|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий |
| Кафедра КБ-2 «Прикладные информационные технологии» |

**ПРАКТИЧЕСКАЯ РАБОТА №1 по дисциплине**

**«Технологии и методы программирования»**

***(наименование дисциплины)***

**Тема практической работы   
«Модельно-ориентированный подход к проектированию»**

Студент группы: БИСО-01-20 Маршев Л.А.

*(шифр учебной группы) (Фамилия И.О)*

Преподаватель: Лесько С.А.

*(Фамилия И.О)*

Москва, 2023

**Порядок выполнения практической работы**

1. При помощи программы PlantUML либо любого редактора построить UML-диаграмму вариантов использования, диаграмму классов проектируемой информационной системы в соответствии с вариантом задания, а также диаграмму последовательности для наиболее часто используемых прецедентов. При построении диаграммы классов нужно добиться достаточной детализации информационной системы. Убедитесь в том, что использовали отношения dependency, aggregation/composition, generalization, описали размещение классов по пакетам проекта.
2. Подготовить отчет с включением диаграмм. Загрузить в GIT. Прикрепить ссылку

**Вариант 1**

**Программное обеспечение банкомата**

Обзор: банкомат по карте позволяет снимать наличные со счета по и/или

печатать справку об остатке на счете.

**Диаграмма вариантов использования**

На диаграмме мы видим актера "Пользователь", который взаимодействует с нашим банкоматом, выполняя два варианта использования: "Снять наличные" и "Получить справку о балансе".

Код:

@startuml

left to right direction

actor User

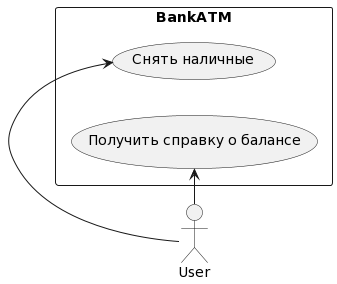
rectangle BankATM {

User -> (Снять наличные)

User -> (Получить справку о балансе)

}

@enduml



**Диаграмма классов**

На диаграмме мы видим четыре класса: "BankAccount", "CashDispenser", "ReceiptPrinter" и "BankATM". "BankAccount" отвечает за хранение информации о банковском счете, "CashDispenser" - за выдачу наличных, "ReceiptPrinter" - за печать справок, а "BankATM" - за взаимодействие со всеми компонентами системы. Классы "CashDispenser" и "ReceiptPrinter" имеют только один экземпляр в системе, поэтому мы использовали агрегацию. Класс "BankATM" использует экземпляры классов "BankAccount", "CashDispenser" и "ReceiptPrinter", поэтому мы использовали зависимость.

Код:

@startuml

package BankATM {

class BankAccount {

- accountNumber : int

- balance : int

+ BankAccount(accountNumber:int, balance:int)

+ getAccountNumber() : int

+ getBalance() : int

+ withdraw(amount:int) : boolean

}

class CashDispenser {

- cashOnHand : int

+ CashDispenser()

+ dispenseCash(amount:int) : boolean

}

class ReceiptPrinter {

+ ReceiptPrinter()

+ printReceipt(accountNumber:int, balance:int) : void

}

class BankATM {

- bankAccount : BankAccount

- cashDispenser : CashDispenser

- receiptPrinter : ReceiptPrinter

+ BankATM(bankAccount:BankAccount, cashDispenser:CashDispenser, receiptPrinter:ReceiptPrinter)

+ withdraw(amount:int) : void

+ getBalance() : int

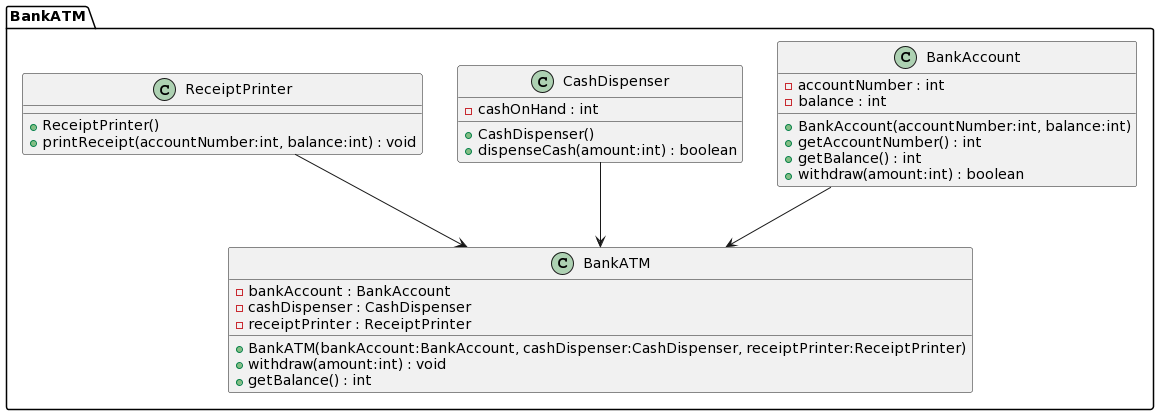
}

BankAccount --> BankATM

CashDispenser --> BankATM

ReceiptPrinter --> BankATM

}



**Диаграмма последовательности**

1. Первоначально, банкомат проверяет существует ли указанный на карте банковский счет в системе.
2. Пользователь вставляет карточку в банкомат и вводит ПИН-код.
3. Пользователь выбирает опцию "Снятие наличных".
4. Банкомат отправляет запрос на списание средств с банковского счета.
5. Банк проверяет возможность списания указанной суммы и, если все в порядке, отправляет команду на выдачу наличных.
6. Банкомат выдает наличные деньги пользователю.
7. Пользователь выбирает опцию "Печать справки об остатке на счете".
8. Банкомат отправляет запрос на получение баланса на банковском счете.
9. Банк отправляет информацию о балансе на банковский счет.
10. Банкомат печатает справку с информацией о балансе на банковском счете.

Диаграмма отображает последовательность действий между актером (клиентом), банкоматом и банковской системой.

Код:

@startuml

actor Customer

participant BankATM

database BankAccount

BankATM -> BankAccount: CheckAccount()

BankATM -> Customer: "Insert ATM card and enter PIN"

activate BankAccount

Customer -> BankATM: "Select Withdrawal option"

BankATM -> BankAccount: Debit(AccountNumber, Amount)

BankAccount -> BankATM: "Dispense cash"

deactivate BankAccount

BankATM -> Customer: "Take cash and card"

Customer -> BankATM: "Select Balance Inquiry option"

BankATM -> BankAccount: CheckBalance(AccountNumber)

BankAccount -> BankATM: "Print balance on receipt"

@enduml

