

设计模式作业 01：虚拟钱包的设计与实现

刘恒星 2022229044

1 贫血设计

贫血钱包主要是将钱包主体和业务分离。钱包中只包含了基本信息，不包含业务逻辑，钱包类设计代码如下

```
1 // wallet\walletADM\VirtualWalletBo.java
2 // 省略getter, setter和constructor
3 public class VirtualWalletBo {
4     private Long id;
5     private Long createTime;
6     private BigDecimal balance;
7 }
```

业务层单独写一个文件，并且把业务的逻辑一并实现

```
1 // wallet\walletADM\VirtualWalletService.java
2 // 省略创建钱包, 获取钱包实体, 获取余额的实现
3 public class VirtualWalletService {
4
5     public boolean debit(Long walletId, BigDecimal amount, ArrayList<VirtualWalletBo>
6         wallets){
7         boolean flg = true;
8         try {
9             VirtualWalletBo virtualWalletBo = getVirtualWallet(walletId, wallets);
10            if(virtualWalletBo == null) return false;
11            BigDecimal balance = getBalance(walletId, wallets);
12            if(balance.compareTo(amount) < 0){
13                flg = false;
14                throw new NoSufficientBalanceException();
15            }
16            System.out.println("Successfully!");
17            virtualWalletBo.setBalance(balance.subtract(amount));
18        } catch (NoSufficientBalanceException e) {
19        }
20        return flg;
21    }
22
23    public void credit(Long walletId, BigDecimal amount, ArrayList<VirtualWalletBo> wallets){
```

```

23     VirtualWalletBo virtualWalletBo = getVirtualWallet(walletId, wallets);
24     if(virtualWalletBo == null) return;
25     System.out.println("Successfully!");
26     BigDecimal balance = getBalance(walletId, wallets);
27     virtualWalletBo.setBalance(balance.add(amount));
28 }
29
30 public void transfer(Long fromWalletId, Long toWalletId, BigDecimal amount,
    ArrayList<VirtualWalletBo> wallets){
31     VirtualWalletBo fromWallet = getVirtualWallet(fromWalletId, wallets);
32     if(fromWallet == null){
33         return ;
34     }
35     VirtualWalletBo toWallet = getVirtualWallet(toWalletId, wallets);
36     if(toWallet == null){
37         return ;
38     }
39     boolean flg = debit(fromWalletId, amount, wallets);
40     if(!flg) return ; // fail debit
41     credit(toWalletId, amount, wallets);
42     System.out.println("Successfully!");
43 }
44
45 }

```

2 充血设计

充血模型中，钱包的设计除了基本的信息，也包括基础的逻辑实现，代码如下

```

1 // wallet\walletRDM\VirtualWalletBo.java
2 // 省略getter, setter和constructor
3 public class VirtualWalletBo {
4     private Long id;
5     private Long createTime;
6     private BigDecimal balance;
7
8     public VirtualWalletBo(Long id) {
9         Date date = new Date();
10        Long Time = date.getTime();
11        this.id = id;
12        createTime = Time;
13        balance = BigDecimal.ZERO;
14    }
15
16    public void debit(BigDecimal amount) throws NoSufficientBalanceException {
17        if(this.balance.compareTo(amount) < 0){

```

```

18         throw new NoSufficientBalanceException();
19     }
20     this.balance = this.balance.subtract(amount);
21 }
22
23 public void credit(BigDecimal amount) throws InvalidAmountException {
24     if(amount.compareTo(BigDecimal.ZERO) < 0){
25         throw new InvalidAmountException();
26     }
27     this.balance = this.balance.add(amount);
28 }
29 }

```

而业务层则是简单的调用实体类实现好的接口来完成相应功能，较为复杂的业务逻辑则是在实体类实现好的功能的基础上进一步实现，代码如下

```

1 // walletRDM\VirtualWalletService.java
2 // 省略创建钱包，获取钱包实体，获取余额的实现
3 public class VirtualWalletService {
4
5     public void debit(Long walletId, BigDecimal amount, ArrayList<VirtualWalletBo> wallets){
6         try {
7             VirtualWalletBo virtualWalletBo = getVirtualWallet(walletId, wallets);
8             if(virtualWalletBo == null) return;
9             virtualWalletBo.debit(amount);
10            System.out.println("Successfully!");
11        } catch (NoSufficientBalanceException e) {
12        }
13    }
14
15    public void credit(Long walletId, BigDecimal amount, ArrayList<VirtualWalletBo> wallets){
16        try{
17            VirtualWalletBo virtualWalletBo = getVirtualWallet(walletId, wallets);
18            if(virtualWalletBo == null) return;
19            virtualWalletBo.credit(amount);
20            System.out.println("Successfully!");
21        }catch (InvalidAmountException e) {
22        }
23
24    }
25
26    public void transfer(Long fromWalletId, Long toWalletId, BigDecimal amount,
27        ArrayList<VirtualWalletBo> wallets){
28        VirtualWalletBo fromWallet = getVirtualWallet(fromWalletId, wallets);
29        if(fromWallet == null){
30            return ;
31        }
32    }
33 }

```

```
31     VirtualWalletBo toWallet = getVirtualWallet(toWalletId, wallets);
32     if(toWallet == null){
33         return ;
34     }
35     try{
36         fromWallet.debit(amount);
37         toWallet.credit(amount);
38         System.out.println("Successfully!");
39     } catch (NoSufficientBalanceException | InvalidAmountException e) {
40
41     }
42 }
43
44 }
```
