|  |  |
| --- | --- |
| Міністерство освіти і науки України  НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  “ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ” | |
| Кафедра управління проєктами в інформаційних технологіях | |
| Звіт з лабораторної № 1  «Основи розробки додатків у .NET»  з дисципліни «Стек технологій***»*** | |
|  | *Виконали*:  студенти  групи КН-322а  Данилова Ангеліна Вадимівна  Непоменко Дмитро Григорович  Лобач Даніїл Артемович |
| Харків, НТУ “ХПІ”  2024 | |

Зміст

[Мета: 2](#_Toc178712414)

[Теоретичні відомості 2](#_Toc178712415)

[Завдання 2](#_Toc178712416)

[Хід роботи 4](#_Toc178712417)

[Висновки 12](#_Toc178712418)

# Мета:

Навчитися створювати консольні та віконні додатки, ознайомитись з використання делегатів та подій у мові програмування C#.

# Теоретичні відомості

У платформі .NET можна виділити наступні основні компоненти:

Common Language Specification (CLS) — набір правил, які мають підтримуватись мовами програмування для сумісності.

Microsoft Intermediate Language (MSIL) — проміжний код, в який компілюється код на C# перед виконанням на платформі.

Common Language Runtime (CLR) — віртуальна машина, яка виконує MSIL, забезпечуючи управління пам'яттю, винятками та безпекою.

.NET Class Library (.NET FCL) — бібліотека класів, яка надає інтерфейси для стандартних операцій, таких як робота з файлами, мережею, базами даних тощо.

# Завдання

1. Опрацювати теорію (див. додаток А)

2. Написати програму для моделювання роботи банкомату.

Рішення має складатися з трьох проектів:

* бібліотека класів;
* віконний проект;
* консольний проект.

У бібліотеці класів реалізуйте загальні типи даних, які будуть використані у віконному та консольному проектах. У віконному додатку розробіть інтерфейс, який наближений до інтерфейсу традиційного банкомату. Консольний додаток повинен забезпечити ті ж самі можливості, що має віконний додаток, але за допомогою консольного меню.

Слід передбачити наступні можливості:

* автентифікація (перевірка введення номеру картки та пін-коду);
* перегляд балансу на картці;
* перегляд історії операцій з фільтруванням по проміжку часу (за поточний день, за поточний тиждень, за поточний місяць)
* зняття коштів;
* зарахування коштів на картку;
* перерахування коштів на картку із заданим номером;
* перегляд найближчих банкоматів;

Повинна бути передбачена можливість додавання власних обробників подій на виконання операцій автентифікації, перегляду балансу, зняття коштів та перерахування коштів на іншу картку.

Додайте власні обробники подій, які будуть виводити повідомлення:

− для віконного додатку - за допомогою виклику:

MessageBox.Show("Повідомлення");

− для консольного додатку - за допомогою виклику

Console.WriteLine("Повідомлення");

При виконанні завдання створіть класи, які забезпечать роботу додатку:

* **Account** - клас, який є поданням карткового рахунку клієнта (властивості: номер картки, прізвище та ім’я власника, баланс на рахунку, тощо);
* **AutomatedTellerMachine** - клас, який моделює роботу банкомата (містить кількість грошей, які містяться у банкоматі, ідентифікатор банкомату, його адресу розташування, тощо);
* **Bank** - клас, що представляє банк (назва банку, перелік банкоматів, тощо).

Це не виключний перелік класів, за необхідності створюйте інші класи, які Вам будуть необхідні при реалізації завдання.

# Хід роботи

Програма для моделювання роботи банкомату.

Рішення складається з трьох проектів:

* бібліотека класів;
* віконний проект;
* консольний проект.

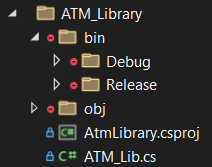


Рисунок 1 – Solution для проекту бібліотека класів.

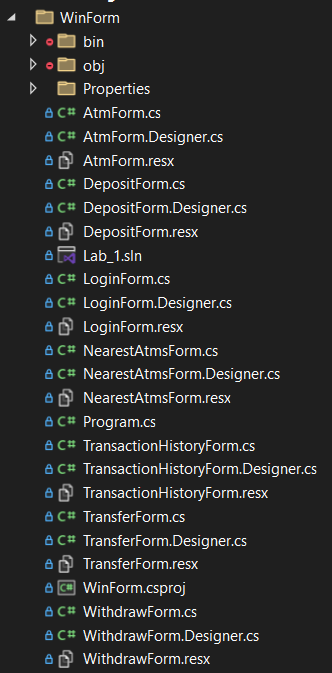


Рисунок 2 – Solution для проекту віконний проект.

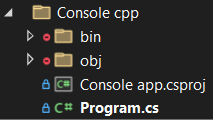


Рисунок 3 – Solution для проекту консольний проект.

У бібліотеці класів реалізовано загальні типи даних, які були використані у віконному та консольному проектах, а саме: Account, Transaction, TransactionFilters, AutomatedTellerMachine, Bank,

У віконному додатку розроблено інтерфейс, який наближений до інтерфейсу традиційного банкомату. Також розроблено консольний додаток, який забезпечує ті ж самі можливості, що має віконний додаток, але за допомогою консольного меню.

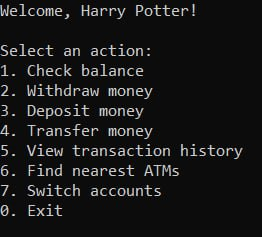


Рисунок 4 – Інтерфейс консольного додатка

Передбачено наступні можливості:

* автентифікація (перевірка введення номеру картки та пін-коду);

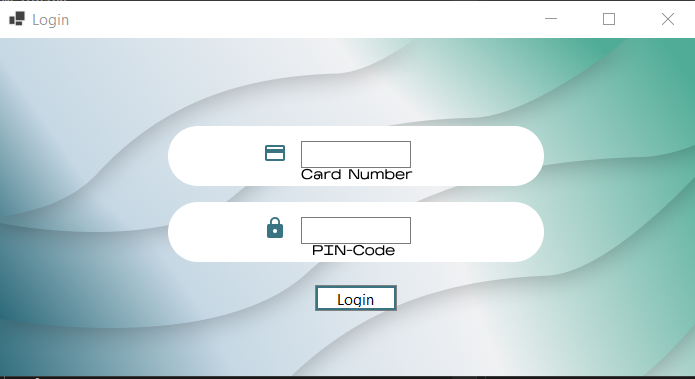


Рисунок 5 – Інтерфейс віконного додатка для функції автентифікації.

* перегляд балансу на картці;

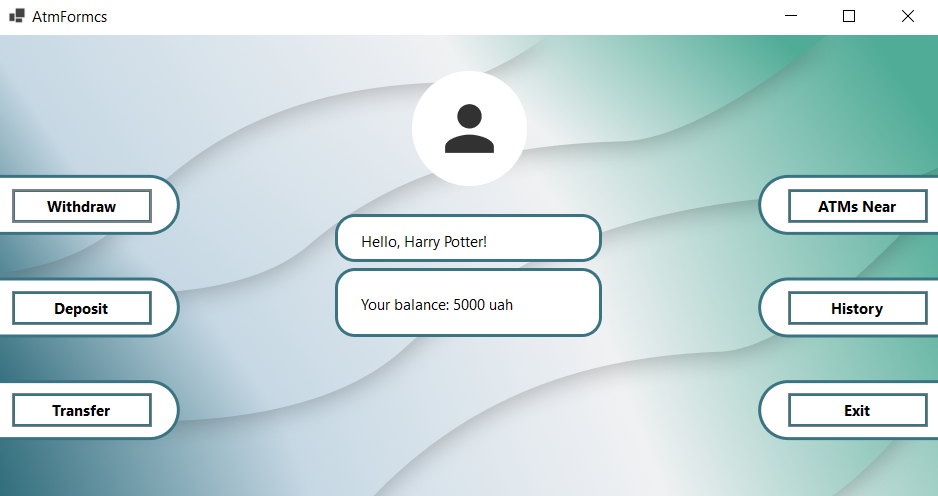


Рисунок 6 – Інтерфейс віконного додатка для функції перегляду балансу на картці

* перегляд історії операцій з фільтруванням по проміжку часу (за поточний день, за поточний тиждень, за поточний місяць)

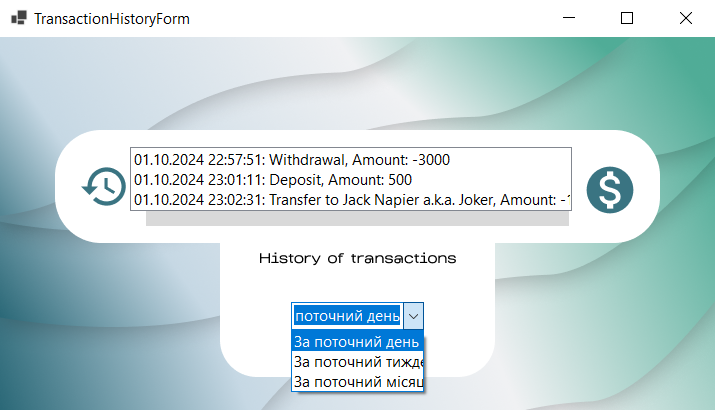


Рисунок 7 – Інтерфейс віконного додатка для функції перегляду історії операцій

* зняття коштів;



Рисунок 8 – Інтерфейс віконного додатка для функції зняття коштів

* зарахування коштів на картку;



Рисунок 9 – Інтерфейс віконного додатка для функції зарахування коштів на картку

* перерахування коштів на картку із заданим номером;

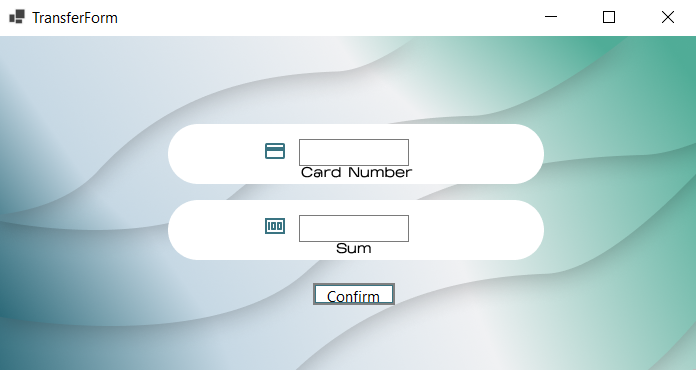


Рисунок 10 – Інтерфейс віконного додатка для функції перерахування коштів на картку

* перегляд найближчих банкоматів;

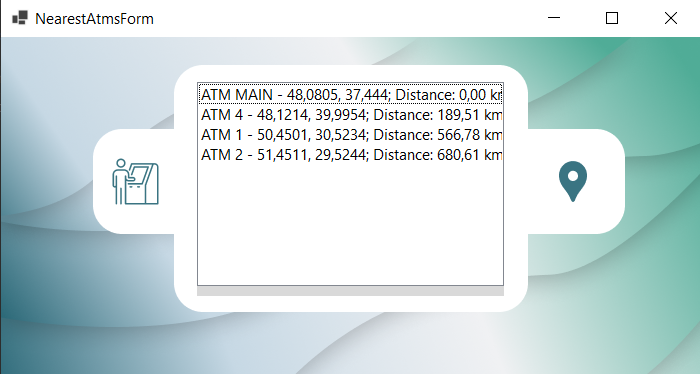
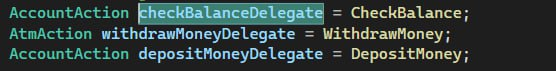
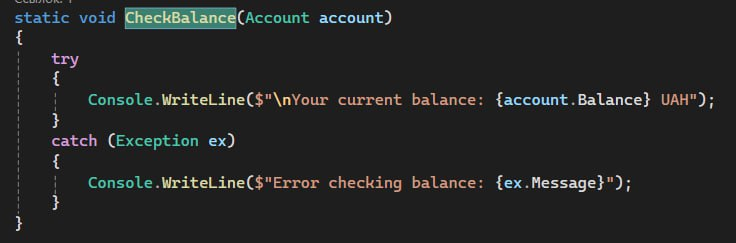


Рисунок 11 – Інтерфейс віконного додатка для функції перегляду найближчих банкоматів

Передбачена можливість додавання власних обробників подій на виконання операцій автентифікації, перегляду балансу, зняття коштів та перерахування коштів на іншу картку.

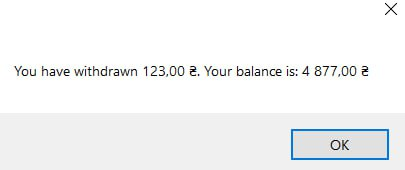




Додайте власні обробники подій, які будуть виводити повідомлення:

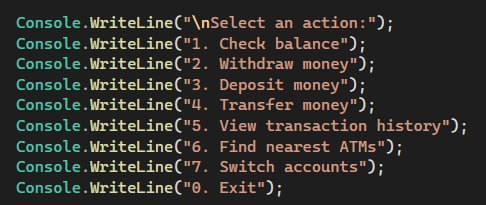
− для віконного додатку - за допомогою виклику:

MessageBox.Show("Повідомлення");



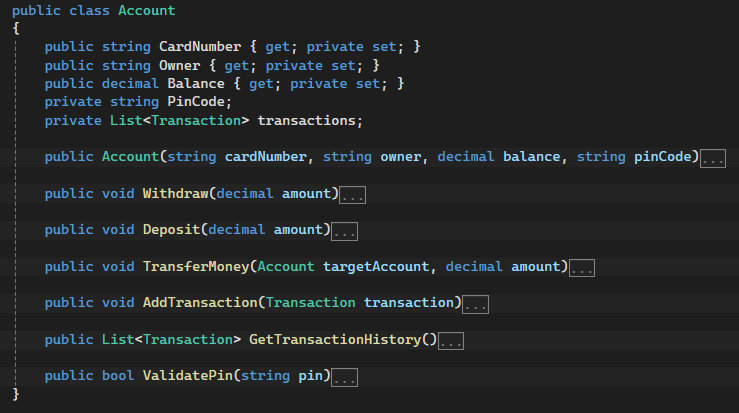
− для консольного додатку - за допомогою виклику

Console.WriteLine("Повідомлення");

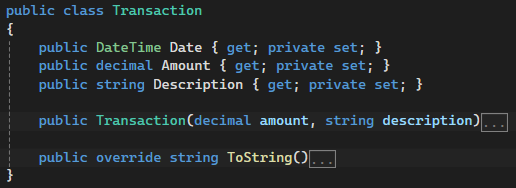


При виконанні завдання було створено наступні класи, які забезпечують роботу додатку:

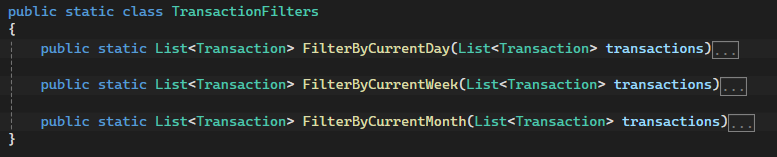
* **Account** - клас, який є поданням карткового рахунку клієнта (властивості: номер картки, прізвище та ім’я власника, баланс на рахунку, тощо);



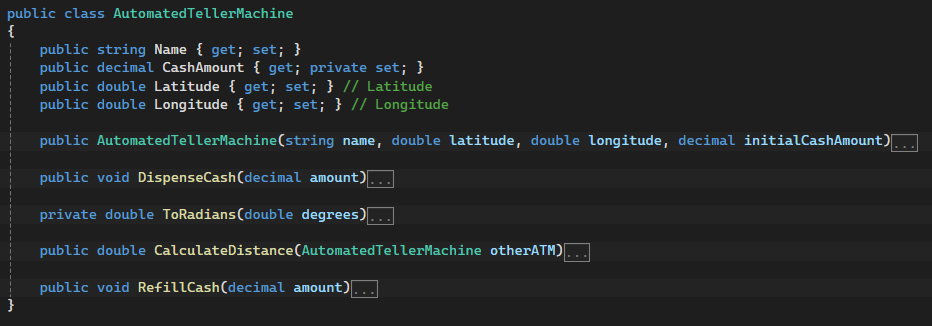
* **Transaction** – клас, який відповідає за усі операції з грошима у банку;

****

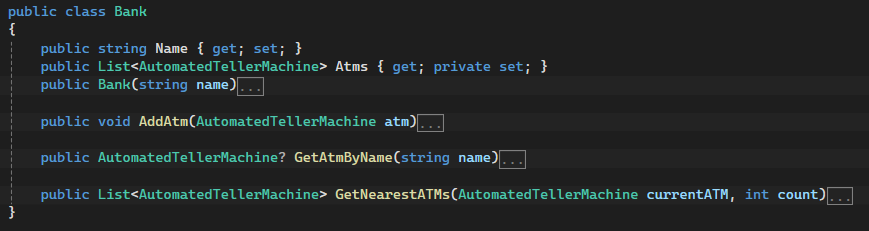
* **TransactionFilters** - клас, відповідає за фільтрування транзакцій за різними періодами часу;



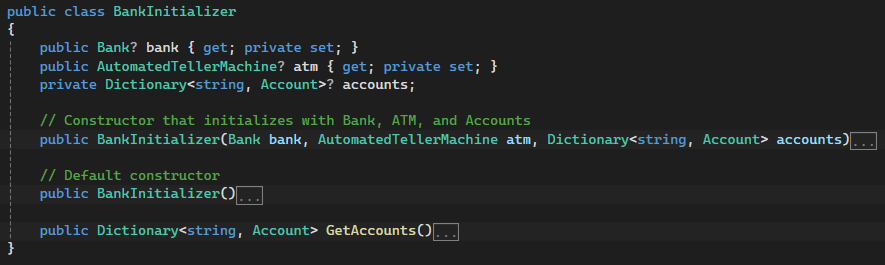
* **AutomatedTellerMachine** - клас, який моделює роботу банкомата (містить кількість грошей, які містяться у банкоматі, ідентифікатор банкомату, його адресу розташування, тощо);



* **Bank** - клас, що представляє банк (назва банку, перелік банкоматів, тощо);



* **BankInitializer** - клас, що ініціалізує усі інші класи .



# Висновки

Під час виконання лабораторної роботи було створено програму для моделювання роботи банкомату. Програма реалізована у вигляді трьох проектів: бібліотеки класів, віконного та консольного додатків. Основні функції, які були реалізовані: автентифікація користувача, перегляд балансу, історія операцій, зняття та зарахування коштів. Було розроблено обробники подій, які виводять повідомлення через MessageBox у віконному додатку та Console.WriteLine у консольному. Ця робота дозволила ознайомитись з основами створення додатків на платформі .NET, застосуванням делегатів та подій у мові програмування C#.