

CORE JAVA PROJECT

PROJECT TITLE: ATM INTERFACE

AIM:

ATM program in Java to display ATM transactions.

LANGUAGES:

JAVA

SOFTWARE REQUIREMENT:

Eclipse for java.

SYNOPSIS:

In this ATM INTERFACE there are 5 main operations are there.
The Main Operations are;

- | | |
|---------------------------|---|
| 1.CHECK AVAILABLE BALANCE | → Customer can perform that operation. |
| 2.WITHDRAW AMOUNT | → Customer can perform that operation. |
| 3.DEPOSIT AMOUNT | → Customer can perform that operation. |
| 4.MINI STATEMENT | → Customer can perform that operation. |
| 5.EXIT | → Customer can exit the current Option. |

INITIAL SETUP:

(IN JAVA):

Project Name: ATM INTERFACE

Package Name: myproject.atm

Classes Name: 1.ATM

2.ATMOperations

3.ATMIplements

4.MainClass

ATMINTERFACE:

AtmMain and AtmOptions CODE :

*****ATM INTERFACE CORE JAVA PROJECT*****

CSR CAPGEMINI TRAINING PROJECT
EDUBRIDGE INDIA PRIVATE - LIMITED.
PROJECT TITLE: ATM INTERFACE
UNDER THE GUIDENCE OF TRAINER,

MRS.INDRAKA MALLI
DONE BY SIVARANJANI.S

In ATM INTERFACE:
MAIN OPERATIONS:

1. CHECK AVAILABLE BALANCE.
2. WITHDRAW AMOUNT.
3. DEPOSIT AMOUNT.
4. MINI STATEMENT.
5. EXIT.

```
package myproject.atm;  
public class ATM {  
    private double balance;  
    private double depositAmount;  
    private double withdrawAmount;  
  
    //default constructor  
    public ATM() {  
    }  
    //getter setter
```

```

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public double getDepositAmount() {
        return depositAmount;
    }

    public void setDepositAmount(double depositAmount) {
        this.depositAmount = depositAmount;
    }

    public double getWithdrawAmount() {
        return withdrawAmount;
    }

    public void setWithdrawAmount(double withdrawAmount) {
        this.withdrawAmount = withdrawAmount;
    }

}

```

CREATE A INTERFACE

```
package myproject.atm;
```

```
//create a interface
```

```

public interface ATMOperation{
    public void viewBalance();
    public void withdrawAmount(double withdrawAmount);
    public void depositAmount(double depositAmount);
    public void viewMiniStatement();
}

```

ATMMAINCLASS:

As in the above program, the ATM class is created which contains withdraw(), deposit() and printbalance() functions. The withdraw() function is used to perform the withdraw operation; this function accepts the balance and the withdrawn amount. Inside the withdraw() function, first check whether the balance is greater than the withdraw amount or not; when it is true, then update the balance by subtracting the withdraw amount from the balance. Next, the function deposit() is used to performs the deposit operation; this function accepts the balance and the deposit amount.

Inside the deposit() function, it updates the balance by adding the deposit amount to the balance. Next, the viewBalance() function is used to print the balance; it accepts the balance. Then, in the main function, a balance variable of an integer is created. Next, printing the selecting opitons for withdrawing, deposit, balance, and exit operations, depending on the specific option selection the case gets to execute, as we can see in the output.

- **Withdraw:** For withdrawing the funds, gets the withdrawal amount from the user, deduct it from the total balance, and display the message.
- **Deposit:** For depositing the funds, gets the deposit amount from the user to add, add it to the total balance, and display the message.
- **Check the balance:** For checking the balance, display the user's total balance.
- **Exit:** Return the user to the home page or initial screen by exiting the current Transaction mode.

```

package myproject.atm;
// Create a template
import java.util.Scanner;
public class MainClass
{
    public static void main(String[] args)
    {
        ATMOperationImpl op=new ATMOperationImpl();
        int atmnumber=637892;
        int atmpin=8668;
        //Get the input from user
        Scanner sc=new Scanner(System.in);
        System.out.println("***** WELCOME TO MY ATM MACHINE*****");
        System.out.println("Enter ATM Number : ");
        int Number=sc.nextInt();
        System.out.println("Enter ATM pin : ");
        int Pin=sc.nextInt();
        //validate the user input same as the configure
        if((atmnumber==Number)&&(atmpin==Pin))
        {
            while(true)
            {
                System.out.println("-----MENU-----");
                System.out.println("1. View Available Balance\n2.Withdraw
Amount\n3.Deposit Amount\n4.View MiniStatement\n5.Exit");
                // ask the user enter the choice
                System.out.println("Enter Choice :");
                int ch=sc.nextInt();
                if(ch==1)
                {
                    op.viewBalance();
                    break;
                }
                else if(ch==2)
                {

```

```

        System.out.println("Enter Amount Withdrawn");
        double withdrawAmount=sc.nextDouble();
        op.withdrawAmount(withdrawAmount);
        //break;
    }
    else if(ch==3)
    {
        System.out.println("Enter Amount to Deposit : ");
        double depositAmount=sc.nextDouble();
        op.depositAmount(depositAmount);
    }
    else if(ch==4)
    {
        op.viewMiniStatement();
        //break;
    }
    else if(ch==5)
    {
        System.out.println("Coollect your ATM card\n *****
Thankyou For using MyATM Machine *****");
        System.exit(o);
    }
    else {
        System.out.println("Please enter valid Choice TryAgain");
    }
}

else {
    System.out.println("Please enter valid ATM number");
    System.exit(o);
}
}

```

ATMIMPLEMENTS:

```
package myproject.atm;
import java.util.HashMap;
import java.util.Map;
public class AtmOperationImpl implements AtmOperationInterf{
    ATM atm=new ATM();
    Map<Double,String> ministmt=new HashMap<>();

    @Override
    public void viewBalance() {
        System.out.println("Available Balance is : "+atm.getBalance());
    }

    @Override
    public void withdrawAmount(double withdrawAmount) {
        if(withdrawAmount%500==0) {
            if (withdrawAmount <= atm.getBalance()) {
                ministmt.put(withdrawAmount, " Amount Withdrawn");
                System.out.println("Collect the Cash " + withdrawAmount);
                atm.setBalance(atm.getBalance() - withdrawAmount);
                viewBalance();
            } else {
                System.out.println("Insufficient Balance !!");
            }
        }
        else {
            System.out.println("Please enter the amount in multupal of 500");
        }
    }

    @Override
    public void depositAmount(double depositAmount) {
        ministmt.put(depositAmount," Amount Deposited");
    }
}
```

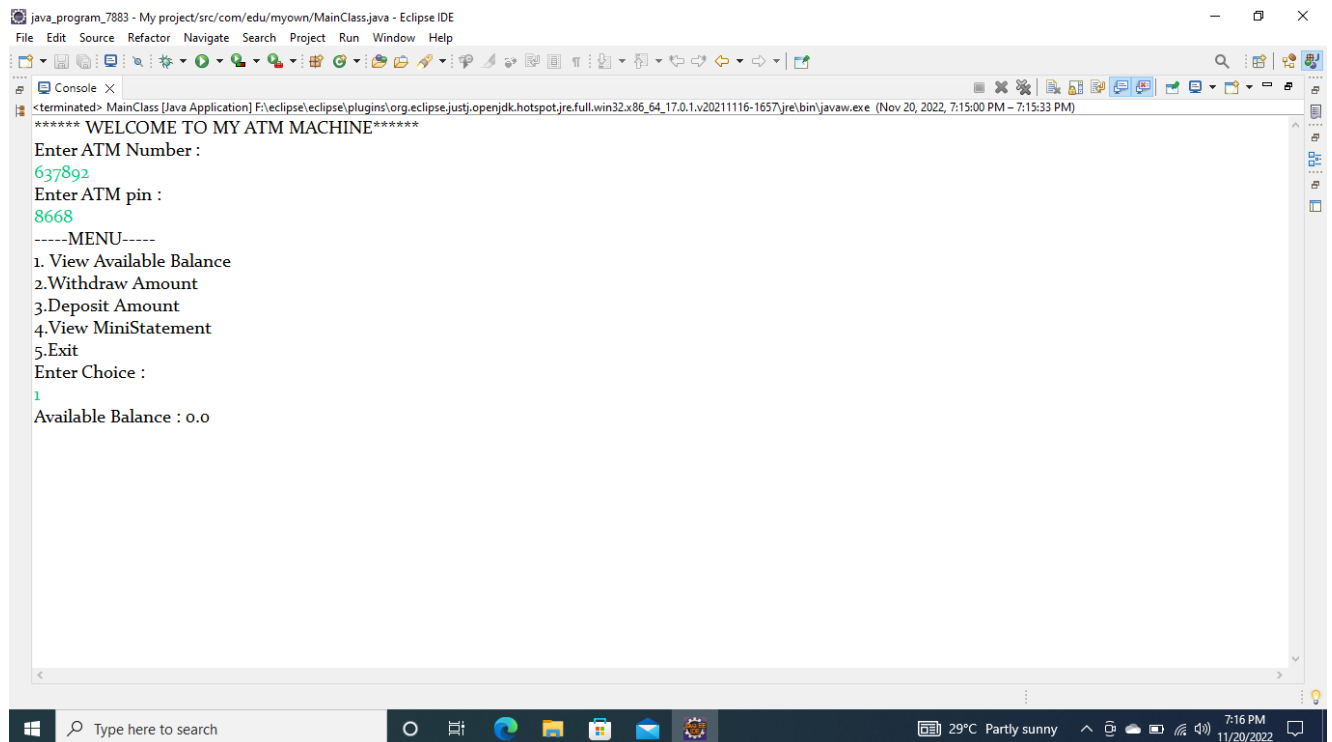
```

        System.out.println(depositAmount+" Deposited Successfully !!");
        atm.setBalance(atm.getBalance()+depositAmount);
        viewBalance();
    }

    @Override
    public void viewMiniStatement() {
        for(Map.Entry<Double,String> m:ministmt.entrySet()){
            System.out.println(m.getKey()+" "+m.getValue());
        }
    }
}

```

CONSOLE OUPUT:



The screenshot shows the Eclipse IDE interface with the console window open. The console output displays the program's execution, including a welcome message, prompts for ATM number and pin, a menu of options, and the selection of 'View Available Balance' which results in an output of 'Available Balance : 0.0'.

```

<terminated> MainClass [Java Application] F:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.x20211116-1657\jre\bin\javaw.exe (Nov 20, 2022, 7:15:00 PM - 7:15:33 PM)
***** WELCOME TO MY ATM MACHINE*****
Enter ATM Number :
637892
Enter ATM pin :
8668
-----MENU-----
1. View Available Balance
2. Withdraw Amount
3. Deposit Amount
4. View MiniStatement
5. Exit
Enter Choice :
1
Available Balance : 0.0

```



```
java_program_7883 - My project/src/com/edu/myown/MainClass.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

MainClass [Java Application] F:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Nov 20, 2022, 7:16:52 PM)
***** WELCOME TO MY ATM MACHINE*****
Enter ATM Number :
637892
Enter ATM pin :
8668
-----MENU-----
1. View Available Balance
2. Withdraw Amount
3. Deposit Amount
4. View MiniStatement
5. Exit
Enter Choice :
3
Enter Amount to Deposit :
10000
10000.0Deposited Successfully!!!!!!
Available Balance : 10000.0
-----MENU-----
1. View Available Balance
2. Withdraw Amount
3. Deposit Amount
4. View MiniStatement
5. Exit
Enter Choice :
```

```
java_program_7883 - My project/src/com/edu/myown/MainClass.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

MainClass [Java Application] F:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Nov 20, 2022, 7:19:12 PM)
***** WELCOME TO MY ATM MACHINE*****
Enter ATM Number :
637892
Enter ATM pin :
8668
-----MENU-----
1. View Available Balance
2. Withdraw Amount
3. Deposit Amount
4. View MiniStatement
5. Exit
Enter Choice :
2
Enter Amount Withdrawn
5000
Insufficient Balance Cannot withdraw!!!!
-----MENU-----
1. View Available Balance
2. Withdraw Amount
3. Deposit Amount
4. View MiniStatement
5. Exit
Enter Choice :
```

```
java_program_7883 - My project/src/com/edu/myown/MainClass.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Console X
<terminated> MainClass [Java Application] F:\eclipse\ eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Nov 20, 2022, 7:22:05 PM - 7:22:56 PM)

50000
50000.0Deposited Successfully!!!!!!
Available Balance : 50000.0
-----MENU-----
1. View Available Balance
2.Withdraw Amount
3.Deposit Amount
4.View MiniStatement
5.Exit
Enter Choice :
2
Enter Amount Withdrawn
10000
Collect the Amount10000.0
Available Balance : 40000.0
-----MENU-----
1. View Available Balance
2.Withdraw Amount
3.Deposit Amount
4.View MiniStatement
5.Exit
Enter Choice :
1
Available Balance : 40000.0
```

```
java_program_7883 - My project/src/com/edu/myown/MainClass.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Console X
MainClass [Java Application] F:\eclipse\ eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Nov 20, 2022, 7:24:14 PM)

5.Exit
Enter Choice :
2
Enter Amount Withdrawn
10000
Collect the Amount10000.0
Available Balance : 40000.0
-----MENU-----
1. View Available Balance
2.Withdraw Amount
3.Deposit Amount
4.View MiniStatement
5.Exit
Enter Choice :
4
10000.0Amount Withdraw
50000.0Amount Deposited
-----MENU-----
1. View Available Balance
2.Withdraw Amount
3.Deposit Amount
4.View MiniStatement
5.Exit
Enter Choice :
```

```
java_program_7883 - My project/src/com/edu/myown/MainClass.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

<terminated> MainClass [Java Application] F:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Nov 20, 2022, 7:24:14 PM - 7:25:49 PM)

10000
Collect the Amount10000.0
Available Balance : 40000.0
-----MENU-----
1. View Available Balance
2.Withdraw Amount
3.Deposit Amount
4.View MiniStatement
5.Exit
Enter Choice :
4
10000.0Amount Withdraw
50000.0Amount Deposited
-----MENU-----
1. View Available Balance
2.Withdraw Amount
3.Deposit Amount
4.View MiniStatement
5.Exit
Enter Choice :
5
Coollect your ATM card
***** Thankyou For using MyATM Machine *****
```

CONCLUSION:

ATM program in Java to display ATM transactions, and the user can withdraw money, deposit money, check the balance, and exit from the ATM.