Evgen959 /
Advanced_Backend

Code Issues Pull requests Actions Projects Wiki Security Insights

Advanced_Backend / Homework / Les019_HW / src / **Bank.java**

Evgen959 newLesen20 9 minutes ago

111 lines (96 loc) · 3.36 KB

Code Blame Raw

```java
import java.util.ArrayList;
import java.util.List;

public class Bank {
    // private int capacity = 6;
    private String title;
    private List<Account> listAccount = new ArrayList<>();
    // private Account[] accounts = new Account[capacity];
    // private int size;
    private TransactionsArray transactionsArray = new TransactionsArray();

    public Bank(String title, Account[] array) {
        this.title = title;
        // size = 0;
        addNewAccount(array);
    }

    public Bank(String title) {
        this.title = title;
        // size = 0;
    }

    @Override
    public String toString() {
        return "Bank:  " + title
                +  "\nВсего счетов: "
                //+ size
                + listAccount.size()
                + "\nСчета:\n"
                + accuntsToString();
    }

    private String accuntsToString() {
        String res = "";
        for (int i = 0; i < listAccount.size() ; i++) {
            res +=listAccount.get(i) + "\n";
        }
        return res;
    }

    public String getTitle() {
        return title;
    }

    public Account getAccuntByOwnerLastName (String lastName) {
        for (int i = 0; i < listAccount.size(); i++) {
            if (listAccount.get(i).getOwner().getLastName().equals(lastName)) {
                return listAccount.get(i);
            }
        }
        return null;
    }

    public void depositAccountByOwnerLastName(String lastName, double amount) {
        Account account = getAccuntByOwnerLastName(lastName);
        if (account != null) {
            account.deposit(amount);
        }
    }

    public void withdrawAccountByOwnerLastName(String lastName, double amount) {
        Account account = getAccuntByOwnerLastName(lastName);
        if (account != null) {
            account.withdraw(amount);
        }
    }

    public  boolean transfer(String lastNameFrom, String lastNameTo, double amount){
        Account accountFrom = getAccuntByOwnerLastName(lastNameFrom);
        Account accountTo = getAccuntByOwnerLastName(lastNameTo);
        if (accountFrom == null || accountTo == null) {
            return false;
        }
        if (accountFrom.withdraw(amount)) {
            accountTo.deposit(amount);
            transactionsArray.addTransaction(new Transaction(accountFrom, accountTo, amount));
            return true;
        }
        return false;
    }

    public TransactionsArray getTransactionsArray() {
           return transactionsArray;
    }

    // addNewAccoun - перегруженный метод (overloading)
    public void addNewAccount(Account[] array) {
        for (int i = 0; i < array.length ; i++) {
            addNewAccount(array[i]);
        }
    }

    public void addNewAccount(Account account){
        /* if (size >= accounts.length) { // проверяем сколько пустых ячеек в массиве
            enlarge();
        }
        accounts[size++] = account;
        */
        listAccount.add(account);
    }

    /* private void enlarge() {
        Account[] temp = new Account[accounts.length*2]; // создаем новый масив
        for (int i = 0; i < accounts.length; i++) {   // переписываем все элементы массва в новый
```

```
            temp[i] = accounts[i];//  переадресовываемся к новаму массиву
        }
        accounts = temp;
    }
    */
}
```