|  |  |
| --- | --- |
|  | **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**

**Название:** Потоки (Threads)

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | И.Д. Капкин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

1. ****Введение и задание****

**Целью лабораторной работы является формирование навыков использования потоков, изучение соответствующих синтаксических конструкций и особенностей работы.**

**Номер в списке – 10.**

**Вариант 1.**

**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.

**Задание 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.

1. ****Ход работы****

**Выполнение задания №1 первого варианта лабораторной работы:**

**import java.util.Calendar;**

**import java.util.Random;**

**import java.util.stream.IntStream;**

**public class Main {**

**static final Random RNG = new Random(Calendar.getInstance().getTimeInMillis());**

**public static void main(String[] args) {**

**Bank account = new Bank(RNG.nextDouble(1000.0));**

**IntStream.range(0, 100).forEach((unused) -> {**

**new Thread(() -> {account.deposit(RNG.nextDouble(2000.0));}).start();**

**new Thread(() -> {account.withdraw(RNG.nextDouble(5000.0));}).start();**

**});**

**}**

**}**

**class Bank {**

**private double balance;**

**Bank(double balance) {**

**this.balance = balance;**

**}**

**public synchronized void deposit(double debet) {**

**this.balance += debet;**

**System.out.println(this.balance);**

**}**

**public synchronized void withdraw(double credit) {**

**if (this.balance > credit) {**

**this.balance -= credit;**

**System.out.println(this.balance);**

**} else {**

**System.out.println("Not enough money to withdraw");**

**}**

**}**

**}**

**Выполнение задания №2 первого варианта лабораторной работы:**

**import java.util.stream.IntStream;**

**public class Main {**

**public static void main(String[] args) {**

**Robot robot = new Robot();**

**IntStream.range(0,20).forEach((unused) -> {**

**new Thread(() -> {**

**try {**

**robot.stepLeft();**

**} catch (InterruptedException e) {**

**throw new RuntimeException(e);**

**}**

**}).start();**

**new Thread(() -> {**

**try {**

**robot.stepRight();**

**} catch (InterruptedException e) {**

**throw new RuntimeException(e);**

**}**

**}).start();**

**});**

**}**

**}**

**class Robot {**

**private Foot forwardFoot;**

**public synchronized void stepLeft() throws InterruptedException {**

**while (this.forwardFoot == Foot.LEFT) {**

**wait();**

**}**

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

**this.forwardFoot = Foot.LEFT;**

**notifyAll();**

**}**

**public synchronized void stepRight() throws InterruptedException {**

**while (this.forwardFoot == Foot.RIGHT) {**

**wait();**

**}**

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

**this.forwardFoot = Foot.RIGHT;**

**notifyAll();**

**}**

**private enum Foot {**

**LEFT, RIGHT**

**}**

**}**

1. ****Заключение****

**В процессе выполнения лабораторной работы были сформированы навыки использования в языке программирования java, были изучены соответствующие синтаксические конструкции и особенности работы.**