Test Question 5(24.7.24)

SET 1

1. Develop a simple banking system that allows users to create accounts, deposit money, withdraw money, and check balance. Implement methods for account creation, deposit, withdrawal, and balance inquiry.

Methods:

- createAccount(String accountHolderName, double initialDeposit)
- depositMoney(String accountNumber, double amount)
- withdrawMoney(String accountNumber, double amount)
- checkBalance(String accountNumber)

code:

```
import java.util.HashMap;

import java.util.Map;



class Account {

   private String accountNumber;

   private String accountHolderName;

   private double balance;



   public Account(String accountNumber, String accountHolderName, double initialDeposit) {

      this.accountNumber = accountNumber;
```

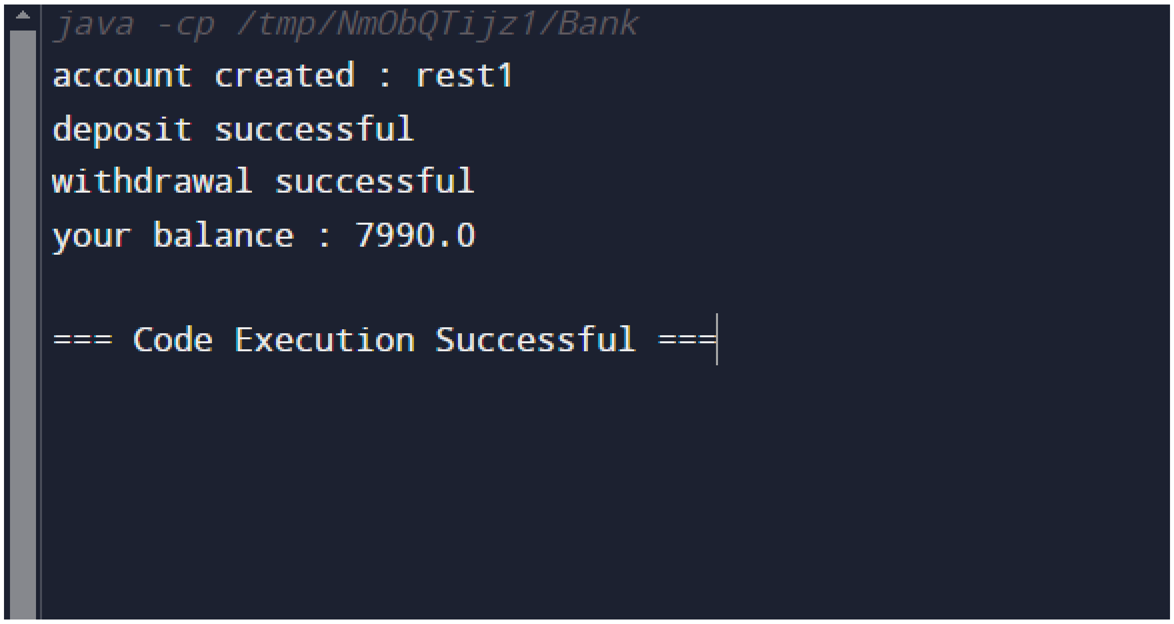
```
    this.accountHolderName = accountHolderName;

    this.balance = initialDeposit;

}



public String getAccountNumber() {

    return accountNumber;

}



public String getAccountHolderName() {

    return accountHolderName;

}



public double getBalance() {

    return balance;

}



public void deposit(double amount) {

    balance += amount;

}
```

```
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
        } else {
            System.out.println("insufficient balance");
        }
    }
}

public class Bank {
    private Map<String, Account> accounts;
    private int accountCount;

    public Bank() {
        accounts = new HashMap<>();
        accountCount = 0;
    }
    public String createAccount(String accountHolderName, double initialDeposit) {

        accountCount++;

        String accountNumber = "rest" + accountCount;

        Account account = new Account(accountNumber, accountHolderName, 
initialDeposit);

        accounts.put(accountNumber, account);

        return accountNumber;

    }



    public void depositMoney(String accountNumber, double amount) {

        if (accounts.containsKey(accountNumber)) {

            accounts.get(accountNumber).deposit(amount);

            System.out.println("deposit successful");

        } else {

            System.out.println("account not found");

        }

    }



    public void withdrawMoney(String accountNumber, double amount) {
```

```
    if (accounts.containsKey(accountNumber)) {

        accounts.get(accountNumber).withdraw(amount);

        System.out.println("withdrawal successful");

    } else {

        System.out.println("account not found");

    }

}



public void checkBalance(String accountNumber) {

    if (accounts.containsKey(accountNumber)) {

        double balance = accounts.get(accountNumber).getBalance();

        System.out.println("your balance : " + balance);

    } else {

        System.out.println("account not found");

    }

}



public static void main(String[] args) {

    Bank bank = new Bank();
```

```
        String accountNumber = bank.createAccount("John Doe", 1000);

        System.out.println("account created : " + accountNumber);

        bank.depositMoney(accountNumber, 7890);

        bank.withdrawMoney(accountNumber, 900);

        bank.checkBalance(accountNumber);

    }

}
```

Output:



```
java -cp /tmp/NmObQTijz1/Bank
account created : rest1
deposit successful
withdrawal successful
your balance : 7990.0

=== Code Execution Successful ===
```

2. Create an expense tracker that allows users to add expenses, categorize them, and view a summary report. Implement methods to add expenses, categorize expenses, and generate reports.

Methods:

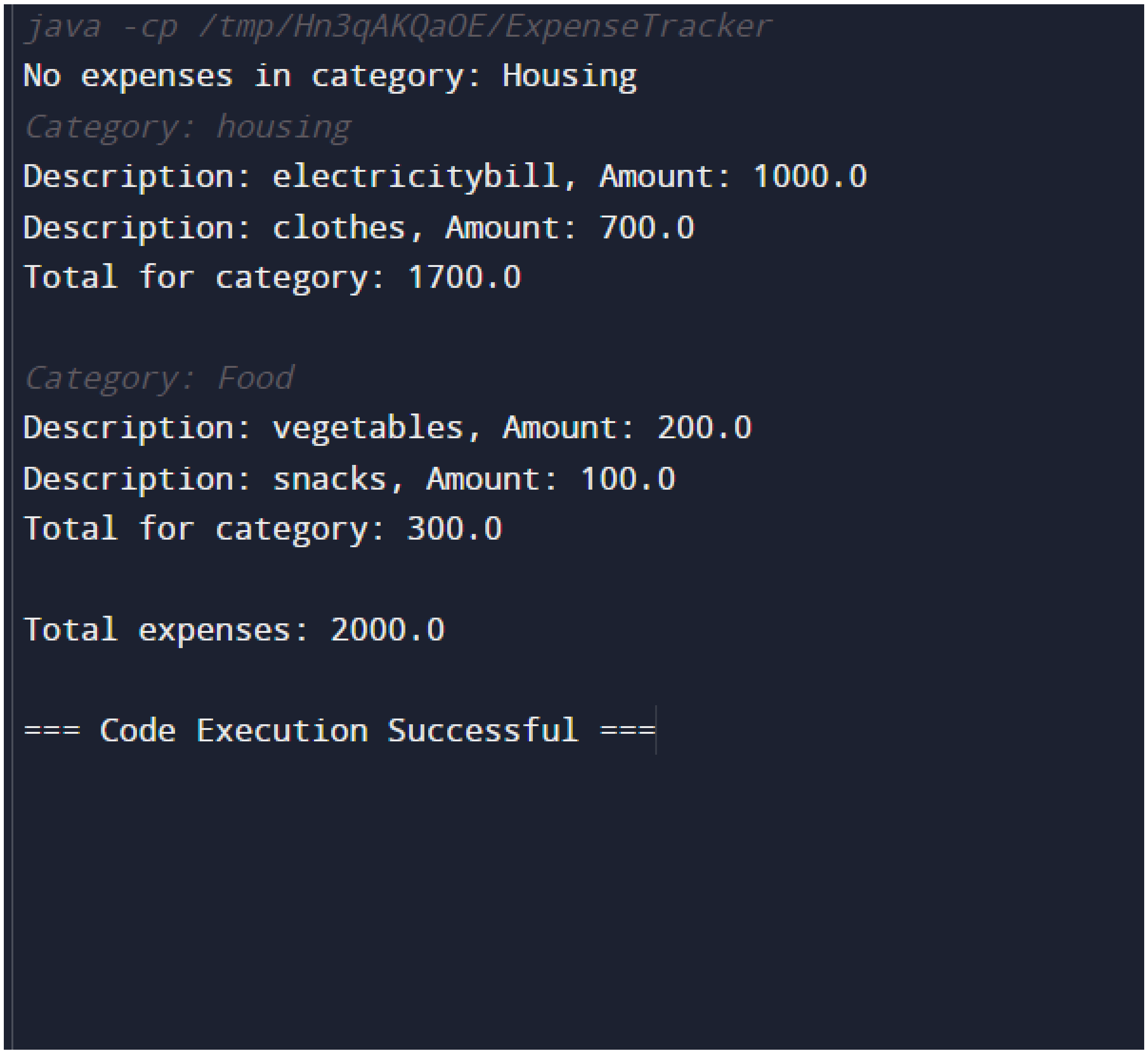
- addExpense(String description, double amount, String category)

- viewExpensesByCategory(String category)
- generateExpenseReport()

code:

```
import java.util.*;



class Expense {

    private String description;

    private double amount;

    private String category;



    public Expense(String description, double amount, String category) {

        this.description = description;

        this.amount = amount;

        this.category = category;

    }



    public String getDescription() {

        return description;

    }
    public double getAmount() {

        return amount;

    }



    public String getCategory() {

        return category;

    }

}



public class ExpenseTracker {

    private Map<String, List<Expense>> expenses;



    public ExpenseTracker() {

        expenses = new HashMap<>();

    }



    public void addExpense(String description, double amount, String category) {

        Expense expense = new Expense(description, amount, category);
        if (expenses.containsKey(category)) {
            expenses.get(category).add(expense);
        } else {
            List<Expense> list = new ArrayList<>();
            list.add(expense);
            expenses.put(category, list);
        }
    }

    public void viewExpensesByCategory(String category) {
        if (expenses.containsKey(category)) {
            List<Expense> list = expenses.get(category);
            System.out.println("Expenses in category: " + category);
            for (Expense expense : list) {
                System.out.println("Description: " + expense.getDescription() + ", Amount: " +
expense.getAmount());
            }
        } else {
            System.out.println("No expenses in category: " + category);
        }
    }



    public void generateExpenseReport() {
        double total = 0;
        for (Map.Entry<String, List<Expense>> entry : expenses.entrySet()) {
            double categoryTotal = 0;
            System.out.println("Category: " + entry.getKey());
            for (Expense expense : entry.getValue()) {
                System.out.println("Description: " + expense.getDescription() + ", Amount: " +
expense.getAmount());
                categoryTotal += expense.getAmount();
            }
            System.out.println("Total for category: " + categoryTotal);
            System.out.println();
            total += categoryTotal;
        }
        System.out.println("Total expenses: " + total);
    }
    public static void main(String[] args) {

        ExpenseTracker tracker = new ExpenseTracker();

        tracker.addExpense("electricitybill", 1000, "housing");

        tracker.addExpense("clothes", 700, "housing");

        tracker.addExpense("vegetables", 200, "Food");

        tracker.addExpense("snacks", 100, "Food");

        tracker.viewExpensesByCategory("Housing");

        tracker.generateExpenseReport();

    }

}
```

Output:



```
java -cp /tmp/Hn3qAKQaOE/ExpenseTracker
No expenses in category: Housing
Category: housing
Description: electricitybill, Amount: 1000.0
Description: clothes, Amount: 700.0
Total for category: 1700.0

Category: Food
Description: vegetables, Amount: 200.0
Description: snacks, Amount: 100.0
Total for category: 300.0

Total expenses: 2000.0

=== Code Execution Successful ===
```