



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 13**

**Название:**                    Потоки

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

Студент                    ИУ6-13М  
(Группа)

\_\_\_\_\_  
(Подпись, дата)                    Е.К. Матвиенко  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)                    П.В. Степанов  
(И.О. Фамилия)

Москва, 2022

**Цель лабораторной работы:** получение первичных навыков работы с потоками языка программирования Java.

### Ход работы:

#### 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.

#### Листинг выполнения задачи 1:

```
package laba_8;
import java.util.Random;
//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.
public class Bank {
    public volatile static int account;
    public volatile static Random random;

    public static void main(String[] args) throws InterruptedException {
        random = new Random();
        account = 0;

        new Thread(income).start();
        Thread.sleep(2000);
        new Thread(outcome).start();
    }

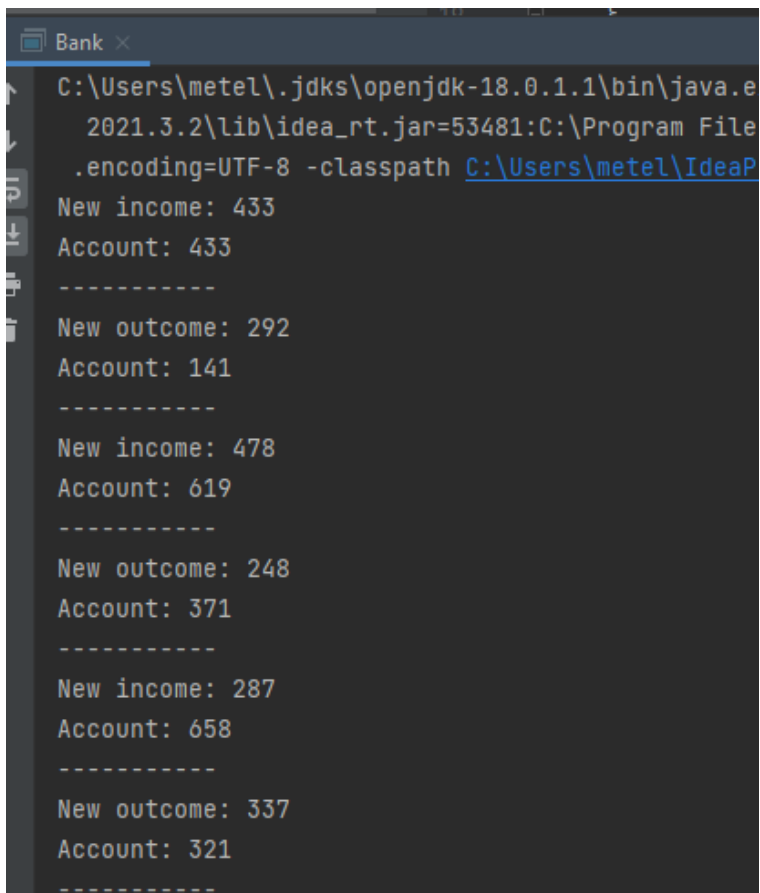
    static Runnable income = new Runnable() {
        @Override
        public void run() {
            while (true) {
                try {
                    Thread.sleep(5000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                int receipt = random.nextInt(100, 500);
                account += receipt;
                System.out.print("New income: ");
                System.out.println(receipt);
                System.out.print("Account: ");
                System.out.println(account);
                System.out.println("-----");
            }
        }
    };
};
```

```

static Runnable outcome = new Runnable() {
    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep(5000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            int withdraw = random.nextInt(200, 400);
            if (withdraw <= account) {
                account -= withdraw;
                System.out.print("New outcome: ");
                System.out.println(withdraw);
                System.out.print("Account: ");
                System.out.println(account);
                System.out.println("-----");
            } else {
                System.out.print("Tried to write-off: ");
                System.out.println(withdraw);
                System.out.print("Account: ");
                System.out.println(account);
                System.out.println("Not enough money!");
                System.out.println("-----");
            }
        }
    }
};
}

```

### Результат выполнения задачи 1:



```

C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.e
2021.3.2\lib\idea_rt.jar=53481:C:\Program File
.encoding=UTF-8 -classpath C:\Users\metel\IdeaP
New income: 433
Account: 433
-----
New outcome: 292
Account: 141
-----
New income: 478
Account: 619
-----
New outcome: 248
Account: 371
-----
New income: 287
Account: 658
-----
New outcome: 337
Account: 321
-----

```

## 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.

### Листинг выполнения задачи 2:

#### Класс Leg:

```
package laba_8;
//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.
public class Robot {
    private Leg legLeft;
    private Leg legRight;

    private Boolean lastLegLeft;

    public Robot() {
        this.legLeft = new Leg("Left");
        this.legRight = new Leg("Right");

        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("there is no such leg");
                    break;
            }
        }
        else {
            if (leg.equals("left") && this.lastLegLeft == false) {
                this.legLeft.makeStep();
                this.lastLegLeft = true;
            }
            else if (leg.equals("right") && this.lastLegLeft == true) {
                this.legRight.makeStep();
                this.lastLegLeft = false;
            }
            else if (leg.equals("left") || leg.equals("right")) {
                System.out.println("this leg made a step in previous time");
            }
            else {

```

```

        System.out.println("There is no such a leg");
    }
}

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

```

## Класс Robot:

```

package laba_8;
//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.
public class Robot {
    private Leg legLeft;
    private Leg legRight;

    private Boolean lastLegLeft;

    public Robot() {
        this.legLeft = new Leg("Left");
        this.legRight = new Leg("Right");

        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("there is no such leg");
                    break;
            }
        }
        else {
            if (leg.equals("left") && this.lastLegLeft == false) {
                this.legLeft.makeStep();
                this.lastLegLeft = true;
            }
            else if (leg.equals("right") && this.lastLegLeft == true) {
                this.legRight.makeStep();
                this.lastLegLeft = false;
            }
            else if (leg.equals("left") || leg.equals("right")) {
                System.out.println("this leg made a step in previous time");
            }
        }
    }
}

```

```

        }
        else {
            System.out.println("There is no such a leg");
        }
    }
}

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

```

### Основная программа MainRobot:

```

package laba_8;
import java.util.Scanner;

public class MainRobot {
    public static void main(String[] args) throws Exception {
        Robot robot = new Robot();

        Scanner in = new Scanner(System.in);

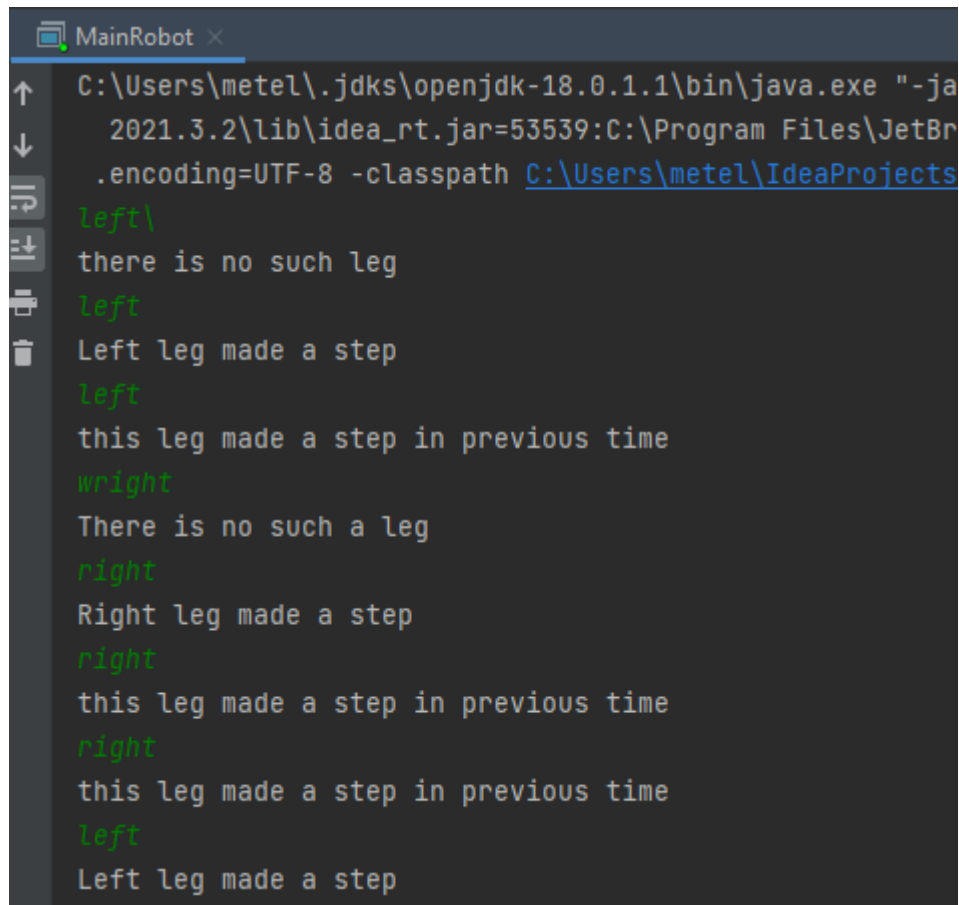
        while (true) {
            String line = in.nextLine();

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

        in.close();
    }
}

```

## Результат выполнения задачи 2:



```
C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.exe "-ja
2021.3.2\lib\idea_rt.jar=53539:C:\Program Files\JetBr
.encoding=UTF-8 -classpath C:\Users\metel\IdeaProjects
left\
there is no such leg
left
Left leg made a step
left
this leg made a step in previous time
wright
There is no such a leg
right
Right leg made a step
right
this leg made a step in previous time
right
this leg made a step in previous time
left
Left leg made a step
```

Программное решение представлено в репозитории распределённой системы управления версиями Git:

[https://github.com/matvilen/BigDataLanguages/tree/main/my\\_lab\\_8/src/laba\\_8](https://github.com/matvilen/BigDataLanguages/tree/main/my_lab_8/src/laba_8)

**Вывод:** при выполнении лабораторной работы были получены навыки работы с потоками в Java.