system("cls");

do {

cout << endl << "Введiть пароль адмiнiстратора: ";

cin >> admin\_password;

cout << endl;

if (admin\_password == apw) {

system("cls");

cout << " " << " --------------------------------- " << endl;

cout << " Вiтаю, адмiнiстраторе!" << endl;

cout << " " << " --------------------------------- " << endl << endl;

cout << "" << " -------------------------------- " << endl;

cout << "" << " 1) - Тест продуктивностi системи" << " 2) - Тест з'єднання з мережею" << " 3) - Вийти до головного меню" << endl;

cout << "" << " -------------------------------- " << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

sys\_test();

break;

case 2:

network\_test();

break;

case 3:

main();

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 3) {

var = false;

}

} while (var);

}

else {

cout << " Невiрний пароль! " << endl;

}

if (admin\_password == apw) {

apwp = false;

}

} while (apwp);

1. **operator\_interface()** – інтерфейс оператора та перевірка його паролю

string operator\_password;

string opw = "oper";

bool opwp = true;

system("cls");

do {

cout << endl << "Введiть пароль оператора: ";

cin >> operator\_password;

cout << endl;

if (operator\_password == opw) {

system("cls");

cout << " --------------------------------- " << endl;

cout << " Вiтаю, операторе!" << endl;

cout << " --------------------------------- " << endl << endl;

cout << " --------------------------------------------- -------------" << endl;

cout << " 1) - Дiзнатися iнформацiю про проданi квитки 2) - Дiзнатися iнформацiю про готiвку в термiналi 3) - Змiнити вартiсть поїздок 4) - Вийти до головного меню" << endl;

cout << " --------------------------------------------- -------"<< endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

tickets\_info();

break;

case 2:

cash\_info();

break;

case 3:

change\_fare();

break;

case 4:

main();

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 4) {

var = false;

}

} while (var);

}

else {

cout << " Невiрний пароль! Перевiрте правильнiсть вводу. " << endl;

}

if (operator\_password == opw) {

opwp = false;

}

} while (opwp);

1. **user\_interface()** – інтерфейс користувача

system("cls");

cout << " " << " --------------------------------- " << endl;

cout << " Вiтаю, користувач!" << endl;

cout << " " << " --------------------------------- " << endl << endl;

cout << " --------------------- ---------------------------- " << endl;

cout << " 1) - Придбати квиток 2) - Дiзнатись баланс картки 3) - Додати користувача 4) - Вийти до головного меню" << endl;

cout << " --------------------- ---------------------------- " << endl << endl;

user\_menu\_switch();

1. **user\_balance()** – інтерфейс перевірки балансу

system("color 06");

system("cls");

int usr\_id;

cout << " ------------------------------" << endl;

cout << " Введiть свiй iдентифiкатор: " << endl;

cout << " ------------------------------" << endl << endl;

cout << " >> ";

cin >> usr\_id;

cout << endl;

cout << "Баланс користувача " << id\_user[usr\_id] << " = " << balance[usr\_id] << " грн." << endl << endl;

cout << " Натснiть 1 щоб повернутися до головного меню: " << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

user\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 1) {

var = false;

}

} while (var);

1. **user\_buy()** – інтерфейс покупки квитків

system("color 02");

system("cls");

acc\_sw();

cout << " --------------------------------------------------------- " << endl;

cout << " Виберiть кiлькiсть поїздок, на яку бажаєте придбати квиток: " << endl;

cout << " ------------------------------------------- " << endl << endl;

cout << " ------------- ----------------- ------------------------ ------" << endl;

cout << " 1) 1 поїздка 2) 2 - 9 поїздок 3) 10 та бiльше поїздок 4) Повернутися до попереднього меню" << endl;

cout << " ------------- ----------------- ----------------------- " << endl << endl;

user\_buy\_switch();

1. **buy\_again()** – запит повторної покупки квитка(ів)

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

user\_buy();

break;

case 2:

main();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

1. **buy\_1\_ticket()** – логічна схема придбання 1 квитка

int cot1;

int var = 1 \* rate[0];

cot1 = 1;

cout << "Куплено 1 квиток за " << var << " грн. Роздрукувати квитанцiю?" << endl;

count\_of\_tickets += cot1;

sum += var;

c\_o\_t += cot1;

cout << "1 - Так; 2 - Hi" << endl << endl;

is\_user\_print\_ticket();

1. **buy\_2\_5\_tickets()** – логічна схема придбання 2 – 9 квитків

int cot2;

cout << "Введiть кiлькiсть квиткiв (2-9): " << endl;

cin >> cot2;

cout << endl;

if (cot2 >= 2 && cot2 <= 9) {

count\_of\_tickets += cot2;

c\_o\_t += cot2;

int var = cot2 \* rate[1];

sum += var;

cout << "Куплено " << cot2 << " квиткiв за " << var << " грн. Роздрукувати квитанцiю?" << endl;

cout << "1 - Так; 2 - Hi" << endl << endl;

is\_user\_print\_ticket();

}

else {

cout << "Невiрна кiлькiсть! Введiть кiлькiсть вiд 2 до 9: " << endl;

buy\_2\_5\_tickets();

}

1. **buy\_10\_more\_tickets()** - логічна схема придбання 10+ квитків

int cot3;

cout << "Введiть кiлькiсть квиткiв (10+): " << endl;

cin >> cot3;

cout << endl;

count\_of\_tickets += cot3;

c\_o\_t += cot3;

int var = cot3 \* rate[2];

sum += var;

cout << "Куплено " << cot3 << " квиткiв за " << var << " грн. Роздрукувати квитанцiю?" << endl;

cout << "1 - Так; 2 - Hi" << endl << endl;

is\_user\_print\_ticket();

1. **print\_sys()** – друк інформації про систему

ofstream sys\_prt("D:\\system\_test.txt");

sys\_prt << " " << " ---------------------------------- " << endl;

sys\_prt << " Поточна характеристика системи: " << endl;

sys\_prt << " " << " ---------------------------------- " << endl << endl;

sys\_prt << " " << " ------------------ " << endl;

sys\_prt << " CPU - Процесор: " << endl;

sys\_prt << " " << " ------------------ " << endl << endl;

sys\_prt << " Поточна швидкiсть: 2.71 ГГц " << endl;

sys\_prt << " Кiлькiсть сокетiв: 1 " << endl;

sys\_prt << " Ядра: 2 " << endl;

sys\_prt << " Логiчнi процеси: 4 " << endl;

sys\_prt << " Вiзуалiзацiя: увiмкнено " << endl;

sys\_prt << " Кеш L1: 128 kB " << endl;

sys\_prt << " Кеш L2: 512 kB " << endl;

sys\_prt << " Кеш L3: 3.0 MB " << endl;

sys\_prt << " " << " -------------------------- " << endl;

sys\_prt << " RAM - Оперативна пам'ять: " << endl;

sys\_prt << " " << " -------------------------- " << endl << endl;

sys\_prt << " Загальна швидкiсть: 2133 MHz " << endl;

sys\_prt << " Форм-фактор: SODIMM " << endl;

sys\_prt << " Зарезервовано: 90.8 Mb " << endl;

sys\_prt << " Використано на даний момент: 3.7/64 Gb " << endl;

sys\_prt << " " << " ------------------ " << endl;

sys\_prt << " Drive C: - Диск С: " << endl;

sys\_prt << " " << " ------------------ " << endl << endl;

sys\_prt << " Ємнiсть: 512 Gb " << endl;

sys\_prt << " Системний: Yes " << endl;

sys\_prt << " Файл пiдкачки: Увiмкнено " << endl;

sys\_prt << " Швидкiсть зчитування даних: 302 kb/s " << endl;

sys\_prt << " Швидкiсть запису даних: 61.6 kb/s " << endl << endl;

if (sys\_prt.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

1. **print\_network()** - друк інформації про мережу

ofstream netw\_prt("D:\\network\_test.txt");

netw\_prt << " " << " --------------------------------------------- " << endl;

netw\_prt << " Поточна характеристика пiдєднаної мережi: " << endl;

netw\_prt << " " << " --------------------------------------------- " << endl << endl;

netw\_prt << " Швидкiсть з'єднання: 320 kb/s " << endl;

netw\_prt << " Пропускна здатнiсть: 24 " << endl;

netw\_prt << " Вiдпралення: 4 mb/s " << endl;

netw\_prt << " Отримання: 120 mb/s " << endl;

netw\_prt << " Адаптер: Ethernet " << endl;

netw\_prt << " IPv4 adress: 192.168.1.1 " << endl << endl;

if (netw\_prt.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

1. **print\_ticket()** – друк квитка

ofstream ticket\_print("D:\\ticket\_0.txt");

// c\_o\_t - кiлькiсть квиткiв

// sum - сума грошей

// acc - аккаунт

ticket\_print << "------------------------------" << endl;

ticket\_print << "| КП Київський метрополiтен |" << endl;

ticket\_print << "------------------------------" << endl;

ticket\_print << "| Квитанцiя |" << endl;

ticket\_print << "------------------------------" << endl;

ticket\_print << " " << asctime(timeinfo) << endl << endl;

ticket\_print << "Акаунт: " << acc << endl;

ticket\_print << "Ви придбали " << c\_o\_t << " квиткiв" << endl << endl;

ticket\_print << "Загальна сума покупки: " << sum << " грн." << endl << endl;

ticket\_print << "Дякуємо за вiзит." << endl << "Щасливої поїздки!" << endl;

if (ticket\_print.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

1. **print\_tickets\_info()** – друк інформації про квитки(для оператора)

ofstream tick\_prt("D:\\tickets\_info.txt");

tick\_prt << " ------------------------------------------ " << endl;

tick\_prt << " Загальна iнформацiя про купленi квитки: " << endl;

tick\_prt << " ------------------------------------------ " << endl << endl;

tick\_prt << " " << endl;

tick\_prt << " Кiлькість проданих квиткiв: " << c\_o\_t << endl;

if (tick\_prt.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

1. **print\_cash\_info()** – друк інформації про готівку в системі(для оператора)

ofstream cash\_prt("D:\\cash\_info.txt");

cash\_prt << " ---------------------------------------------- " << endl;

cash\_prt << " Загальна iнформацiя про наявнiсть готiвки: " << endl;

cash\_prt << " ---------------------------------------------- " << endl << endl;

cash\_prt << " " << endl;

cash\_prt << " Внесено персоналом готівки в термінал: 5000 грн." << endl;

cash\_prt << " Внесено користувачами готівки в термінал: 2318 грн." << endl;

cash\_prt << " Видано як решта: 1305 грн." << endl;

if (cash\_prt.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

1. **change\_fare()** – зміна інформації про тарифи та самих тарифів

int num\_t;

int new\_price\_t;

system("cls");

cout << " --------------------------------------------------------------- " << endl;

cout << " Щоб змiнити вартiсть квитка введiть його порядковий номер : " << endl;

cout << " --------------------------------------------------------------- " << endl << endl;

cout << " >> ";

cin >> num\_t;

// ПРОЦЕДУРУ ПЕРЕВiРКИ ВiРНОСТi НОМЕРУ БУДЕ ДОДАНО ЗГОДОМ

cout << endl;

cout << "Зараз квитку пiд номером " << num\_t << " назначена цiна " << rate[num\_t] << " грн." << endl << endl;

cout << "Введiть нову цiну: ";

cin >> new\_price\_t;

cout << endl;

rate[num\_t] = new\_price\_t;

cout << "Цiну змiнено усппiшно!" << endl << endl;

cout << "Натиснiть 1 щоб повернутися до попереднього меню " << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

operator\_interface();

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 1) {

var = false;

}

} while (var);

1. **is\_user\_print\_ticket()** – перевірка можливості друку і вибір меню для різних випадків

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

print\_ticket();

break;

case 2:

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

1. **user\_buy\_switch()** – пропозиція придбати більше квитків

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

buy\_1\_ticket();

cout << "Купити ще?" << endl;

cout << "1 - Так; 2 - Hi, вийти до головного меню" << endl << endl;

buy\_again();

break;

case 2:

buy\_2\_5\_tickets();

cout << "Купити ще?" << endl;

cout << "1 - Так; 2 - Hi, вийти до головного меню" << endl << endl;

buy\_again();

break;

case 3:

buy\_10\_more\_tickets();

cout << "Купити ще?" << endl;

cout << "1 - Так; 2 - Hi, вийти до головного меню" << endl << endl;

buy\_again();

case 4:

user\_interface();

//user\_menu\_switch();

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 4) {

var = false;

}

} while (var);

1. **ПРОЕКТУВАННЯ СИСТЕМИ**
2. **Проектування інтерфейсу системи**

Основні функціональні можливості реалізованого в системі інтерфейсу користувача виконують функції розроблені для оптимальної роботи з програмою. При запуску програми користувач має нагоду вибрати один з інтерфейсів представлених у головному меню. Після вибору інтерфейсу користувача, спрацьовує функція, яка надає вибір подальших дій спрямованих на отримання конкретної інформації. Вибір проводиться для переходу з інтерфейсу користувача до подальших інтерфейсів, які містять детальну інформацію на вибрану користувачем тему. Серед вибіркових інтерфейсів представлено інтерфейс, який містить інформацію про групи, з подальшим переходом до сегментів з більш докладнішою інформацією; інтерфейс, який містить інформацію про вартість квитків; інтерфейс, який містить інформацію про баланс того чи іншого користувача; також маємо змогу повернутися до головного меню. При обранні одного із запропонованих інтерфейсів спрацьовують допоміжні функції, які містять сегменти вибору SWITCH та можливість виведення отриманої інформації у файл текстового типу. В результаті інтерфейс користувача має універсальну модуляцію програми яку при бажанні можна розширити або ускладнити.

Основні функціональні можливості розробленого інтерфейсу оператора:

- доступ до інформації про наявність готівки в терміналі;

- доступ до інформації про продані квитки;

- перегляд наявності інформації у файлі текстового типу;

- вихід у головне меню

Основні функціональні можливості розробленого інтерфейсу адміністратора:

* тест продуктивності системи;
* тест з’єднання з мережею;
* перегляд наявності інформації у файлі текстового типу;
* вихід у головне меню

Вище представлений короткий перелік можливостей оператора, який надає програмне забезпечення, розроблене з метою оснащення апарату типу «iBox» для обслуговування пасажирів комунального підприємства «Київський метрополітен». Інтерфейс оператора працює завдяки функціям, які розроблялися спеціально під дану програму. Після переходу до інтерфейсу оператора, необхідно ввести пароль, який надає доступ до подальших дій. Після введення пароля ви маєте змогу виконати три дії, але не одночасно. Наприклад ви можете переглянути кількість проданих квитків, або змінити ціну на певний тип квитка. Після виконання однієї із операцій, програма автоматично завершить свою роботу, і для того щоб зробити знову якісь дії вам потрібно запустити програму спочатку.

1. **Тестування системи**

В розробленому програмному забезпеченні організовано та передбачено роботу таких основних інтерфейсів:

- інтерфейсу головного меню;

- інтерфейсу користувача;

- інтерфейсу оператора;

Протестуємо роботу інтерфейсу головного меню, який надає можливість обрати або інтерфейс користувача, або інтерфейс оператора. Ми маємо можливість протестувати його за поданим нижче алгоритмом його дій:

1. натиснути кнопку для початку роботи програми;

2. після запуску програми здійснити вибір для переходу до головного меню;

3. обрати або інтерфейс користувача або інтерфейс оператора;

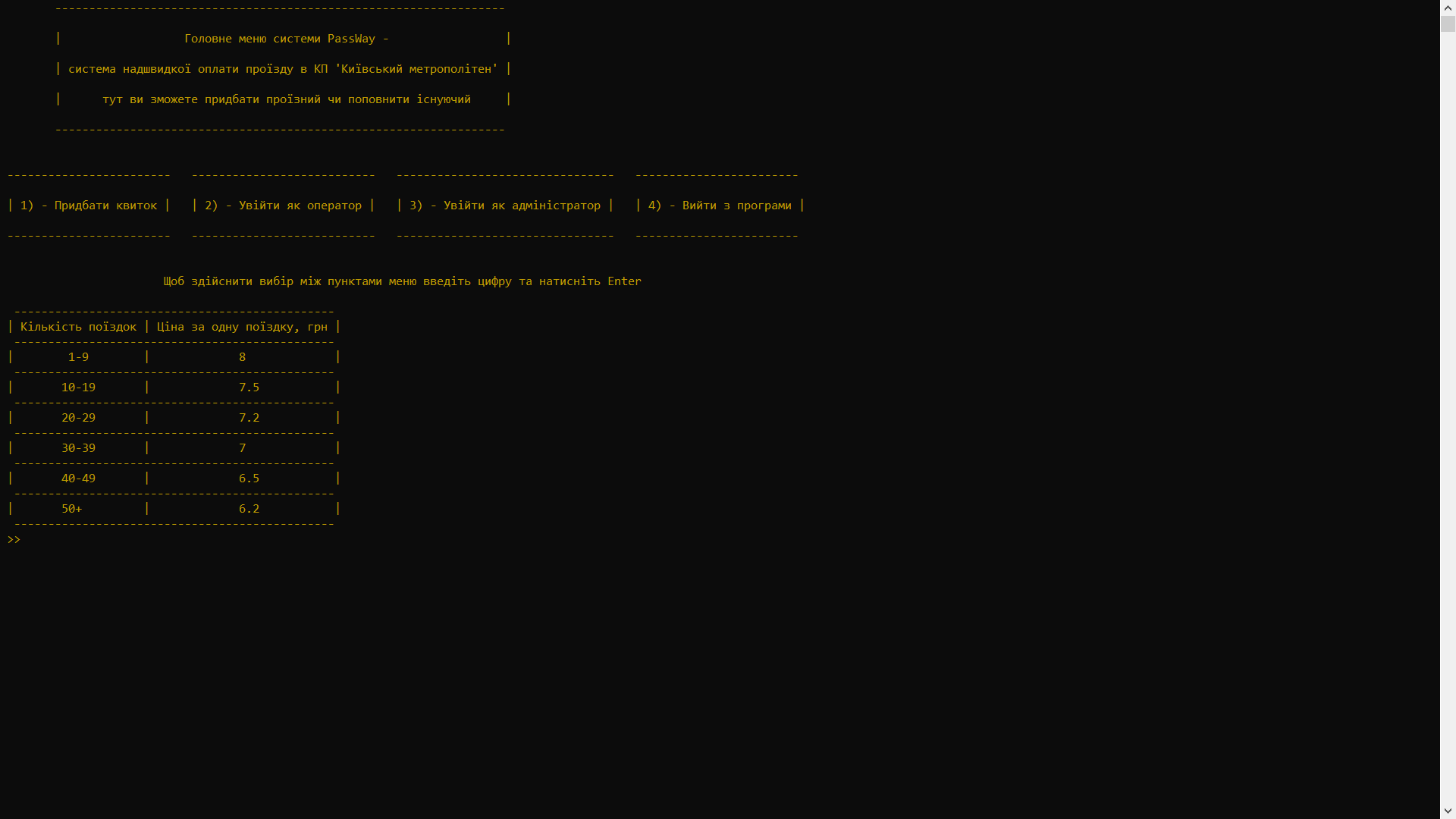


Рисунок 3.1 – головне меню програми

Таку саму схему тестування функцій програми можна застосувати і до інтерфейсів користувача, адміністратора та оператора.

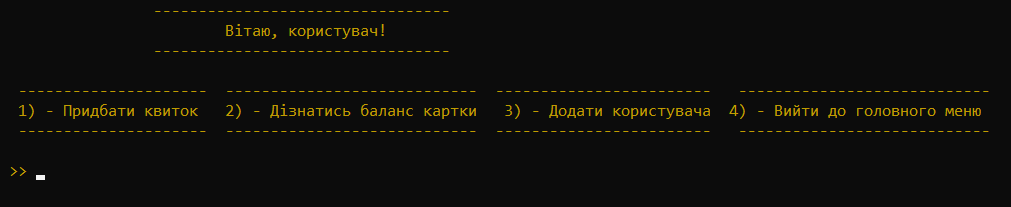


Рисунок 3.2 – Інтерфейс користувача

Під час тестування можливостей придбання квитків ми можемо скористатися тестовим акаунтом з певною сумою коштів на рахунку. Тестування здійснюється вибором різноманітної кількості можливих і неможливих варіантів вводу даних.

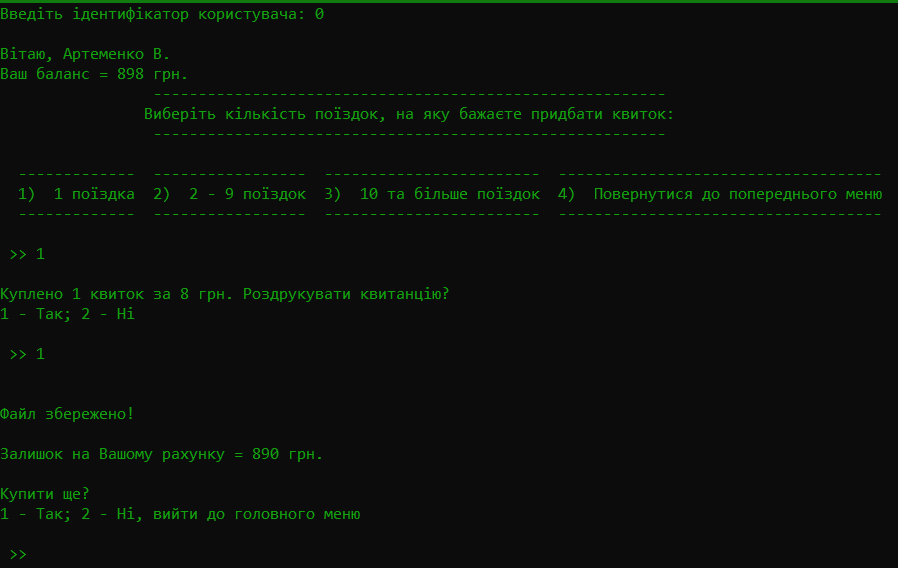


Рисунок 3.3 – Приклад використання меню вибору квитків ціною 8 гривень

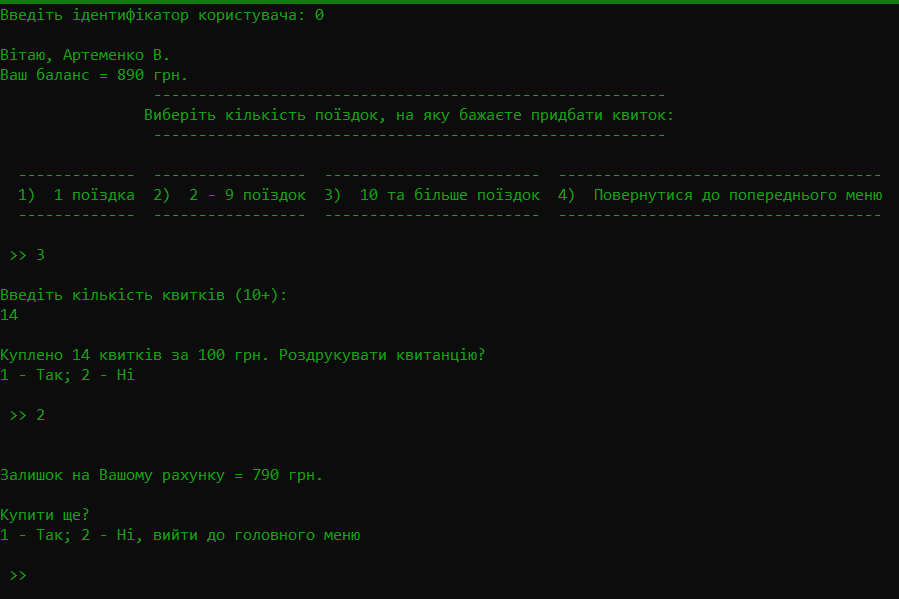


Рисунок 3.4 – Приклад використання меню вибору квитків в різних цінових діапазонах

Під час тестування інтерфейсів, що потребують вводу паролю ми можемо спробувати підібрати пароль та впевнитися у безпеці збереження даних.

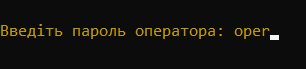


Рисунок 3.5 – перевірка паролю оператора

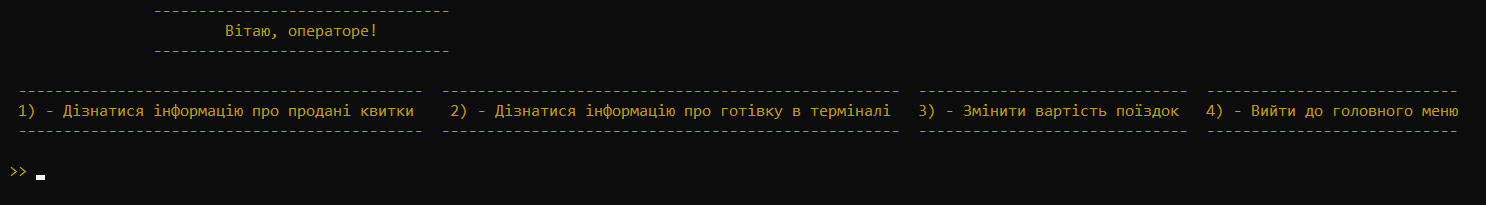


Рисунок 3.6 – інтерфейс оператора

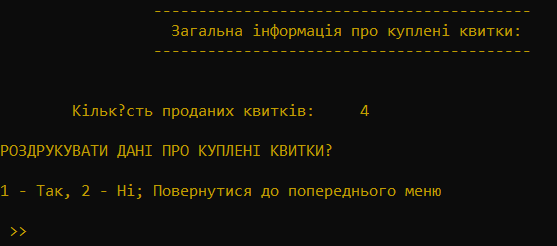


Рисунок 3.7 – меню інформації про квитки

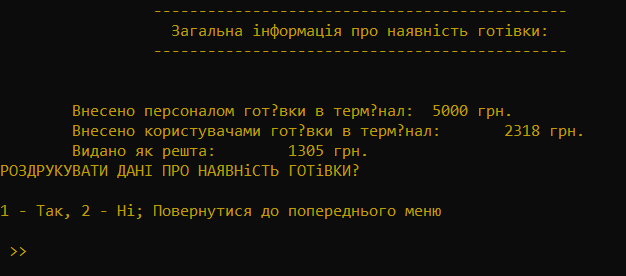


Рисунок 3.8 – меню інформації про готівку

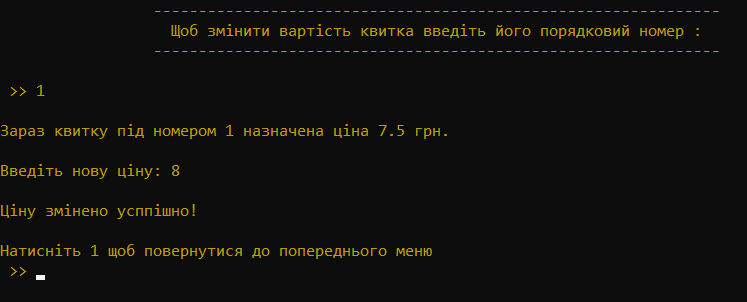


Рисунок 3.9 – меню зміни вартості квитків

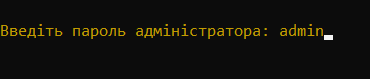


Рисунок 3.10 – перевірка паролю адміністратора

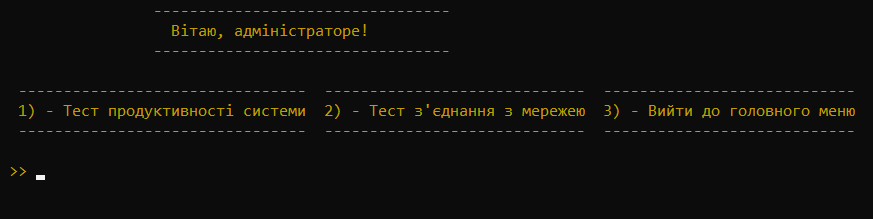


Рисунок 3.11 – інтерфейс адміністратора

Тестуючи функції які передбачають виведення масивів з інформацією про систему чи мережу ми можемо змінити чи вилучити деякі фізичні складові апаратів для перевірки вірності даних, що виводяться адміністратору.

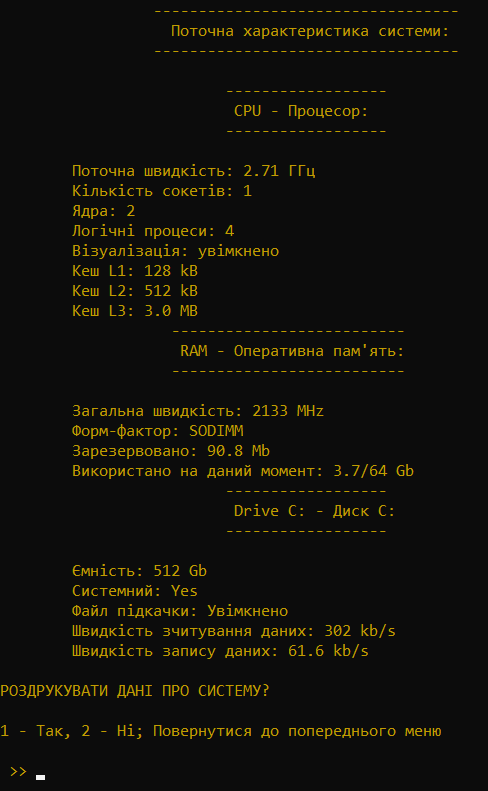


Рисунок 3.12 – перегляд загальної інформації про систему

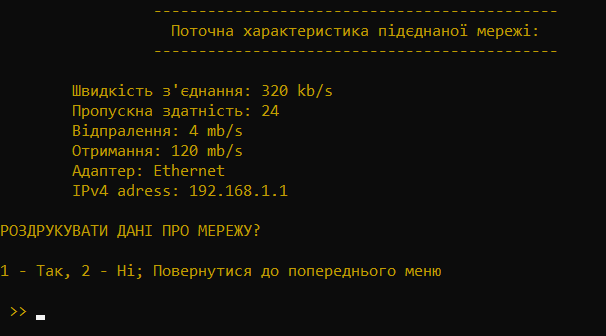


Рисунок 3.13 – перегляд інформації про мережу

1. **Вимоги щодо встановленого програмного забезпечення**

При встановленні розробленого програмного забезпечення необхідно врахувати наступні вимоги до операційної системи:

- Windows 7, 8, 8.1, 10;

- тип операційної системи x64-bit;

- мінімальна частота процесора 2,0 ГГц;

- встановлена оперативна пам’ять (ОЗП): 2,00 Гб;

- кількість вільного місця на HDD: 500 Мб;

Для коректної роботи та правильності функціонування проектованої системи, необхідно врахувати додаткові вимоги щодо встановленого програмного забезпечення:

- Microsoft Visual Studio 2019(X64-bit);

- Пакет основних бібліотек мови С++;

Склад програмного забезпечення необхідного для роботи проектної системи, а саме:

1. Локальні апарати доступу користувачів:
   1. Термінали типу «iBox»
   2. Термінали типу «PassWay»
2. Серверне устаткування:
   1. Мережеві сервери обробки даних
   2. Сервери із збільшеним можливим об’ємом збереження даних
   3. Сервери, що забезпечуватимуть роботу у разі виходу з ладу основних серверів

**ВИСНОВКИ**

В даній роботі були проаналізовані питання відносно теоретичної основи побудови програм з простим інтерфейсом користувача в середовищі Microsoft Visual Studio 2019, а також описані проблеми, які виникають при цьому.

Система може з успіхом використовуватися на будь-яких станціях КП «Київський метрополітен». Спрямованість на зменшення витрат користувачів стає одним із ключових факторів встановлення цієї системи.

При розробці концепції системи були вивчені принципи побудови аналогічних систем провідними європейськими виробниками і максимально врахована специфіка її експлуатації в вітчизняних умовах. Це, насамперед, відноситься до локалізації та типу розрахунків.

Гранична увага була приділена захищеності системи, в якій використовуються над технологічні засоби захисту персональної інформації провідних фірм ряду європейських держав.

Відмітні особливості системи:

* + компактність пристроїв, на яких вона може запуститися;
  + швидкість роботи;
  + можливість всебічної діагностики;
  + можливість оплати за разовими квитками, абонементними та дебетними картками;
  + гнучка система завдання тарифів, механізм надання знижок;
  + захищеність від зловживань з боку користувачів і персоналу.

**ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ПОСИЛАНЬ**

1. Система оплати проїзду КП "Київський метрополітен" – Режим доступу: <http://www.metro.kiev.ua/node/106>, 2019
2. Київський метрополітен - Вікіпедія – Режим доступу: <http://tiny.cc/kievmetro>, 2019
3. Термінали типу "iBox" - Вікіпедія – Режим доступу: <http://tiny.cc/terminalss>, 2019
4. Функціональні можливості системи "iBox" – Режим доступу: <https://ibox.ua/>, 2018
5. Синтаксична конструкція циклів в С++ – Режим доступу: <http://tiny.cc/g7sw7y>, 2017
6. Синтаксис функцій в С++ – Режим доступу: <http://tiny.cc/3atw7y>, 2017
7. Масиви та контейнери – Режим доступу: <http://tiny.cc/lgtw7y>, 2017
8. Робота з файлами в С++ – Режим доступу: <http://cppstudio.com/uk/post/446/>, 2017
9. Локальні і глобальні змінні в С++ – Режим доступу: <http://cppstudio.com/uk/post/415/>, 2018
10. Прототипи функцій в С++ – Режим доступу: <http://cppstudio.com/uk/post/404/>, 2018
11. Оператори управління потоком виконання програм – Режим доступу: <http://tiny.cc/potokk>, 2018

**ДОДАТКИ**

**Додаток А. Лістинг розробленого програмного забезпечення**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <cmath>

#include <ctime>

#include <iomanip>

#include <fstream>

#include <windows.h>

#include <array>

#include <vector>

using namespace std;

// announcement of function

int main();

void main\_menu\_interface();

void main\_menu\_switch();

void user\_menu\_switch();

void user\_interface();

void admin\_interface();

void user\_buy();

void user\_balance();

void user\_interface();

void operator\_interface();

void tickets\_info();

void cash\_info();

void ticket\_info\_switch();

void cash\_info\_switch();

void print\_tickets\_info();

void print\_cash\_info();

void add\_user();

void is\_user\_print\_ticket();

void print\_ticket();

void change\_fare();

void go\_back\_cash();

// GLOBAL VARIABLES & ARRAYS

int c\_o\_t;

int sum;

string acc;

int bal;

int urr;

int count\_of\_tickets = 0;

float rate[] = { 8, 7.5, 7.2 };

vector<string> id\_user = { "Артеменко В.", "Маудза О.", "Паплiнський В.", "Летич А.", "Жидков Д.", "Кисiль Т.М." };

int balance[] = { 898, 315, 391, 213, 781, 1251 };

int c\_bal = sizeof(balance)/sizeof(balance[0]);

int sum\_bal;

time\_t seconds = time(NULL);

tm\* timeinfo = localtime(&seconds);

int vneseno = 5000;

int reshta = 1305;

// FILES TO PRINT

void print\_sys() {

ofstream sys\_prt("D:\\system\_test.txt");

sys\_prt << " " << " ---------------------------------- " << endl;

sys\_prt << " Поточна характеристика системи: " << endl;

sys\_prt << " " << " ---------------------------------- " << endl << endl;

sys\_prt << " " << " ------------------ " << endl;

sys\_prt << " CPU - Процесор: " << endl;

sys\_prt << " " << " ------------------ " << endl << endl;

sys\_prt << " Поточна швидкiсть: 2.71 ГГц " << endl;

sys\_prt << " Кiлькiсть сокетiв: 1 " << endl;

sys\_prt << " Ядра: 2 " << endl;

sys\_prt << " Логiчнi процеси: 4 " << endl;

sys\_prt << " Вiзуалiзацiя: увiмкнено " << endl;

sys\_prt << " Кеш L1: 128 kB " << endl;

sys\_prt << " Кеш L2: 512 kB " << endl;

sys\_prt << " Кеш L3: 3.0 MB " << endl;

sys\_prt << " " << " -------------------------- " << endl;

sys\_prt << " RAM - Оперативна пам'ять: " << endl;

sys\_prt << " " << " -------------------------- " << endl << endl;

sys\_prt << " Загальна швидкiсть: 2133 MHz " << endl;

sys\_prt << " Форм-фактор: SODIMM " << endl;

sys\_prt << " Зарезервовано: 90.8 Mb " << endl;

sys\_prt << " Використано на даний момент: 3.7/64 Gb " << endl;

sys\_prt << " " << " ------------------ " << endl;

sys\_prt << " Drive C: - Диск С: " << endl;

sys\_prt << " " << " ------------------ " << endl << endl;

sys\_prt << " Ємнiсть: 512 Gb " << endl;

sys\_prt << " Системний: Yes " << endl;

sys\_prt << " Файл пiдкачки: Увiмкнено " << endl;

sys\_prt << " Швидкiсть зчитування даних: 302 kb/s " << endl;

sys\_prt << " Швидкiсть запису даних: 61.6 kb/s " << endl << endl;

if (sys\_prt.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

}

void print\_network() {

ofstream netw\_prt("D:\\network\_test.txt");

netw\_prt << " " << " --------------------------------------------- " << endl;

netw\_prt << " Поточна характеристика пiдєднаної мережi: " << endl;

netw\_prt << " " << " --------------------------------------------- " << endl << endl;

netw\_prt << " Швидкiсть з'єднання: 320 kb/s " << endl;

netw\_prt << " Пропускна здатнiсть: 24 " << endl;

netw\_prt << " Вiдпралення: 4 mb/s " << endl;

netw\_prt << " Отримання: 120 mb/s " << endl;

netw\_prt << " Адаптер: Ethernet " << endl;

netw\_prt << " IPv4 adress: 192.168.1.1 " << endl << endl;

if (netw\_prt.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

}

void print\_ticket() {

ofstream ticket\_print("D:\\ticket\_0.txt");

// c\_o\_t - кiлькiсть квиткiв

// sum - сума грошей

// acc - аккаунт

ticket\_print << "------------------------------" << endl;

ticket\_print << "| КП Київський метрополiтен |" << endl;

ticket\_print << "------------------------------" << endl;

ticket\_print << "| Квитанцiя |" << endl;

ticket\_print << "------------------------------" << endl;

ticket\_print << " " << asctime(timeinfo) << endl << endl;

ticket\_print << "Акаунт: " << acc << endl;

ticket\_print << "Ви придбали " << c\_o\_t << " квиткiв" << endl << endl;

ticket\_print << "Загальна сума покупки: " << sum << " грн." << endl << endl;

ticket\_print << "Дякуємо за вiзит." << endl << "Щасливої поїздки!" << endl;

if (ticket\_print.is\_open())

cout << endl << "Файл збережено!" << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl;

};

void print\_tickets\_info() {

ofstream tick\_prt("D:\\tickets\_info.txt");

tick\_prt << " ------------------------------------------ " << endl;

tick\_prt << " Загальна iнформацiя про купленi квитки: " << endl;

tick\_prt << " ------------------------------------------ " << endl << endl;

tick\_prt << " " << endl;

tick\_prt << " Кiлькiсть проданих квиткiв: " << c\_o\_t << endl;

if (tick\_prt.is\_open())

cout << endl << "Файл збережено!" << endl << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl << endl;

};

void print\_cash\_info() {

ofstream cash\_prt("D:\\cash\_info.txt");

cash\_prt << " ---------------------------------------------- " << endl;

cash\_prt << " Загальна iнформацiя про наявнiсть готiвки: " << endl;

cash\_prt << " ---------------------------------------------- " << endl << endl;

cash\_prt << " " << endl;

cash\_prt << " Внесено персоналом готiвки в термiнал: " << vneseno << " грн." << endl;

cash\_prt << " Внесено користувачами готiвки в термiнал: " << sum\_bal << " грн." << endl;

cash\_prt << " Видано як решта: " << reshta << " грн." << endl << endl;

if (cash\_prt.is\_open())

cout << endl << "Файл збережено!" << endl << endl;

else

cout << endl << "Неможливо зберегти файл!" << endl << endl;

};

// TESTS OF SYSTEM & NETWORK

void sys\_test() {

system("cls");

cout << " " << " ---------------------------------- " << endl;

cout << " Поточна характеристика системи: " << endl;

cout << " " << " ---------------------------------- " << endl << endl;

cout << " " << " ------------------ " << endl;

cout << " CPU - Процесор: " << endl;

cout << " " << " ------------------ " << endl << endl;

cout << " Поточна швидкiсть: 2.71 ГГц " << endl;

cout << " Кiлькiсть сокетiв: 1 " << endl;

cout << " Ядра: 2 " << endl;

cout << " Логiчнi процеси: 4 " << endl;

cout << " Вiзуалiзацiя: увiмкнено " << endl;

cout << " Кеш L1: 128 kB " << endl;

cout << " Кеш L2: 512 kB " << endl;

cout << " Кеш L3: 3.0 MB " << endl;

cout << " " << " -------------------------- " << endl;

cout << " RAM - Оперативна пам'ять: " << endl;

cout << " " << " -------------------------- " << endl << endl;

cout << " Загальна швидкiсть: 2133 MHz " << endl;

cout << " Форм-фактор: SODIMM " << endl;

cout << " Зарезервовано: 90.8 Mb " << endl;

cout << " Використано на даний момент: 3.7/64 Gb " << endl;

cout << " " << " ------------------ " << endl;

cout << " Drive C: - Диск С: " << endl;

cout << " " << " ------------------ " << endl << endl;

cout << " Ємнiсть: 512 Gb " << endl;

cout << " Системний: Yes " << endl;

cout << " Файл пiдкачки: Увiмкнено " << endl;

cout << " Швидкiсть зчитування даних: 302 kb/s " << endl;

cout << " Швидкiсть запису даних: 61.6 kb/s " << endl << endl;

cout << "РОЗДРУКУВАТИ ДАНI ПРО СИСТЕМУ?" << endl << endl;

cout << "1 - Так, 2 - Нi; Повернутися до попереднього меню" << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

print\_sys();

break;

case 2:

admin\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

void network\_test() {

system("cls");

cout << " " << " --------------------------------------------- " << endl;

cout << " Поточна характеристика пiдєднаної мережi: " << endl;

cout << " " << " --------------------------------------------- " << endl << endl;

cout << " Швидкiсть з'єднання: 320 kb/s " << endl;

cout << " Пропускна здатнiсть: 24 " << endl;

cout << " Вiдпралення: 4 mb/s " << endl;

cout << " Отримання: 120 mb/s " << endl;

cout << " Адаптер: Ethernet " << endl;

cout << " IPv4 adress: 192.168.1.1 " << endl << endl;

cout << "РОЗДРУКУВАТИ ДАНI ПРО МЕРЕЖУ?" << endl << endl;

cout << "1 - Так, 2 - Нi; Повернутися до попереднього меню" << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

print\_network();

break;

case 2:

admin\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

// FOR OPERATOR

void tickets\_info() {

system("cls");

cout << " ------------------------------------------ " << endl;

cout << " Загальна iнформацiя про купленi квитки: " << endl;

cout << " ------------------------------------------ " << endl << endl;

cout << " " << endl;

cout << " Кiлькiсть проданих квиткiв: " << c\_o\_t << endl << endl;

ticket\_info\_switch();

};

void cash\_info() {

system("cls");

for (int i = 0; i < 6; i++) {

sum\_bal += balance[i];

}

cout << " ---------------------------------------------- " << endl;

cout << " Загальна iнформацiя про наявнiсть готiвки: " << endl;

cout << " ---------------------------------------------- " << endl << endl;

cout << " " << endl;

cout << " Внесено персоналом готiвки в термiнал: " << vneseno << " грн." << endl;

cout << " Внесено користувачами готiвки в термiнал: " << sum\_bal << " грн." << endl;

cout << " Видано як решта: " << reshta << " грн." << endl << endl;

cash\_info\_switch();

}

void change\_fare() {

int num\_t;

float new\_price\_t;

system("cls");

cout << " --------------------------------------------------------------- " << endl;

cout << " Щоб змiнити вартiсть квитка введiть його порядковий номер : " << endl;

cout << " --------------------------------------------------------------- " << endl << endl;

cout << " >> ";

cin >> num\_t;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

change\_fare();

return;

}

// ПРОЦЕДУРУ ПЕРЕВiРКИ ВiРНОСТi НОМЕРУ БУДЕ ДОДАНО ЗГОДОМ

cout << endl;

cout << "Зараз квитку пiд номером " << num\_t << " назначена цiна " << rate[num\_t] << " грн." << endl << endl;

cout << "Введiть нову цiну: ";

cin >> new\_price\_t;

cout << endl;

rate[num\_t] = new\_price\_t;

cout << "Цiну змiнено успiшно!" << endl << endl;

cout << "Натиснiть 1 щоб повернутися до попереднього меню: " << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

operator\_interface();

/\*case 2:

change\_fare();\*/

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 1) {

var = false;

}

} while (var);

}

// FOR USER

void add\_user() {

// НЕ ПРАЦЮЄ

string add\_nm;

int add\_bal;

cout << "Введіть ім'я: ";

cin >> add\_nm;

cout << endl;

cout << "Введіть баланс: ";

cin >> add\_bal;

cout << endl;

id\_user[c\_bal+1] += add\_nm;

balance[c\_bal+1] = add\_bal;

}

void buy\_again() {

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

user\_buy();

break;

case 2:

main();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

void buy\_1\_ticket() {

int cot1;

int var = 1 \* rate[0];

if (bal >= var) {

cot1 = 1;

cout << "Куплено 1 квиток за " << var << " грн. Роздрукувати квитанцiю?" << endl;

count\_of\_tickets += cot1;

sum += var;

c\_o\_t += cot1;

bal -= var;

cout << "1 - Так; 2 - Hi" << endl << endl;

is\_user\_print\_ticket();

balance[urr] = bal;

cout << endl << "Залишок на Вашому рахунку = " << bal << " грн." << endl << endl;

}

else {

cout << "Недостатньо коштiв на рахунку!" << endl << endl;

}

}

void buy\_2\_5\_tickets() {

int cot2;

cout << "Введiть кiлькiсть квиткiв (2-9): " << endl;

cin >> cot2;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

buy\_2\_5\_tickets();

return;

}

int var = cot2 \* rate[1];

if (bal >= var) {

if (cot2 >= 2 && cot2 <= 9) {

count\_of\_tickets += cot2;

c\_o\_t += cot2;

sum += var;

bal -= var;

cout << "Куплено " << cot2 << " квиткiв за " << var << " грн. Роздрукувати квитанцiю?" << endl;

cout << "1 - Так; 2 - Hi" << endl << endl;

is\_user\_print\_ticket();

balance[urr] = bal;

cout << endl << "Залишок на Вашому рахунку = " << bal << " грн." << endl << endl;

}

else {

cout << "Невiрна кiлькiсть! Введiть кiлькiсть вiд 2 до 9: " << endl;

buy\_2\_5\_tickets();

}

}

else {

cout << "Недостатньо коштiв на рахунку!" << endl << endl;

}

};

void buy\_10\_more\_tickets() {

int cot3;

cout << "Введiть кiлькiсть квиткiв (10+): " << endl;

cin >> cot3;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

buy\_10\_more\_tickets();

return;

}

int var = cot3 \* rate[2];

if (bal >= var) {

if (cot3 >= 10) {

count\_of\_tickets += cot3;

c\_o\_t += cot3;

sum += var;

bal -= var;

cout << "Куплено " << cot3 << " квиткiв за " << var << " грн. Роздрукувати квитанцiю?" << endl;

cout << "1 - Так; 2 - Hi" << endl << endl;

is\_user\_print\_ticket();

balance[urr] = bal;

cout << endl << "Залишок на Вашому рахунку = " << bal << " грн." << endl << endl;

}

else {

cout << "Невiрна кiлькiсть! " << endl;

buy\_10\_more\_tickets();

}

}

else {

cout << "Недостатньо коштiв на рахунку!" << endl << endl;

}

}

// SWITCHES

void is\_user\_print\_ticket() {

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

print\_ticket();

break;

case 2:

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

void user\_buy\_switch() {

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

buy\_1\_ticket();

cout << "Купити ще?" << endl;

cout << "1 - Так; 2 - Hi, вийти до головного меню" << endl << endl;

buy\_again();

break;

case 2:

buy\_2\_5\_tickets();

cout << "Купити ще?" << endl;

cout << "1 - Так; 2 - Hi, вийти до головного меню" << endl << endl;

buy\_again();

break;

case 3:

buy\_10\_more\_tickets();

cout << "Купити ще?" << endl;

cout << "1 - Так; 2 - Hi, вийти до головного меню" << endl << endl;

buy\_again();

case 4:

user\_interface();

//user\_menu\_switch();

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 4) {

var = false;

}

} while (var);

};

void user\_menu\_switch() {

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

user\_buy();

break;

case 2:

user\_balance();

break;

case 3:

cout << "Ця функцiя буде доступна у найближчих оновленнях програми. Чекайте!" << endl << endl;

user\_menu\_switch();

//add\_user();

break;

case 4:

main();

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 3) {

var = false;

}

} while (var);

}

void main\_menu\_switch() {

system("color 06");

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

user\_interface();

break;

case 2:

operator\_interface();

break;

case 3:

admin\_interface();

break;

case 4:

exit(0);

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 3) {

var = false;

}

} while (var);

}

void go\_back\_ticket() {

cout << "Натиснiть 1 щоб повернутися до попереднього меню: " << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

operator\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

void ticket\_info\_switch() {

cout << "РОЗДРУКУВАТИ ДАНI ПРО КУПЛЕНI КВИТКИ?" << endl << endl;

cout << "1 - Так, 2 - Нi; Повернутися до попереднього меню" << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

print\_tickets\_info();

go\_back\_ticket();

break;

case 2:

operator\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

void go\_back\_cash() {

cout << "Натиснiть 1 щоб повернутися до попереднього меню: " << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

operator\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

void cash\_info\_switch() {

cout << "РОЗДРУКУВАТИ ДАНI ПРО НАЯВНIСТЬ ГОТIВКИ?" << endl << endl;

cout << "1 - Так, 2 - Нi; Повернутися до попереднього меню" << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

print\_cash\_info();

go\_back\_cash();

break;

case 2:

operator\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 2) {

var = false;

}

} while (var);

}

void acc\_sw() {

int acc\_idd;

cout << "Введiть iдентифiкатор користувача: ";

cin >> acc\_idd;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

acc\_sw();

return;

}

if (acc\_idd < c\_bal) {

acc = id\_user[acc\_idd];

cout << "Вiтаю, " << id\_user[acc\_idd] << endl;

bal = balance[acc\_idd];

cout << "Ваш баланс = " << bal << " грн." << endl;

urr = acc\_idd;

}

else {

cout << "Такого користувача не iснує!" << endl << endl;

acc\_sw();

}

}

// INTERFACES

// USER

void user\_buy() {

system("color 02");

system("cls");

acc\_sw();

cout << " --------------------------------------------------------- " << endl;

cout << " Виберiть кiлькiсть поїздок, на яку бажаєте придбати квиток: " << endl;

cout << " --------------------------------------------------------- " << endl << endl;

cout << " ------------- ----------------- ------------------------ ------------------------------------ " << endl;

cout << " 1) 1 поїздка 2) 2 - 9 поїздок 3) 10 та бiльше поїздок 4) Повернутися до попереднього меню" << endl;

cout << " ------------- ----------------- ------------------------ ------------------------------------ " << endl << endl;

user\_buy\_switch();

}

void user\_balance() {

system("color 06");

system("cls");

int usr\_id;

cout << " ------------------------------" << endl;

cout << " Введiть свiй iдентифiкатор: " << endl;

cout << " ------------------------------" << endl << endl;

cout << " >> ";

cin >> usr\_id;

cout << endl;

cout << "Баланс користувача " << id\_user[usr\_id] << " = " << balance[usr\_id] << " грн." << endl << endl;

cout << " Натснiть 1 щоб повернутися до головного меню: " << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

user\_interface();

break;

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 1) {

var = false;

}

} while (var);

}

void user\_interface() {

system("cls");

cout << " " << " --------------------------------- " << endl;

cout << " Вiтаю, користувач!" << endl;

cout << " " << " --------------------------------- " << endl << endl;

cout << " --------------------- ---------------------------- ------------------------ ---------------------------- " << endl;

cout << " 1) - Придбати квиток 2) - Дiзнатись баланс картки 3) - Додати користувача 4) - Вийти до головного меню" << endl;

cout << " --------------------- ---------------------------- ------------------------ ---------------------------- " << endl << endl;

user\_menu\_switch();

};

// OPERATOR

void operator\_interface() {

string operator\_password;

string opw = "oper";

bool opwp = true;

system("cls");

do {

cout << endl << "Введiть пароль оператора: ";

cin >> operator\_password;

cout << endl;

if (operator\_password == opw) {

system("cls");

cout << " --------------------------------- " << endl;

cout << " Вiтаю, операторе!" << endl;

cout << " --------------------------------- " << endl << endl;

cout << " --------------------------------------------- --------------------------------------------------- ------------------------------ ----------------------------" << endl;

cout << " 1) - Дiзнатися iнформацiю про проданi квитки 2) - Дiзнатися iнформацiю про готiвку в термiналi 3) - Змiнити вартiсть поїздок 4) - Вийти до головного меню" << endl;

cout << " --------------------------------------------- --------------------------------------------------- ------------------------------ ----------------------------" << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

switch (menu\_answer)

{

case 1:

tickets\_info();

break;

case 2:

cash\_info();

break;

case 3:

change\_fare();

break;

case 4:

main();

/\*case 5:

add\_user();\*/

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 4) {

var = false;

}

} while (var);

}

else {

cout << " Невiрний пароль! Перевiрте правильнiсть вводу. " << endl;

}

if (operator\_password == opw) {

opwp = false;

}

} while (opwp);

}

// ADMIN

void admin\_interface() {

string admin\_password;

string apw = "admin";

bool apwp = true;

system("cls");

do {

cout << endl << "Введiть пароль адмiнiстратора: ";

cin >> admin\_password;

cout << endl;

if (admin\_password == apw) {

system("cls");

cout << " " << " --------------------------------- " << endl;

cout << " Вiтаю, адмiнiстраторе!" << endl;

cout << " " << " --------------------------------- " << endl << endl;

cout << "" << " -------------------------------- " << " ----------------------------- " << " ---------------------------- " << endl;

cout << "" << " 1) - Тест продуктивностi системи" << " 2) - Тест з'єднання з мережею" << " 3) - Вийти до головного меню" << endl;

cout << "" << " -------------------------------- " << " ----------------------------- " << " ---------------------------- " << endl << endl;

int menu\_answer = 0, again\_menu = 0;

bool var = true, var1 = true;

do {

cout << " >> ";

cin >> menu\_answer;

cout << endl;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

continue;

}

switch (menu\_answer)

{

case 1:

sys\_test();

break;

case 2:

network\_test();

break;

case 3:

main();

default:

cout << endl << "Невiрно вибраний " << menu\_answer << "-й пункт!" << endl << endl;

}

if (menu\_answer >= 1 && menu\_answer <= 3) {

var = false;

}

} while (var);

}

else {

cout << " Невiрний пароль! " << endl;

}

if (admin\_password == apw) {

apwp = false;

}

} while (apwp);

}

//MENU

void main\_menu\_interface() {

system("cls");

system("color 06");

cout << " ------------------------------------------------------------------ " << endl << endl;

cout << " | Головне меню системи PassWay - | " << endl << endl;

cout << " | система надшвидкої оплати проїзду в КП 'Київський метрополiтен' | " << endl << endl;

cout << " | тут ви зможете придбати проїзний чи поповнити iснуючий | " << endl << endl;

cout << " ------------------------------------------------------------------ " << endl << endl << endl;

cout << " ------------------------ --------------------------- -------------------------------- ------------------------ " << endl << endl;

cout << " | 1) - Придбати квиток | | 2) - Увiйти як оператор | | 3) - Увiйти як адмiнiстратор | | 4) - Вийти з програми |" << endl << endl;

cout << " ------------------------ --------------------------- -------------------------------- ------------------------ " << endl << endl << endl;

cout << " Щоб здiйснити вибiр мiж пунктами меню введiть цифру та натиснiть Enter " << endl << endl;

cout << " -----------------------------------------------" << endl;

cout << " | Кiлькiсть поїздок | Цiна за одну поїздку, грн |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 1-9 | 8 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 10-19 | 7.5 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 20-29 | 7.2 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 30-39 | 7 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 40-49 | 6.5 |" << endl;

cout << " -----------------------------------------------" << endl;

cout << " | 50+ | 6.2 |" << endl;

cout << " -----------------------------------------------" << endl;

}

int main()

{

SetConsoleDisplayMode(GetStdHandle(STD\_OUTPUT\_HANDLE), CONSOLE\_FULLSCREEN\_MODE, 0);

system("cls");

setlocale(0, "ukr");

main\_menu\_interface();

main\_menu\_switch();

return 0;

}