import java.util.ArrayList;

import java.util.List;

interface Account {

void deposit(double amount);

void withdraw(double amount);

void calculateInterest();

double viewBalance();

}

abstract class BankAccount implements Account {

protected double balance;

public BankAccount(double initialBalance) {

this.balance = initialBalance;

}

@Override

public void deposit(double amount) {

balance += amount;

}

@Override

public void withdraw(double amount) {

if (amount <= balance) {

balance -= amount;

} else {

System.out.println("Insufficient funds");

}

}

@Override

public double viewBalance() {

return balance;

}

@Override

public abstract void calculateInterest();

}

class SavingsAccount extends BankAccount {

private static final double INTEREST\_RATE = 0.05; // 5% annual

interest rate

public SavingsAccount(double initialBalance) {

super(initialBalance);

}

@Override

public void calculateInterest() {

balance += balance \* INTEREST\_RATE;

}

}

class CurrentAccount extends BankAccount {

private static final double OVERDRAFT\_LIMIT = 500.0;

public CurrentAccount(double initialBalance) {

super(initialBalance);

}

@Override

public void withdraw(double amount) {

if (amount <= balance + OVERDRAFT\_LIMIT) {

balance -= amount;

} else {

System.out.println("Overdraft limit exceeded");

}

}

@Override

public void calculateInterest() {

// Current accounts do not earn interest

}

}

class Bank {

private List<Account> accounts = new ArrayList<>();

public void addAccount(Account account) {

accounts.add(account);

}

public void displayBalances() {

for (Account account : accounts) {

System.out.println("Balance: " + account.viewBalance());

}

}

}

public class BankingSystem {

public static void main(String[] args) {

Bank bank = new Bank();

SavingsAccount savingsAccount = new SavingsAccount(1000);

CurrentAccount currentAccount = new CurrentAccount(500);

bank.addAccount(savingsAccount);

bank.addAccount(currentAccount);

savingsAccount.deposit(200);

currentAccount.withdraw(100);

savingsAccount.calculateInterest();

currentAccount.calculateInterest();

bank.displayBalances();

}

}

OUTPUT:

