JAVA PROJECT ON ATM APPLICATION BY USING OOPs CONCEPT

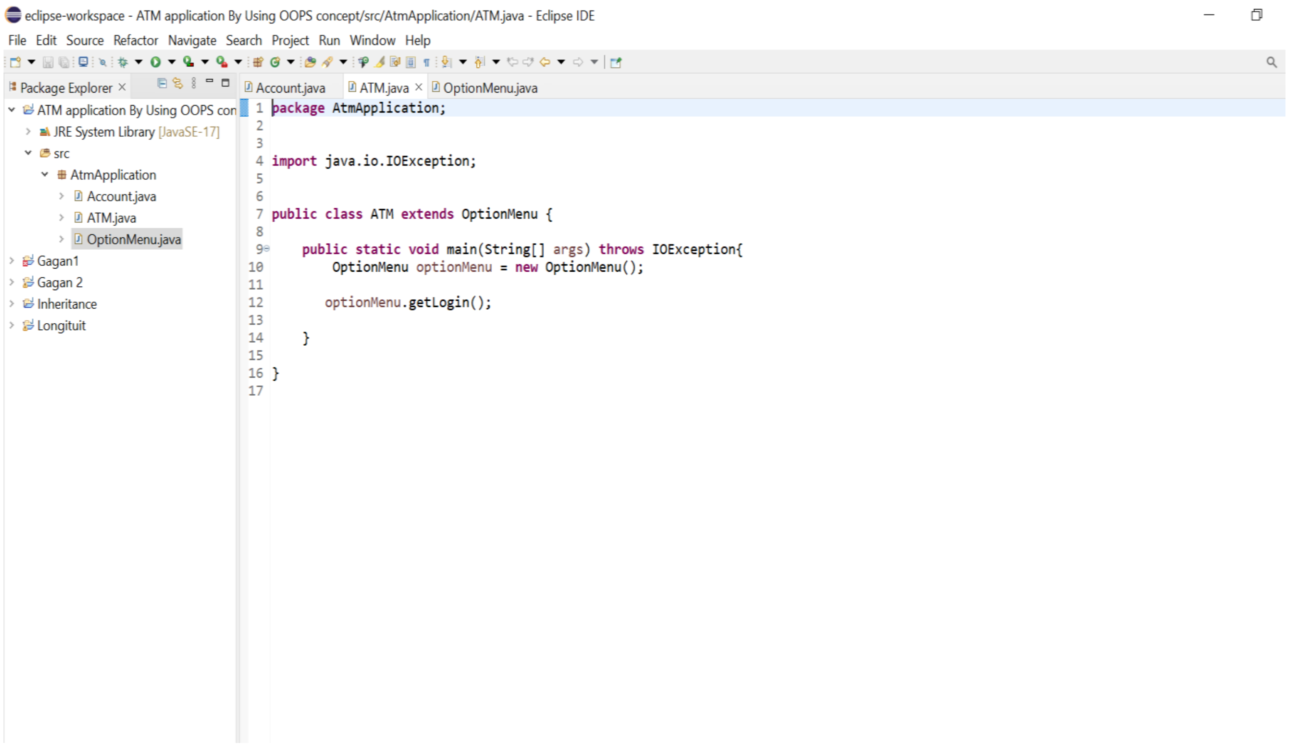
PROJECT STATEMENT:

This project is to make an automated teller machine (ATM) with customer number and customer pin number with that customer are able to withdraw, deposit and view their account balance.

Basically, I created three main classes:

1) **ATM Class**
2) **OptionMenu Class**
3) **Account Class**

1) **ATM Class:**
   It is mainly an entry point class which basically initializes OptionMenu class and calls the get login method. It is a first method in my ATM application program in which ATM extends OptionMenu class by using EXTENDS keyword.


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/ATM.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java

```java
package AtmApplication;


import java.io.IOException;


public class ATM extends OptionMenu {


    public static void main(String[] args) throws IOException{
        OptionMenu optionMenu = new OptionMenu();

        optionMenu.getLogin();

    }

}
```

2) **OptionMenu Class:**
   It will give the understanding of what kind of functionality iam using,
   Here mainly iam using three main class functions:
   1) Java.io.IOException
   2) Java.text.DecimalFormat
   3) Java.util.Hashmap
   4) Java.util.Scanner

- IOException class is used to check exception that occurs during the compilation time of the java program.
- HashMap class is used to provide functionality of the hash table data structure and it is used to store key/value pairs.
- We use Decimal Format class in order to denote what kind of format of money we have.
- We use HashMap for storing couple of records of account number to pin mapping.

First Method I used is Here mainly we use getLogin() method where we had couple of entries to map, Where I entered customer number with customer pin number and I done same for 2 more entries(It can be done for infinity numbers).

If we call the getLogin() method it will insert the customer number and customer pin number.
If we give wrong customer number it would give wrong customer number or Invalid which I was written in catch block

Second Method I used is getaccount type(). In this where the program asks the user what type of account you want to access.

There are mainly two type of account I initialized,

1) **Checking Account**
2) **Saving Account**
3) **Exit**

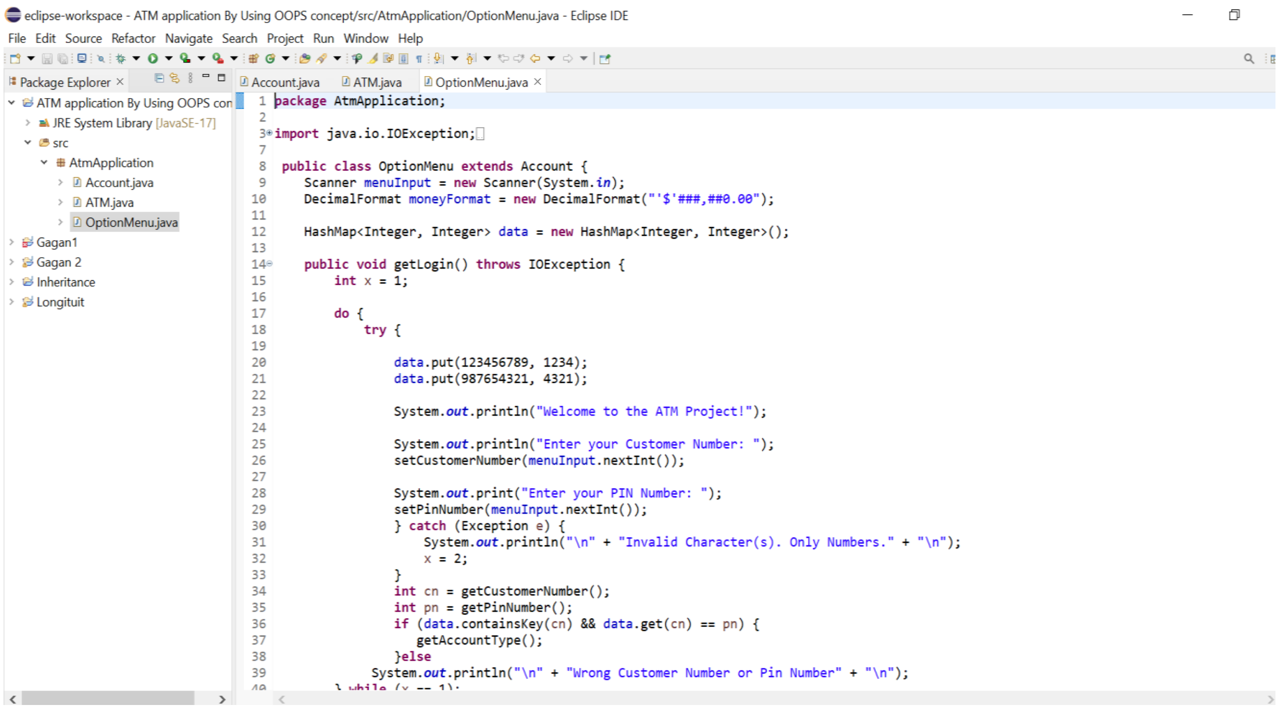
Here I used the Switch condition to perform some operations where if I press 1 it goes to Checking account, similarly if I press 2 it goes to Saving Account and if I press 3 it Exits from the ATM application.

Third Method and Fourth Method is getChecking method and getSaving method

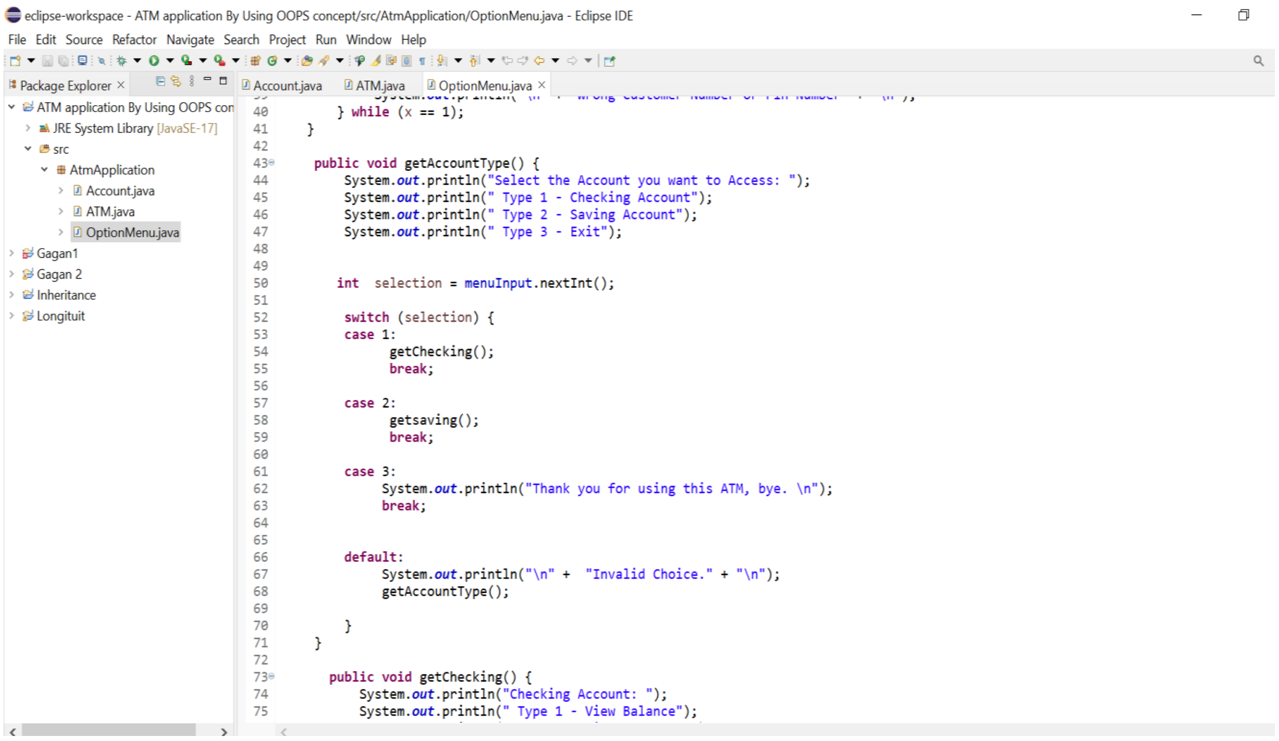
In this it performs three different Account Functions

1) **View Balance**
2) **Withdraw Funds**
3) **Deposit Funds**
4) **Exit**

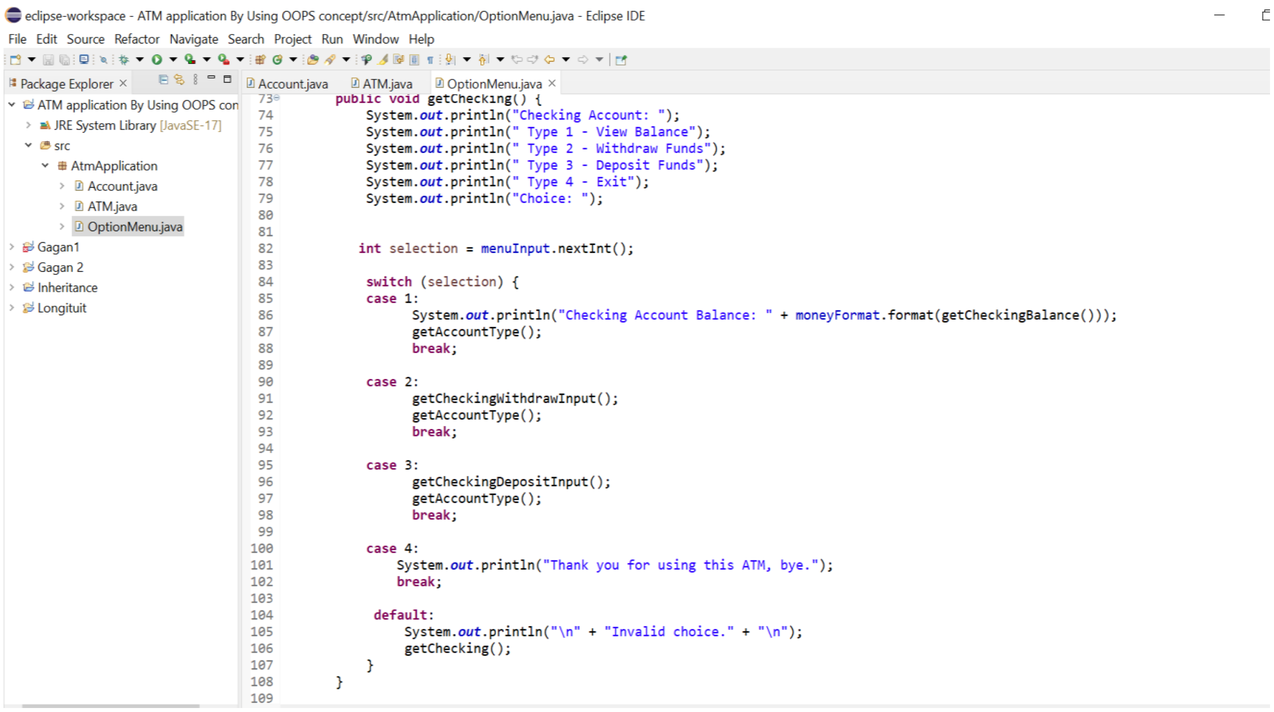
Here I used the Switch condition to perform some operations where if I press 1 it checks account balance, if I press 2 it debits money from account, similarly if I press 3 it credits money to account and if I press 4 it Exits from the ATM account Functions (This method is same for both Account Types).


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/OptionMenu.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java

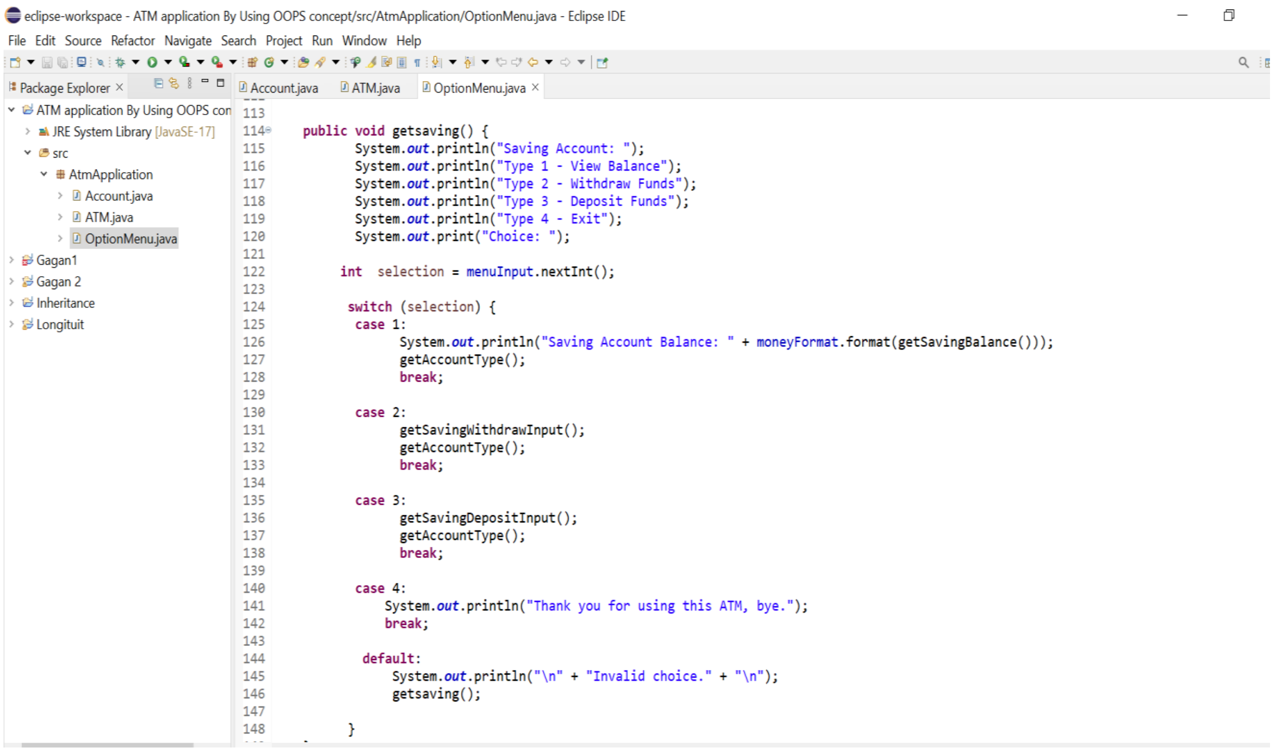
```java
package AtmApplication;

import java.io.IOException;

 public class OptionMenu extends Account {
    Scanner menuInput = new Scanner(System.in);
    DecimalFormat moneyFormat = new DecimalFormat("'$'###,##0.00");

    HashMap<Integer, Integer> data = new HashMap<Integer, Integer>();

    public void getLogin() throws IOException {
        int x = 1;

        do {
            try {

                data.put(123456789, 1234);
                data.put(987654321, 4321);

                System.out.println("Welcome to the ATM Project!");

                System.out.println("Enter your Customer Number: ");
                setCustomerNumber(menuInput.nextInt());

                System.out.print("Enter your PIN Number: ");
                setPinNumber(menuInput.nextInt());
                } catch (Exception e) {
                    System.out.println("\n" + "Invalid Character(s). Only Numbers." + "\n");
                    x = 2;
                }
                int cn = getCustomerNumber();
                int pn = getPinNumber();
                if (data.containsKey(cn) && data.get(cn) == pn) {
                   getAccountType();
                }else
             System.out.println("\n" + "Wrong Customer Number or Pin Number" + "\n");
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/OptionMenu.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java

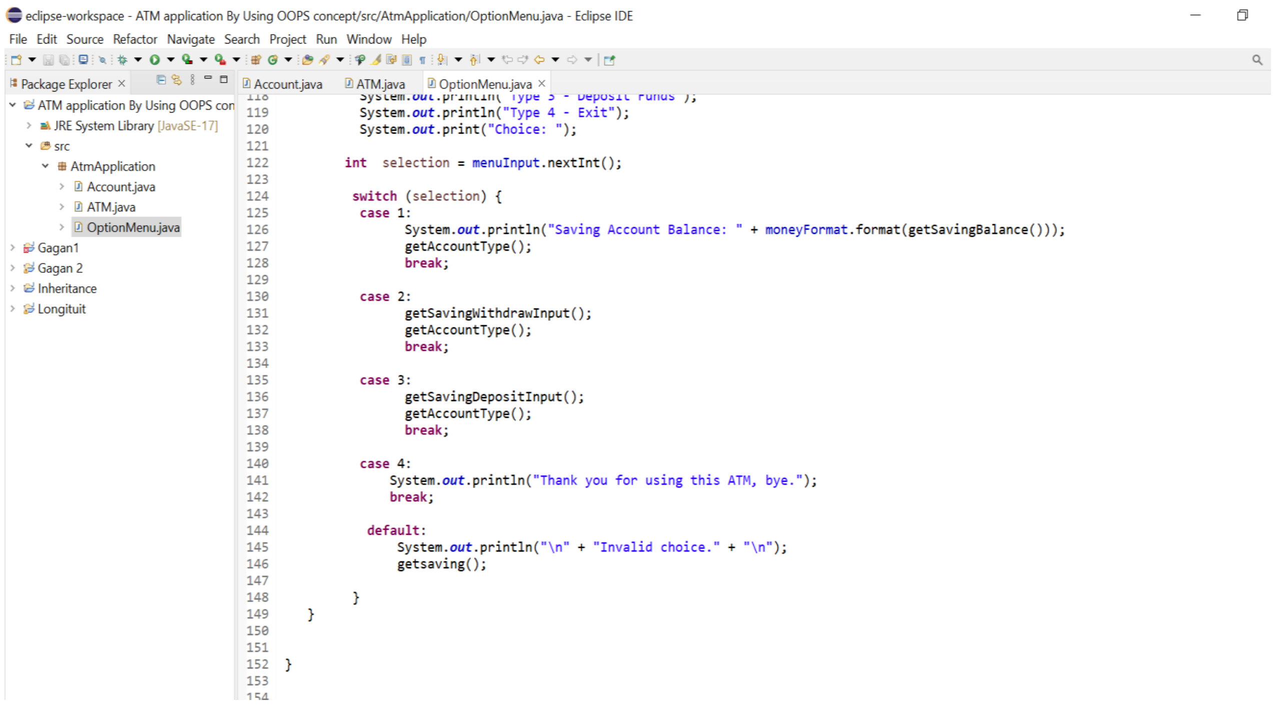
```java
        } while (x == 1);
    }

    public void getAccountType() {
        System.out.println("Select the Account you want to Access: ");
        System.out.println(" Type 1 - Checking Account");
        System.out.println(" Type 2 - Saving Account");
        System.out.println(" Type 3 - Exit");


        int  selection = menuInput.nextInt();

        switch (selection) {
        case 1:
              getChecking();
              break;

        case 2:
              getsaving();
              break;

        case 3:
             System.out.println("Thank you for using this ATM, bye. \n");
             break;


        default:
             System.out.println("\n" +  "Invalid Choice." + "\n");
             getAccountType();

        }
    }

      public void getChecking() {
          System.out.println("Checking Account: ");
          System.out.println(" Type 1 - View Balance");
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/OptionMenu.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Account.java
ATM.java
OptionMenu.java
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit

```java
    public void getChecking() {
        System.out.println("Checking Account: ");
        System.out.println(" Type 1 - View Balance");
        System.out.println(" Type 2 - Withdraw Funds");
        System.out.println(" Type 3 - Deposit Funds");
        System.out.println(" Type 4 - Exit");
        System.out.println("Choice: ");


       int selection = menuInput.nextInt();

        switch (selection) {
        case 1:
              System.out.println("Checking Account Balance: " + moneyFormat.format(getCheckingBalance()));
              getAccountType();
              break;

        case 2:
              getCheckingWithdrawInput();
              getAccountType();
              break;

        case 3:
              getCheckingDepositInput();
              getAccountType();
              break;

        case 4:
            System.out.println("Thank you for using this ATM, bye.");
            break;

         default:
             System.out.println("\n" + "Invalid choice." + "\n");
             getChecking();
        }
    }
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/OptionMenu.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Account.java
ATM.java
OptionMenu.java
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit

```java

    public void getsaving() {
           System.out.println("Saving Account: ");
           System.out.println("Type 1 - View Balance");
           System.out.println("Type 2 - Withdraw Funds");
           System.out.println("Type 3 - Deposit Funds");
           System.out.println("Type 4 - Exit");
           System.out.print("Choice: ");

         int  selection = menuInput.nextInt();

          switch (selection) {
           case 1:
                 System.out.println("Saving Account Balance: " + moneyFormat.format(getSavingBalance()));
                 getAccountType();
                 break;

           case 2:
                 getSavingWithdrawInput();
                 getAccountType();
                 break;

           case 3:
                 getSavingDepositInput();
                 getAccountType();
                 break;

           case 4:
               System.out.println("Thank you for using this ATM, bye.");
               break;

            default:
                System.out.println("\n" + "Invalid choice." + "\n");
                getsaving();

          }
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/OptionMenu.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java

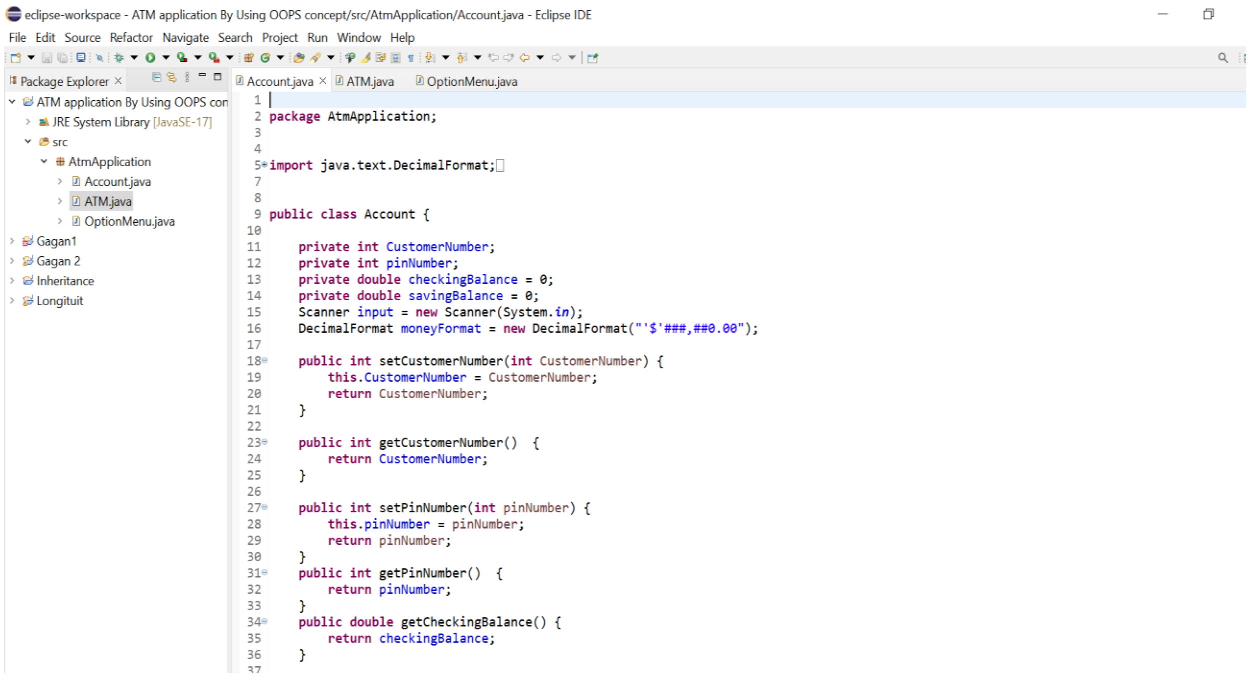
```java
        System.out.println("Type 3 - Deposit Funds");
        System.out.println("Type 4 - Exit");
        System.out.print("Choice: ");

      int  selection = menuInput.nextInt();

       switch (selection) {
        case 1:
              System.out.println("Saving Account Balance: " + moneyFormat.format(getSavingBalance()));
              getAccountType();
              break;

        case 2:
              getSavingWithdrawInput();
              getAccountType();
              break;

        case 3:
              getSavingDepositInput();
              getAccountType();
              break;

        case 4:
            System.out.println("Thank you for using this ATM, bye.");
            break;

         default:
             System.out.println("\n" + "Invalid choice." + "\n");
             getsaving();

       }
    }


}
```

3) Account Class:

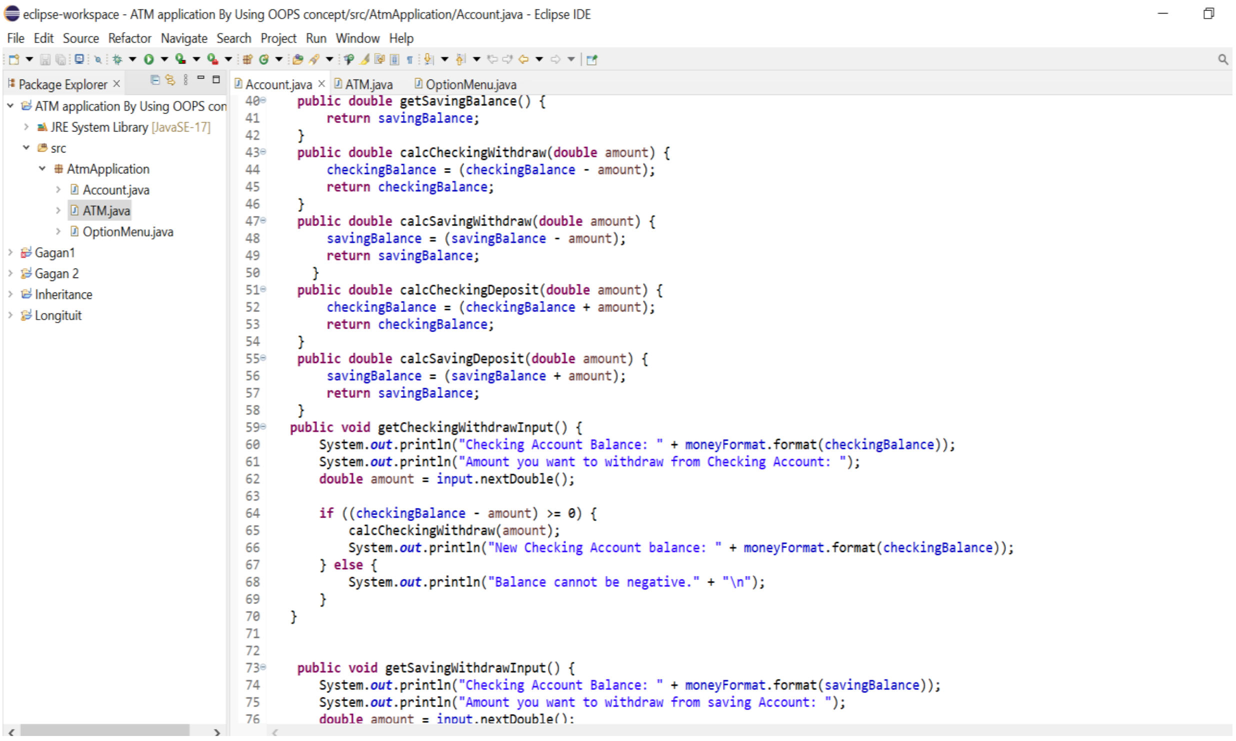
The entire transaction calculation part of a program happens in the account class and Here I defined private member variables like Customer number, Pin number, checking balance and Saving balance.

Then mainly I used Getters and Setters method for what I specified in private member variables.

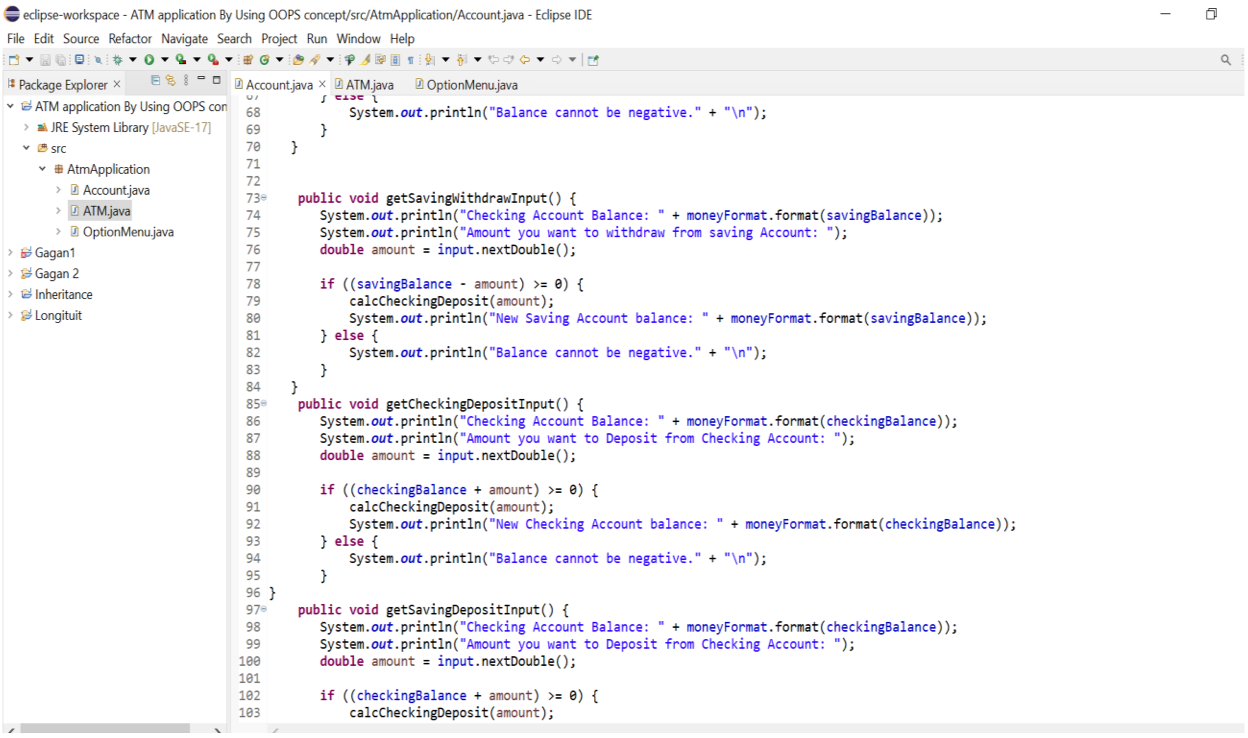
\


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/Account.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java

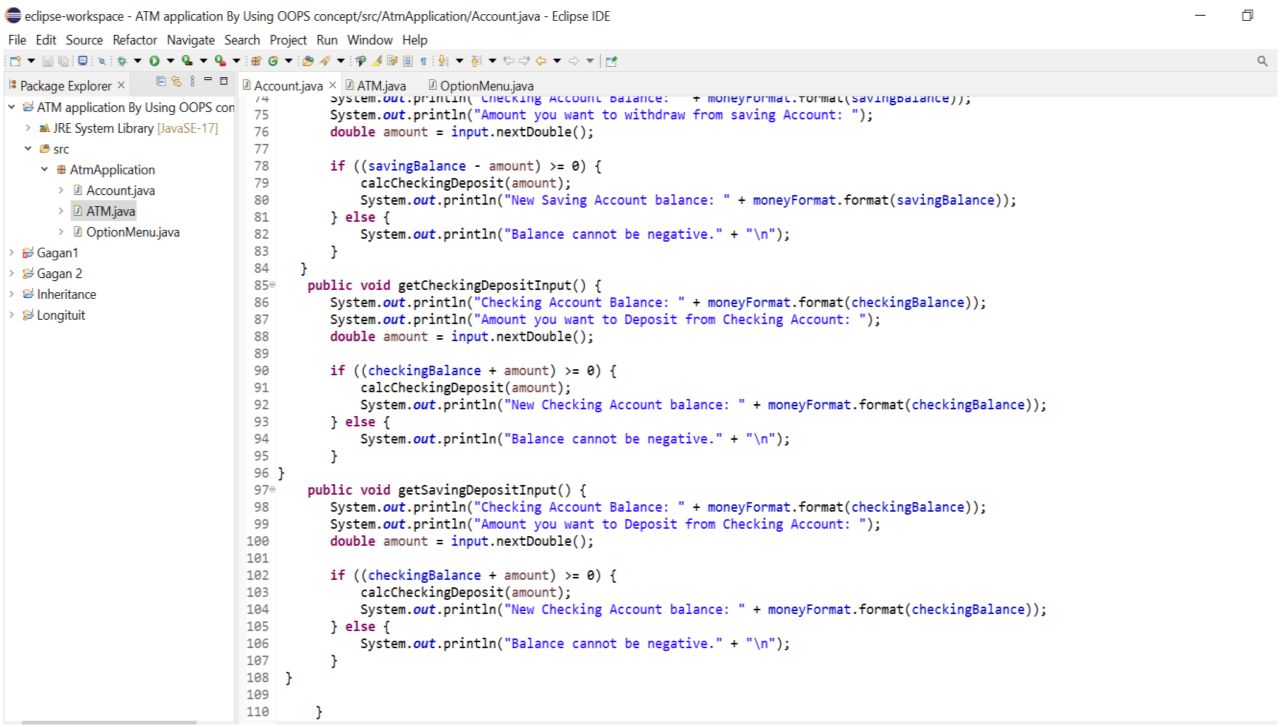
```java

package AtmApplication;


import java.text.DecimalFormat;


public class Account {

    private int CustomerNumber;
    private int pinNumber;
    private double checkingBalance = 0;
    private double savingBalance = 0;
    Scanner input = new Scanner(System.in);
    DecimalFormat moneyFormat = new DecimalFormat("'$'###,##0.00");

    public int setCustomerNumber(int CustomerNumber) {
        this.CustomerNumber = CustomerNumber;
        return CustomerNumber;
    }

    public int getCustomerNumber()  {
        return CustomerNumber;
    }

    public int setPinNumber(int pinNumber) {
        this.pinNumber = pinNumber;
        return pinNumber;

    }
    public int getPinNumber()  {
        return pinNumber;
    }
    public double getCheckingBalance() {
        return checkingBalance;
    }
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/Account.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java

```java
    public double getSavingBalance() {
        return savingBalance;
    }
    public double calcCheckingWithdraw(double amount) {
        checkingBalance = (checkingBalance - amount);
        return checkingBalance;
    }
    public double calcSavingWithdraw(double amount) {
        savingBalance = (savingBalance - amount);
        return savingBalance;
      }
    public double calcCheckingDeposit(double amount) {
        checkingBalance = (checkingBalance + amount);
        return checkingBalance;
    }
    public double calcSavingDeposit(double amount) {
        savingBalance = (savingBalance + amount);
        return savingBalance;
    }
   public void getCheckingWithdrawInput() {
       System.out.println("Checking Account Balance: " + moneyFormat.format(checkingBalance));
       System.out.println("Amount you want to withdraw from Checking Account: ");
       double amount = input.nextDouble();

       if ((checkingBalance - amount) >= 0) {
           calcCheckingWithdraw(amount);
           System.out.println("New Checking Account balance: " + moneyFormat.format(checkingBalance));
       } else {
           System.out.println("Balance cannot be negative." + "\n");
       }
   }


    public void getSavingWithdrawInput() {
       System.out.println("Checking Account Balance: " + moneyFormat.format(savingBalance));
       System.out.println("Amount you want to withdraw from saving Account: ");
       double amount = input.nextDouble();
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/Account.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java

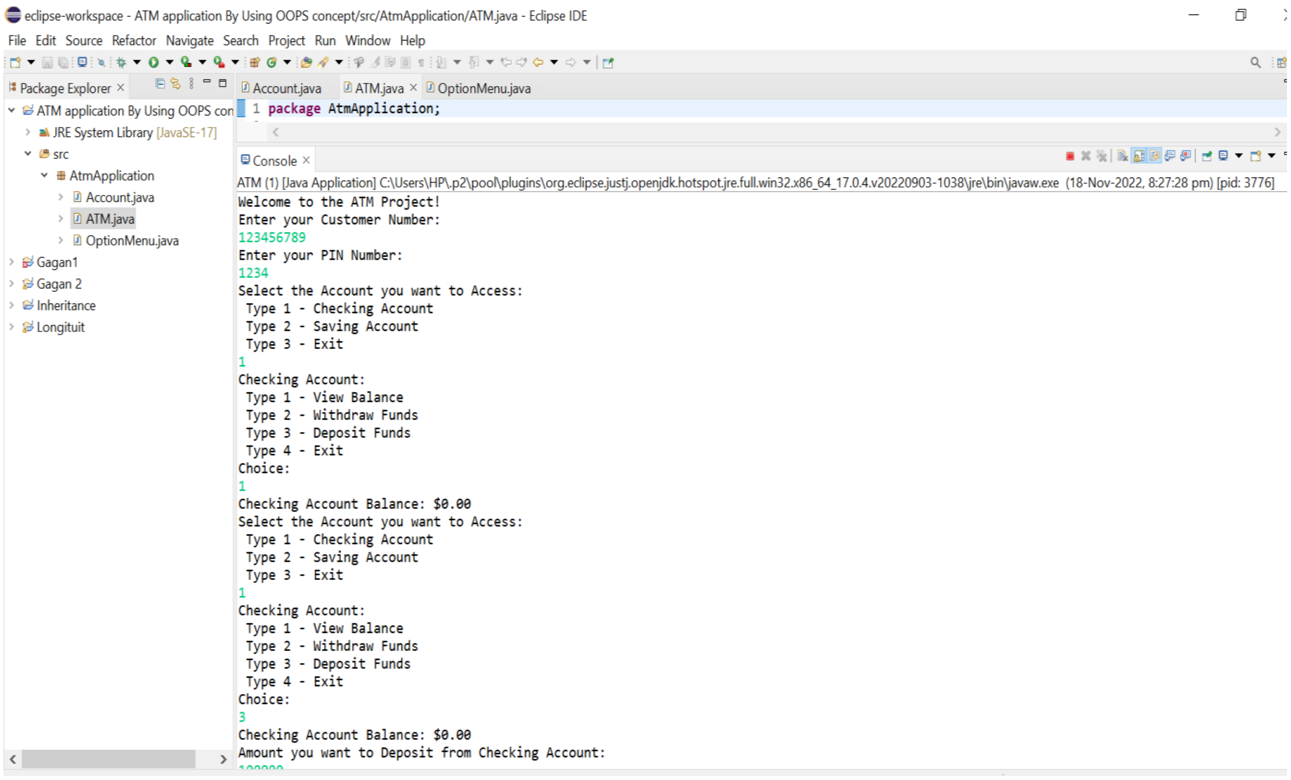
```java
       } else {
           System.out.println("Balance cannot be negative." + "\n");
       }
   }


    public void getSavingWithdrawInput() {
       System.out.println("Checking Account Balance: " + moneyFormat.format(savingBalance));
       System.out.println("Amount you want to withdraw from saving Account: ");
       double amount = input.nextDouble();

       if ((savingBalance - amount) >= 0) {
           calcCheckingDeposit(amount);
           System.out.println("New Saving Account balance: " + moneyFormat.format(savingBalance));
       } else {
           System.out.println("Balance cannot be negative." + "\n");
       }
   }
    public void getCheckingDepositInput() {
       System.out.println("Checking Account Balance: " + moneyFormat.format(checkingBalance));
       System.out.println("Amount you want to Deposit from Checking Account: ");
       double amount = input.nextDouble();

       if ((checkingBalance + amount) >= 0) {
           calcCheckingDeposit(amount);
           System.out.println("New Checking Account balance: " + moneyFormat.format(checkingBalance));
       } else {
           System.out.println("Balance cannot be negative." + "\n");
       }
}
    public void getSavingDepositInput() {
       System.out.println("Checking Account Balance: " + moneyFormat.format(checkingBalance));
       System.out.println("Amount you want to Deposit from Checking Account: ");
       double amount = input.nextDouble();

       if ((checkingBalance + amount) >= 0) {
           calcCheckingDeposit(amount);
```



eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/Account.java - Eclipse IDE

```java
        System.out.println("Checking Account Balance: " + moneyFormat.format(savingBalance));
        System.out.println("Amount you want to withdraw from saving Account: ");
        double amount = input.nextDouble();

        if ((savingBalance - amount) >= 0) {
            calcCheckingDeposit(amount);
            System.out.println("New Saving Account balance: " + moneyFormat.format(savingBalance));
        } else {
            System.out.println("Balance cannot be negative." + "\n");
        }
    }
    public void getCheckingDepositInput() {
        System.out.println("Checking Account Balance: " + moneyFormat.format(checkingBalance));
        System.out.println("Amount you want to Deposit from Checking Account: ");
        double amount = input.nextDouble();

        if ((checkingBalance + amount) >= 0) {
            calcCheckingDeposit(amount);
            System.out.println("New Checking Account balance: " + moneyFormat.format(checkingBalance));
        } else {
            System.out.println("Balance cannot be negative." + "\n");
        }
}
    public void getSavingDepositInput() {
        System.out.println("Checking Account Balance: " + moneyFormat.format(checkingBalance));
        System.out.println("Amount you want to Deposit from Checking Account: ");
        double amount = input.nextDouble();

        if ((checkingBalance + amount) >= 0) {
            calcCheckingDeposit(amount);
            System.out.println("New Checking Account balance: " + moneyFormat.format(checkingBalance));
        } else {
            System.out.println("Balance cannot be negative." + "\n");
        }
 }

     }
```

OUTPUT:

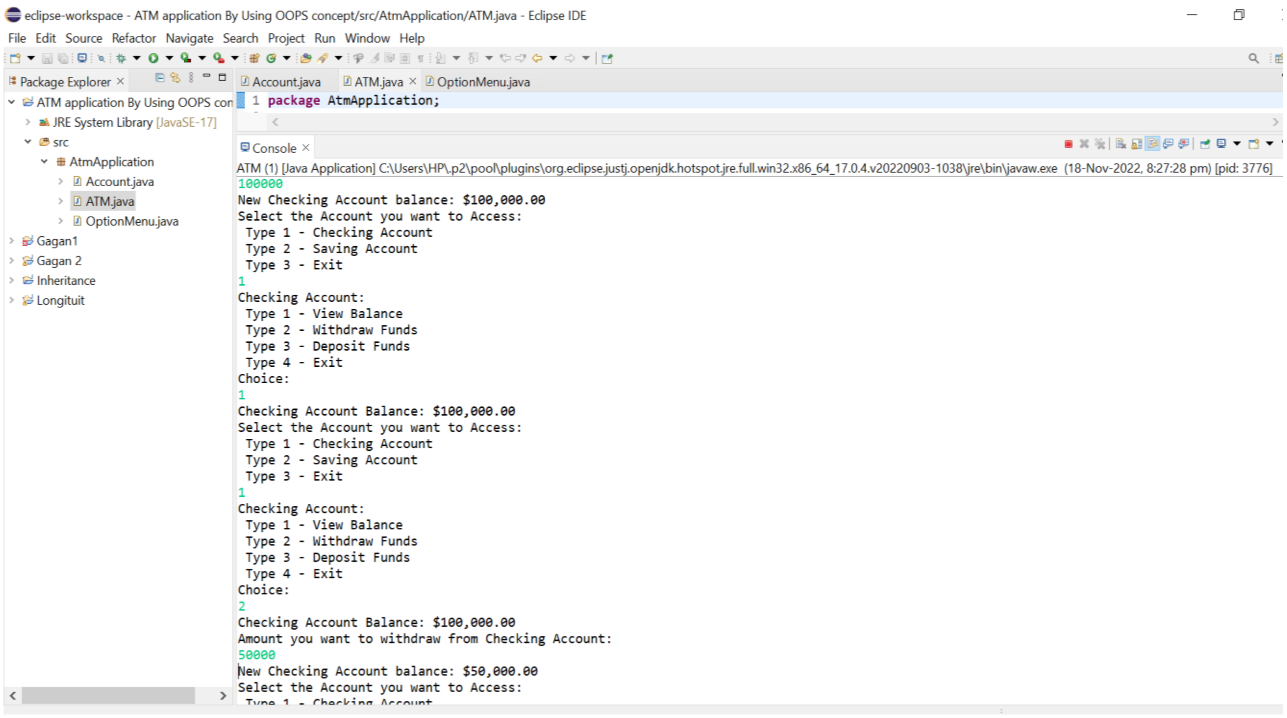


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/ATM.java - Eclipse IDE

```java
package AtmApplication;
```

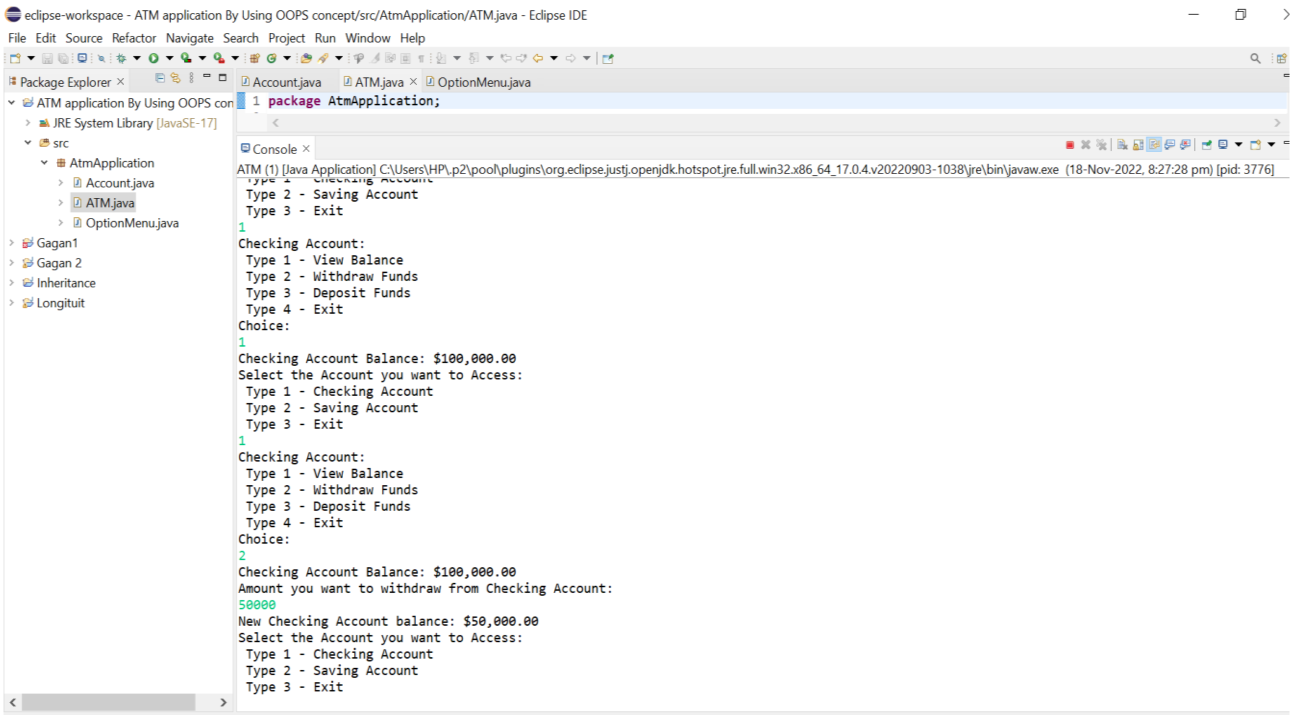
ATM (1) [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe (18-Nov-2022, 8:27:28 pm) [pid: 3776]

```
Welcome to the ATM Project!
Enter your Customer Number:
123456789
Enter your PIN Number:
1234
Select the Account you want to Access:
 Type 1 - Checking Account
 Type 2 - Saving Account
 Type 3 - Exit
1
Checking Account:
 Type 1 - View Balance
 Type 2 - Withdraw Funds
 Type 3 - Deposit Funds
 Type 4 - Exit
Choice:
1
Checking Account Balance: $0.00
Select the Account you want to Access:
 Type 1 - Checking Account
 Type 2 - Saving Account
 Type 3 - Exit
1
Checking Account:
 Type 1 - View Balance
 Type 2 - Withdraw Funds
 Type 3 - Deposit Funds
 Type 4 - Exit
Choice:
3
Checking Account Balance: $0.00
Amount you want to Deposit from Checking Account:
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/ATM.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java
Console

```
package AtmApplication;
```

```
ATM (1) [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe  (18-Nov-2022, 8:27:28 pm) [pid: 3776]
100000
New Checking Account balance: $100,000.00
Select the Account you want to Access:
 Type 1 - Checking Account
 Type 2 - Saving Account
 Type 3 - Exit
1
Checking Account:
 Type 1 - View Balance
 Type 2 - Withdraw Funds
 Type 3 - Deposit Funds
 Type 4 - Exit
Choice:
1
Checking Account Balance: $100,000.00
Select the Account you want to Access:
 Type 1 - Checking Account
 Type 2 - Saving Account
 Type 3 - Exit
1
Checking Account:
 Type 1 - View Balance
 Type 2 - Withdraw Funds
 Type 3 - Deposit Funds
 Type 4 - Exit
Choice:
2
Checking Account Balance: $100,000.00
Amount you want to withdraw from Checking Account:
50000
New Checking Account balance: $50,000.00
Select the Account you want to Access:
 Type 1 - Checking Account
```


eclipse-workspace - ATM application By Using OOPS concept/src/AtmApplication/ATM.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ATM application By Using OOPS con
JRE System Library [JavaSE-17]
src
AtmApplication
Account.java
ATM.java
OptionMenu.java
Gagan1
Gagan 2
Inheritance
Longituit
Account.java
ATM.java
OptionMenu.java
Console

```
package AtmApplication;
```

```
ATM (1) [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe  (18-Nov-2022, 8:27:28 pm) [pid: 3776]
 Type 2 - Saving Account
 Type 3 - Exit
1
Checking Account:
 Type 1 - View Balance
 Type 2 - Withdraw Funds
 Type 3 - Deposit Funds
 Type 4 - Exit
Choice:
1
Checking Account Balance: $100,000.00
Select the Account you want to Access:
 Type 1 - Checking Account
 Type 2 - Saving Account
 Type 3 - Exit
1
Checking Account:
 Type 1 - View Balance
 Type 2 - Withdraw Funds
 Type 3 - Deposit Funds
 Type 4 - Exit
Choice:
2
Checking Account Balance: $100,000.00
Amount you want to withdraw from Checking Account:
50000
New Checking Account balance: $50,000.00
Select the Account you want to Access:
 Type 1 - Checking Account
 Type 2 - Saving Account
 Type 3 - Exit
```

***** THANK YOU *****