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**

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

Студент ИУ6-22М _____ Т.И. Кадыров
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2024

Цель работы: освоить базовые принципы работы с потоками на языке Java.

Вариант: 8.

Задание 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.

Код решения приведен в листинге 1.

Листинг 1 — реализация решения

```
class Robot extends Thread {
   private String direction;

   public Robot(String direction) {
      this.direction = direction;
   }

   @Override
   public void run() {
      walk();
   }

   public void walk() {
      System.out.println(Thread.currentThread().getName() + " starts walking " + direction);

      for (int i = 0; i < 5; i++) {
         System.out.println(Thread.currentThread().getName() + " steps " + direction);

         try {
            Thread.sleep(1000);
         } catch (InterruptedException e) {
            System.out.println(e.getMessage());
         }
      }

      System.out.println(Thread.currentThread().getName() + " finishes walking " + direction);
   }
}

public class Main extends Thread {

   public static void main(String[] args) {
      Robot leftLeg = new Robot("LEFT");
      Robot rightLeg = new Robot("RIGHT");

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

Задание 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.

Код решения приведен в листинге 2.

Листинг 2 — реализация решения

```
import java.util.concurrent.locks.*;

class Shop {
  private int stock = 0;
  private Lock lock = new ReentrantLock();
  private Condition condition = lock.newCondition();

  public void produce(int quantity) {
    lock.lock();
    try {
      stock += quantity;
      System.out.println("Производитель поставил " + quantity + " товар(ов). Всего на складе: " + stock);
      condition.signalAll();
    } finally {
      lock.unlock();
    }
  }

  public void consume(int quantity) throws InterruptedException {
    lock.lock();
    try {
      while (stock < quantity) {
        System.out.println("Товара в магазине не хватает. Подождите...");
        condition.await();
      }
      stock -= quantity;
      System.out.println("Покупатель забрал " + quantity + " товар(ов). Всего на складе: " + stock);
    } finally {
      lock.unlock();
    }
  }
}

class Producer implements Runnable {
  private Shop shop;

  public Producer(Shop shop) {
    this.shop = shop;
  }

  @Override
  public void run() {
```

```
        try {
            while (true) {
                int quantity = (int) (Math.random() * 5) + 1; // случайное количество товара от 1 до 5
                shop.produce(quantity);
                Thread.sleep(1000); // пауза для имитации времени производства
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

class Consumer implements Runnable {
    private Shop shop;

    public Consumer(Shop shop) {
        this.shop = shop;
    }

    @Override
    public void run() {
        try {
            while (true) {
                int quantity = (int) (Math.random() * 3) + 1; // случайное количество товара от 1 до 3
                shop.consume(quantity);
                Thread.sleep(1500); // пауза для имитации времени покупки
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Shop shop = new Shop();
        Producer producer = new Producer(shop);
        Consumer consumer = new Consumer(shop);

        Thread producerThread = new Thread(producer);
        Thread consumerThread = new Thread(consumer);

        producerThread.start();
        consumerThread.start();
    }
}
```

Вывод: в ходе лабораторной работы были освоены базовые принципы работы с потоками на языке Java.