

Evgen959 /
Advanced_Backend[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Advanced_Backend](#) / [Lesen 020](#) / [code](#) / [hw_na_les019](#) / [src](#) / [Bank.java](#) 

Evgen959 newLesen20

15 minutes ago



92 lines (78 loc) · 2.6 KB

Code

Blame

Raw



```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  public class Bank {
5      private String title;
6      private List<Account> listAccount = new ArrayList<>();
7      private TransactionsList transactionsList = new TransactionsList();
8
9      public Bank(String title, Account[] array) {
10         this.title = title;
11         addNewAccount(array);
12     }
13
14     public Bank(String title) {
15         this.title = title;
16     }
17
18     @Override
19     public String toString() {
20         return "Bank: " + title
21             + "\nBcero cчeтoв: "
22             + listAccount.size()
23             + "\nCчeтa:\n"
24             + accountsToString();
25     }
26
27     private String accountsToString() {
28         String res = "";
29         for (int i = 0; i < listAccount.size(); i++) {
30             res += listAccount.get(i) + "\n";
31         }
32         return res;
33     }
34
35     public String getTitle() {
36         return title;
37     }
38
39     public Account getAccountByOwnerLastName (String lastName) {
40         for (int i = 0; i < listAccount.size(); i++) {
41             Account account = listAccount.get(i);
42             if (account.getOwner().getLastName().equals(lastName)) {
43                 return account;
44             }
45         }
46     }
```

```
45     }
46     return null;
47 }
48
49 ✓ public void depositAccountByOwnerLastName(String lastName, double amount) {
50     Account account = getAccountByOwnerLastName(lastName);
51     if (account != null) {
52         account.deposit(amount);
53     }
54 }
55
56 ✓ public void withdrawAccountByOwnerLastName(String lastName, double amount) {
57     Account account = getAccountByOwnerLastName(lastName);
58     if (account != null) {
59         account.withdraw(amount);
60     }
61 }
62
63 ✓ public boolean transfer(String lastNameFrom, String lastNameTo, double amount){
64     Account accountFrom = getAccountByOwnerLastName(lastNameFrom);
65     Account accountTo = getAccountByOwnerLastName(lastNameTo);
66     if (accountFrom == null || accountTo == null) {
67         return false;
68     }
69     if (accountFrom.withdraw(amount)) {
70         accountTo.deposit(amount);
71         transactionsList.addTransaction(new Transaction(accountFrom, accountTo, amount));
72         return true;
73     }
74     return false;
75 }
76
77 public TransactionsList getTransactionsList() {
78     return transactionsList;
79 }
80
81 // addNewAccoun - перегруженный метод (overloading)
82 ✓ public void addNewAccount(Account[] array) {
83     for (int i = 0; i < array.length ; i++) {
84         addNewAccount(array[i]);
85     }
86 }
87
88 public void addNewAccount(Account account){
89     listAccount.add(account);
90 }
91 }
```