



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

Вариант 14

Название: Потоки (Threads)

Дисциплина: Языки программирования для работы с большими
данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

В.Е. Санталов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы: получение навыков работы с потоками в Java.

Задание 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.
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.

Подзадача 1.

Код программы:

```

import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;

public class Leg extends Thread {
    private final String side;
    private final BlockingQueue<String> orders;

    public Leg(String side) {
        this.side = side;
        this.orders = new LinkedBlockingQueue<String>();
    }

    public void makeStep() {
        this.orders.add("step");
    }

    public void stopLeg() {
        this.orders.add("stop");
    }

    @Override
    public void run() {
        loop: while (true) {
            try {
                String order = orders.take();

                switch (order) {
                    case "step":
                        System.out.println(this.side + " нога сделала шаг");
                        break;

                    default:
                        break loop;
                }
            } catch (InterruptedException e) {
                System.err.println("Error occured: " + e);
            }
        }
    }
}

```

```

public class Robot {
    private Leg legLeft;
    private Leg legRight;

    private Boolean lastLegLeft;

    public Robot() {
        this.legLeft = new Leg(side: "Левая");
        this.legRight = new Leg(side: "Правая");

        this.legLeft.start();
        this.legRight.start();

        this.lastLegLeft = null;
    }

    public void walk(String leg) {
        if (lastLegLeft == null) {
            switch (leg) {
                case "left":
                    this.legLeft.makeStep();
                    this.lastLegLeft = true;
                    break;

                case "right":
                    this.legRight.makeStep();
                    this.lastLegLeft = false;
                    break;

                default:
                    System.out.println(x: "У робота нет такой ноги");
                    break;
            }
        }
        else {
            if (leg.equals(anObject: "left") && this.lastLegLeft == false) {
                this.legLeft.makeStep();
                this.lastLegLeft = true;
            }
            else if (leg.equals(anObject: "right") && this.lastLegLeft == true) {
                this.legRight.makeStep();
                this.lastLegLeft = false;
            }
            else if (leg.equals(anObject: "left") || leg.equals(anObject: "right")) {
                System.out.println(x: "Данная нога сделала шаг в прошлый раз");
            }
            else {
                System.out.println(x: "У робота нет такой ноги");
            }
        }
    }
}

```

```

        System.out.println(x: "у робота нет такой ноги ");
    }
}

public void stop() {
    this.legLeft.stopLeg();
    this.legRight.stopLeg();
}
}

```

```

import java.util.Scanner;

public class task1 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        Robot robot = new Robot();

        Scanner in = new Scanner(System.in);

        while (true) {
            String line = in.nextLine();

            if (line.equals(anObject: "stop")) {
                robot.stop();
                break;
            }
            else robot.walk(line.replaceAll(regex: "[ \n]", replacement: ""));
        }

        in.close();
    }
}

```

Результат выполнения программы:

```
left
Левая нога сделала шаг
right
Правая нога сделала шаг
left
Левая нога сделала шаг
right
Правая нога сделала шаг
right
Данная нога сделала шаг в прошлый раз
left
Левая нога сделала шаг
left
Данная нога сделала шаг в прошлый раз
left
Данная нога сделала шаг в прошлый раз
stop
```

Подзадача 2.

Код программы:

```

import java.util.ArrayList;
import java.util.Random;

public class Provider extends Thread {
    public ArrayList<String> shopStock;

    final String[] items = {
        "Сухарики 3 корочки",
        "Чипсы Lays",
        "Чипсы Русская Картошка",
        "Coca-Cola",
        "Monster BLACK",
        "Балтика 9"
    };

    Provider(ArrayList<String> shopstock) {
        this.shopStock = shopstock;
    }

    @Override
    public void run() {
        Random random = new Random();

        try {
            while (true) {
                Thread.sleep(random.nextInt(bound: 2) * 1000 + 2000);
                shopStock.add(items[random.nextInt(items.length)]);
            }
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
            e.printStackTrace();
        }
    }
}

```

```

import java.util.ArrayList;

public class Shop extends Thread {
    final ArrayList<String> stock;
    int stockCount;

    Shop(ArrayList<String> stock) {
        this.stock = stock;
        stockCount = 0;
    }

    public String getProduct(int index) {
        stockCount--;

        return stock.remove(index);
    }

    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep(1);
                if (stockCount != stock.size()) {
                    System.out.println(x: "В магазине появился новый товар!");
                    stockCount++;
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

```



```

import java.util.ArrayList;
import java.util.Scanner;

public class task2 {
    Run | Debug
    public static void main(String[] args) {
        ArrayList<String> stock = new ArrayList<String>();

        Shop shop = new Shop(stock);
        Provider provider = new Provider(stock);

        shop.start();
        provider.start();

        Scanner in = new Scanner(System.in);

        while (true) {
            String line = in.nextLine();

            if (line.equals(anObject: "stop")) {
                shop.stop();
                provider.stop();
                System.out.println(x: "Вы ушли из магазина. До скорой встречи!");
                break;
            }
            else if (line.equals(anObject: "contents")) {
                System.out.println("В магазине есть: " + stock);
            }
            else if (line.startsWith(prefix: "get")) {
                System.out.println("Вы купили: " + shop.getProduct(Integer.parseInt(line.split(regex: " ")[1])));
            }
        }

        in.close();
    }
}

```

Результат выполнения программы:

```

В магазине появился новый товар!
В магазине появился новый товар!
В магазине появился новый товар!
В магазине появился новый товар!
contents
В магазине есть: [Coca-Cola, Coca-Cola, Monster BLACK, Чипсы Lays]
В магазине появился новый товар!
В магазине появился новый товар!
contents
В магазине есть: [Coca-Cola, Coca-Cola, Monster BLACK, Чипсы Lays, Monster BLACK, Сухарики 3 корочки]
В магазине появился новый товар!
В магазине появился новый товар!
get 4
Вы купили: Monster BLACK
В магазине появился новый товар!
В магазине появился новый товар!
contents
В магазине есть: [Coca-Cola, Coca-Cola, Monster BLACK, Чипсы Lays, Сухарики 3 корочки, Балтика 9, Coca-Cola, Чипсы Lays, Чипсы Lays]
В магазине появился новый товар!
stoВ магазине появился новый товар!
r
Вы ушли из магазина. До скорой встречи!

```

Ссылка на программное решение:

<https://github.com/Time2HackJS/BigDataLanguages/tree/master/lr8>

Вывод: в ходе лабораторной работы были получены навыки работы с потоками в Java.