МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра МСС

Гилевич Павел Геннадьевич

«Банковская система»

Отчет по лабораторной работе
студента 3 курса 12 группы
по дисциплине «Технологии программирования»

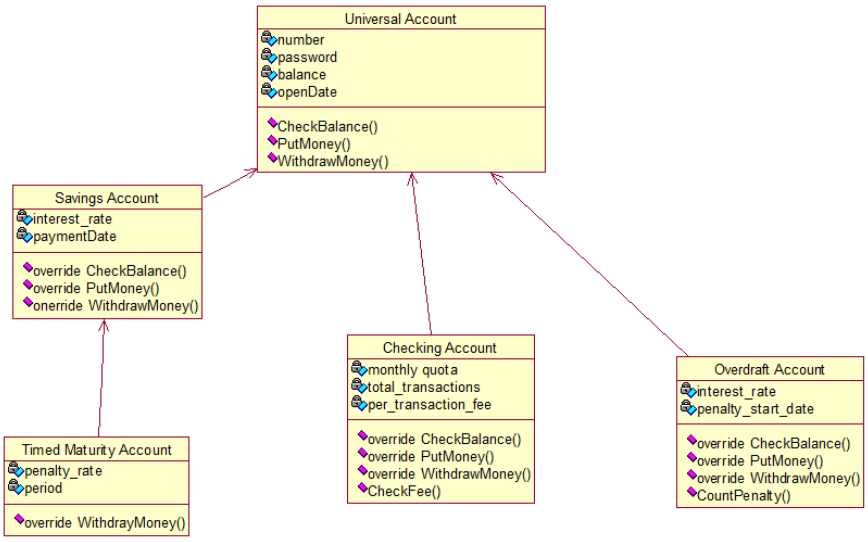
Преподаватель
Довнар С.Е.

Минск 2013

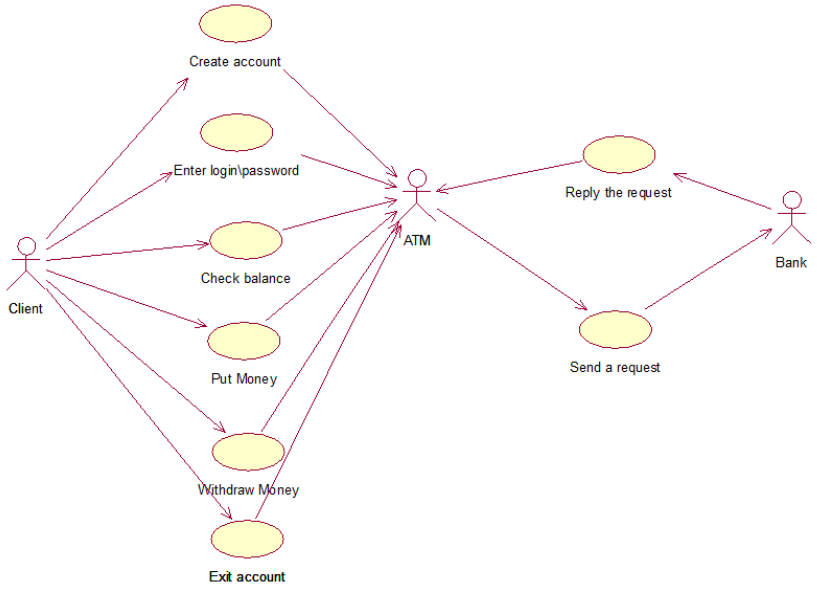
I. Структура модели

Для наглядной иллюстрации структуры разработанной модели приведем 3 различных UML-диаграммы.

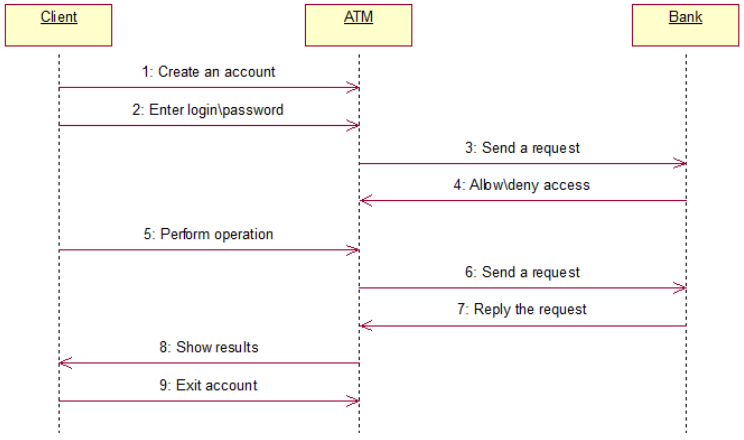
1. Диаграмма классов (Class Diagram)



2. Диаграмма вариантов использования (Use Case Diagram)



3. Диаграмма последовательности (Sequence Diagram)



II. Программная реализация

Для реализации данной модели был выбран язык программирования C#. Приведем некоторые фрагменты кода.

- Реализация абстрактного класса *UniversalAccount*, являющегося вершиной иерархии классов

```
abstract class UniversalAccount
    {
        protected int number;                   //account's number
        protected String password;              //account's password
        protected double balance;               //current balance
        protected DateTime openDate;            //account's opening date

        //abstract methods
        public abstract void PutMoney(double sum, DateTime currDate);
        public abstract double WithdrawMoney(double sum, DateTime currDate);
        public abstract double CheckBalance(DateTime currDate);

        //method CountMonths calculates the number of months between 2 dates
        public int CountMonths(DateTime currentDate, DateTime paymentDate)
        {
            int months = (currentDate.Year - paymentDate.Year) * 12 + (currentDate.Month -
paymentDate.Month);
            if (currentDate.Day < paymentDate.Day)
                months--;
            return months;
        }
    }
```

- Приведем также пример реализации одного из дочерних классов, для примера - класс *SavingsAccount*.

```
class SavingsAccount:UniversalAccount
    {
        protected double interest_rate;         //annual interest rate
        protected DateTime paymentDate;         //date of last percents' payment

        public SavingsAccount(int number, string password, double balance, double rate,
DateTime openDate)
        {
            this.number = number;
            this.password = password;
            this.balance = balance;
            this.interest_rate = rate;
            this.openDate = openDate;
            this.paymentDate = openDate;
        }

        public override double CheckBalance(DateTime currentDate)
        {
            double currentBalance = this.balance;
            int months = CountMonths(currentDate, paymentDate);

            for (int i = 0; i < months; i++)
                currentBalance *= (1 + interest_rate / 100 / 12);

            return currentBalance;
        }

        public override void PutMoney(double sum, DateTime currentDate)
        {
            balance = CheckBalance(currentDate);
            paymentDate = new DateTime(currentDate.Year, currentDate.Month, openDate.Day);
            balance += sum;
        }
```

```
        public override double WithdrawMoney(double sum, DateTime currentDate)
        {
            this.balance = CheckBalance(currentDate);
            paymentDate = new DateTime(currentDate.Year, currentDate.Month, openDate.Day);

            if (balance >= sum)
            {
                balance -= sum;
                return sum;
            }
            else
            {
                Console.WriteLine("Not enough money");
                return 0;
            }
        }
    }
```

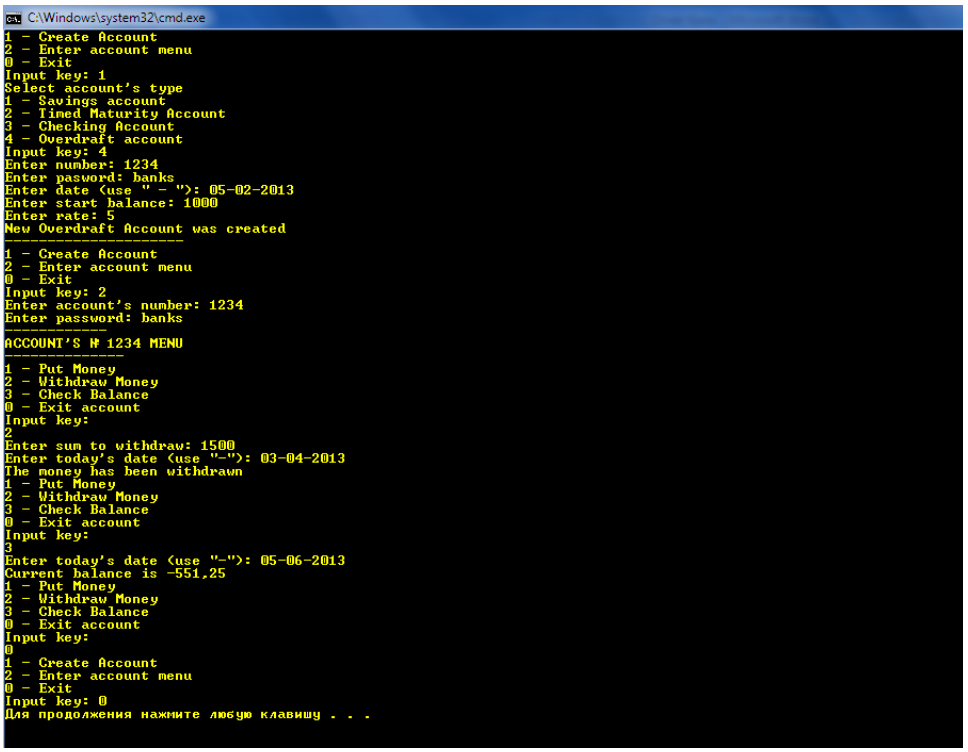
- А также приведем некоторые моменты работы функции *main* и класса *Program*.

```
class Program
    {
        static List<UniversalAccount> Accounts;                 //list of created accounts
        static UniversalAccount SearchByLogin(int number)       //searches account by its
login
        {
            foreach(UniversalAccount acc in Accounts)
                if (acc.Number == number)
                    return acc;

            return null;
        }

        static void Main(string[] args)
        {
            Accounts = new List<UniversalAccount>();
            for(;;)
            {
                //start menu
                Console.Write("1 - Create Account\n2 - Enter account menu\n0 - Exit\nInput
key: ");
                int key = Convert.ToInt32(Console.ReadLine());
                switch (key)
                {
                    case 1:
                        Console.WriteLine("Select account's type\n1 - Savings account\n2 -
Timed Maturity Account\n3 - Checking Account");
                        Console.Write("4 - Overdraft account\nInput key: ");
                        int acc_key;
                        acc_key = Convert.ToInt32(Console.ReadLine());
                        int acc_number;
                        for (; ;)
                        {
                            Console.Write("Enter number: "); acc_number =
Convert.ToInt32(Console.ReadLine());
                            if (SearchByLogin(acc_number) != null)
                            {
                                Console.WriteLine("Account with this number already exists.
Try again");
                            }
                            else
                                break;
                        }
Console.Write("Enter pasword: ");
String acc_password = Console.ReadLine();
Console.Write("Enter date (use \" - \"): ");
String date = Console.ReadLine();
String[] split_date = date.Split('-');
DateTime acc_date = new
DateTime(Convert.ToInt32(split_date[2]),Convert.ToInt32(split_date[1]),
Convert.ToInt32(split_date[0]));
```

```
Console.Write("Enter start balance: ");
double acc_balance = Convert.ToDouble(Console.ReadLine());

                        switch (acc_key)
                        {
                            case 1:
                                //creating of Savings Account
                                break;
                            case 2:
                                //creating of Timed Maturity Account
                                break;
                            case 3:
                                //creating of Checking Account
                                break;
                            case 4:
                                //creating of Overdraft Account
                                break;
                        }
                        break;

                    case 2:
                        UniversalAccount search_result;
                        //searching existing account by login, then checking its password

Console.WriteLine("------------\nACCOUNT'S № " + search_result.Number+ " MENU\n--------------
");
                        for (; ; )
                        {
                            Console.WriteLine("1 - Put Money\n2 - Withdraw Money\n3 - Check
Balance\n0 - Exit account");
                            int key_menu = Convert.ToInt32(Console.ReadLine());
                            switch (key_menu)
                            {
                                case 1:
                                    //putting money
                                    break;
                                case 2:
                                    //withdrawing money
                                  break;

                                case 3:
                                    //checking balance
                                  break;

                                default:
                                    break;
                            }
                                if (key_menu == 0)
                                break;
                        }
                        break;

                    case 0:
                        return;
                }
            }
        }
    }
```

III. Пример работы программы

Приведем скриншот, демонстрирующий работу разработанного приложения.


C:\Windows\system32\cmd.exe
1 - Create Account
2 - Enter account menu
0 - Exit
Input key: 1
Select account's type
1 - Savings account
2 - Timed Maturity Account
3 - Checking Account
4 - Overdraft account
Input key: 4
Enter number: 1234
Enter pasword: banks
Enter date (use " - "): 05-02-2013
Enter start balance: 1000
Enter rate: 5
New Overdraft Account was created
1 - Create Account
2 - Enter account menu
0 - Exit
Input key: 2
Enter account's number: 1234
Enter password: banks
ACCOUNT'S № 1234 MENU
1 - Put Money
2 - Withdraw Money
3 - Check Balance
0 - Exit account
Input key:
2
Enter sum to withdraw: 1500
Enter today's date (use "-"): 03-04-2013
The money has been withdrawn
1 - Put Money
2 - Withdraw Money
3 - Check Balance
0 - Exit account
Input key:
3
Enter today's date (use "-"): 05-06-2013
Current balance is -551,25
1 - Put Money
2 - Withdraw Money
3 - Check Balance
0 - Exit account
Input key:
0
1 - Create Account
2 - Enter account menu
0 - Exit
Input key: 0
Для продолжения нажмите любую клавишу . . .