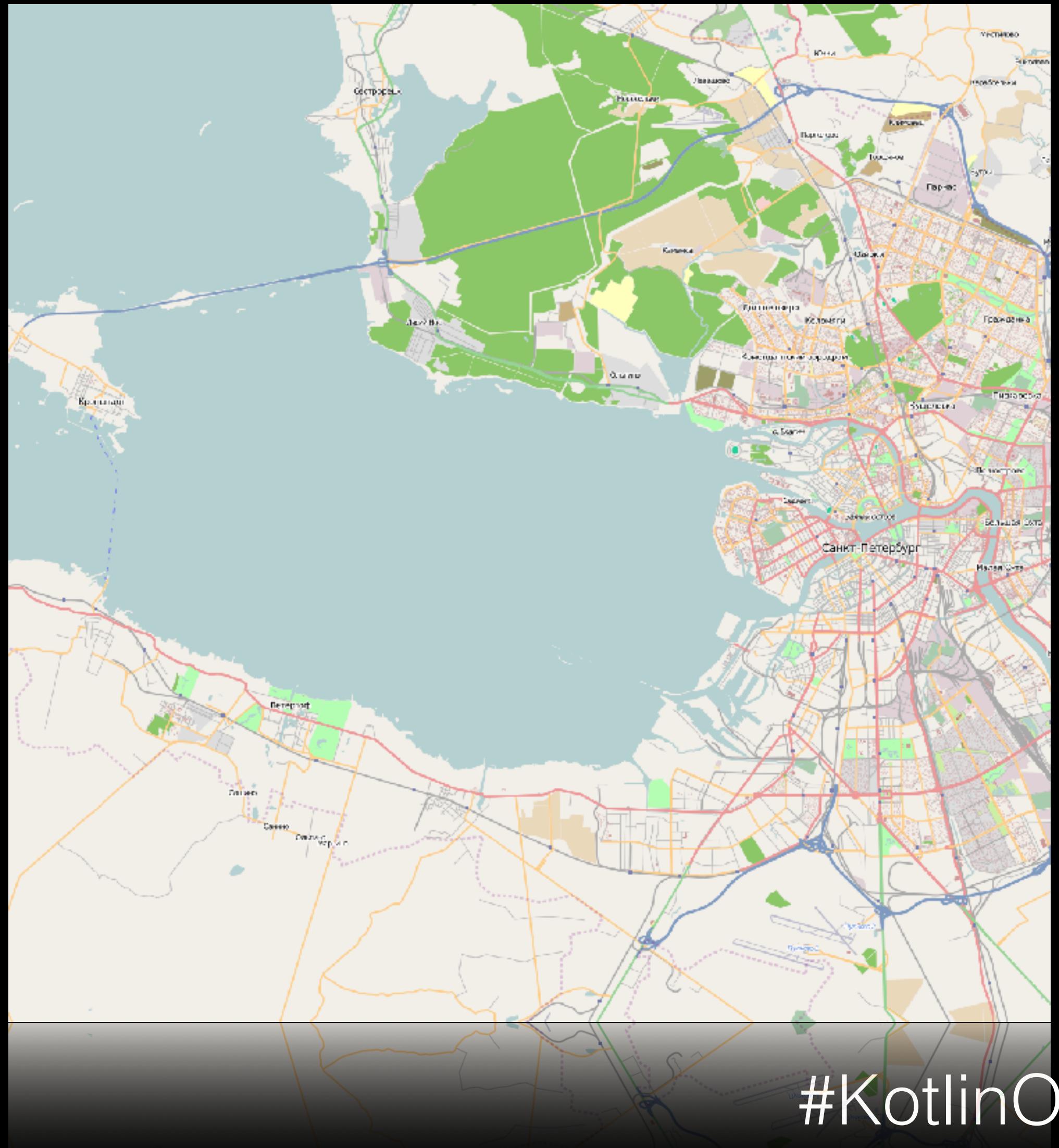


Why Kotlin?

Jeff Sheets - @sheetsj
OJUG
October 2016

Kotlin?

- Statically-typed JVM/Android language
- Created by JetBrains in 2011
- Has a “Static” Groovy feel to it
- v1.0 released 02/2016



#KotlinOJUG



#KotlinOJUG

Gradle Moving to Kotlin

Big Deal? Or Not?

 **Informationized** @BlueAquilae · May 20
@danveloper Not a fatal blow but sure the bright future of @gradle put a little @ApacheGroovy on the sideway and @kotlin in front

 **Guillaume Laforge** @glaforge

@blueaquilae @danveloper @gradle and it's sad as there's nothing Kotlin can do that Groovy can't

 **Cédric Champeau** @CedricChampeau 

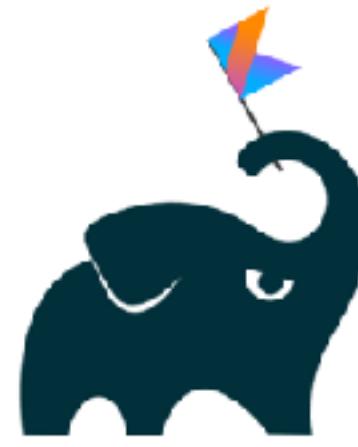
#Kotlin meets #gradle bit.ly/25aWhUi And for those who worry, no, we're not dropping #groovylang support.

RETWEETS LIKES
18 **17**

2:25 AM - 18 May 2016

#KotlinOJUG

Gradle + Kotlin?



- Performance
 - Groovy use of Exceptions for control flow
- IDE Support
 - Pivotal left Groovy Eclipse - nobody picked it up yet
 - Kotlin static easy to build completion into IDE
 - Groovy could be done - but needs someone to develop it

Class with a primary
constructor and a member
function

```
— class Greeter(val name: String) {  
    fun greet() {  
        println("Hello, $name")  
    }  
}
```

```
fun main(args: Array<String>) {  
    Greeter(args[0]).greet()  
}
```

Note that there's no new
keyword used to create an
object

<https://kotlinlang.org/>

#KotlinOJUG



Mike Plummer
@plummer_mike



Following

Some of the highlights of [@kotlin](#) when coming from a Java background.

Object Partners, Inc @objectpartners

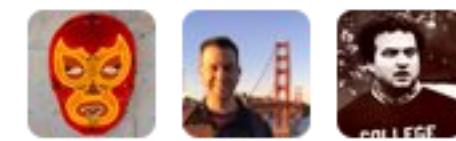
New Blog Post: An Introduction to Kotlin ow.ly/3bEnNL

RETWEETS

2

LIKE

1



12:40 PM - 23 Feb 2016



2



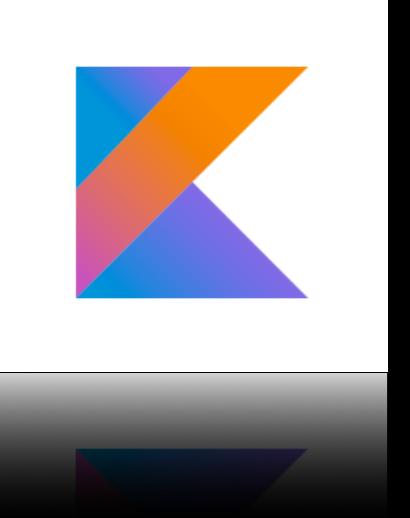
1

...

https://twitter.com/plummer_mike/status/702201485812219904

<https://objectpartners.com/2016/02/23/an-introduction-to-kotlin/>

#KotlinOJUG



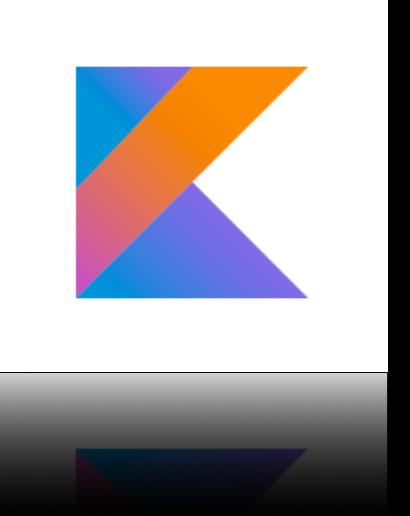
Immutability Checks

- val vs var

A screenshot of an IDE showing a code editor and a message bar. The code editor contains the following Kotlin code:

```
fun varVersusVal() {  
    var four = 4  
    four = 5 // OK  
  
    val five = 5  
    five = 6 // ERROR
```

The line "five = 6" is highlighted with a red underline, indicating an error. A message box at the bottom displays the error message: "Val cannot be reassigned".



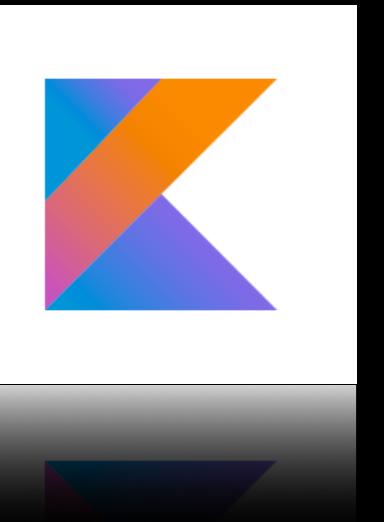
Null Safety

- Compile Time Null Checks

```
fun nullChecks() {  
    var three: Int? = 3  
    three = null //OK  
  
    var four: Int = 4  
    four = null //ERROR  
}
```

Null can not be a value of a non-null type Int

- Also null-safe syntax like:
 `println user?.getMiddleName()`

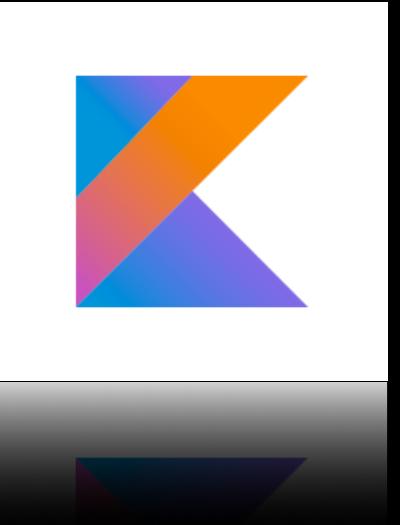


Strings

- String Variable Replacement - and Multi-line Strings

```
fun strings() {  
    val year = 2016  
  
    println("Year is $year")  
  
    println("""  
        The Year is $year  
        and it will soon be winter  
    """)  
}
```

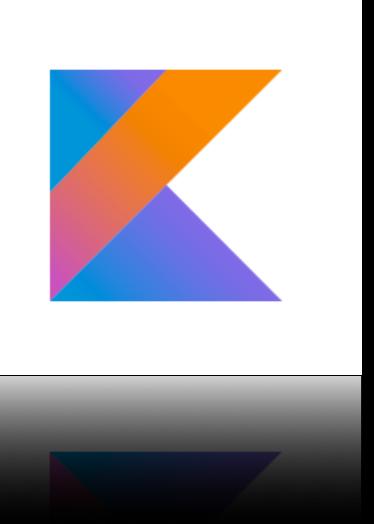
Object Properties



- Data Classes - and Simple Properties

```
/**  
 * https://kotlinlang.org/docs/reference/data-classes.html  
 *  
 * data classes get  
 *   - equals/hashCode  
 *   - toString  
 *   - copy()  
 *   - destructuring componentN() functions  
 *     - https://kotlinlang.org/docs/reference/multi-declarations.html  
 *     - val (value, left, right) = node  
 */  
data class Node(var value: String,  
               val left: Node? = null,  
               val right: Node? = null) {  
    var visited: Boolean = false  
}
```

Spek Tests



- JS Jasmine-like syntax - describe/it
- Used with an assertion library (JUnit, or Expekt, ...)

```
describe("Kotlin Binary Search tests") {  
    val bs = KotlinBinarySearch()  
  
    it("should be correct for breadth first search") {  
        val result = bs.bfs(tree())  
        println(result)  
  
        expect(result).to.equal mutableListOf("1", "2", "3", "4", "5", "6", "7", "8", "9")  
    }  
}
```

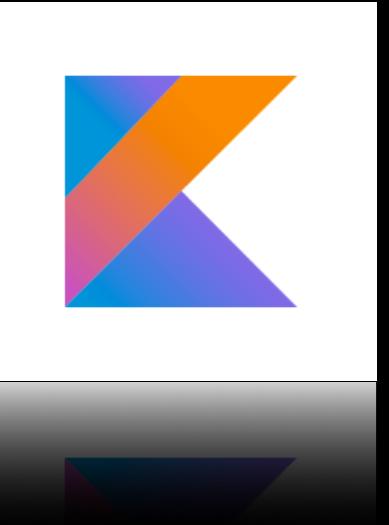
Spek - Dynamic Tests



- Do Spock-like “where” assertions

```
describe("binarySearch") {  
    val primes = listOf(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,  
        53, 59, 61, 67, 71, 73, 79, 83, 89)  
  
    it("should work for 67") {  
        expect(bs.binarySearch(67, primes)).to.equal(18)  
    }  
  
    //Nice trick – https://realm.io/news/kau-rob-fletcher-kotlin-testing/  
    for ((num, primeLocation) in mapOf(8 to -1, 2 to 0, 89 to 23, 97 to -1)) {  
        describe("binarySearch of $num") {  
            val result = bs.binarySearch(num, primes)  
            it("has location $primeLocation in array") {  
                expect(result).to.equal(primeLocation)  
            }  
        }  
    }  
}
```

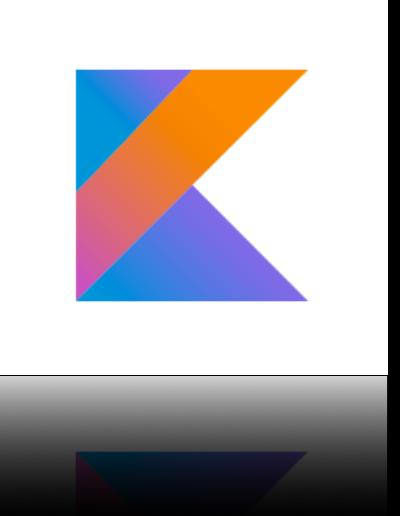
Initial Gotchas



- List initializers - `listOf(1, 2, 3)` vs `mutableListOf(1, 2, 3)` vs `[1,2,3]`
- Integer vs Int
- Return type of ‘Unit’
 - vs type of “Nothing”



Other things



- Operator Overloading (.plus() +)
- Extension Functions (add methods to existing classes)
- Delegates (for composition without inheritance)
- Destructuring - val (first, last) = name
- No checked exceptions
- == vs === (like == vs .is() in groovy)



Jeff Sheets - @sheetsj

OPI Tech Talk - Thursday - 10/20 - Mike Plummer - "Getting Started with Angular 2"

OPI Open Office Co-Working - Tuesday - 10/25

<https://objectpartners.com/events/>