|  |  |
| --- | --- |
| 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 И СИСТЕМЫ УПРАВЛЕНИЯ**

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

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

**Отчет по лабораторной работе № 8**

***по дисциплине «Языки программирования для работы с большими данными»***

Студент ИУ6-21М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Щербакова**

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

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

(Подпись, дата) (И.О.Фамилия)

*2020 г.*

**Задание 1**

**Вариант 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.

**Код программы:**

**Файл Main.java**

package Robot;

import java.util.concurrent.TimeUnit;

public class Main {

public static void main(String[] args) throws InterruptedException {

Object lock = new Object();

int count = 5;

Runnable leftStep = () -> {

synchronized(lock) {

for (int i = 0; i < count; i++) {

try {

*TimeUnit*.***SECONDS***.sleep(1);

System.***out***.println("LEFT");

lock.notify();

lock.wait();

} catch (InterruptedException e) {

System.***out***.println("Interrupted");

}

}

}

};

Runnable rightStep = () -> {

synchronized(lock) {

for (int i = 0; i < count; i++) {

try {

*TimeUnit*.***SECONDS***.sleep(1);

System.***out***.println("RIGHT");

lock.notify();

lock.wait();

} catch (InterruptedException e) {

System.***out***.println("Interrupted");

}

}

}

};

Thread leftThread = new Thread(leftStep);

Thread rightThread = new Thread(rightStep);

leftThread.start();

rightThread.start();

leftThread.join();

rightThread.join();

}

}

**Вывод программы**

LEFT

RIGHT

LEFT

RIGHT

LEFT

RIGHT

LEFT

RIGHT

LEFT

RIGHT

**Вывод:** Программы работают корректно.