

ASSIGNMENT-4

SUBJECT: FPL-2

U.NO.- UEC2021350

NAME: MANASI MILIND SANGAMNERKAR

SECTION-C

BATCH-C/3

DATE OF SUBMISSION: 22ND JUNE,2022

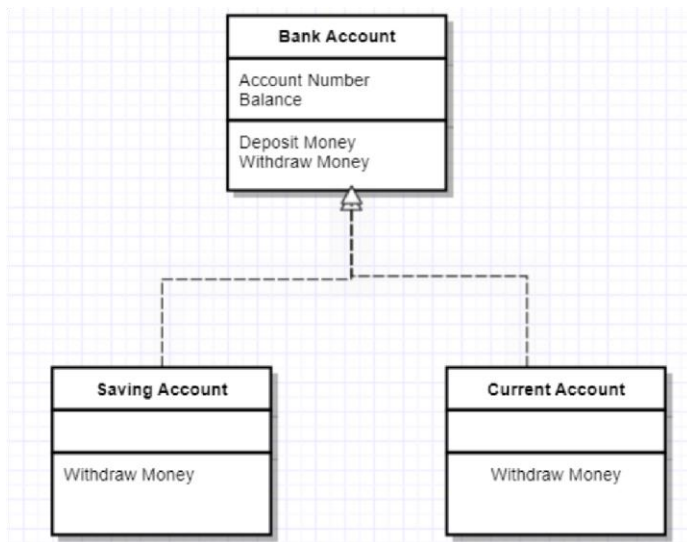
Assignment No. 4

Aim – Find appropriate class hierarchy, polymorphic behavior in applications like banking and implement it.

Input – Account No and balance for two customer

Output –

Theory – Polymorphism is the ability of an entity to behave in different forms. The same concept is used in the programming world where software entities need to show some different behaviors while they are in the same type and sharing all other common features. Take the bank account as an example. Current account and a savings account are two subtypes of bank account and they have almost the same attributes and behaviors, but the withdrawal limit is different based on the account type. Savings account customers cannot withdraw more than their balance but the current account holders can withdraw more than their account balance. Here the attributes of Bank account class can be reused for Current account class and a savings account class using Inheritance as shown below.



Algorithm :

1. In this program create four classes as Bank Account, Saving Account, Current Current and BankingTest.
2. Create a Bank account class, with variable account number and balance and two methods for depositing money and Withdraw money(abstract method).

3. Extend '**Bank Account**' class to **Saving Account class** 'and **Current Account class**.
4. Both of these classes will have their own implementation of the '**Withdraw money**' method.
5. The Saving account will validate if sufficient balance is available before withdrawing while the current account will not have any such validation.
6. Create an instance of saving account class .
7. Create an instance of a current account class.
8. Using deposit and withdrawal methods ,deposit and withdraw money from savings and current account.

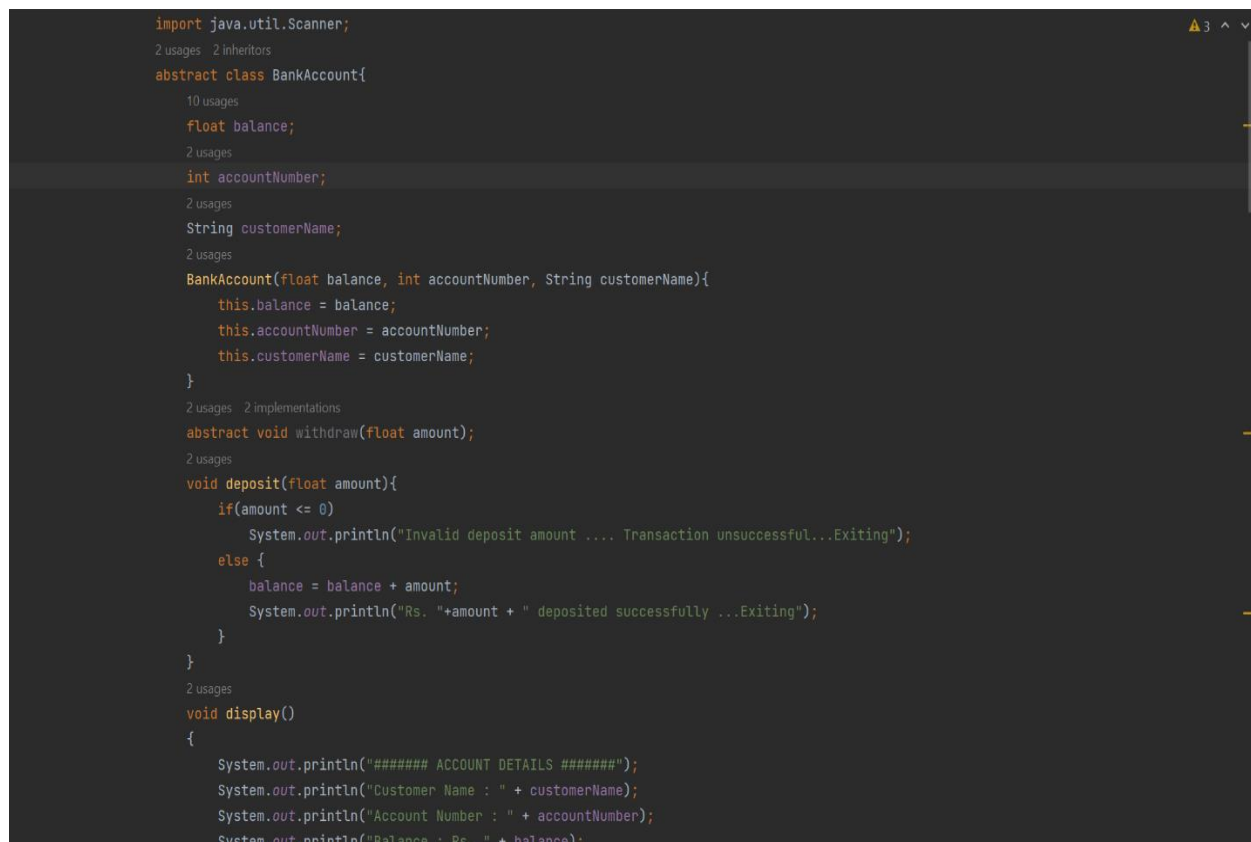
Outcomes –

The learner should be able to

1. Abstract class
2. use if else
3. Method overriding

SCREENSHOT OF INPUT AND OUTPUT: –

1)SCREENSHOT OF THE INPUT:



```
import java.util.Scanner;

2 usages 2 inheritors
abstract class BankAccount{
    10 usages
    float balance;
    2 usages
    int accountNumber;
    2 usages
    String customerName;
    2 usages
    BankAccount(float balance, int accountNumber, String customerName){
        this.balance = balance;
        this.accountNumber = accountNumber;
        this.customerName = customerName;
    }
    2 usages 2 implementations
    abstract void withdraw(float amount);
    2 usages
    void deposit(float amount){
        if(amount <= 0)
            System.out.println("Invalid deposit amount .... Transaction unsuccessful...Exiting");
        else {
            balance = balance + amount;
            System.out.println("Rs. "+amount + " deposited successfully ...Exiting");
        }
    }
    2 usages
    void display()
    {
        System.out.println("##### ACCOUNT DETAILS #####");
        System.out.println("Customer Name : " + customerName);
        System.out.println("Account Number : " + accountNumber);
        System.out.println("Balance : Rs. " + balance);
    }
}
```

```

        System.out.println("Account Number : " + accountNumber);
        System.out.println("Balance : Rs. " + balance);
    }
}
2 usages
class SavingsAccount extends BankAccount{
    1 usage
    SavingsAccount(float balance, int accountNumber, String customerName) {
        super(balance, accountNumber, customerName);
    }
}
2 usages
void withdraw(float amount)
{
    if(amount <= 0)
        System.out.println("Invalid withdraw amount....Transaction Unsuccessful....Exiting");
    else if( amount > 0 && amount < balance){
        balance = balance - amount;
        System.out.println("Rs. "+amount+ " withdrawn successfully.....Exiting");
    }
    else
        System.out.println("Insufficient Funds....Transaction Unsuccessful....Exiting");
}
}
2 usages
class CurrentAccount extends BankAccount{
    1 usage
    CurrentAccount(float balance, int accountNumber, String customerName) {
        super(balance, accountNumber, customerName);
    }
}
2 usages
void withdraw(float amount)
{
    if(amount <= 0)
        System.out.println("Invalid withdraw amount....Transaction Unsuccessful....Exiting");
    else if( amount > 0 && amount <= balance){
        balance = balance - amount;
        System.out.println("Rs. "+amount+ " withdrawn successfully.....Exiting");
    }
    else
        System.out.println("Insufficient Funds....Transaction Unsuccessful....Exiting");
}
}
}
}

```

```

    void withdraw(float amount)
    {
        if(amount <= 0)
            System.out.println("Invalid withdraw amount....Transaction Unsuccessful....Exiting");
        else if( amount > 0 && amount <= balance){
            balance = balance - amount;
            System.out.println("Rs. "+amount+ " withdrawn successfully.....Exiting");
        }
        else
            System.out.println("Insufficient Funds....Transaction Unsuccessful....Exiting");
    }
}

public class Main_1 {
    public static void main(String args[]){
        int ch, accountType;
        float amt;
        SavingsAccount sa = null; // sa ---> NULL location
        CurrentAccount ca = null; // ca ---> NULL location
        Scanner in = new Scanner(System.in);

        System.out.println("Enter account type : savings - 1 / current - 2");
        accountType = in.nextInt();

        System.out.println("Enter the Account Number : ");
        int accNo = in.nextInt();
        in.nextLine();

        System.out.println("Enter the Customer Name : ");
        String cusName = in.nextLine();

        System.out.println("Enter the Balance in Rs : ");
        float bal = in.nextFloat();
    }
}

```

```

if(accountType == 1)
    sa = new SavingsAccount(bal, accNo, cusName); // sa --> [balance, accountName, customerName]
else
    ca = new CurrentAccount(bal, accNo, cusName); // sa --> [balance, accountName, customerName]

do{
    System.out.println("*****MENU*****");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display");
    System.out.println("4. Exit");
    System.out.println("Enter your choice : ");
    ch = in.nextInt();

```

```

switch(ch)
{
    case 1:
        System.out.println("Enter the amount to be deposited : ");
        amt = in.nextFloat();
        if(accountType == 1)
            sa.deposit(amt);
        else
            ca.deposit(amt);
        break;
    case 2:
        System.out.println("Enter the amount to be withdrawn : ");
        amt = in.nextFloat();
        if(accountType == 1)
            sa.withdraw(amt);
        else
            ca.withdraw(amt);
        break;

```

```

        System.out.println("Enter the amount to be withdrawn : ");
        amt = in.nextFloat();
        if(accountType == 1)
            sa.withdraw(amt);
        else
            ca.withdraw(amt);
        break;
    case 3:
        if(accountType == 1)
            sa.display();
        else
            ca.display();
        System.out.println("Account Type : " + accountType);
        break;
    case 4:
        System.out.println("Exiting Application !!!!!!!");
        System.exit( status: 0);
        break;

```

```

    default:
        System.out.println("Invalid choice");

```

```

    }
}while(ch<=4);
}
}

```

SCREENSHOT OF OUTPUT:-

1)SAVINGS ACCOUNT:

```
Enter account type : savings - 1 / current - 2
1
Enter the Account Number :
4112
Enter the Customer Name :
tony stark
Enter the Balance in Rs :
52000
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
1
Enter the amount to be deposited :
500
Rs. 500.0 deposited successfully ...Exiting
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
2
```

```
Enter the amount to be withdrawn :
4500
Rs. 4500.0 withdrawn successfully.....Exiting
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
3
##### ACCOUNT DETAILS #####
Customer Name : tony stark
Account Number : 4112
Balance : Rs. 52000.0
Account Type : 1
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
4
Exiting Application !!!!!!!

Process finished with exit code 0
```

2] CURRENT ACCOUNT:-

```
Enter account type : savings - 1 / current - 2
2
Enter the Account Number :
415
Enter the Customer Name :
tony stark
Enter the Balance in Rs :
4000
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
1
Enter the amount to be deposited :
400
Rs. 400.0 deposited successfully ...Exiting
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
2
```

```
Enter the amount to be withdrawn :
4400
Rs. 4400.0 withdrawn successfully.....Exiting
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
3
##### ACCOUNT DETAILS #####
Customer Name : tony stark
Account Number : 415
Balance : Rs. 0.0
Account Type : 2
*****MENU*****
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice :
4
Exiting Application !!!!!!!

Process finished with exit code 0
```

*******END OF PRACTICAL-4*******