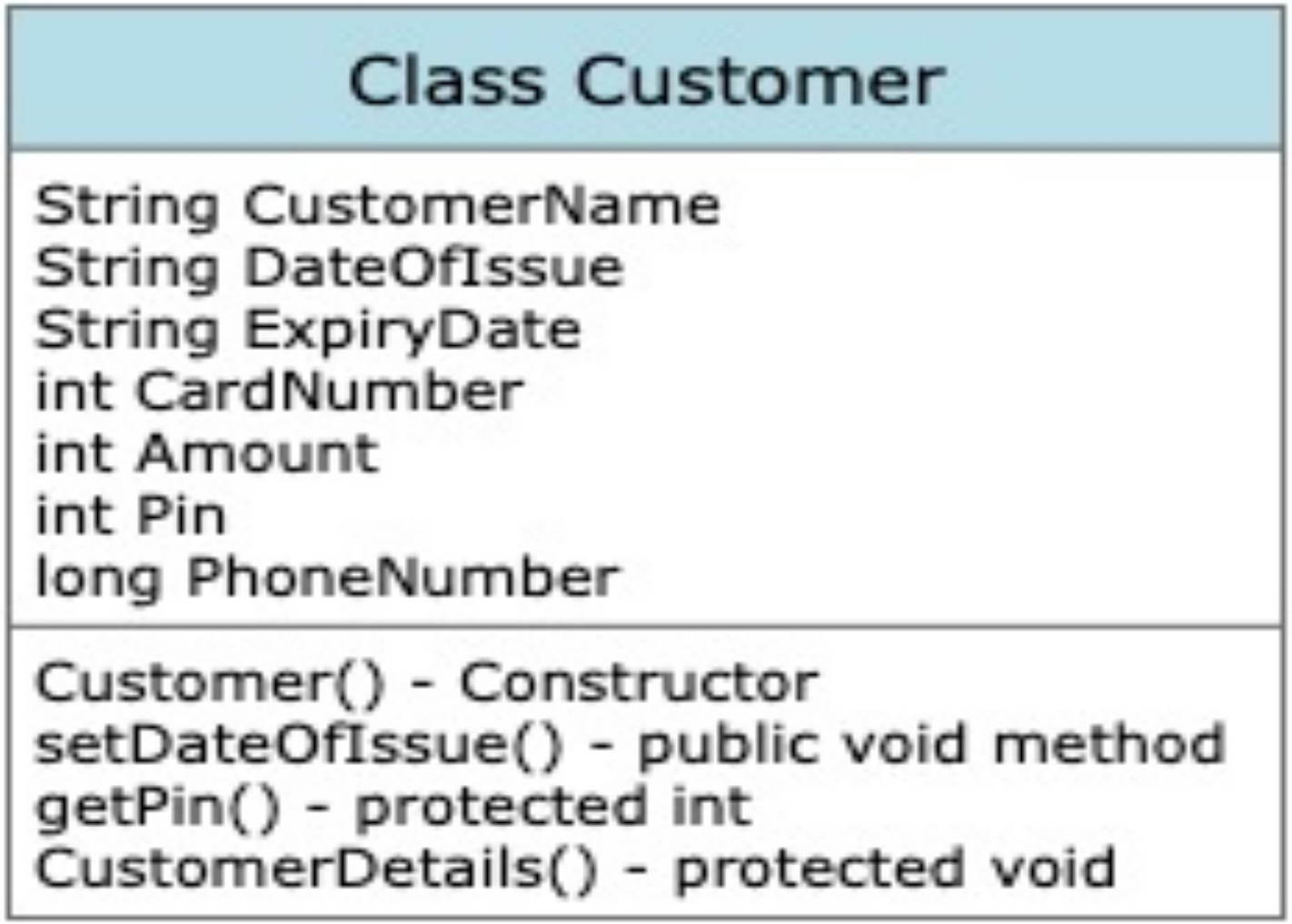
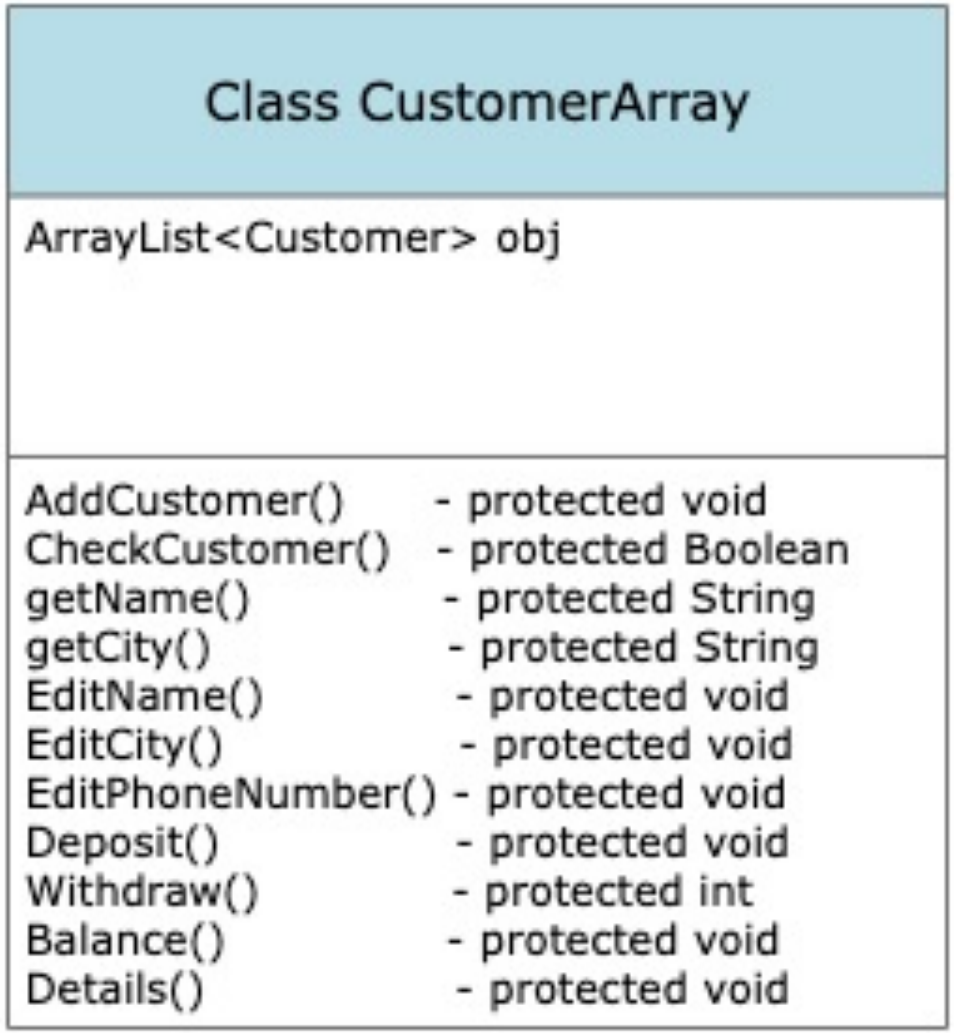
CASE STUDY 2

ATM Machine Case study in Java


Class Customer
String CustomerName
String DateOfIssue
String ExpiryDate
int CardNumber
int Amount
int Pin
long PhoneNumber
Customer() - Constructor
setDateOfIssue() - public void method
getPin() - protected int
CustomerDetails() - protected void

Class Customer

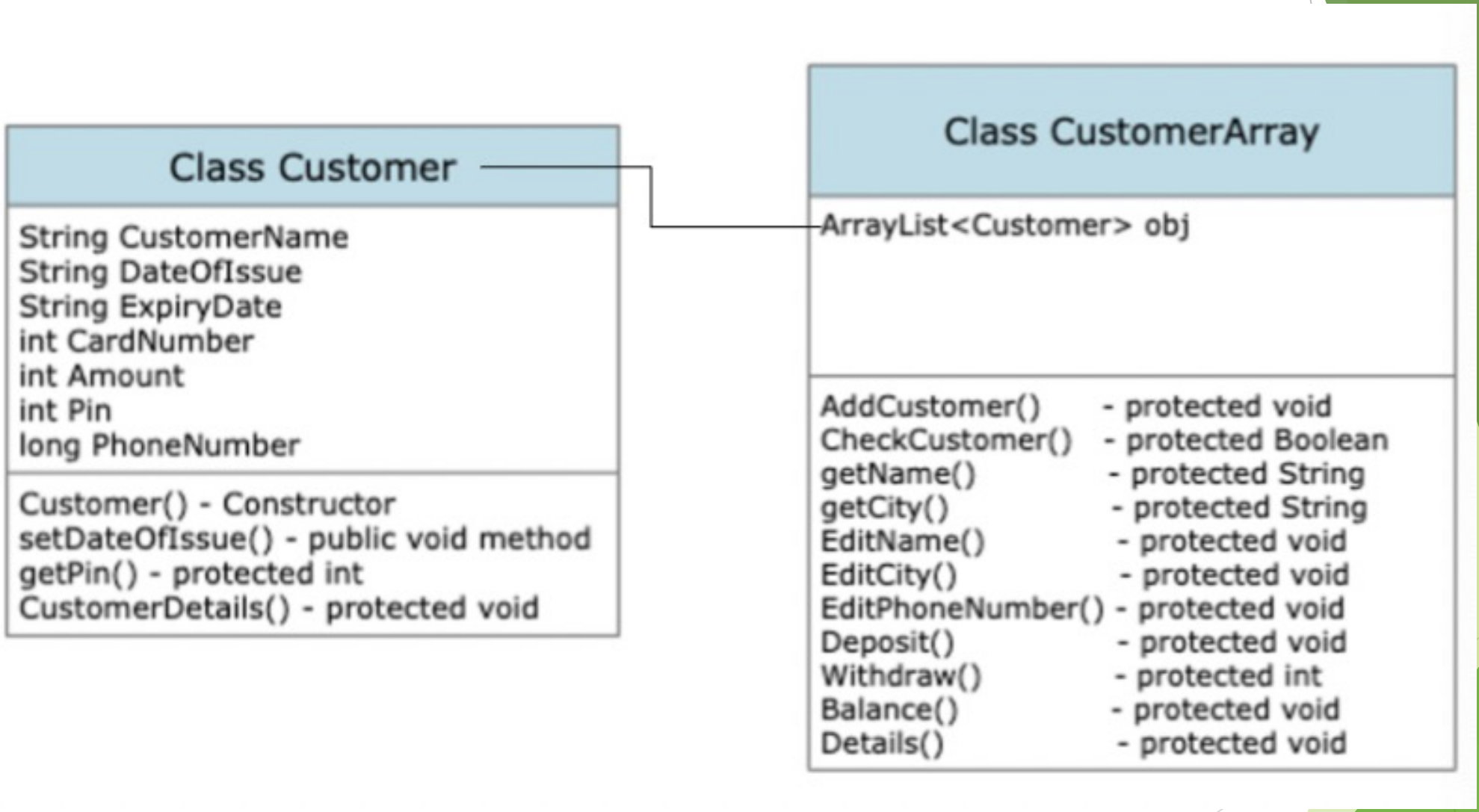
- This class is the basis of the customer accounts. This contains the customer details like Customer name, Phone number, Date of issue, Expiry Date, Card Number, Pin and amount in the account.
- The above info about the customer is initialized using the constructor Customer()
- To take the day's date as date of issue, a method setDateOfIssue() is created. It also gives the expiry date(which is 4 years from Date of issue).
- getPin() method is created to get the pin of the customer.
- CustomerDetails() method gives the mini statement.

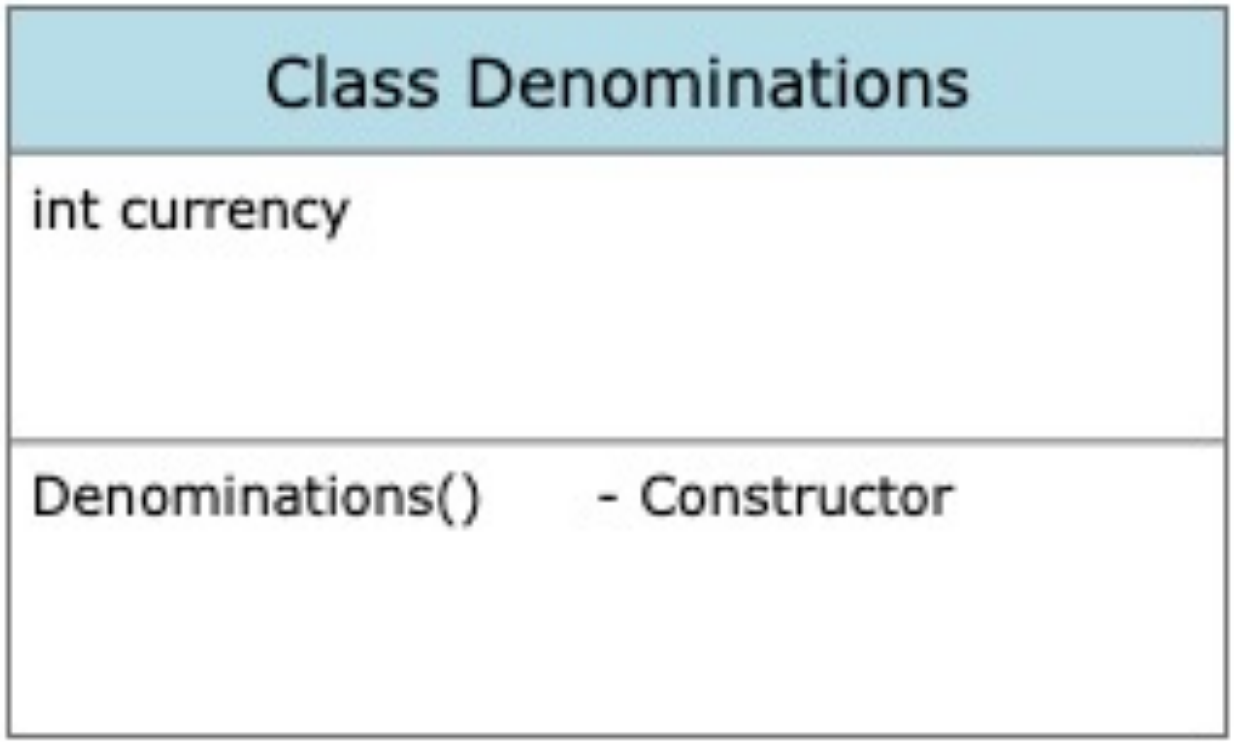

Class CustomerArray
ArrayList<Customer> obj
AddCustomer() - protected void
CheckCustomer() - protected Boolean
getName() - protected String
getCity() - protected String
EditName() - protected void
EditCity() - protected void
EditPhoneNumber() - protected void
Deposit() - protected void
Withdraw() - protected int
Balance() - protected void
Details() - protected void

Class CustomerArray

- This class is used to add or delete accounts to the bank. An ArrayList of Customers are taken. Arraylist is used for the dynamic allocation of memory.
- Each element of the ArrayList is initialized using the addDenominations() method.
- checkCustomer() method ius used to check if a given account number and pin are in the accounts of bank or not.
- getName() and getCity() are encapsulating methods used to return Name of user and City where the user resides in.
- EditName(), EditCity() and EditPhoneNumber() are methods used to edit the Customer Details.

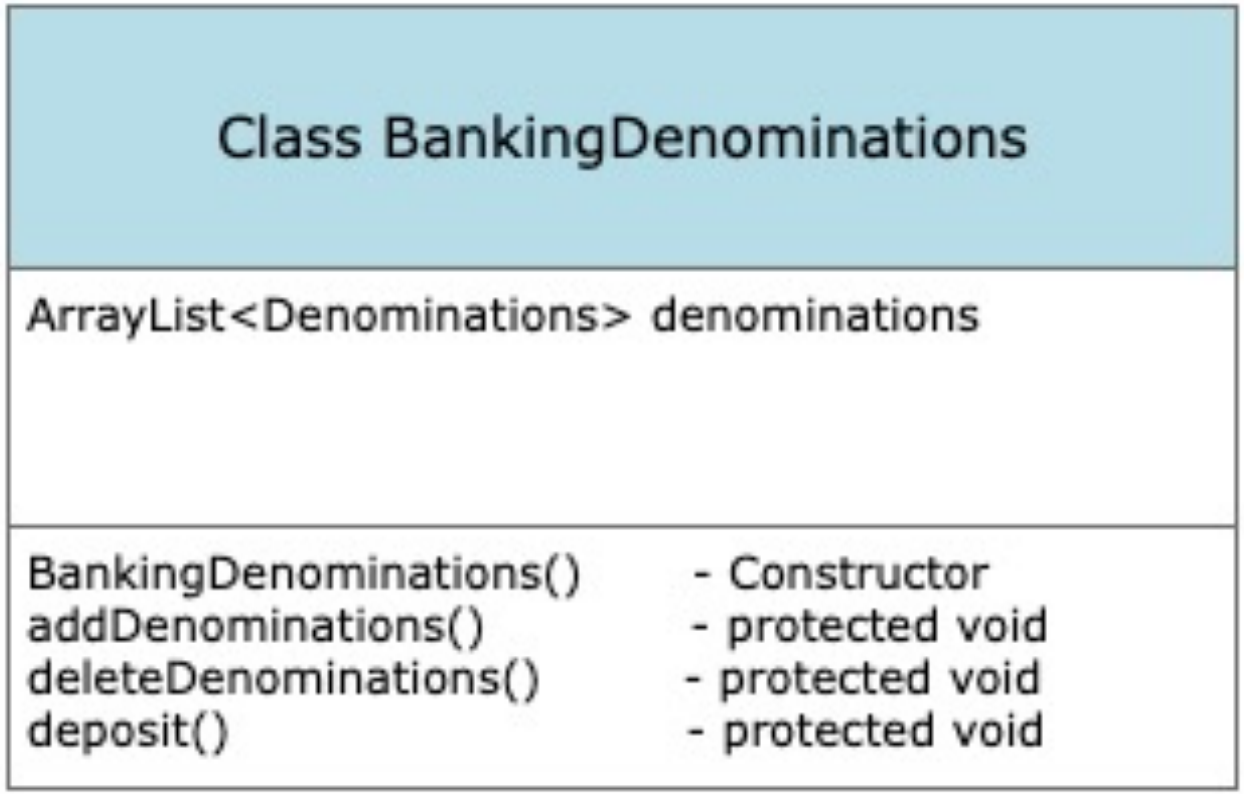
- Deposit() method is used for depositing money to user’s account through money vender.
- Withdraw() method is used for withdrawing money from user’s account through the ATM.
- Balance() method is used to check the remining balance in account.
- Details() method is used to print the user’s details.


Class Customer
String CustomerName
String DateOfIssue
String ExpiryDate
int CardNumber
int Amount
int Pin
long PhoneNumber
Customer() - Constructor
setDateOfIssue() - public void method
getPin() - protected int
CustomerDetails() - protected void
Class CustomerArray
ArrayList<Customer> obj
AddCustomer() - protected void
CheckCustomer() - protected Boolean
getName() - protected String
getCity() - protected String
EditName() - protected void
EditCity() - protected void
EditPhoneNumber() - protected void
Deposit() - protected void
Withdraw() - protected int
Balance() - protected void
Details() - protected void


Class Denominations
int currency
Denominations() - Constructor

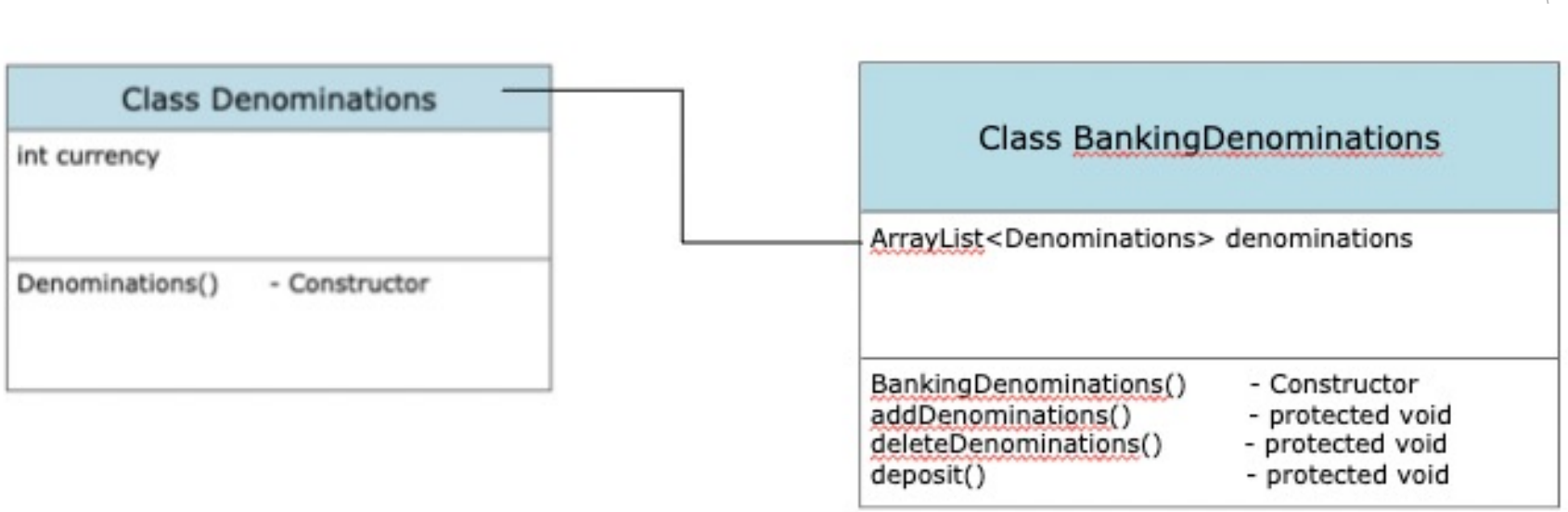
Class Denominations

- This class is used to know about the currency denominations of the state of the art currency.
- The constructor initializes the currency.


Class BankingDenominations
ArrayList<Denominations> denominations
BankingDenominations() - Constructor
addDenominations() - protected void
deleteDenominations() - protected void
deposit() - protected void

Class BankingDenominations

- This class is used to initialise the currency notes by using the object of Denominations class.
- addDenominations() is a method used to add a new currency note to the system.
- deleteDenominations() serves the purpose od deleting a currency not not in use.


Class Denominations
int currency
Denominations() - Constructor
Class BankingDenominations
ArrayList<Denominations> denominations
BankingDenominations() - Constructor
addDenominations() - protected void
deleteDenominations() - protected void
deposit() - protected void

Design of System

- The system is designed so as to ensure maximum dynamic allocation of memory.
- An arrayList implementation of the bank accounts is used for the system design which makes it easy to do the necessary transactions.
- An object oriented approach towards the problem is ensured at every stage of the code.
- Encapsulation, Polymorphism and other object oriented features are used in the code writing.

Functionalities of ATM System

- The ATM system is designed dynamically. This implementation is easier for the access of accounts.
- The system is made user-friendly for the user to easily access the code.
- This ATM system has functionalities of depositing, withdrawing, printing mini statement and to ensure more safety an OTP generator is also used for this code.
- A currency denominations class is also included for the dynamic usage of the denominations. For example, if there is a new currency note coming into existence, then it is easier to add the currency to the system.

Advantages of the System

- This system is designed for dynamic allocation of memory using array list implementation.
- An OTP generator is used in main class to generate OTP to ensure safety.
- Documentation for every part of the code is present. It ensures readability of the code.
- All OOPS concepts and features are present in the code.
- It has a great user interface which guides the user to different options by an integrated Menu system.

Limitations of System

- There is a limit to the number of accounts in the system.
- The number of accounts can not exceed 100000 in this system. It can be extended if necessary.

Sri Vibhav J
CS20B047