МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра ЕОМ

ЗВІТ

з лабораторної роботи №1

з дисципліни «C# та платформа .NET»

на тему: «C# Basics»

Варіант №5

Виконав: Прийняв:

ст. гр. КІУКІу-21-2 Лукашов С.А

Кірієнко Д.В

Харків 2022

* 1. Мета роботи

Ознайомлення з основними особливостями мови C# та платформи .NET.

1.2 Хід роботи

1.2.1 Завдання

Забезпечити у проекті розподілення частини інтерфейсу ті частини логіки типів даних, які будуть створені під час виконання завдання.

Дані типу:

* назва тарифу, визначається за допомогою enum з 3-4 значень;
* вартість хвилини розмови;
* поточна сума коштів на рахунку;
* номер телефону, має 10 цифр;
* масив підключених послуг, послуги визначаються за допомогою enum.

Функціональність типу:

* ініціалізуючий конструктор та конструктор без аргументів;
* метод переходу на новий тарифний план, тариф є аргументом;
* метод здійснення дзвінка, кількість хвилин та номер, куди відбувається дзвінок є вхідними аргументами, вартість дзвінку повертається з методу;
* методи підключення/відключення послухи, послуга має мати ціну, яка знімається з рахунку;
* методи поповнення рахунку на вказану суму, отримання поточного стану рахунку;
* метод отримання історії дзвінків.

1.2.2 Виконання завдання

Лістинг 1.1 – Код класу Phone

class Phone

{

private List<string> calls = new List<string>(); // список, в який записуються дзвінки

// Перелік тарифів

public enum Tariff : int

{

Bronze = 1,

Silver = 2,

Gold = 3,

Platinum = 4,

}

// Перелік послуг

public enum Services : int

{

FreeCall = 1,

FreeSMS = 2,

ExtraMB = 3,

ExtraMinutes = 4

}

Services[] services = new Services[4];

Tariff tariffPlan; // тарифний план

double pricePerMinute; // вартість хвилини розмови

double money; // поточна сума коштів на рахунку

string phoneNumber; // номер телефону

// Конструктор за замовчуванням

public Phone()

{

tariffPlan = Tariff.Bronze;

pricePerMinute = 0.50;

money = 40;

phoneNumber = "0000000001";

}

// Конструктор з параметрами

public Phone(Tariff tariff, double money, string phoneNumber)

{

tariffPlan = tariff;

pricePerMinute = 0.50;

this.money = money;

this.phoneNumber = phoneNumber;

}

// Метод переходу на новий тарифний план

public void newTariff(Tariff tariff)

{

tariffPlan = tariff;

}

// Функція здійснення дзвінка

public double call(int minutes, string phoneNumber)

{

double pricePerCall = minutes \* pricePerMinute;

if (money > 0)

{

string call = phoneNumber + " " + minutes + " хвилин";

calls.Add(call);

money -= pricePerCall;

}

else

{

money = 0;

Console.WriteLine("Недостатньо коштiв для дзвiнка!");

}

return pricePerCall;

}

// Метод підключення послуги

public void activateService(Services service)

{

if (money >= 10)

{

switch (service)

{

case Services.FreeCall:

money -= 10;

services[0] = Services.FreeCall;

Console.WriteLine("Послугу \"Безкоштовні дзвінки\" підключено!");

break;

case Services.FreeSMS:

money -= 10;

services[1] = Services.FreeSMS;

Console.WriteLine("Послугу \"Безкоштовні СМС\" підключено!");

break;

case Services.ExtraMB:

if (money >= 20)

{

money -= 20;

services[2] = Services.ExtraMB;

Console.WriteLine("Послугу \"Додаткові мегабайти\" підключено!");

}

break;

case Services.ExtraMinutes:

if (money >= 20)

{

money -= 20;

services[3] = Services.ExtraMinutes;

Console.WriteLine("Послугу \"Додаткові хвилини\" підключено!");

}

break;

}

}

}

// Метод відключення послуги

public void deactivateService(Services service)

{

switch (service)

{

case Services.FreeCall:

services[0] = 0;

Console.WriteLine("Послугу \"Безкоштовні дзвінки\" відключено!");

break;

case Services.FreeSMS:

services[1] = 0;

Console.WriteLine("Послугу \"Безкоштовні СМС\" відключено!");

break;

case Services.ExtraMB:

services[2] = 0;

Console.WriteLine("Послугу \"Додаткові мегабайти\" відключено!");

break;

case Services.ExtraMinutes:

services[3] = 0;

Console.WriteLine("Послугу \"Додаткові хвилини\" відключено!");

break;

}

}

//Метод поповнення поточного рахунку

public void putMoney(int amount)

{

if (amount > 0) money += amount;

Console.WriteLine("На рахунок зараховано " + amount);

}

// Метод отримання поточного стану рахунку

public void showMoney()

{

Console.WriteLine("На рахунку " + money);

}

// Метод перегляду історії дзвінків

public void showCalls()

{

Console.WriteLine("Iсторiя викликiв:");

foreach (var calls in calls)

{

Console.WriteLine(calls);

}

}

}

Лістинг 1.2 – Код основної функції Main

static void Main()

{

Phone phone = new Phone(); // створення екземпляру класу Phone

phone.showMoney(); // перевірка стану рахунку

phone.newTariff(Phone.Tariff.Gold); // зміна тарифу

phone.call(14, "4523155325"); // здійснення дзвінка

phone.activateService(Phone.Services.FreeCall); // підключення послуги

phone.call(7, "2325890394"); // здійснення дзвінка

phone.deactivateService(Phone.Services.FreeCall); // відключення послуги

phone.putMoney(200); // поповнення рахунку

phone.showMoney();

phone.showCalls(); // перегляд історії дзвінків

}

В лістингу 1.2 можна побачити, що створюється екземпляр класу Phone з стандартними параметрами. Роботу програми представлено на рисунку 1.1. З рисунку 1.1 можна побачити, що спочатку перевіряється баланс, далі підключається та відключається послуга «Безкоштовні дзвінки». Потім рахунок поповнюється на 200 та перевіряється баланс. В кінці викликається історія викликів.

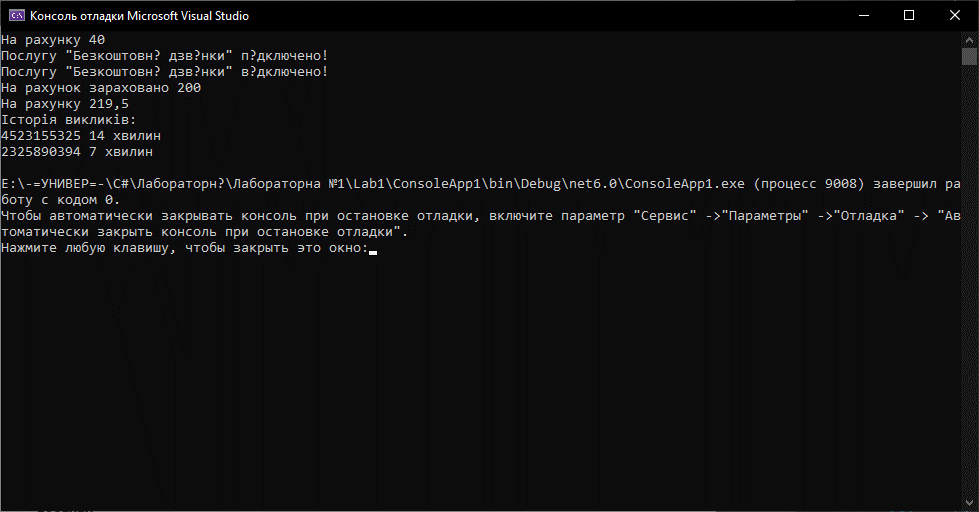


Рисунок 1.1 – Результат роботи програми

1.3 Висновки

В даній лабораторній роботі було розроблено клас Phone, який має два конструктори: конструктор за замовчуванням, та конструктор з параметрами. Клас має методи: переходу на новий тарифний план, здійснення дзвінка, підключення послуг, відключення послуг, поповнення поточного рахунку, отримання поточного стану рахунку, перегляду історії дзвінків. Клас протестовано у головній функції Main.