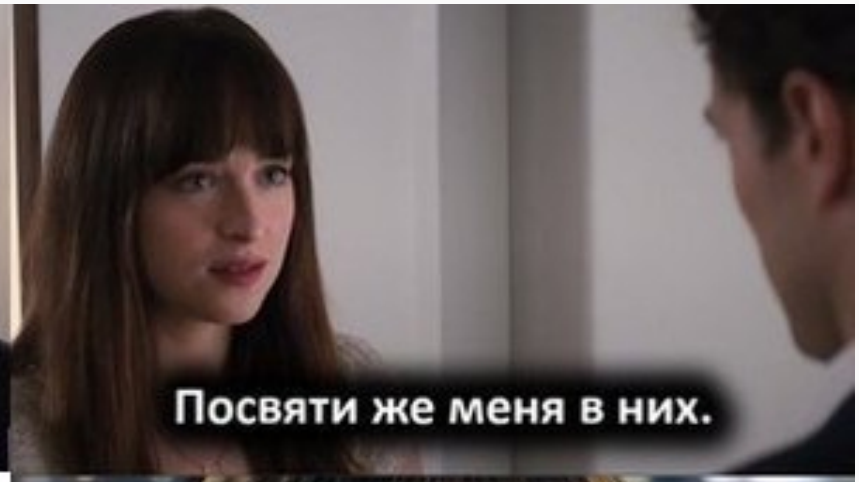
50 shades of errors

Марат Ахин

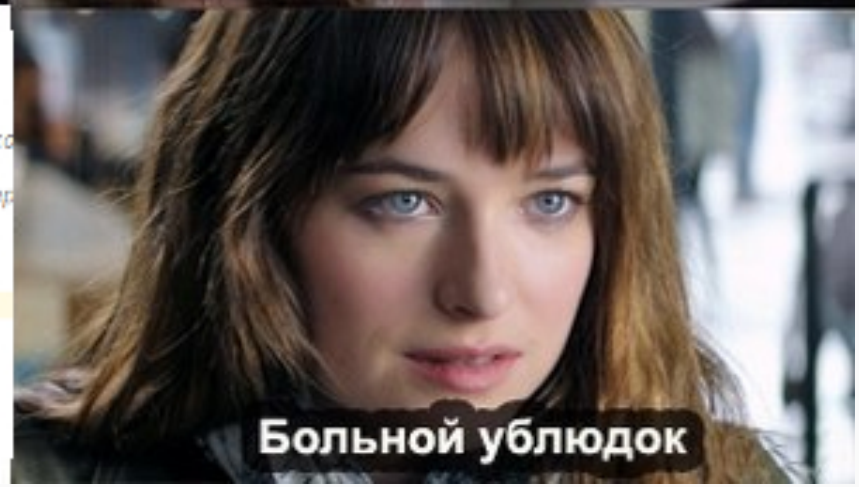
22 ноября 2016 г.

Санкт-Петербургский политехнический университет

Прелюдия




```
/**
 * Простая
 *
 * Для заданных чисел m и n, m <= n, определить, имеется ли хо
 * то есть, существует ли такое целое k, что m <= k*k <= n.
 * Например, для интервала 21..28 21 <= 5*5 <= 28, а для интер
 */
fun squareBetweenExists(m: Int, n: Int): Boolean {
    for (i in m..n) {
        return Math.sqrt(i.toDouble()) ==
                Math.sqrt(i.toDouble()).toInt().toDouble()
    }
    return false
}
```



Disclaimer

- Все, представленное в данной презентации, является личным мнением состава преподавателей, пропущенным через призму их многолетнего опыта разработки и преподавания
- Несмотря на это, данная презентация может содержать как формальные, так и фактические ошибки
- Не следует воспринимать все, что следует далее, как совершенный абсолют

Disclaimer


"ONLY A SITH DEALS IN ABSOLUTES."
MAKES ABSOLUTE STATEMENTS
quickmeme.com

Зачем все это?

- Студенты делают очень много ошибок
- Студенты делают одни и те же ошибки снова и снова
- Преподаватели устали писать одно и то же ☺

Глупый учится на своих ошибках, а умный — на чужих (с)

Типы ошибок

- Ошибки “тридевятого царства”
- Непонимание языка Kotlin
- Подход “костыли и палки”
- Игнорирование процесса
- Прочее

Ошибки “тридевятого царства”

Игнорирование форматирования

```
fun lengthInMeters(sagenes: Int, arshins: Int, vershoks: Int): Double {
    val s = sagenes * 48 * 0.04445
 val a = arshins * 16 * 0.04445
    val v = vershoks * 0.04445
    return s + a + v}
```

```
fun rookOrBishopThreatens(kingX: Int, kingY: Int,
                          rookX: Int, rookY: Int,
                        bishopX: Int, bishopY: Int): Int {
    if ((kingX==rookX||kingY==rookY)&&
      (Math.abs(kingX-bishopX)==Math.abs(kingY-bishopY)))
        return 3
    else if ((kingX==rookX||kingY==rookY)&&
      (Math.abs(kingX-bishopX)!=Math.abs(kingY-bishopY)))
        return 1
    else if (Math.abs(kingX-bishopX)==Math.abs(kingY-bishopY)&&
      (kingX!=rookX&&kingY!=rookY))
        return 2
    else return 0
}
```

Игнорирование форматирования

```
fun whichRookThreatens(kingX: Int, kingY: Int,
                       rookX1: Int, rookY1: Int,
                       rookX2: Int, rookY2: Int): Int {
   val rook1: Boolean = kingX == rookX1 || kingY == rookY1
   val rook2: Boolean = kingX == rookX2 || kingY == rookY2
   if (!rook1 && !rook2) return 0
       else if (rook1 && rook2) return 3
          else if (rook1) return 1
               return 2
}
```

Плохие имена переменных

```
fun convertToString(n: Int, base: Int): String {
    var list: MutableList<Int> = convert(n, base).toMutableList()
    var list2 = mutableListOf<Char>()
    for (i in 0..list.size - 1) {
        when {
            list[i] <= 9 -> list2.add('0' + list[i])
            else -> list2.add('a' + list[i] - 10)
        }
    }
    var str = list2.joinToString("")
    return str
}
```

Лишние скобки

```
fun triangleKind(a: Double, b: Double, c: Double): Int {
    val maxSide = max(max(a, b), c)
    val minSide = min(min(a, b), c)
    val anotherSide = (a + b + c) - (minSide + maxSide)
    return when {
        maxSide > (minSide + anotherSide) -> -1
        maxSide == sqrt((minSide * minSide + anotherSide * anotherSide)) -> 1
        maxSide < sqrt((minSide * minSide + anotherSide * anotherSide)) -> 0
        else -> 2
    }
}
```

Длинные строки

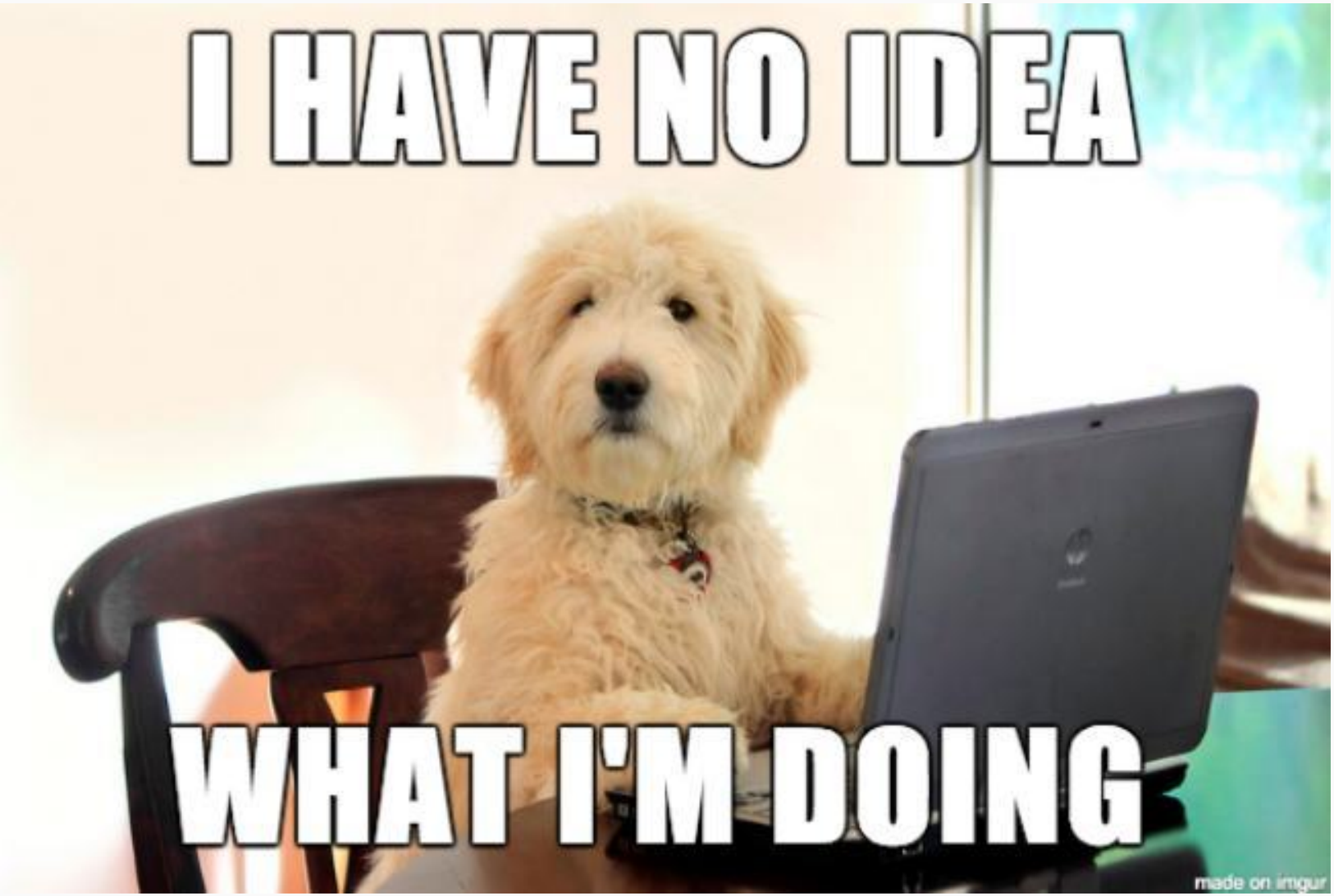
```
fun crossPoint(other: Line): Point {
    val sin1 = sin(angle)
    val cos1 = cos(angle)
    val sin2 = sin(other.angle)
    val cos2 = cos(other.angle)
    val crossX = (other.point.x * sin2 * cos1 - point.x * sin1 * cos2 + point.y * cos1
    val crossY = (crossX - other.point.x) * tan(other.angle) + other.point.y
    return Point(crossX, crossY)
}
```

Резюме

"Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live."

~ John Woods

Непонимание языка Kotlin



Ненужное указание типа

```
fun fib(n: Int): Int {
    if (n in 1..2) return 1
    var fib1: Int = 1
    var fib2: Int = 1
    var fin3: Int = 0
    for (i in 3..n){
        fin3 = fib2 + fib1
        fib1 = fib2
        fib2 = fin3
    }
    return fin3
}
```

Разделение объявления и определения переменной

```
fun decimalFromString(str: String, base: Int): Int {
    var list: List<Int>
    list = listOf()
    val str1 = str
    for (i in 0..str1.length - 1)
        if (str1[i] in '0'..'9') list += str1[i].toInt() - 48
        else list += str[i].toInt() - 87
    return decimal(list, base)
}
```

Разделение объявления и определения переменной

```
fun factorize(n: Int): List<Int> {
    var nn = n
    var minDivisor: Int
    var result = listOf<Int>()
    while (nn > 1) {
        minDivisor = minDivisor(nn)
        result += minDivisor
        nn /= minDivisor
    }
    return result
}
```

Полное описание функции

```
fun seconds(hours: Int, minutes: Int, seconds: Int): Int {
    val a = hours * 3600 + minutes * 60 + seconds
    return a
}
```

```
fun queenThreatens(x1: Int, y1: Int, x2: Int, y2: Int): Boolean {
    if (x1 == x2) return true
    if (y1 == y2) return true
    if (abs(x2 - x1) == abs(y2 - y1)) return true
    return false
}
```

Лишние проверки списка на пустоту

```
fun polynom(p: List<Double>, x: Double): Double {
    var px = 0.0
    var i = 0.0
    if (p.isEmpty()) return px
    else {
        for (element in p) {
            px += element * Math.pow(x, i)
            i++
        }
    }
    return px
}
```

Лишние проверки списка на пустоту

```
fun accumulate(list: MutableList<Double>): MutableList<Double> {
    if (list.isEmpty() || list.size == 1) return list
    else {
        for (i in 1..list.size - 1) {
            list[i] += list[i - 1]
        }
        return list
    }
}
```

Неправильный тип цикла

```
fun polynom(p: List<Double>, x: Double): Double {
    var px = 0.0
    var i = 0.0
    if (p.isEmpty()) return px
    else {
        for (element in p) {
            px += element * Math.pow(x, i)
            i++
        }
    }
    return px
}
```

&&/|| vs and/or

```
fun brickPasses(a: Int, b: Int, c: Int, r: Int, s: Int): Boolean {
        if (r >= a and (s >= b or s >= c)) return true
    else if (r >= b and (s >= c or s >= a)) return true
    else if (r >= c and (s >= b or s >= a)) return true
    else return false
}
```

if vs when

```
fun brickPasses(a: Int, b: Int, c: Int, r: Int, s: Int): Boolean {
         if (r >= a and (s >= b or s >= c)) return true
    else if (r >= b and (s >= c or s >= a)) return true
    else if (r >= c and (s >= b or s >= a)) return true
    else return false
}
```

```
fun rookOrBishopThreatens(kingX: Int, kingY: Int,
                          rookX: Int, rookY: Int,
                          bishopX: Int, bishopY: Int): Int {
    if (kingX == rookX || kingY == rookY) {
        if (Math.abs(kingX - bishopX) == Math.abs(kingY - bishopY)) return 3
        else return 1
    }
    if (Math.abs(kingX - bishopX) == Math.abs(kingY - bishopY)) {
        if (kingX == rookX || kingY == rookY) return 3
        else return 2
    } else return 0
}
```

if (bool == true)

```
fun whichRookThreatens(kingX: Int, kingY: Int,
                       rookX1: Int, rookY1: Int,
                       rookX2: Int, rookY2: Int): Int {
    val danger1 = kingX == rookX1 || kingY == rookY1
    val danger2 = kingX == rookX2 || kingY == rookY2
    if (danger1 == true && danger2 == true) return (3)
    else if (danger1 == true) return (1)
    else if (danger2 == true) return (2)
    else return (0)
}
```

if (cond) true else false

```
fun isPalindrome(n: Int): Boolean {
    var a = 0
    var m = n
    while (m > 0) {
        a = a * 10 + m % 10
        m /= 10
    }
    if (a == n) return true else return false
}
```

Лишние вызовы функций

```
fun convertToString(n: Int, base: Int): String {
    val list = convert(n, base).toMutableList()
    var res = listOf<Char>()
    for (i in 0..list.size - 1) {
        if (list[i] >= 10) {
            res += 'a' + (list[i] - 10)
        } else {
            res += '0' + list[i]
        }
    }
    return res.joinToString(separator = "")
}
```

Лишние вызовы функций

```
fun abs(v: List<Double>): Double {
    var number = 0.0
    for (i in 0..v.size - 1) {
        number += sqr(v[i].toString().toDouble())
    }
    return Math.sqrt(number)
}
```

Неправильные вызовы функций

```
fun decimalFromString(str: String, base: Int): Int {
    var res = listOf<Int>()
    for (i in 0..str.length - 1) {
        if (str[i].toInt() <= '9'.hashCode()) {
            res += str[i].hashCode() - '0'.hashCode()
        } else {
            res += str[i].hashCode() - 'a'.hashCode() + 10
        }
    }
    return decimal(res, base)
}
```

very + “long” + s.tring + “$concatenation”

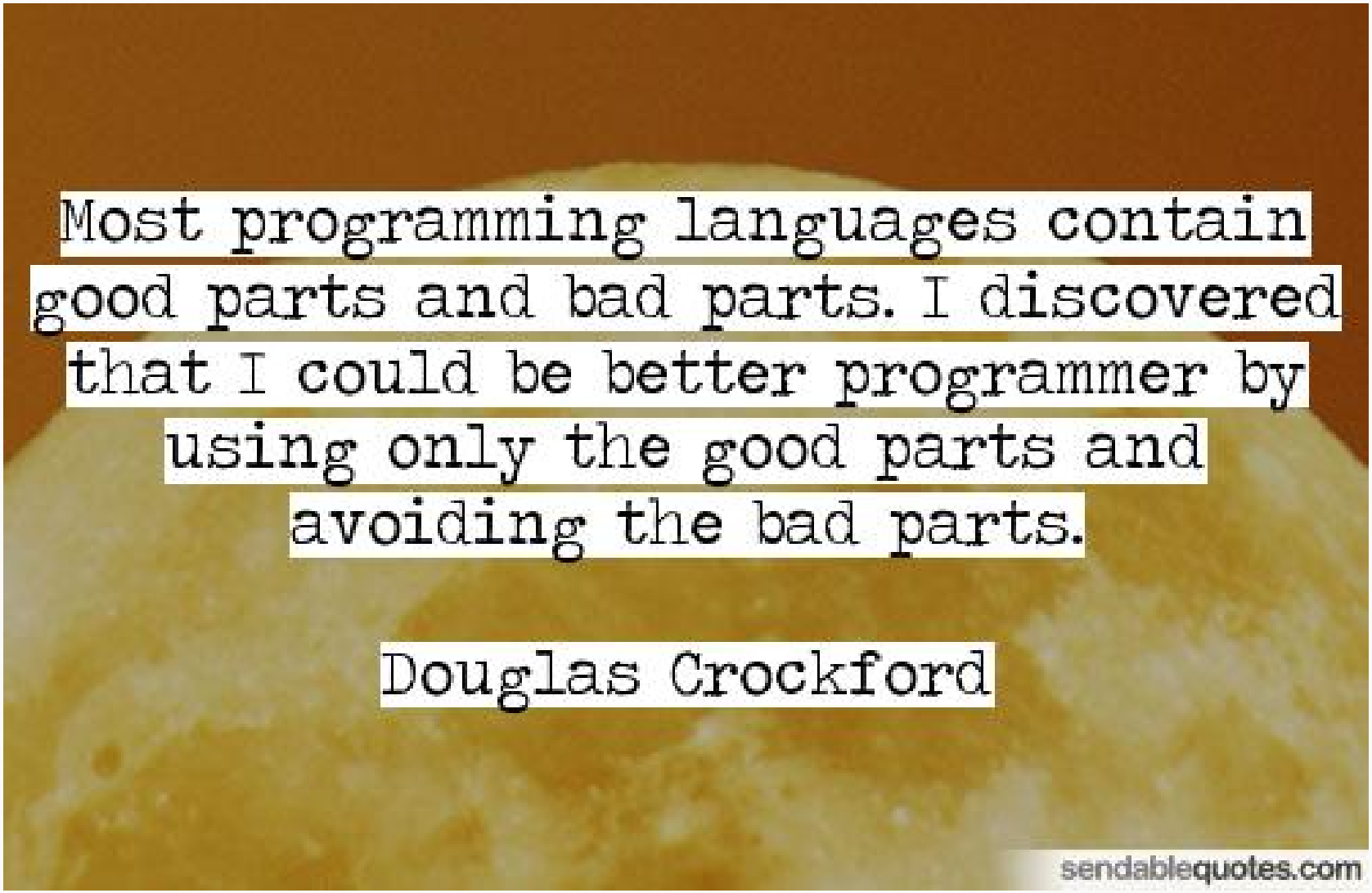
```
override fun toString(): String {
    var str = ""
    for (i in 0..height - 1) {
        for (j in 0..width - 1) {
            str += this[i,j]
            str += "\t"
        }
        str += "\n"
    }
    return str
}
```

Магические константы

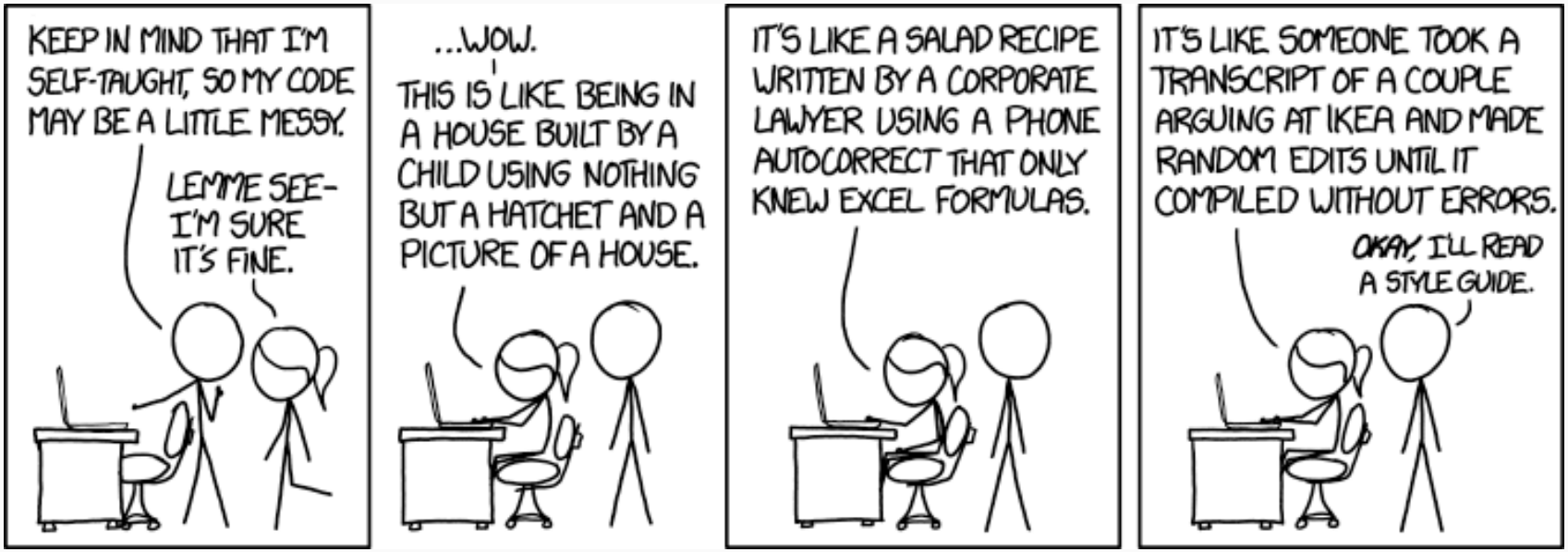
```
fun decimalFromString(str: String, base: Int): Int {
    var list = listOf<Int>()
    for (i in 0..str.length - 1)
        if (str[i] in '0'..'9') list += str[i].toInt() - 48
        else list += str[i].toInt() - 87
    return decimal(list, base)
}
```

```
fun roman(n: Int): String {
    val listRoman = listOf("I", "IV", "V", "IX", "X", "XL",
        "L", "XC", "C", "CD", "D", "CM", "M")
    val listArabic = listOf(1, 4, 5, 9, 10, 40,
        50, 90, 100, 400, 500, 900, 1000)
    var number = n
    var res = ""
    if (number <= 0) return ""
    while (number > 0) {
        for (i in listArabic.size - 1 downTo 0) {
            if (number - listArabic[i] >= 0) {
                res += listRoman[i]
                number -= listArabic[i]
                break
            }
        }
    }
    return res
}
```

Резюме


Most programming languages contain good parts and bad parts. I discovered that I could be better programmer by using only the good parts and avoiding the bad parts.
Douglas Crockford
sendablequotes.com

Подход “костыли и палки”



Дублирование кода

```
fun revert(n: Int): Int {
    var a = 0
    var m = n
    while (m > 0) {
        a = a * 10 + m % 10
        m /= 10
    }
    return a
}
```

Дублирование кода

```
fun isPalindrome(n: Int): Boolean {
    var a = 0
    var m = n
    while (m > 0) {
        a = a * 10 + m % 10
        m /= 10
    }
    if (a == n) return true else return false
}
```

Дублирование кода

```
fun fibSequenceDigit(n: Int): Int {
    fun count(n: Int): Int {
        var i = 0
        var nn = n
        while (nn > 0) {
            nn = nn / 10
            i++
        }
        return i
    }

    var i = 0
    var nn = 0
    ...
```

“Лишь бы тесты прошли...”

```
fun fibSequenceDigit(n: Int): Int {
    var count = 1
    var fib1 = 0
    var fib2 = 0
    var currentFib = 1

    while (count < n) {
        fib1 = fib2
        fib2 = currentFib
        currentFib = fib1 + fib2
        count += digitNumber(currentFib)
    }
    return currentFib / pow(10, count - n) % 10
}
```

“Лишь бы тесты прошли...”

```
fun sin(x: Double, eps: Double): Double {
    var n = 0
    var sin = x
    var part = Double.POSITIVE_INFINITY
    while (Math.abs(part) * 1000 > Math.abs(eps)) {
        n++
        part = Math.pow(x, n * 2.0 + 1) / factorial(n * 2 + 1)
        if (n % 2 == 0) sin += part
        else sin -= part
    }
    return sin
}
```

Don’t stop me now!

```
fun isCoPrime(m: Int, n: Int): Boolean {
    var k = 1
    for (i in 2..m) {
        if (m % i == 0 && n % i == 0)
            k = 0
    }
    return k != 0
}
```

Игнорирование сути задачи

```
fun squareBetweenExists(m: Int, n: Int): Boolean {
    for (i in m..n) {
        if (Math.sqrt(i.toDouble()) ==
            Math.sqrt(i.toDouble()).toInt().toDouble()) {
            return true
        }
    }
    return false
}
```

Игнорирование сути задачи

```
fun circleByThreePoints(a: Point, b: Point, c: Point): Circle {
    val distanceOne = a.distance(b)
    val distanceTwo = b.distance(c)
    val distanceThree = a.distance(c)
    val distanceSum = distanceOne + distanceTwo + distanceThree
    val distanceMax = max(distanceOne, max(distanceTwo, distanceThree))
    if (distanceMax >= distanceSum - distanceMax)
        throw IllegalArgumentException()
    val center = bisectorByPoints(a, b).crossPoint(bisectorByPoints(b, c))
    val radius = center.distance(a)
    return Circle(center, radius)
}
```

“Когда у тебя в руках молоток…”

```
fun russian(n: Int): String {
    ...
    for (i in 0..2) {
        when (i) {
            0 -> {
                ...
            }
            1 -> {
                ...
            }
            2 -> {
                ...
            }
        }
    }
    ...
}
```

Резюме

I'm not a real programmer. I throw together things until it works then I move on. The real programmers will say Yeah it works but you're leaking memory everywhere. Perhaps we should fix that. I'll just restart Apache every 10 requests.

(Rasmus Lerdorf)

izquotes.com

Игнорирование процесса

Не проходят локальные тесты

Succeeded:

- [Trivial] lesson1.task1/seconds
- [Trivial] lesson1.task1/lengthInMeters
- [Example] lesson1.task1/sqRoot
- [Easy] lesson1.task1/thirdDigit
- [Example] lesson1.task1/sqr
- [Trivial] lesson1.task1/angleInRadian
- [Example] lesson1.task1/discriminant
- [Trivial] lesson1.task1/trackLength
- [Easy] lesson1.task1/numberRevert
- [Easy] lesson1.task1/accountInThreeYears
- [Example] lesson1.task1/quadraticRootProduct

Failed:

- [Easy] lesson1.task1/travelMinutes
  - org.opentest4j.AssertionFailedError : expected: <216> but was: <-216>
- [Easy] lesson1.task1/travelMinutes
  - Expected: 179
  - Actual: -179
  - Inputs:
    - hoursDepart -> 1

Projects

None yet

Labels

bad style

waiting for

Milestone

No milestone

Assignees

ice-phoe

4 participant

Notifications

You're receivi
you were assi

Проект не собирается

```
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:compile (default-compile) @ kfirst ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- kotlin-maven-plugin:1.0.3:test-compile (test-compile) @ kfirst ---
[INFO] Kotlin Compiler version 1.0.3
[INFO] Compiling Kotlin sources from [/var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test]
[INFO] Module name is kfirst
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson3/task1/RandomTests.kt: (198, 29) Too
many arguments for @Test @Tag public final fun digitNumber(): Unit defined in lesson3.task1.RandomTests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson3/task1/Tests.kt: (67, 38) Too many
arguments for @Test @Tag public final fun digitNumber(): Unit defined in lesson3.task1.Tests
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 20.753 s
[INFO] Finished at: 2016-09-29T01:03:02+03:00
[INFO] Final Memory: 94M/467M
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.jetbrains.kotlin:kotlin-maven-plugin:1.0.3:test-compile (test-compile) on
project kfirst: Compilation error. See log for more details -> [Help 1]
```

Проект не собирается

```
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/RandomTests.kt: (559, 109) Too many arguments for @Test @Tag public final fun plusMinus(): Unit defined in lesson5.task1.RandomTests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (57, 40) Too many arguments for @Test @Tag public final fun bestLongJump(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (58, 39) Too many arguments for @Test @Tag public final fun bestLongJump(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (59, 40) Too many arguments for @Test @Tag public final fun bestLongJump(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (60, 39) Too many arguments for @Test @Tag public final fun bestLongJump(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (67, 40) Too many arguments for @Test @Tag public final fun bestHighJump(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (68, 39) Too many arguments for @Test @Tag public final fun bestHighJump(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (69, 40) Too many arguments for @Test @Tag public final fun bestHighJump(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (75, 35) Too many arguments for @Test @Tag public final fun plusMinus(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (76, 35) Too many arguments for @Test @Tag public final fun plusMinus(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (77, 35) Too many arguments for @Test @Tag public final fun plusMinus(): Unit defined in lesson5.task1.Tests
[ERROR] /var/lib/jenkins/workspace/kotlin-as-first.fall-2016/test/lesson5/task1/Tests.kt: (78, 36) Too many arguments for @Test @Tag public final fun plusMinus(): Unit defined in lesson5.task1.Tests
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
```

None yet

Labels

bad sty

tests fa

waiting

Mileston

No milest

Assigne

mglu

3 particip

Notificat

You're re

you're su

Lock

Отсутствие проверки изменений

```
87  98                <id>compile</id>
88     -              <phase>process-sources</phase>
       99  +          <phase>compile</phase>
89  100               <goals>
90  101                   <goal>compile</goal>
       102 +              <goal>js</goal>
91  103               </goals>
92  104           </execution>
93  105           <execution>
94  106               <id>test-compile</id>
95     -              <phase>process-test-sources</phase>
       107 +          <phase>test-compile</phase>
96  108               <goals>
97  109                   <goal>test-compile</goal>
       110 +              <goal>test-js</goal>
98  111               </goals>
99  112           </execution>
100 113       </executions>
```

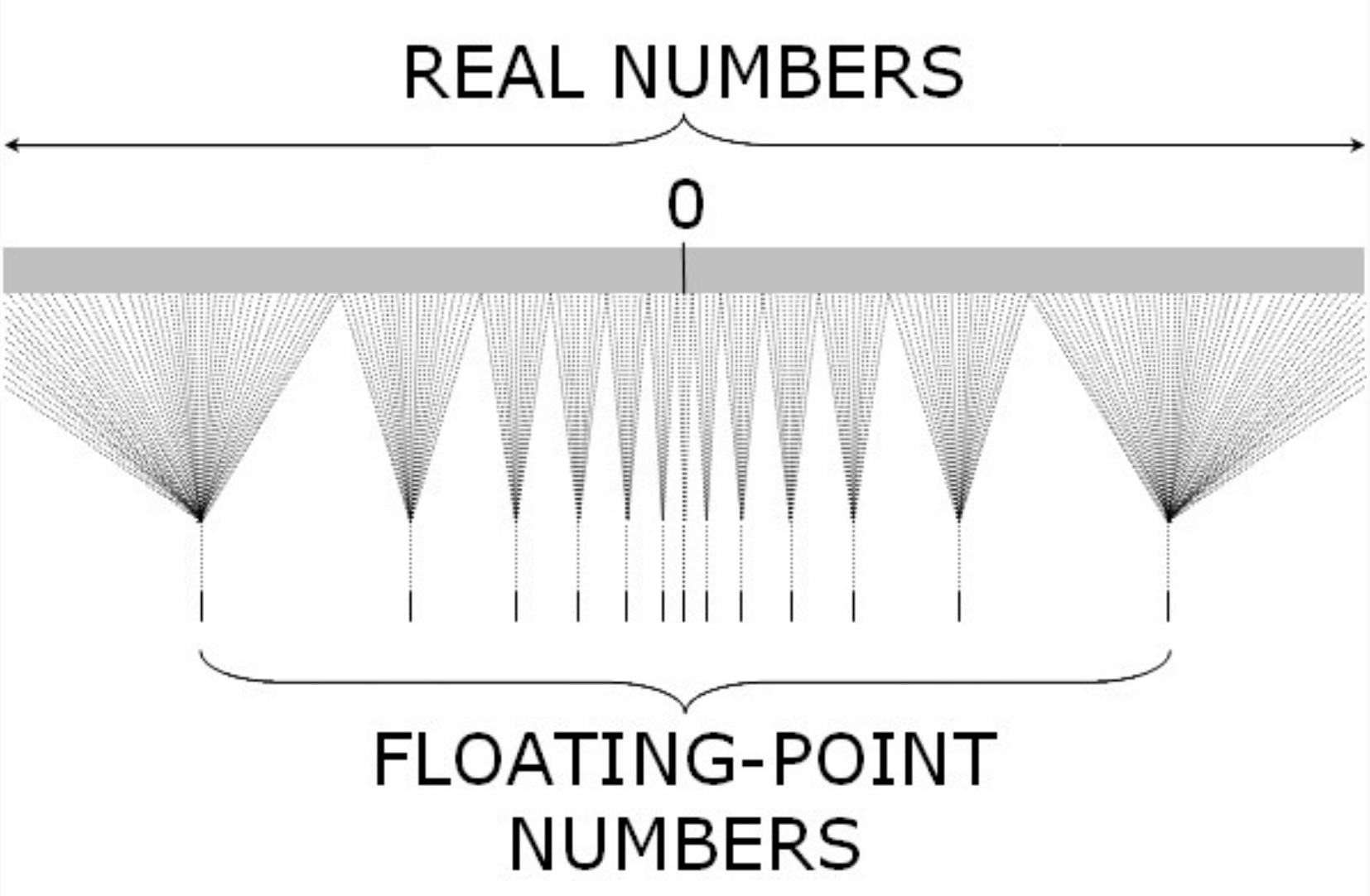
Отсутствие проверки изменений

```
@file:Suppress("UNUSED_PARAMETER")

package lesson3.task1

import com.sun.xml.internal.ws.runtime.config.TubelineFeatureReader
import java.lang.Math.*
import kotlin.concurrent.timer

/**
 * Пример
 *
 ...
```

Шум в сообщениях / комментариях / ...

TODO

Проблемы с плавающей запятой



Double/Float вместо Long/Int

```
fun revert(n: Int): Int {
    var res = 0
    var count = digitNumber(n)
    var t = 0
    while (count != 0) {
        res += n
          / pow(10.0, count * 1.0 - 1).toInt()
          % 10
          * pow(10.0, t * 1.0).toInt()
        count--
        t++
    }
    return res
}
```

Сравнение Double на равенство

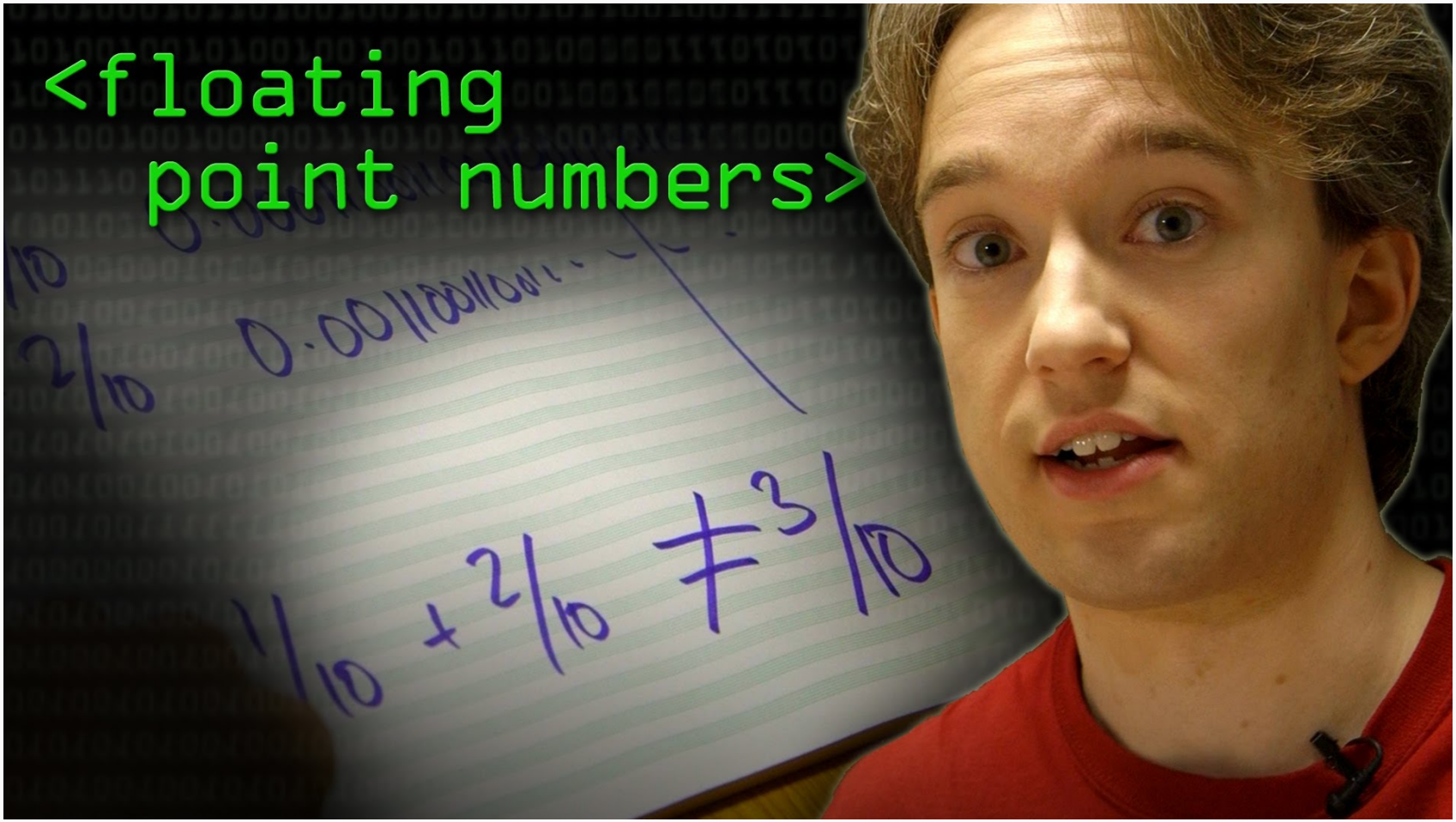
```
fun squareBetweenExists(m: Int, n: Int): Boolean {
    for (i in m..n) {
        if (Math.sqrt(i.toDouble()) % 1 == 0.0)
            return true
    }
    return false
}
```

Huh?

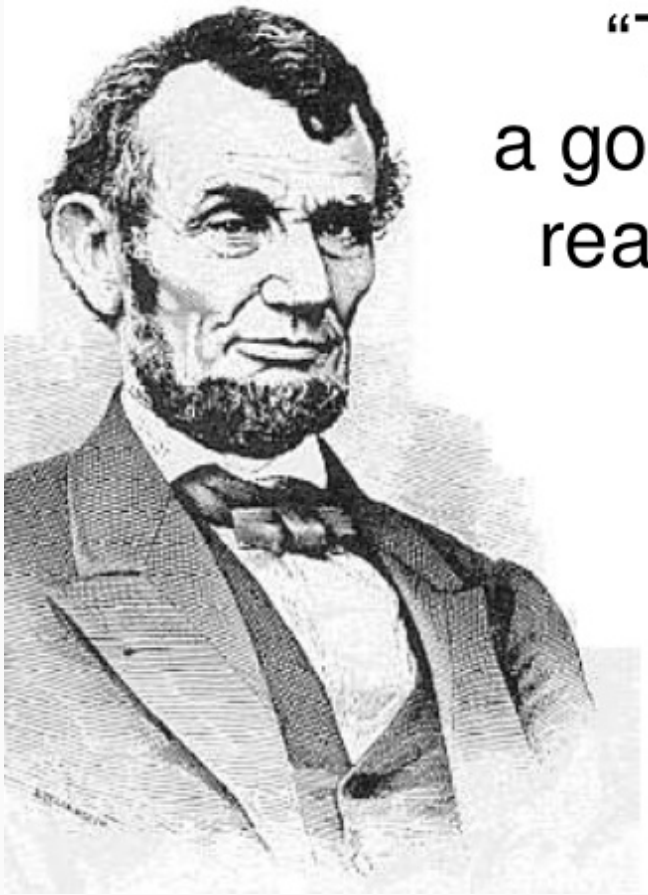
```
fun foo(a: Double): Double {
    val b = 10 * a - 10
    val c = a - 0.1 * b
    return c
}
```

- foo(5)
- foo(100)
- foo(2987154209766221.0)
- foo(2987154209766221.6)
- foo(719525522284533115.3)

Резюме



Что же делать?



“The key to being
a good developer is to be
really good at Google.”

- Abraham Lincoln

Что же делать?

To Do:

1. Build stuff

2.