

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

**Лабораторная работа №3 по курсу**  
**«Проектирование программ в интеллектуальных системах»**

**Выполнили студенты группы**  
**921701:**

**Соловьев А.**

**Проверил:**

**Садовский М.Е**

**МИНСК**

**2020**

**Цель работы:** Изучить основы наследование на языке программирования C++.

### **Задание:**

При выборе этого варианта студенту необходимо изучить работу банкомата «Беларусь-

банк». После этого необходимо выявить взаимодействующие объекты в рамках сессии работы банкомата. На основе выявленных объектов сформировать объектную модель, которая описывает работу банкомата. В данном варианте необходимо обратить особое внимание на классы для представления экранов банкомата. Именно они будут формировать основную иерархию классов данного варианта.

### **Ход и результаты работы:**

В лабораторной работе реализован банкомат.

Лабораторная включает в себя такие классы как Card, Bank и Session, наследуемый от двух первых, а так же отдельную иерархию экранов банкомата.

```
class Card {
protected:
    string data[4];
public:
    Card(int number);
};

class Bank {
protected:
    vector<Data_of_card> database;
public:
    Bank();
};

class Session :public Card, public Bank {
    int position_of_card = 0;
public:
    Session(int number);
    bool initialization();
    bool get_pincode();
    bool withdraw_money(int sum);
    int& put_money();
    void check_balance();
    Data_of_card& get_card();
    void end_of_session();
};
```

```

class Screen {
protected:
    int language = 0;
    string message[3][18];
public:
    Screen() { system("cls"); }
    virtual int turn_on() = 0;
};

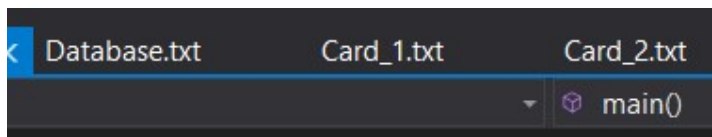
class Options :Screen {
public:
    Options(Session& session){ processing(session); }
    int turn_on();
    void processing(Session& session);
};

class Check :Screen {
public:
    Check(Session& session) { print_a_check(session); }
    int turn_on();
    void print_a_check(Session& session);
};

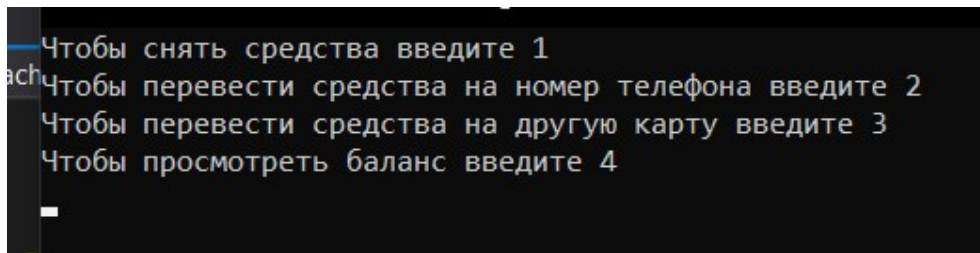
```

В ходе программы создаётся объект класса Session, с которым в дальнейшем мы будем работать.

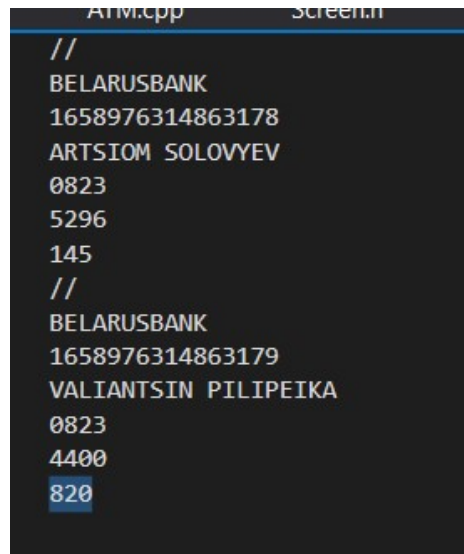
При создании объекта программа берёт данные из файлов (данные карты и базу данных банка).



В дальнейшем нам предлагается выбор действий с картой.



После окончания работы алгоритма, программа перезаписывает изменённые данные в базу данных банка в таком формате:



```
Atm.cpp screen.1
//
BELARUSBANK
1658976314863178
ARTSIOM SOLOVYEV
0823
5296
145
//
BELARUSBANK
1658976314863179
VALIANTSIN PILIPEIKA
0823
4400
820
```

### Вывод

В ходе лабораторной работы я научился проводить объектно-ориентированный анализ предметной области.

Объектно-ориентированное программирование помогает программистам бороться со сложностью программ. Данный тип представления схож с типом представления окружающего мира человеком. Так код становится более понятным на смысловом уровне.

Так же при таком типе работы удобно разделять поставленные задачи. Например, при работе команды программистов они разделяют задачи между собой, что позволяет сделать работу быстрее и эффективнее.

Так же к плюсам ООП можно возмoжность дoстраивать код просто добавив новые классы, приспособив их к уже существующим.