



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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ 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

Название: Потоки

Дисциплина: Языки программирования для работы с большими
данными

Вариант: 2

Студент

ИУ6-22М

(Группа)

(Подпись, дата)

Н.А. Аскерова

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Вариант 1

2. Реализовать многопоточное приложение “Робот”. Надо написать робота, который умеет ходить. За движение каждой его ноги отвечает отдельный поток. Шаг выражается в выводе в консоль LEFT или RIGHT.

Листинг 1 – Код программы

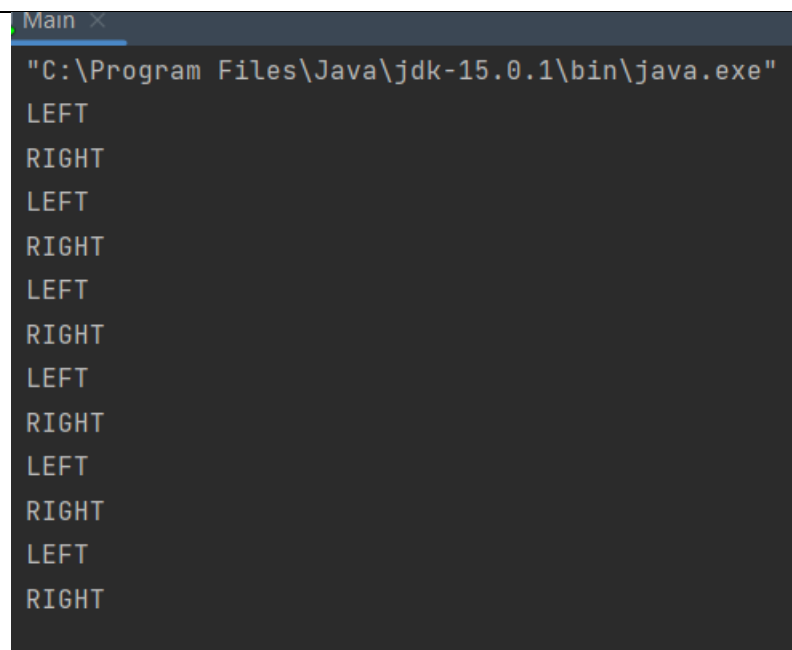
```
public class Main {
    public static void main(String[] args) {
        Thread leftLeg = new Thread(new Leg("LEFT"));
        Thread rightLeg = new Thread(new Leg("RIGHT"));

        leftLeg.start();
        rightLeg.start();
    }

    private static class Leg implements Runnable {
        private final String name;

        public Leg(String name) {
            this.name = name;
        }

        @Override
        public void run() {
            while (true) {
                System.out.println(name);
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
Main ×
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe"
LEFT
RIGHT
LEFT
RIGHT
LEFT
RIGHT
LEFT
RIGHT
LEFT
RIGHT
LEFT
RIGHT
```

Рисунок 1 – Результат работы программы

3. 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 2 – Код программы

```
import java.util.Random;

public class Store {
    private int inventory = 0;

    public synchronized void addInventory(int quantity) {
        inventory += quantity;
        System.out.println("Manufacturer delivered " + quantity + " products to the store. Total inventory: " +
inventory);
        notifyAll();
    }

    public synchronized void sellInventory(int quantity) {
        while (inventory < quantity) {
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted while waiting for inventory");
            }
        }
        inventory -= quantity;
        System.out.println("Buyer purchased " + quantity + " products from the store. Total inventory: " + inventory);
    }

    public static void main(String[] args) {
        Store store = new Store();

        Thread manufacturerThread = new Thread() -> {
            Random random = new Random();
            while (true) {
                try {
                    Thread.sleep(random.nextInt(5000) + 1000); // Simulate buyer arrival time
                } catch (InterruptedException e) {
                    throw new RuntimeException(e);
                }
                int quantity = random.nextInt(10) + 1; // Random number of products delivered
                store.addInventory(quantity);
            }
        };

        Thread buyerThread = new Thread() -> {
            Random random = new Random();
            while (true) {
                try {
                    Thread.sleep(random.nextInt(5000) + 1000); // Simulate buyer arrival time
                } catch (InterruptedException e) {
                    throw new RuntimeException(e);
                }
                int quantity = random.nextInt(10) + 1; // Random number of products to buy
                store.sellInventory(quantity);
            }
        };
    }
}
```

```
    }  
    });  
  
    manufacturerThread.start();  
    buyerThread.start();  
}  
}
```

```
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe" "-javaagent:C:\Program Files  
Manufacturer delivered 3 products to the store. Total inventory: 3  
Manufacturer delivered 5 products to the store. Total inventory: 8  
Manufacturer delivered 5 products to the store. Total inventory: 13  
Buyer purchased 9 products from the store. Total inventory: 4  
Buyer purchased 3 products from the store. Total inventory: 1  
Manufacturer delivered 8 products to the store. Total inventory: 9  
Buyer purchased 6 products from the store. Total inventory: 3  
Manufacturer delivered 3 products to the store. Total inventory: 6  
Manufacturer delivered 2 products to the store. Total inventory: 8  
Buyer purchased 8 products from the store. Total inventory: 0  
Manufacturer delivered 7 products to the store. Total inventory: 7  
Buyer purchased 6 products from the store. Total inventory: 1  
Manufacturer delivered 10 products to the store. Total inventory: 11  
Buyer purchased 4 products from the store. Total inventory: 7  
Buyer purchased 7 products from the store. Total inventory: 0  
Manufacturer delivered 3 products to the store. Total inventory: 3  
Manufacturer delivered 7 products to the store. Total inventory: 10  
Buyer purchased 9 products from the store. Total inventory: 1  
Manufacturer delivered 1 products to the store. Total inventory: 2  
Manufacturer delivered 6 products to the store. Total inventory: 8
```

Рисунок 2 – Результат работы программы

Вывод: приобретен навык работы с потоками.

Ссылка на репозиторий с программами: <https://github.com/nargi3/BigData>