



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 09.04.01/12 Интеллектуальный анализ больших
данных в системах поддержки принятия решений.

О Т Ч Е Т

по лабораторной работе № 8

Вариант № 4

Название: потоки

Дисциплина: языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

А.А.Клушина

(И.О. Фамилия)

Преподаватель

П.В. Степанов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2024

Цель: изучить потоки в java

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 снятия) вывести текущий баланс счета. В том случае, если денежных средств недостаточно – вывести сообщение.

Код класса Bank:

```
import java.util.Random;

public class Bank {
    private int balance;

    public Bank(int balance) {
        this.balance = balance;
    }

    public synchronized void deposit(int amount) {
        balance += amount;
        System.out.println("Вклад: " + amount + " | Текущий
баланс: " + balance);
    }

    public synchronized void withdraw(int amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Снятие: " + amount + " | Текущий
баланс: " + balance);
        } else {
            System.out.println("Недостаточно средств! ");
        }
    }

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

```

Bank account = new Bank(100);

Runnable depositTask = () -> {
    Random random = new Random();
    int amount = random.nextInt(50) + 1;
    account.deposit(amount);
};

Runnable withdrawTask = () -> {
    Random random = new Random();
    int amount = random.nextInt(50) + 1;
    account.withdraw(amount);
};

for (int i = 0; i < 10; i++) {
    new Thread(depositTask).start();
    new Thread(withdrawTask).start();
}
}
}

```

Работа программы показана на рисунке 1.

```
7/31/2017 3:11:31 PM java 11/31/2017 java 3/11/2017 java
Снятие: 8 | Текущий баланс: 92
Снятие: 40 | Текущий баланс: 52
Вклад: 38 | Текущий баланс: 90
Снятие: 16 | Текущий баланс: 74
Снятие: 15 | Текущий баланс: 59
Вклад: 23 | Текущий баланс: 82
Вклад: 33 | Текущий баланс: 115
Снятие: 20 | Текущий баланс: 95
Вклад: 15 | Текущий баланс: 110
Снятие: 34 | Текущий баланс: 76
Снятие: 12 | Текущий баланс: 64
Вклад: 35 | Текущий баланс: 99
Снятие: 45 | Текущий баланс: 54
Вклад: 12 | Текущий баланс: 66
Вклад: 39 | Текущий баланс: 105
Снятие: 16 | Текущий баланс: 89
Снятие: 22 | Текущий баланс: 67
Вклад: 49 | Текущий баланс: 116
Вклад: 32 | Текущий баланс: 148
Вклад: 46 | Текущий баланс: 194

Process finished with exit code 0
```

Рисунок 1 – Работа программы

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.

Код класса Main:

```
public class Robot {

    private static Object lock = new Object();
    private static boolean isLeftLegTurn = true;

    public static void main(String[] args) {
        Thread leftLeg = new Thread(() -> {
            while (true) {
                synchronized (lock) {
                    if (isLeftLegTurn) {
                        System.out.println("LEFT");
                        isLeftLegTurn = false;
                    }
                }
            }
        });
        leftLeg.start();
    }
}
```

```

        }
    }

    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

});

Thread rightLeg = new Thread(() -> {
    while (true) {
        synchronized (lock) {
            if (!isLeftLegTurn) {
                System.out.println("RIGHT");
                isLeftLegTurn = true;
            }
        }
    }

    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

});

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

```

Работа программы показана на рисунке 2.



```
LEFT  
RIGHT  
LEFT  
RIGHT  
LEFT  
RIGHT  
LEFT  
RIGHT
```

Рисунок 2 – Работа программы

Вывод: во время выполнения лабораторной работы были изучены потоки в java.