|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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**

**Отчет**

**по лабораторной работе №8**

**Название:** Потоки

**Дисциплина:** Языки программирования для работы с большими данными

Вариант: 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | Н.А. Аскерова |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (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();

}

}

}

}

}

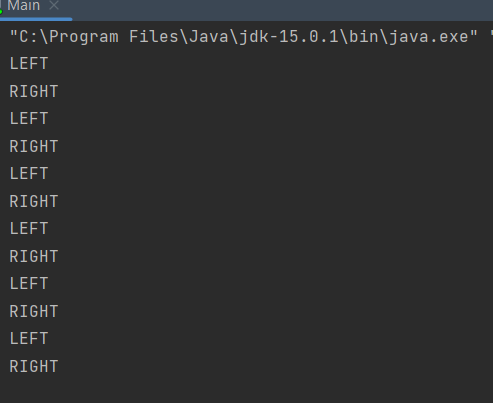


Рисунок 1 – Результат работы программы

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();

}

}

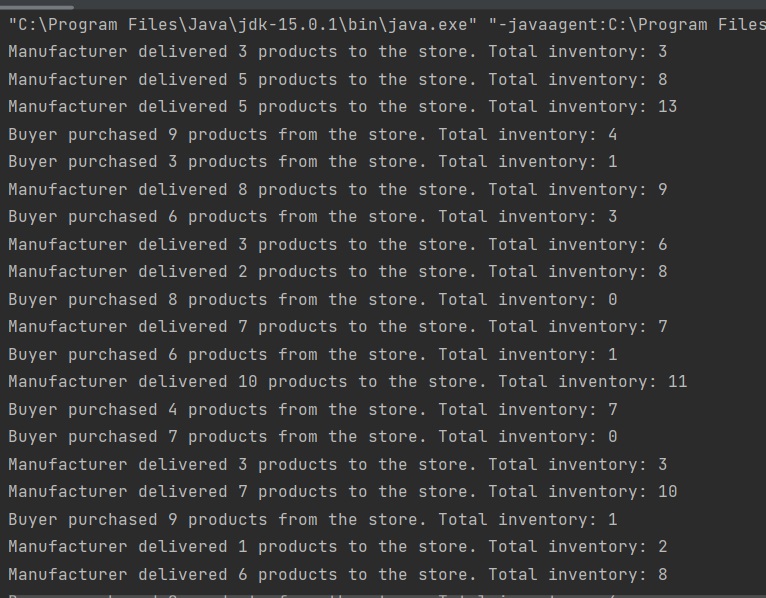


Рисунок 2 – Результат работы программы

**Вывод:** приобретен навык работы с потоками.

**Ссылка на репозиторий с программами:** [**https://github.com/nargi3/BigData**](https://github.com/nargi3/BigData)