|  |  |
| --- | --- |
|  | **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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 8**

**Название:** Потоки

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

Студент \_\_\_ИУ6-23М\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**А.Д. Зеленский**\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**П. В. Степанов**\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

**Цель:** ознакомиться с базовыми принципами языка Java для работы с большими данными.

**Вариант 1:**

1. 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.
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.**Решение:**

Ниже представлен код класса первой программы, а на рисунке 1 – результат работы программы.

public class Main {

public static void main(String[] args) {

System.out.println("Hello world!");

Thread leftLeg = new Thread(new Leg("LEFT", 3));

Thread rightLeg = new Thread(new Leg("RIGHT",2.5));

leftLeg.start();

rightLeg.start();

}

static class Leg implements Runnable {

private String legName;

private int slee;

public Leg(String legName, double slee) {

this.legName = legName;

this.slee= (int) (slee\*1000);

}

public void run() {

while (true) {

System.out.println(legName);

try {

Thread.sleep(slee); // пауза

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

}

Рисунок 1 – Результат работы программы

Ниже представлен код класса второй программы, а на рисунке 2 – результат работы программы.

import java.util.Random;

public class Main {

public static void main(String[] args) {

System.out.println("Hello world!");

Random random = new Random();

Shop shop = new Shop();

Thread delivery = new Thread(new Runnable() {

@Override

public void run() {

while (true) {

try {

shop.delivery();

Thread.sleep(5000);

} catch (InterruptedException e) {

throw new RuntimeException(e);

}

System.out.println("Товар поставлен в магазин. В наличие " + shop.get\_count() + " штук.");

}

}

});

Thread buyer = new Thread(new Runnable() {

@Override

public void run() {

while (true) {

try {

String rez = shop.buyer(random.nextInt(0, 100));

Thread.sleep(5000);

System.out.println(rez);

} catch (InterruptedException e) {

throw new RuntimeException(e);

}

}

}

});

// buyer.start();

delivery.start();

buyer.start();

}

static class Shop {

private int count\_t;

boolean aval\_flg;

Random random = new Random();

public Shop() {

this.count\_t = 0;

this.aval\_flg = false;

}

public void delivery() {

this.count\_t =this.count\_t + random.nextInt(1, 100);

this.aval\_flg = true;

}

public int get\_count() {

return this.count\_t;

}

public String buyer(int c) {

String ret;

if (this.aval\_flg = false) {

ret = "Тавара нет в наличии. Дождитесь поставки!";

return ret;

} else {

if (c > this.count\_t) {

ret = "Тавара нет в наличии в данном объеме " + c +" штук. " +

"Дождитесь поставки или запросите меньше! В наличие: " + this.count\_t + " штук.";

return ret;

}else {

this.count\_t= this.count\_t-c;

ret = "Тавар выдан в запрошенном объеме " + c +" штук. " +

" В наличие осталось: " + this.count\_t + " штук.";

return ret;

}

}

}

}

}

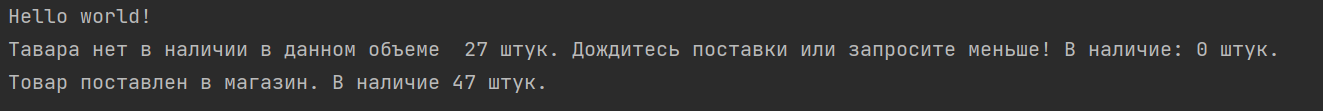


Рисунок 2 – Результат работы программы

**Вывод:** были получены базовые представления о работе языка java с потоками. Были написаны программы согласно вариантам.