

Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ №4-5

ДИСЦИПЛІНА: «КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ»

Виконав: студентка групи КС22
Мазуренко Анжеліки
Перевірив: Спорів Олександр
Євгенович

Харків
2024

Лабораторна робота No4-5 Java Net Programming (TCP Sockets)

Git Repository: <https://github.com/miorezu/LB4-5.JavaNet>

Завдання №1

Напишіть програму, яка моделює роботу «карткової системи» метро.

Розглянемо таку

ситуацію. Студентам *ВНЗ* видають іменні штриховані пластикові проїзні картки (така картка несе лише ідентифікаційний код – ідентифікатор студента). Для того, щоб програма вийшла не надто громіздкою, наша система буде підтримувати обмежений набір операцій:

- ☐ операцію видачі клієнту нової картки та її реєстрації в системі;
- ☐ операцію отримання інформації про клієнта;
- ☐ операцію поповнення рахунку;
- ☐ операцію оплати поїздки (зняття деякої кількості коштів з рахунку);
- ☐ операцію отримання залишку коштів на картці.

У ході виконання завдання необхідно:

- ☐ розробити сервер, який зберігає інформацію про видані пластикові картки і

організовує виконання основних операцій;

- ☐ створити клієнта, який за допомогою пластикової картки з ідентифікатором надсилає

запити на виконання потрібних йому операцій і отримує повідомлення про виконання;

- ☐ з метою підвищення надійності системи реалізувати взаємодію комп'ютерів за

допомогою протоколу *TCP*.

Операції, які ми можемо здійснювати над карткою знаходяться у пакеті operations. У пакеті data знаходяться класи, які зберігають у собі інформацію про картки та користувачів. У пакеті main клас сервера та клієнта, які нам потрібно запустити та ще один клас.

Для демонстрації роботи програми запустимо клас MetroServer та після Client.

Лістинг 1. Фрагмент коду

```
Client client = new Client("localhost", 7891);
AddMetroCardOperation operation = new AddMetroCardOperation();
operation.getCard().setUser(new User("Anzhelika", "Mazurenko",
"F", "31.03.2005"));
operation.getCard().setSerialNumber("00001");
operation.getCard().setEstablishment("KhNU");
operation.getCard().setBalance(9);
client.applyOperation(operation);
client.finish();
client = new Client("localhost", 7891);
client.applyOperation(new AddMoneyOperation("00001", 100));
client.applyOperation(new ShowBalanceOperation("00001"));
client.applyOperation(new PayMoneyOperation("00001", 50));
client.applyOperation(new ShowBalanceOperation("00001"));
client.applyOperation(new PayMoneyOperation("00001", 1000));
client.applyOperation(new ShowCardInfoOperation("00001"));
client.applyOperation(new XMLOperation());

client.finish();
```

```
Card added
Finish Work Socket[addr=/127.0.0.1,port=51274,localport=7891]
Success. Money added
Success. Your Balance: 109.0
Success. Money payed
Success. Your Balance: 59.0
Error. Failed to pay
MetroCard{serialNumber='00001', user=User{name='Anzhelika', surname='Mazurenko', sex='F', birthday=Thu Mar 31 00:00:00 EEST 2005}, establishment='KhNU', balance=59.0}
Success
Finish Work Socket[addr=/127.0.0.1,port=51276,localport=7891]
```

Рисунок 1 – скріншот виконання фрагменту лістинга 1

Додаємо карту клієнта, заповнюючи всі данні про користувача та баланс картки. Завершуємо роботу.

Розпочинаємо роботу та тестуємо операції над карткою. Додаємо гроші, знімаємо та показуємо баланс. Бачим, що все працює, але коли ми

хочемо зняти суму більше за наш баланс видає помилку та гроші не знімаються. Виводимо інформацію картки та завершуємо роботу.

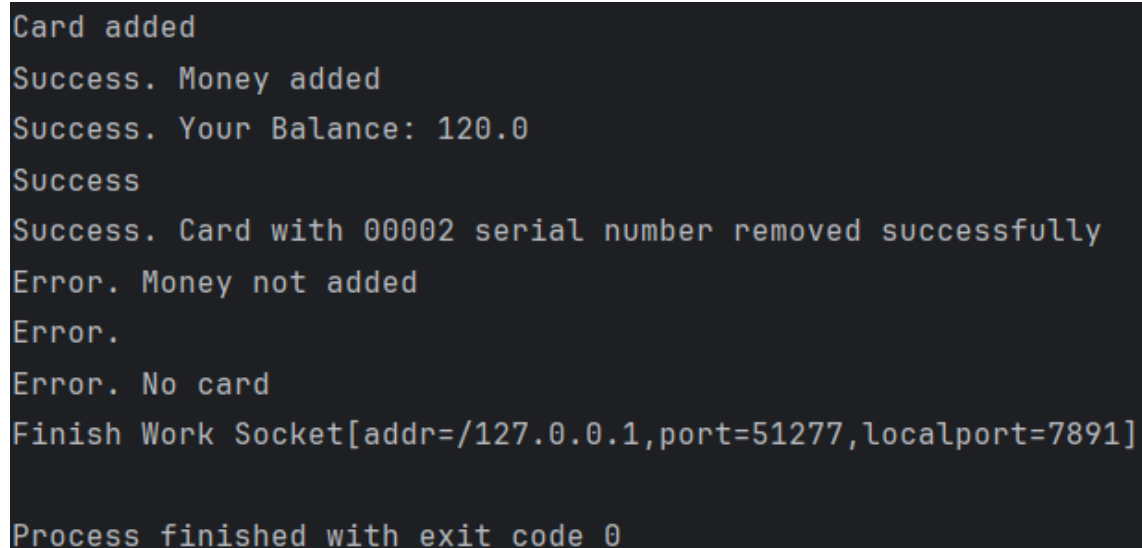
Лістинг 2. Фрагмент коду

```
client = new Client("localhost", 7891);
operation = new AddMetroCardOperation();
operation.getCard().setUser(new User("Erwin", "Smit", "M",
"25.09.1999"));
operation.getCard().setSerialNumber("00002");
operation.getCard().setEstablishment("");
operation.getCard().setBalance(20);
client.applyOperation(operation);

client.applyOperation(new AddMoneyOperation("00002", 100));
client.applyOperation(new ShowBalanceOperation("00002"));
client.applyOperation(new XMLOperation());

client.applyOperation(new RemoveCardOperation("00002"));
client.applyOperation(new AddMoneyOperation("00002", 100));
client.applyOperation(new ShowBalanceOperation("00002"));
client.applyOperation(new ShowCardInfoOperation("00002"));

client.finish();
```



```
Card added
Success. Money added
Success. Your Balance: 120.0
Success
Success. Card with 00002 serial number removed successfully
Error. Money not added
Error.
Error. No card
Finish Work Socket[addr=/127.0.0.1,port=51277,localport=7891]
Process finished with exit code 0
```

Рисунок 2 - скріншот виконання фрагменту лістинга 2

Знову розпочинаємо роботу й додаємо нового користувача і картку для нього. Додаємо кошти й показуємо баланс. Також записуємо дані про

наші картки в xml документ (авжеж, краще зробити, щоб дані про картки записував сервер, а не клієнт).

```
<?xml version="1.0" encoding="Windows-1251" standalone="no"?>
<CardsList>
  <InfoCard Balance="59.0" Establishment="KhNU" SerialNumber="00001">
    <User Name="Anzhelika" Sex="F" Surname="Mazurenko" birthdayDate="Thu Mar 31 00:00:00 EEST 2005" />
  </InfoCard>
  <InfoCard Balance="120.0" Establishment="" SerialNumber="00002">
    <User Name="Erwin" Sex="M" Surname="Smit" birthdayDate="Sat Sep 25 00:00:00 EEST 1999" />
  </InfoCard>
</CardsList>
```

Рисунок 3 - xml документ

Після цього видаляємо картку й бачимо, що при спробі здійснити над нею операції видає помилку. Закінчуємо роботу.

В операціях скоротити велику кількість if else допоміг command pattern, використовуючи абстрактний метод execute().

Всі класи та їх взаємодії можна передивитись у [репозиторії](#).