**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра МСС**

**Понтелей Виталий Витальевич**

Отчет по лабораторной работе №4

(«Технологии программирования»)

студента 3 курса 12 группы

**Преподаватель**

*Довнар С.Е.*

Минск 2013

**Диаграмма использования**



Клиент работающий с банкоматом имеет возможность:

- узнать баланс счета

- вкладывать фонды

- снимать фонды

- снимать определенную сумму денег

- добавлять определенную сумму денег

**Диаграмма классов**



На данной диаграмме представлены классы счетов, все классы наследуются от класса Account, который представляет собой универсальный счет. Возможные операции со счетом:

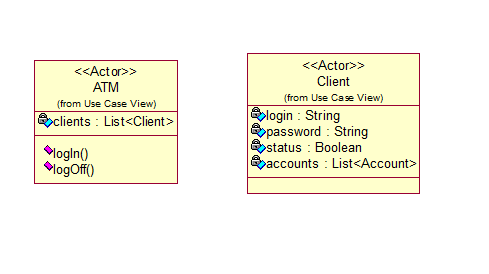
- getBalance() –получение денежного остатка на счету

- investFunds() – вкладывать фонды

- takeFunds() – снимать фонды

- takeMoney() – снять денежную сумму со счета

- addMoney() – добавить денежную сумму на счет



Класс Client, содержит поля:

- login – логин пользователя

- password – пароль пользователя

- status – пользуется ли клиент в данный момент банкоматом

- accounts – счета клиента

Класс ATM, содержит одно единственное поле:

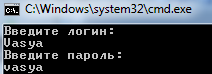
- clients – содержащее, зарегистрированных пользователей.

Пользователь при использовании банкомата может использовать методы logIn – вход в личный кабинет и logoff – выход из личного кабинета.

**Диаграмма последовательностей**



Чтобы работать с банкоматом клиент должен ввести свой логин и пароль:



При этом вызывается метод logIn, который проверяет наличие пользователя в базе данных и устанавливает status = true, если проверка прошла успешно.

Далее клиент может выбрать свой счет, что происходит с помощью следующего кода:

Console.WriteLine("1 - выбрать счет\n0 - выход");

ans = Console.ReadLine();

if (ans == "1")

{

Client cl = atm.getClient(num);

for (int i = 0; i != cl.accountCount(); i++)

{

if (cl.getAccount(i).GetType() == typeof(Account)) Console.WriteLine("{0} - Универсальный счет", i + 1);

else if (cl.getAccount(i).GetType() == typeof(SavingAccount)) Console.WriteLine("{0} - Сберегательный аккаунт", i + 1);

else if (cl.getAccount(i).GetType() == typeof(TimedMaturityAccount)) Console.WriteLine("{0} - Счет со временем погашения платежа", i + 1);

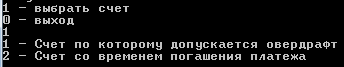
else if (cl.getAccount(i).GetType() == typeof(CheckingAccount)) Console.WriteLine("{0} - Текущий счет", i + 1);

else if (cl.getAccount(i).GetType() == typeof(OverdraftAccount)) Console.WriteLine("{0} - Счет по которому допускается овердрафт", i + 1);

}

int numAcc = Convert.ToInt32(Console.ReadLine());

}



После выбора счета пользователь может выбирать стандартные операции. Ниже приведен код, который реализует это:

do

{

Console.WriteLine("1 - Узнать баланс\n2 - Вложить фонды\n3 - Снять фонды\n4 - Снять деньги\n5 - Положить деньги\n0 - Выход");

res = Convert.ToInt32(Console.ReadLine());

if (res == 1) Console.WriteLine(acc.getBalance());

else if (res == 2)

{

Console.WriteLine("Введите сумму: ");

double sum = Convert.ToDouble(Console.ReadLine());

if (acc.investFunds(sum)) Console.WriteLine("Вложение прошло успешно");

else Console.WriteLine("Ошибка не хватает средств");

}

else if (res == 3)

{

Console.WriteLine("Введите сумму: ");

double sum = Convert.ToDouble(Console.ReadLine());

if (acc.takeFunds(sum)) Console.WriteLine("Снятие прошло успешно");

else Console.WriteLine("Ошибка не хватает средств");

}

else if (res == 4)

{

Console.WriteLine("Введите сумму: ");

double sum = Convert.ToDouble(Console.ReadLine());

if (acc.takeMoney(sum) != 0) Console.WriteLine("Снятие прошло успешно");

else Console.WriteLine("Ошибка не хватает средств");

}

else if (res == 5)

{

Console.WriteLine("Введите сумму: ");

double sum = Convert.ToDouble(Console.ReadLine());

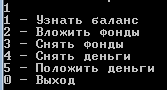
acc.addMoney(sum);

Console.WriteLine("Положено " + sum + " денег");

}

}

while (res != 0);



После выхода из программы выполняется метод logoff, который устанавливает status = false, что значит, что пользователь не использует банкомат.