

# Странная джава

2018



# Привет!

## Константин Грошев

Ведущий программист / Luxoft

Тимлид, отец, мечтатель

# Тихая гавань

---

Все, что будет показано на этом и последующих слайдах является точкой зрения автора и не является официальной позицией той или иной компании. Все возможные совпадения случайны.

Пожалуйста, не принимайте решений на основе этого доклада. В противном случае воспользуйтесь услугами профессионалов.



# Что будет происходить дальше

---

Небольшие задачи

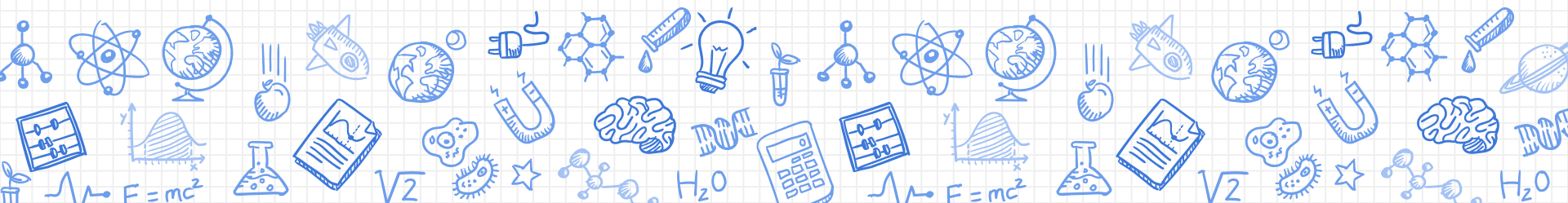
Формат:

- X Задача
- X Обсуждение
- X Правильный ответ
- X Пояснение



# Неравное равенство

Все мы одинаково равны,  
но некоторые равны равнее



```
public class IntegerPuzzle {  
    public static void main(String[] args) {  
        Integer a = 12;  
        Integer b = 12;  
        if (a == b) System.out.println("bingo!");  
    }  
}
```

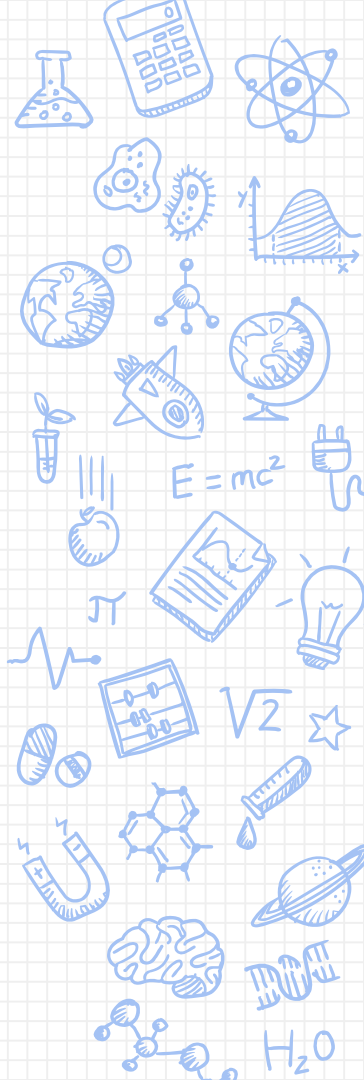
**Вывод:**

**bingo!**

# Почему?

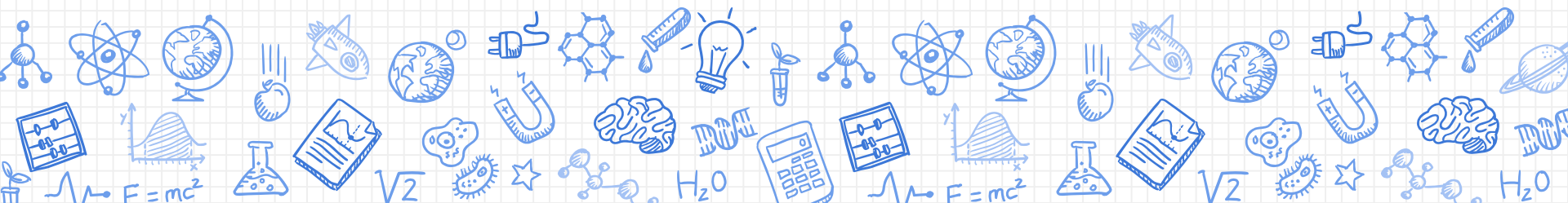
---

```
/**  
 * ...  
 * This method will always cache values in the range -128 to 127,  
 * inclusive, and may cache other values outside of this range.  
 * ...  
 * @since 1.5  
 */  
public static Integer valueOf(int i) {  
    if (i >= IntegerCache.low && i <= IntegerCache.high)  
        return IntegerCache.cache[i + (-IntegerCache.low)];  
    return new Integer(i);  
}
```



# Не совсем статические типы

Java – это язык со  
статической типизацией





```
public class StringPuzzle {  
    public static void main(String[] args) {  
        String s = null;  
        s += 10;  
        System.out.println(s);  
    }  
}
```

**Вывод:**

**null10**

# Почему?

---

JLS, 5.1.11. String Conversion:

...Now only reference values need to be considered. **If the reference is null, it is converted to the string "null"** (four ASCII characters n, u, l, l). Otherwise, the conversion is performed as if by an invocation of the toString method of the referenced object with no arguments; but if the result of invoking the toString method is null, then the string "null" is used instead.

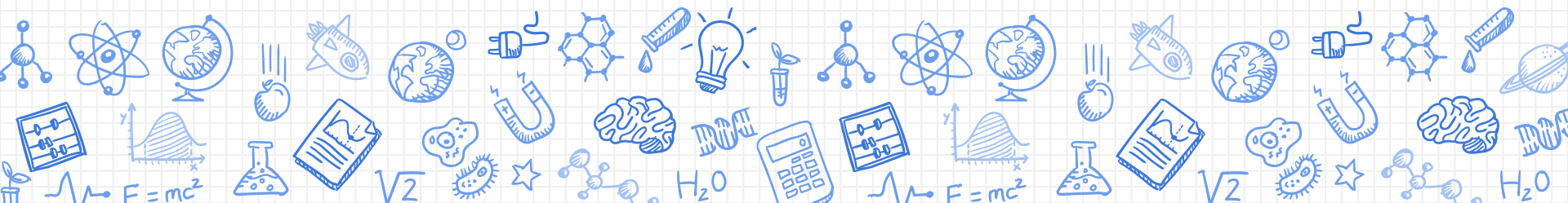
JLS, 15.18.1. String Concatenation Operator +

```
s = new StringBuilder(String.valueOf(s))  
                .append(String.valueOf(10)).toString();
```



# Очарованный стрим

Не верь глазам своим



```
public class Twuzzler {  
    public static void main(String[] args) {  
        "Hello world!".chars().forEach(System.out::print);  
    }  
}
```

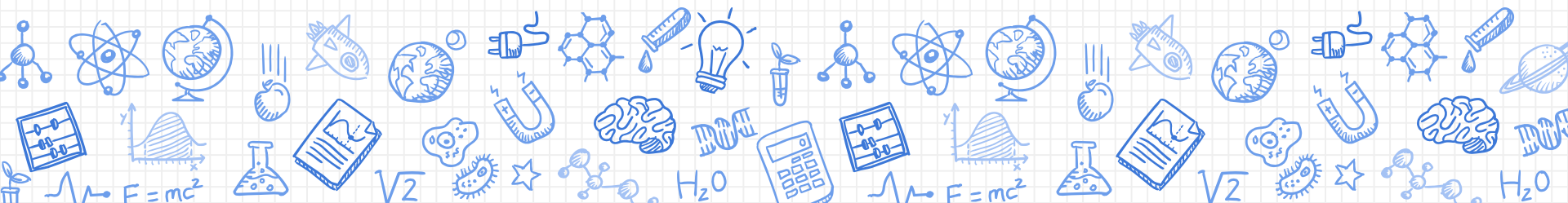
**Вывод:**

**72101108108111321191111410810033**



# Непонятный инкремент

Все просто как  $2+2!$



```
public class Increment {  
    public static void main(String[] args) {  
        int j = 0;  
        for (int i = 0; i < 100; i++)  
            j = j++;  
        System.out.println(j);  
    }  
}
```

**Вывод:**

0

## Почему?

---

### *JLS 15.14.2 postfix increment operator*

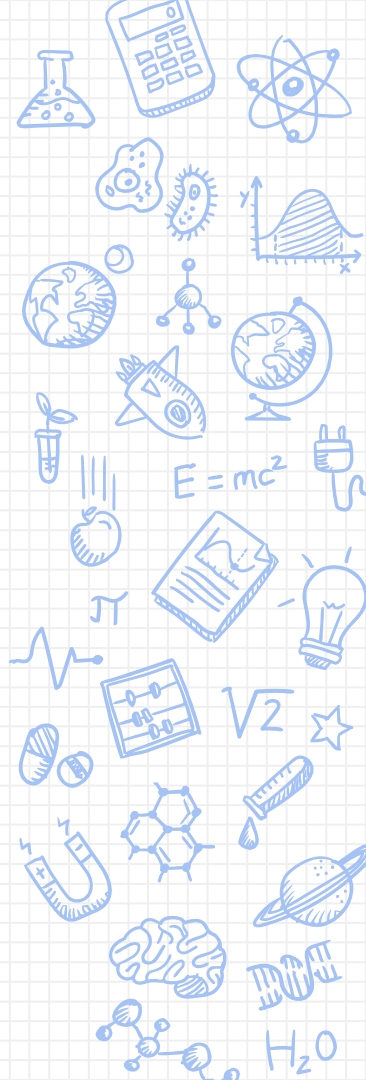
At run time, if evaluation of the operand expression completes abruptly, then the postfix increment expression completes abruptly for the same reason and no incrementation occurs. Otherwise, the value 1 is added to the value of the variable and the sum is stored back into the variable.

```
int tmp = j;
```

```
j = j + 1;
```

```
j = tmp;
```

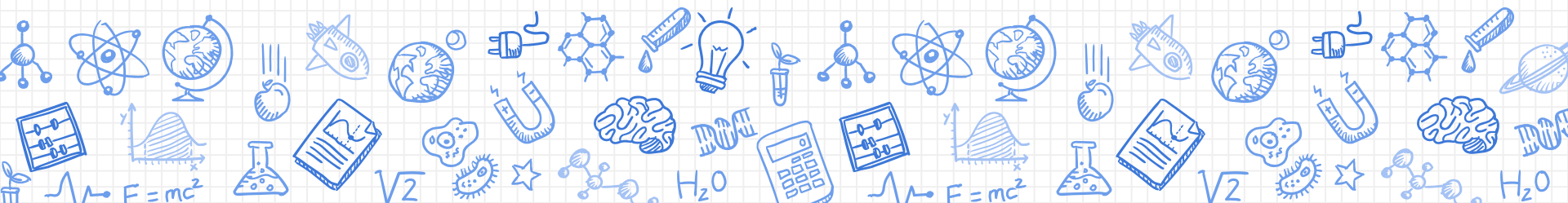
*... и никакой магии!*





# Нелогичное решение

Лапки у бегемота круглые, чтобы было удобно  
порхать с кувшинки на кувшинку



```
public class Indecisive {  
    public static void main(String[] args) {  
        System.out.println(decision());  
    }  
  
    static boolean decision() {  
        try {  
            return true;  
        } finally {  
            return false;  
        }  
    }  
}
```

Вывод: false

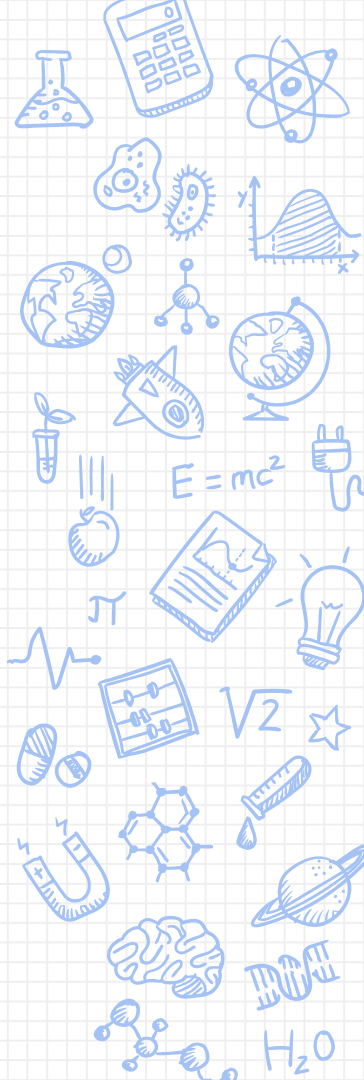
## Почему?

---

### *JLS 14.20.2 Execution of try-finally and try-catch-finally*

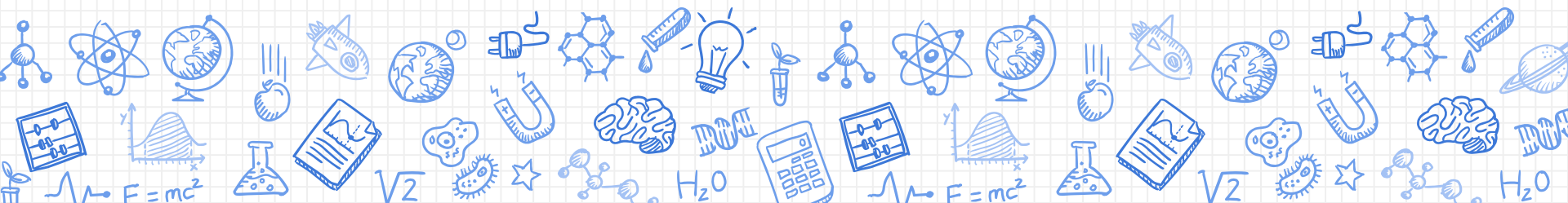
If execution of the try block completes normally, then the finally block is executed, and then there is a choice:

- X** If the finally block completes normally, then the try statement completes normally
- X** If the finally block completes abruptly for reason  $S$ , then the try statement completes abruptly for reason  $S$ .



# Назойливый NPE

Чтобы засунуть жирафа в холодильник мы просто открываем холодильник и кладем туда жирафа



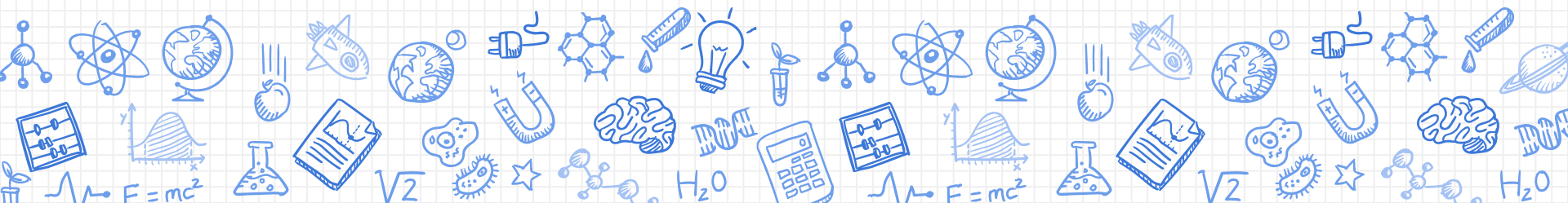
```
public class NullPointerExceptionPuzzle {  
    public static void main(String[] args) {  
        try {  
            throw new NullPointerException("NullPointerException 1");  
        } catch (NullPointerException e) {  
            throw new NullPointerException("NullPointerException 2");  
        } finally {  
            return;  
        }  
    }  
}
```

**Вывод:**



# Большие проблемы малых чисел

Если то, что вы видите – очевидно, это не значит, что так оно и есть. Возможно, вы чего-то не знаете



```
public class BigProblem {  
    public static void main(String[] args) {  
        BigInteger firstParam = new BigInteger("2");  
        BigInteger secondParam = new BigInteger("2");  
  
        BigInteger total = BigInteger.ZERO;  
        total.add(firstParam);  
        total.add(secondParam);  
        System.out.println(total);  
    }  
}
```

**Вывод: 0**



## Почему?

---

BigInteger, String, BigDecimal, а также обертки Integer, Long, Short, Byte, Character, Boolean, Float, и Double – это immutable типы!

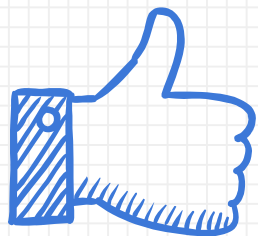


Хочу знать больше!

---

- X JLS для вашей платформы и версии Java
- X Joshua Bloch, Neal Gafter. Java Puzzlers. Traps, Pitfalls, and Corner Cases
- X Joshua Bloch. Effective Java: Second Edition
- X Gayle Laakmann McDowell. Cracking the Coding Interview: 189 Programming Questions and Solutions





# СПАСИБО!

## У вас есть вопросы?

Ищите меня в интернете

- X [@kgroshev](#)
- X [mail@groshev.net](mailto:mail@groshev.net)
- X [www.groshev.net](http://www.groshev.net)