Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

О Т Ч Е Т

по лабораторной работе № 8

Вариант 15

Название: Потоки в Java

Дисциплина: Языки программирования для работы с большими данными

Студент ИУ6-23М _____ А.Р. Степаненко
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2022

Цель работы:

Получение навыков работы с потоками в Java.

Выполнение:

Задание:

1. Реализовать многопоточное приложение “Банк”. Имеется банковский счет. Сделать синхронным пополнение и снятие денежных средств на счет/со счет случайной суммой. При каждой операции (пополнения или снятие) вывести текущий баланс счета. В том случае, если денежных средств недостаточно – вывести сообщение.
2. Реализовать многопоточное приложение “Магазин”. Вся цепочка: производитель-магазин-покупатель. Пока производитель не поставит на склад продукт, покупатель не может его забрать. Реализовать приход товара от производителя в магазин случайным числом. В том случае, если товара в магазине не хватает– вывести сообщение.

Листинг выполнения подзадачи 1 (файл lr811.java)

```
package lr81;


import java.util.concurrent.ThreadLocalRandom;

import static lr81.lr811.flag;
import static lr81.lr811.account;

public class lr811 {
    public volatile static int flag;
    public volatile static int account;

    public static void main(String[] args) throws InterruptedException {
        account = 0;
        flag = 0;
        Adder add = new Adder();
        Subber sub = new Subber();
        add.start();
        sub.start();
        while (true){
            Thread.sleep(5000);
            flag = ThreadLocalRandom.current().nextInt(0, 2 + 1);
        }
    }
}
class Adder extends Thread {
    public Adder() {
    }
    @Override
```

```
    public void run() {
        while (true) {
            if (flag == 1) {
                int n = ThreadLocalRandom.current().nextInt(1, 100 + 1);
                account += n;
                System.out.println("New income " + n);
                System.out.println("Current account = " + account);
                flag = 0;
            }
        }
    }
}

class Subber extends Thread {
    public Subber() {
    }
    @Override
    public void run() {
        while (true) {
            if (flag == 2) {
                int n = ThreadLocalRandom.current().nextInt(1, 100 + 1);
                System.out.println("Trying to withdraw " + n);
                if (account < n){
                    System.out.println("Not enough money :c");
                } else {
                    account -= n;
                    System.out.println("Success");
                }
                System.out.println("Current account = " + account);
                flag = 0;
            }
        }
    }
}
```

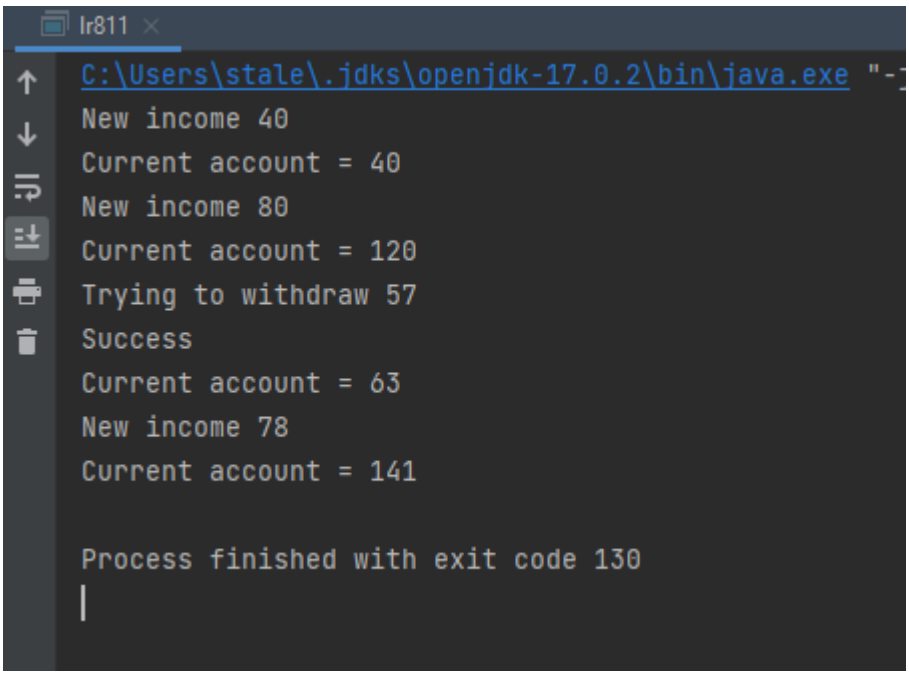

Рисунок 1 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл lr813.java)

```
package lr81;
```

```java
import java.util.HashMap;
import java.util.concurrent.ThreadLocalRandom;

import static lr81.lr813.flag;
import static lr81.lr813.storage;

public class lr813 {

    public volatile static boolean flag;
    public volatile static HashMap<Integer, Integer> storage;

    public static void main(String[] args) {
        storage = new HashMap<>();
        Supplier sup = new Supplier();
        Customer cus = new Customer();
        sup.start();
        cus.start();
    }
}


class Supplier extends Thread {
    public Supplier() {
    }
    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep(5000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            int goods = ThreadLocalRandom.current().nextInt(1, 10 + 1);
            int amount = ThreadLocalRandom.current().nextInt(1, 100 + 1);
            System.out.println("New supply " + goods + ". Amount = " + amount);
            if (storage.containsKey(goods)) {
                storage.put(goods, storage.get(goods) + amount);
            } else {
                storage.put(goods, amount);
            }
            flag = true;
        }
    }
}

class Customer extends Thread {
    public Customer() {
    }
    @Override
    public void run() {
        int good_i_want = ThreadLocalRandom.current().nextInt(1, 10 + 1);
        int amount_i_want = ThreadLocalRandom.current().nextInt(1, 100 + 1);
        System.out.println("order = " + amount_i_want + " pieces of " + good_i_want);
        while (true) {
            while (!flag) {
                // беск цикл
            }
            if (storage.containsKey(good_i_want) && storage.get(good_i_want) >=
amount_i_want) {
                storage.put(good_i_want, storage.get(good_i_want) - amount_i_want);
                System.out.println("Just took " + amount_i_want + " of " +
good_i_want);
                good_i_want = ThreadLocalRandom.current().nextInt(1, 10 + 1);
                amount_i_want = ThreadLocalRandom.current().nextInt(1, 100 + 1);
                System.out.println("order = " + amount_i_want + " pieces of " +
good_i_want);
```

```
            } else {
                System.out.println("No goods I want :c");
            }
            flag = false;
        }
    }
}
```

```
C:\Users\stale\.jdks\openjdk-17.0.2\bin\java.exe "-javaag
order = 55 pieces of 6
New supply 6. Amount = 80
Just took 55 of 6
order = 54 pieces of 10
New supply 7. Amount = 79
No goods I want :c
New supply 1. Amount = 55
No goods I want :c
New supply 3. Amount = 23
No goods I want :c
```

Рисунок 2 - Результат выполнения кода решения подзадачи 2

Ссылка на программное решение:

Программное решение представлено в репозитории распределённой системы управления версиями Git:

https://github.com/stalekc/java_magister/tree/main/lr8

Вывод:

При выполнении лабораторной работы были получены навыки работы с потоками в Java.