



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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И
СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

ОТЧЕТ

по лабораторной работе № 8

Вариант 13

Название: Потоки

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-22М

(Группа)

(Подпись, дата)

В.А.Ловцов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

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

Код:

```
import java.util.Random;

class BankAccount {
    private int balance;

    public BankAccount(int initialBalance) {
        this.balance = initialBalance;
    }

    public synchronized void debit(int amount) {
        balance += amount;
        System.out.println("\tПополнение: +" + amount);
        System.out.println("Баланс: " + balance);
    }

    public synchronized void credit(int amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("\tСнятие: -" + amount);
            System.out.println("Баланс: " + balance);
        } else {
            System.out.println("Недостаточно средств для
снятия!");
        }
    }
}
```

```

        public int getBalance(){
            return this.balance;
        }
    }

    public class First {
        public static void main(String[] args) {
            // Создание объекта банка с начальным балансом 1000
            BankAccount account = new BankAccount(10000);
            System.out.println("Баланс: " + account.getBalance());

            Thread debitThread = new Thread(() -> {
                Random random = new Random();
                while (true) {
                    int amount = random.nextInt(1000) + 1;
                    account.debit(amount);
                    try {
                        Thread.sleep(3000);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            });

            Thread creditThread = new Thread(() -> {
                Random random = new Random();
                while (true) {
                    int amount = random.nextInt(1000) + 1;
                    account.credit(amount);
                    try {
                        Thread.sleep(3500);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            });
        }
    }

```

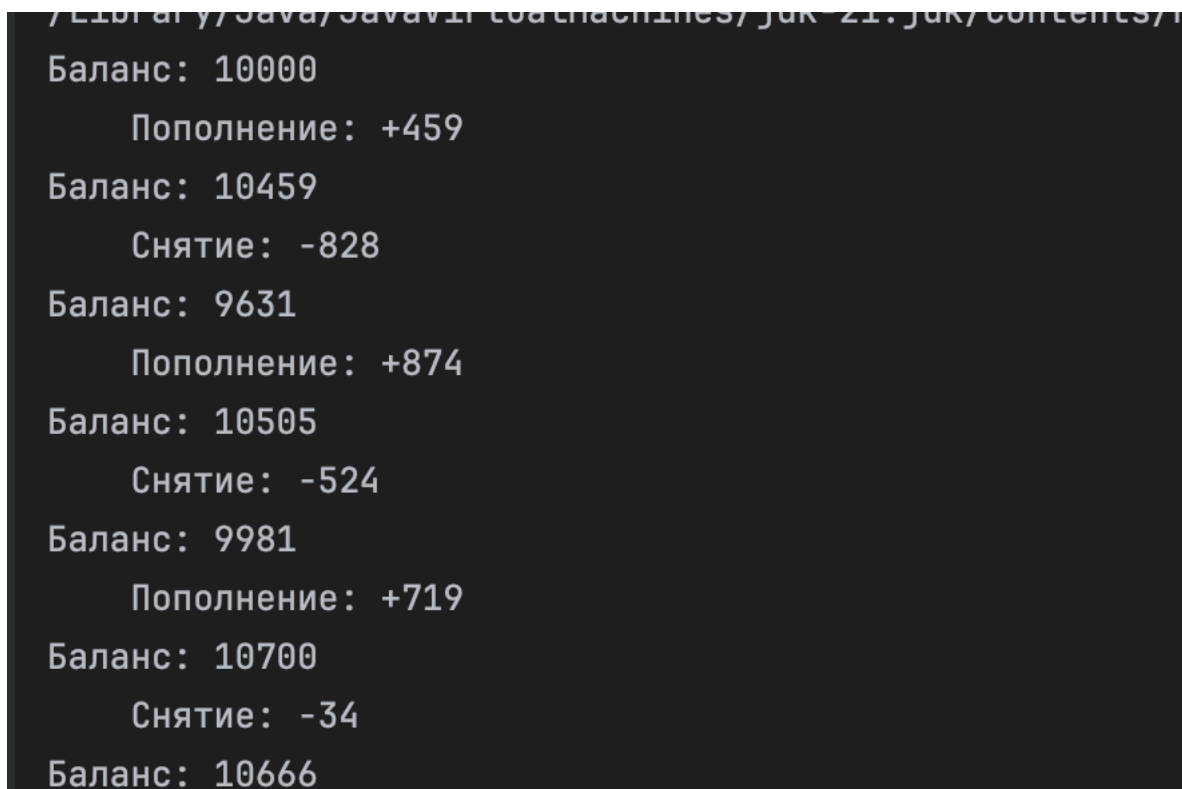
```

        }
    }
});

debitThread.start();
creditThread.start();
}
}

```

Работа программы показана на рисунке 1.



```

/.../java/javav1-console/jak-21.jar/console/1
Баланс: 10000
    Пополнение: +459
Баланс: 10459
    Снятие: -828
Баланс: 9631
    Пополнение: +874
Баланс: 10505
    Снятие: -524
Баланс: 9981
    Пополнение: +719
Баланс: 10700
    Снятие: -34
Баланс: 10666

```

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:

```

class Robot {
    private Thread moveLeftThread;

```

```

private Thread moveRightThread;

Robot() {
    moveLeftThread = new Thread(() -> {
        while (true) {
            System.out.println("left");
            try {
                Thread.sleep(2000); // Пауза 1 секунда
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    });

    moveRightThread = new Thread(() -> {
        while (true) {
            System.out.println("right");
            try {
                Thread.sleep(2000); // Пауза 1 секунда
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    });
}

public void startRight() {
    moveRightThread.start();
}

public void startLeft() {
    moveLeftThread.start();
}

```

```

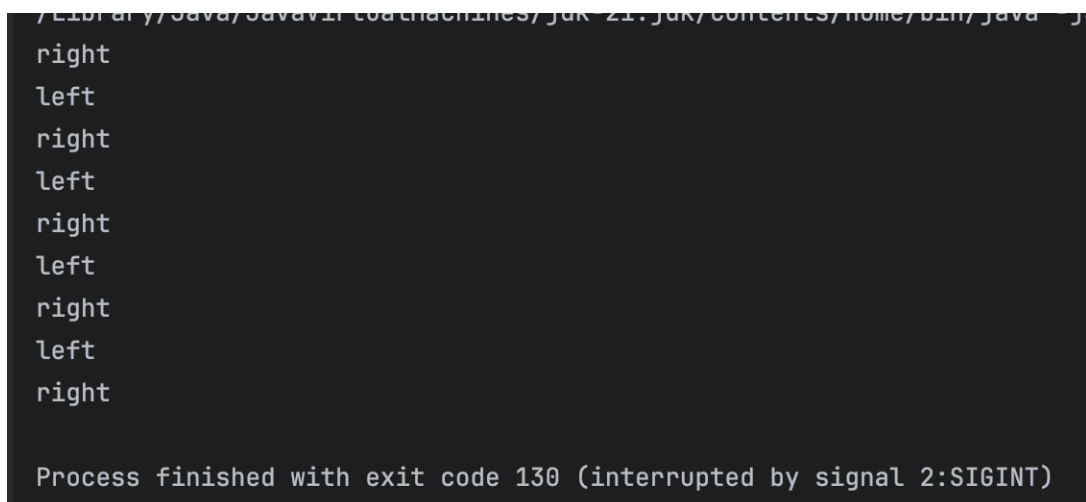
    public void pause(int time){
        try {
            Thread.sleep(time);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

}

public class Second {
    public static void main(String[] args) throws
InterruptedException {
        Robot robot = new Robot();
        robot.startRight();
        robot.pause(1000);
        robot.startLeft();
    }
}

```

Работа программы показана на рисунке 2.



```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -J
right
left
right
left
right
left
right
left
right
left
right
Process finished with exit code 130 (interrupted by signal 2:SIGINT)

```

Рисунок 2 – Работа программы

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

