

Code:-

import java.util.Scanner;  
  
class Account {  
    String customerName;  
    long accountNumber;  
    String accountType;  
    double balance;  
  
    public Account(String customerName, long accountNumber, String accountType, double balance) {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.accountType = accountType;  
        this.balance = balance;  
    }  
  
    public void deposit(double amount) {  
        balance += amount;  
        System.out.println("Deposit of $" + amount + " successful. Updated balance: $" + balance);  
    }  
  
    public void displayBalance() {  
        System.out.println("Account Balance: $" + balance);  
    }  
}  
  
class CurAccount extends Account {  
    double minBalance;  
    double serviceCharge;  
  
    public CurAccount(String customerName, long accountNumber, double balance, double minBalance, double serviceCharge) {  
        super(customerName, accountNumber, "Current", balance);  
        this.minBalance = minBalance;  
        this.serviceCharge = serviceCharge;  
    }  
  
    public void checkMinBalance() {  
        if (balance < minBalance) {  
            balance -= serviceCharge;  
            System.out.println("Minimum balance not maintained. Service charge of $" + serviceCharge + " imposed.");  
            displayBalance();  
        }  
    }  
  
    public void withdraw(double amount) {  
        if (amount > balance) {  
            System.out.println("Insufficient funds. Withdrawal failed.");  
        } else {  
            balance -= amount;  
            System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" + balance);  
            checkMinBalance();  
        }  
    }  
}  
  
class SavAccount extends Account {  
    double interestRate;  
  
    public SavAccount(String customerName, long accountNumber, double balance, double interestRate) {  
        super(customerName, accountNumber, "Savings", balance);  
        this.interestRate = interestRate;  
    }  
  
    public void computeInterest() {  
        double interest = balance \* (interestRate / 100);  
        balance += interest;  
        System.out.println("Interest computed and deposited: $" + interest);  
        displayBalance();  
    }  
  
    public void withdraw(double amount) {  
        if (amount > balance) {  
            System.out.println("Insufficient funds. Withdrawal failed.");  
        } else {  
            balance -= amount;  
            System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" + balance);  
        }  
    }  
}  
  
public class Bank {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        CurAccount currentAccount = new CurAccount("John Doe", 123456789, 1000, 500, 10);  
        SavAccount savingsAccount = new SavAccount("Jane Doe", 987654321, 2000, 5);  
  
        int choice;  
        do {  
            System.out.println("\nSelect an option:");  
            System.out.println("1. Deposit");  
            System.out.println("2. Display Balance");  
            System.out.println("3. Compute Interest (Savings Account only)");  
            System.out.println("4. Withdraw");  
            System.out.println("5. Exit");  
            System.out.print("Enter your choice: ");  
            choice = scanner.nextInt();  
  
            switch (choice) {  
                case 1:  
                    System.out.print("Enter amount to deposit: ");  
                    double depositAmount = scanner.nextDouble();  
                    System.out.print("Select account (1. Current, 2. Savings): ");  
                    int accountType = scanner.nextInt();  
                    if (accountType == 1) {  
                        currentAccount.deposit(depositAmount);  
                    } else if (accountType == 2) {  
                        savingsAccount.deposit(depositAmount);  
                    } else {  
                        System.out.println("Invalid account type.");  
                    }  
                    break;  
                case 2:  
                    System.out.print("Select account (1. Current, 2. Savings): ");  
                    int accType = scanner.nextInt();  
                    if (accType == 1) {  
                        currentAccount.displayBalance();  
                    } else if (accType == 2) {  
                        savingsAccount.displayBalance();  
                    } else {  
                        System.out.println("Invalid account type.");  
                    }  
                    break;  
                case 3:  
                    if (savingsAccount instanceof SavAccount) {  
                        ((SavAccount) savingsAccount).computeInterest();  
                    } else {  
                        System.out.println("Invalid option for current account.");  
                    }  
                    break;  
                case 4:  
                    System.out.print("Enter amount to withdraw: ");  
                    double withdrawAmount = scanner.nextDouble();  
                    System.out.print("Select account (1. Current, 2. Savings): ");  
                    int accTyp = scanner.nextInt();  
                    if (accTyp == 1) {  
                        currentAccount.withdraw(withdrawAmount);  
                    } else if (accTyp == 2) {  
                        savingsAccount.withdraw(withdrawAmount);  
                    } else {  
                        System.out.println("Invalid account type.");  
                    }  
                    break;  
                case 5:  
                    System.out.println("Exiting the program. Thank you!");  
                    break;  
                default:  
                    System.out.println("Invalid choice. Please enter a valid option.");  
            }  
        } while (choice != 5);  
  
        scanner.close();  
    }  
}

output:-

