Muhammad Ali Jinnah University, Karachi

Project Report

Course Code: CE 2421

Course Title: Object Oriented Programming Lab

Project Name: ATM Machine using JAVA(OOP).

Submitted by:
Muzaffar Rafique Burdi (FA20-BECE-0007)

ATM(Automated Teller Machine)

Acknowledgement:

I am very thankful to Sir Muzamil Shaikh give us some findings /errors and he assist us. due to his assistance, we have completed our project because of there was some errors in our project and sir muzamil shaikh guide us regarding our project.

We have found Sir Muzamil Shaikh very helpful and very good guide to complete and fulfil the project.

ABSTRACT:

Automated Teller Machine (ATM), a device used by bank customers to process account transactions. In modern world the bank is playing very important roll in in our life cycle. The bank gives transaction facility to the costumer like ATM. The ATM user insert atm card in the atm machine and type pin code.
In my atm project ATM user type account number and password then options come on the screen like fast cash, withdraw, transfer amount , balance check and exit.
I am try to cover all concepts of OOP in our project like Encapsulation ,Abstraction, Inheritance, Polymorphism.

Code:

```
package lab;
 import
java.util.InputMismatchException; import
java.util.Scanner;
abstract class atm {  // Abstraction
      private String [] account = {"maju41204-1","maju41204-2","maju41204-3"};
private String [] pin = {"1233","1244","1255"};     private long [] money =
```

```
{10000,20000,3000};   // Encapsulation        private String [] transfer =
{"meezan123-1","hbl123-2","mcb123-3"};        private String yes = "yes";
      int x = 0;

      boolean login(String loginaccount,String loginpin){
for(int i = 0;i<account.length;i++){
                  if(loginaccount.equals(account[i])&&loginpin.equals(pin[i])){
                        x = (i);
                        return true;
                  }
            }
            return false;
      }

      public long[] getMoney() {
            return money;
      }

      public void setMoney(long[] money) {
this.money = money;
      }

void transfer(){

      for(int i = 0;i<=transfer.length;i++){
Scanner sc1 = new Scanner(System.in);
            System.out.print("Please Enter correct Account Number of the other party:");
            String taccount = sc1.next();
            System.out.print("Please Enter  the transfer amount:");
            int c = sc1.nextInt();
if(taccount.equals(transfer[i])){
                  if(i== i){
      if(c <= money[x]){
                              System.out.println("Please check the account name:"+taccount);
                              System.out.println("Are you sure to transfer money: yes / no?");
                              String yesNo = sc1.next();
                              if(yesNo.equals(yes)) {
                                    System.out.println(" Amount Transfer succeeded ");
                                    money[x]= money[x]-c;
                                    System.out.println("Your current balance is:"+money[x]);
                        break;
                                    }else {
                                    System.out.println("--- Transfer failed ---");
                                    System.out.println("Your current balance is:"+money[x]);
                                    break;
                              }
                        }else {
                              System.out.println("Transfer failed, your balance is insufficient !!");
                        break;
                        }
                  }else{
                         System.out.println("Account Number entered incorrectly! Cannot transfer money
!!!");
                        break;
                  }
            }
            else if(i == 2){
                  System.out.println("The Account Number you entered is wrong!");
                  System.out.println("Please Try agian later!");
```

```
                    break;

             }
             System.out.println("The Account Number you entered is wrong! or Amount!");

       }

             System.out.println("Invalid Input");

}
abstract void fastcash();
}
class fastcash extends atm {     // Inheritance
       void fastcash(){
             try {
            Scanner sc = new Scanner(System.in);
             System.out.println("Please select choice:\n1. 500 withdrawal amount\n2. 1000 withdrawal
amount\n3. 5000 withdrawal amount\n4. 10000 withdrawal amount\n5. 20000 withdrawal amount");
             System.out.print("Please enter choise:");
             int s = sc.nextInt();
             if (s>=1 &&  s<=5) {
       switch(s){          case
1:
                    int fastcash1=500;
       if(fastcash1 <= getMoney()[x]) {
                           getMoney()[x] = getMoney()[x] - fastcash1;
System.out.println("Withdrawal amount:"+fastcash1);
                           System.out.println("balance:"+getMoney()[x]);
                    }else {
                           System.out.println("Insufficient balance");
                    }

                    break;
       case 2:
                    int fastcash2=1000;
       if(fastcash2 <= getMoney()[x]) {
                           getMoney()[x] = getMoney()[x] - fastcash2;
System.out.println("Withdrawal amount:"+fastcash2);
                           System.out.println("balance:"+getMoney()[x]);
                    }else {
                           System.out.println("Insufficient balance");
                    }
                    break;
       case 3:
                    int fastcash3=5000;
       if(fastcash3 <= getMoney()[x]) {
                           getMoney()[x] = getMoney()[x] -fastcash3;
System.out.println("Withdrawal amount:"+fastcash3);
                           System.out.println("balance:"+getMoney()[x]);
                    }else {
                           System.out.println("Insufficient balance");
                    }
                    break;
       case 4:
                    int fastcash4=10000;
                    if(fastcash4 <= getMoney()[x]) {
                           getMoney()[x] = getMoney()[x] - fastcash4;
                    System.out.println("Withdrawal amount:"+fastcash4);
                           System.out.println("balance:"+getMoney()[x]);
```

```
                }else {
                        System.out.println("Insufficient balance");
                }
                break;
            case 5:
                int fastcash5=20000;
            if(fastcash5 <= getMoney()[x]) {
                        getMoney()[x] = getMoney()[x] - fastcash5;
            System.out.println("Withdrawal amount:"+fastcash5);
                        System.out.println("balance:"+getMoney()[x]);
                }else {
                        System.out.println("Insufficient balance or invalid input");
                }
                break;
            }


            }
               else {
                   System.out.println("---Invalid Input---");
               }
            }
            catch(InputMismatchException exception)
            {
            System.out.println("---Wrong Input---");
            }

}
}

class withdraw extends fastcash{
void withdraw(){
            Scanner sc5 = new Scanner(System.in);
            try {
            System.out.print("Enter the withdrawal amount:");
            int withdraw = sc5.nextInt();
      if(withdraw >=0) {
if(withdraw <= getMoney()[x]) {
                getMoney()[x] = getMoney()[x] - withdraw;
System.out.println("Withdrawal amount:"+withdraw);
                System.out.println("balance:"+getMoney()[x]);
            }else {
                System.out.println("Insufficient balance!!!");
            }
            }
            else {
                System.out.println("---Invalid Input---");
            }
            }
            catch(InputMismatchException exception)
            {
            System.out.println("---Wrong Input---");
            }
      }
}

class deposit extends withdraw{
      void deposit() { try {
            Scanner sc = new Scanner(System.in);
```

```
            System.out.print("Enter the deposit amount:");
            long deposit = sc.nextLong(); if(deposit >=0)
            {
            getMoney()[x] = getMoney()[x] + deposit;
            System.out.println("The deposit amount is:"+deposit+"! Deposit successful!");
            System.out.println("The current balance is:"+getMoney()[x]);
      }
            else {
                  System.out.println("---Invalid Input---");
            }
            }
            catch(InputMismatchException exception)
            {
            System.out.println("---Wrong Input---");
            }
      }

      void cbalance() {
            System.out.println("The current balance is:");      // Polymorphism
      }



}
class cbalance extends deposit{
void cbalance() {
            System.out.println("The current balance is:"+getMoney()[x]);
      }
}

public class Main{
      public static void main(String[]args){
cbalance a1 = new cbalance();
             System.out.println("----------ATM Project by Muzaffar Rafique Burdi & Karan Kumar---------
-");
            System.out.println("********************** Welcome to MAJU ATM**********************");
      for(int i = 1;i <= 3;i++) {
                  Scanner sc = new Scanner(System.in);
                  System.out.print("Enter your Bank Account Number:");
                  String account= sc.next();
                  System.out.println("Enter your Bank account PIN:");
                  String pin= sc.next();
while(a1.login(account, pin)){
                        if(a1.login(account, pin)){
                              try
                              {
                              System.out.println("Please select your choice: \n1. transfer \n2. Fast
cash \n3. Withdrawal \n4. Deposit \n5.Check balance \n6. Exit");
                                    System.out.print("Please enter choise:");
                              int number = sc.nextInt();
                              if(number>=0 && number<=6) {
                        switch(number){
      case 1:
                                    a1.transfer();
                                    break;
                        case 2:
                                    a1.fastcash();
                                    break;
                        case 3:

      a1.withdraw();
```

```
break;                      case 4:
      a1.deposit();
break;                      case 5:

            a1.cbalance();
            break;                      case 6:
                                System.out.print("*** Thanks for using MAJU ATM ***");
                                return;

                                }
                        }
                        else {
                                System.out.println("---Invalid Input---");

                        }


      }
                                catch(InputMismatchException exception)
                                {
                                System.out.println("---Wrong Input---");
                                }

                        }

}


}
      System.out.println("Invalid Input");

}
}
```

Code Explanation:

I have use multi-inheritance.

First class in atm is abstract and in abtract class have account array there have account numbers and pin code array there have account number pin code stored and money array there have money for account numbers. And also transfer array for transfers account numbers. the atm user transfer amount to other account.

```
boolean login(String loginaccount,String loginpin){
    for(int i = 0;i<account.length;i++){
        if(loginaccount.equals(account[i])&&loginpin.equals(pin[i])){
            x = (i);
            return true;
        }
```

I have use Boolean login method to judge whether the user's name and password are correct. And also, transfer method it has user enter other account number and amount to transfer before transfer the amount program should show account number of other that he has transfer amount then the user enters yes, he wants to transfer amount then amount will transfer in case he types amount greater then balance the transition will failed and it shows message (your balance is insufficient). I also use exception. if user input wrong data there program should show exception.

There has another abstract fast cash method. This method defines in another class.

The class fast cash its extends from atm. there have method fast cash in this method is use for fast cash the user withdraw cash in selected option like press 1 for 500 withdraw, press 2 for 1000 withdraw, press 3 for 10000 withdraw and press 4 for 20000 withdraw. If users select option (amount) greater then balance the program should show (`Insufficient balance or invalid input`).). I also use exception. if user input wrong data the program should show exception.

Other class is withdrawn extends from fast cash in this class has withdrawn method. in this method the users has own choice to enter amount for withdraw but less then balance.

Other class is deposit extends from withdraw in this class has deposit method in this method user enter amount for deposit and also other method balance but is method does not use because of we use only for apply polymorphism.

Last class is cbalance ins this cbalance class has cbalance method in this method user will see your remaining Balance

Entire program we use exception because if user enter invalid input the program does not terminated.

Output With Some Explanation:

```
Main (3) [Java Application] C:\Users\Muzaffar Baloch\.p2\pool\plugins\org.eclipse.justj.openjdk.h
----------ATM Project by Muzaffar Rafique Burdi & Karan Kumar----------
************************ Welcome to MAJU ATM************************
Enter your Bank Account Number:
```

User enter his account number

```
Main (3) [Java Application] C:\Users\Muzaffar Baloch\.p2\pool\plugins\org.eclipse.justj.openjdk
----------ATM Project by Muzaffar Rafique Burdi & Karan Kumar----------
************************ Welcome to MAJU ATM************************
Enter your Bank Account Number:maju41204-1
Enter your Bank account PIN:
```

After enter account number also enter pin code his account number

```
Main (3) [Java Application] C:\Users\Muzaffar Baloch\.p2\pool\plugins\org.eclipse.justj.openj
----------ATM Project by Muzaffar Rafique Burdi & Karan Kumar----------
************************ Welcome to MAJU ATM************************
Enter your Bank Account Number:maju41204-1
Enter your Bank account PIN:
1233
Please select your choice:
1. transfer
2. Fast cash
3. Withdrawal
4. Deposit
5.Check balance
6. Exit
Please enter choise:
```

After correct account number and pin then there have some option shows on display .

```
Please enter choise:1
Please Enter correct Account Number of the other party:
```

User chose 1 for transfer money, the program should show enter other party account number

```
Please enter choise:1
Please Enter correct Account Number of the other party:meezan123-1
Please Enter  the transfer amount:2000
```

User enter account number and amount

```
Please Enter correct Account Number of the other party:meezan123-1
Please Enter  the transfer amount:2000
Please check the account name:meezan123-1
Are you sure to transfer money: yes / no?
```

For conformation they should show detail of other account.

```
Please Enter correct Account Number of the other party:meezan123-1
Please Enter  the transfer amount:2000
Please check the account name:meezan123-1
Are you sure to transfer money: yes / no?
yes
 Amount Transfer succeeded
Your current balance is:8000
Please select your choice:
1. transfer
2. Fast cash
3. Withdrawal
4. Deposit
5.Check balance
6. Exit
Please enter choise:
```

user enter yes amount will transfer it shown current balance and option for other choice.

```
Please enter choise:2
Please select choice:
1. 500 withdrawal amount
2. 1000 withdrawal amount
3. 5000 withdrawal amount
4. 10000 withdrawal amount
5. 20000 withdrawal amount
Please enter choise:
```

User chose 2 it shows some fat cash withdraw option.

```
Please select choice:
1. 500 withdrawal amount
2. 1000 withdrawal amount
3. 5000 withdrawal amount
4. 10000 withdrawal amount
5. 20000 withdrawal amount
Please enter choise:3
Withdrawal amount:5000
balance:3000
Please select your choice:
1. transfer
2. Fast cash
3. Withdrawal
4. Deposit
5.Check balance
6. Exit
Please enter choise:
```

User chose 3 for withdraw amount

```
balance:3000
Please select your choice:
1. transfer
2. Fast cash
3. Withdrawal
4. Deposit
5.Check balance
6. Exit
Please enter choise:4
Enter the deposit amount:3000
The deposit amount is:3000! Deposit successful!
The current balance is:6000
```

And 4 for deposit money

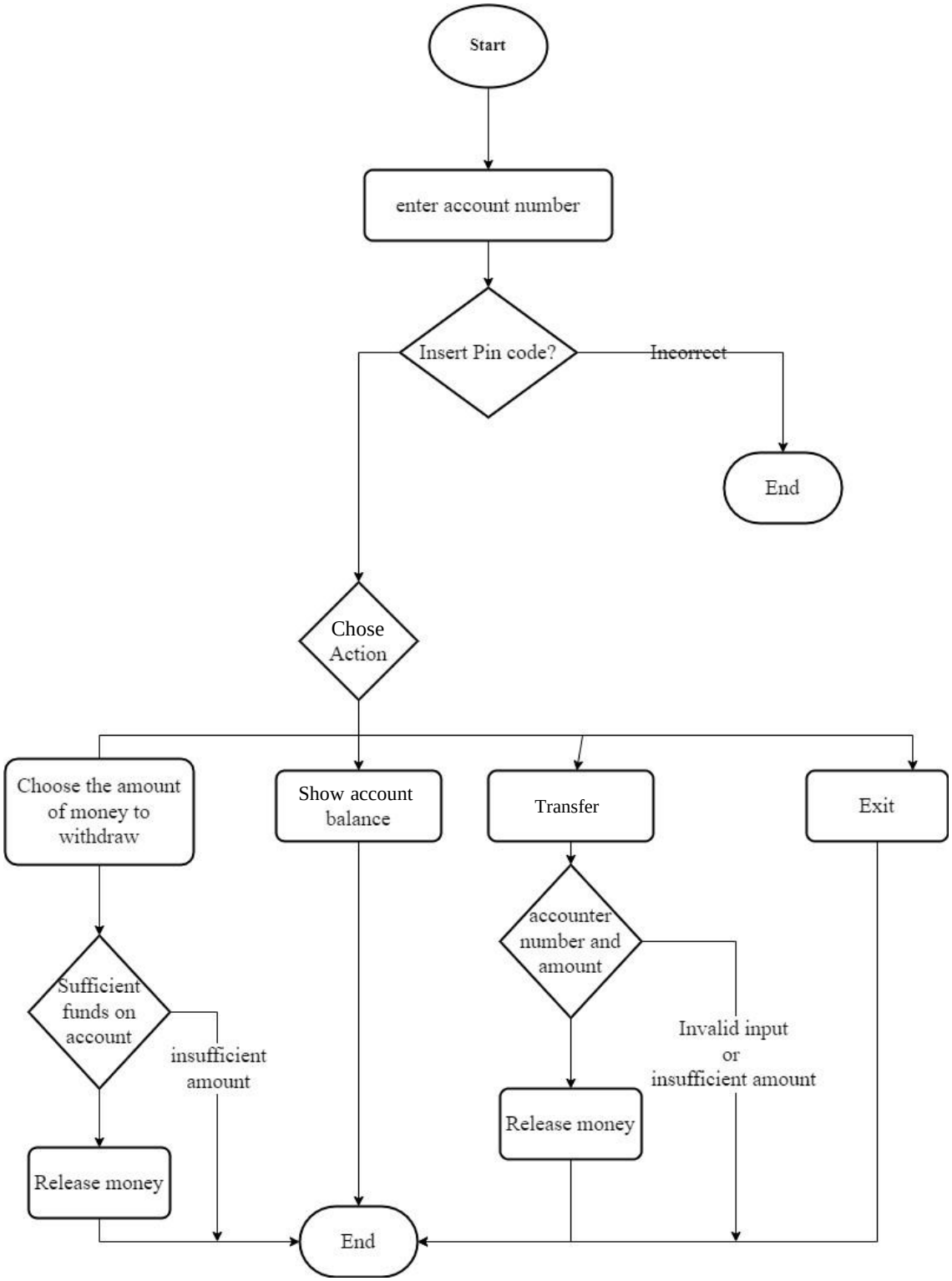
```
Please select your choice:
1. transfer
2. Fast cash
3. Withdrawal
4. Deposit
5.Check balance
6. Exit
Please enter choise:5
The current balance is:6000
```

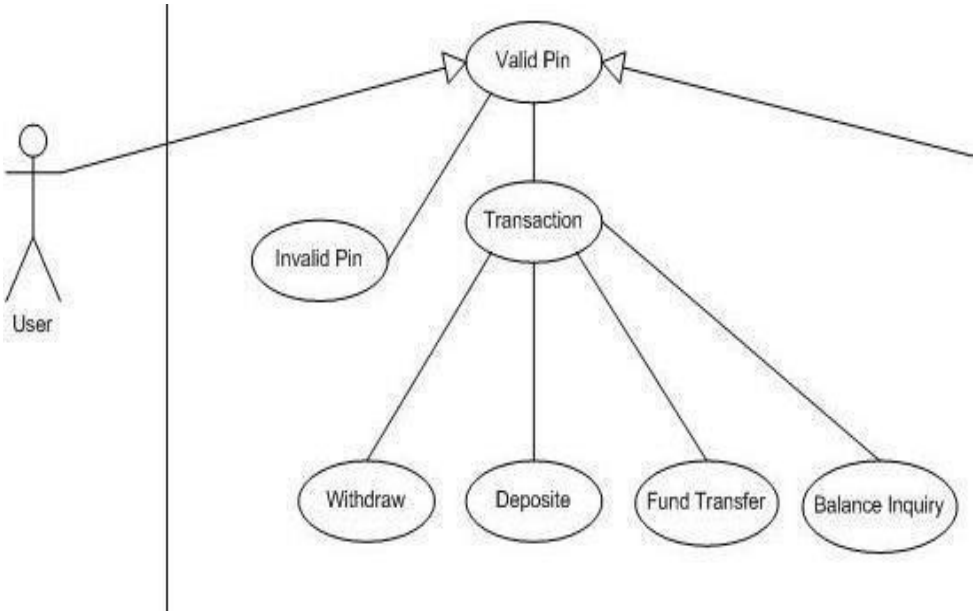
5 for check balance

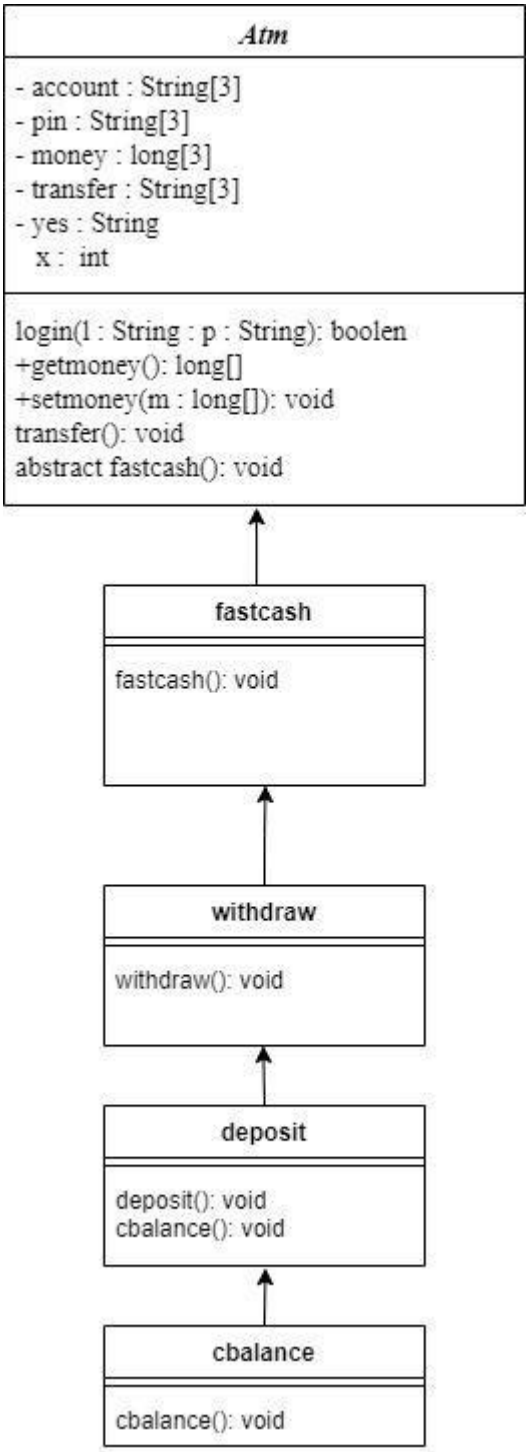
```
Please enter choise:5
The current balance is:6000
Please select your choice:
1. transfer
2. Fast cash
3. Withdrawal
4. Deposit
5.Check balance
6. Exit
Please enter choise:6
*** Thanks for using MAJU ATM ***
```

6 for exit

block diagram/UML:


Start
enter account number
Insert Pin code?
Incorrect
End
Chose Action
Choose the amount of money to withdraw
Show account balance
Transfer
Exit
Sufficient funds on account
insufficient amount
Release money
accounter number and amount
Invalid input or insufficient amount
Release money
End


Valid Pin
User
Invalid Pin
Transaction
Withdraw
Deposite
Fund Transfer
Balance Inquiry


Atm
- account : String[3]
- pin : String[3]
- money : long[3]
- transfer : String[3]
- yes : String
x : int
login(l : String : p : String): boolen
+getmoney(): long[]
+setmoney(m : long[]): void
transfer(): void
abstract fastcash(): void
fastcash
fastcash(): void
withdraw
withdraw(): void
deposit
deposit(): void
cbalance(): void
cbalance
cbalance(): void