Name: Sahil Hedau

Sec: A (A3)

Roll No.: 56

Date: 24/5/2023

OOPs Practical 4

Aim:

Write a program to implement multiple inheritance.
Consider a class BankAccount with data members as account number, aadhar number, owner name, ROI and balance with member functions openAccount(), deposit(amount),closeAccount() and updateInterest()(this method deposits 4% interest to the account). Create an interface Debitable which has method withdraw().
Derive a class FixedDepositAccount from BankAccount having data member lockInPeriod. Override methods updateInterest() to update Simple Interest, and method closeAccount() to charge 5 % for closure of FD Account before lockInPeriod.
Derive a class SavingAccount from class BankAccount and interface Debitable. The account numbers should be serial numbers of 5 digit and automatically assigned on object creation, such that all FDAccounts start with 55 and all Saving account start with 11.
[ROI for Saving Account is 4% and for FD – 1-2yrs-6% ; 2-5yrs-6.5% ; >5yrs- 7%]

Code:

main.java

```java
public class main{
    public static void main(String[] args) {

        // Saving Account 1
        SavingAccount s1 = new SavingAccount(123456, "Sahil", 500);
        s1.withdraw(100);
```

```
        s1.deposit(2000);
        s1.updateInterest();
        s1.withdraw(500);
        s1.closeAccount();

        // Fixed Deposite
        FixedDepositeAccount f1 = new FixedDepositeAccount(987654, "Samrat",
5, 100000);
        f1.updateInterest();
        f1.closeAccount(3);

        // Saving Account 2
        SavingAccount s2 = new SavingAccount(123456, "Sahil", 5000);
        s2.withdraw(500);
        s2.closeAccount();
    }
}
```

BankAccount.java

```
public class BankAccount {
    int accountNumber;
    int aadharNumber;
    String name;
    double roi = 0;
    double balance;

    BankAccount(int aadharNumber, String name, double amount){
        openAccount(aadharNumber, name, amount);
        accountNumber = 99999;
    }

    private void openAccount(int aadharNumber, String name , double amount){
        this.aadharNumber = aadharNumber;
        this.name = name;
        this.balance = amount;
    }

    void deposit(int amount){
        balance += amount;
        System.out.println("\n"+amount + "Rs. Deposited!\n");
    }

    void updateInterest(){
        double interest = 0.04*balance;
        balance += interest;
    }
```

```
    void closeAccount(){
        System.out.println("\nAccount Number: "+accountNumber);
        System.out.println("Aadhar Number: "+aadharNumber);
        System.out.println("Name: "+name);
        System.out.println("Balance: "+balance);
        System.out.println("Account is Closed!");
    }
}
```

SavingAccount.java

```
public class SavingAccount extends BankAccount implements Debitable{
    static int base = 11001;

    SavingAccount(int aadharNumber, String name, int amount){
        super(aadharNumber, name, amount);
        accountNumber = base++;
    }

    public int withdraw(int amount){
        balance -= amount;
        System.out.println("Account Number: "+accountNumber);
        System.out.println("Name: "+name);
        System.out.println("Current Balance: "+balance);
        System.out.println("Rs. "+amount+" Withdrawed!");
        return amount;
    }
}
```

FixedDepositeAccount.java

```
public class FixedDepositeAccount extends BankAccount{
    double lockInPeriod;
    static int base = 55001;

    FixedDepositeAccount(int aadharNumber, String name, int lockInPeriod, int
amount){
        super(aadharNumber, name, amount);
        this.lockInPeriod = lockInPeriod;
        accountNumber = base++;
        if(lockInPeriod >=1 && lockInPeriod <= 2)
            roi = 6;
        else if (lockInPeriod >=2 && lockInPeriod <= 5)
            roi = 6.5;
        else
            roi = 7;
    }
```

```
    void updateInterest(){
        double interest = roi*balance/100;
        balance += interest;
    }

    void closeAccount(double endTime){
        double charge = 0;
        if(endTime<lockInPeriod){
            charge = 0.05*balance;
        }
        balance -= charge;
        System.out.println("\nAccount Number: "+accountNumber);
        System.out.println("Aadhar Number: "+aadharNumber);
        System.out.println("Name: "+name);
        System.out.println("Balance: "+balance);
        System.out.println("Account is Closed! With "+ charge +" Charges.\n");
    }
}
```

Debitable.java

```
public interface Debitable {
    int withdraw(int amount);
}
```

Terminal Output: