Introduction to Kotlin

Tobias Schürg - 6.12.2017

Kotlin Intro

The new fancy, opensource programming language

- Runs on the Java Virtual Machine
    - Can be compiled to JavaScript
- Statically typed
- Interoperable with Java code
- Developed by JetBrains (IntelliJ)
    - First appeared 2011 (already developed for a year)
    - Kotlin v1.0 was released on February 15, 2016
    - Goals: compile as quickly as Java, more compact, more cocise, easier to read
- Kotlin Native
    - Windows (x86_64 only at the moment), Linux (x86_64, arm32, MIPS, MIPS little endian), MacOS (x86_64), iOS (arm64 only), Android (arm32 and arm64)

Adoption (source https://en.wikipedia.org/wiki/Kotlin_(programming_language)#Adoption)

- According to the Kotlin website
  - Prezi is using Kotlin in the backend.
  - DripStat has done a writeup of their experience with Kotlin.[28]
- According to Jetbrains blog Kotlin is used among others by
  - Amazon Web Services
  - Pinterest
  - Coursera
  - Netflix
  - Uber
  - Square
  - Trello
  - Basecampand
- According to Google, Kotlin has already been adopted by several major developers
  - Expedia, Flipboard, Pinterest, Square, and others—for their Android production apps

Corda, a distributed ledger developed by a consortium of well-known banks (such as Goldman Sachs, Wells Fargo, J.P. Morgan, Deutsche Bank, UBS, HSBC, BNP Paribas, Société Générale), has over 90% Kotlin in its codebase.

Kotlin - Basics

Defining Variables

Control Flow

Loops and ranges

Smart casts

Interoperability with Java

Null Safety

Avoiding the billion dollar mistake

- NPE will only occur if
    - Explicitely thrown
    - The `!!`-operator is used
    - External Java code caused it
- Safe calls

```
b?.length
bob?.department?.head?.name
```

- Elvis Operator

```
val l = b?.length ?: -1
```

- !! Operator

```
val l = b!!.length
```

Classes

- Instance creation without “new” keyword
- Data Classes
- Object Classes (Singletons)
- Sealed Classes
- Default Values
- Companion Objects

There is a lot of *fun* in Kotlin

- Default arguments
- Named arguments
- Single expression functions
- Inline functions
- Infix functions

- Extension functions

Higher-Order Functions and Lambdas

- Functions which can take one or more functions as arguments

```
fun <T, R> List<T>.map(transform: (T) -> R): List<R> {
   val result = arrayListOf<R>()
   for (item in this) result.add(transform(item))
   return result
}
```

- “Alternative to interfaces and BiFunctions”
- Implicit name of single parameter: it

Kotlin – Advanced

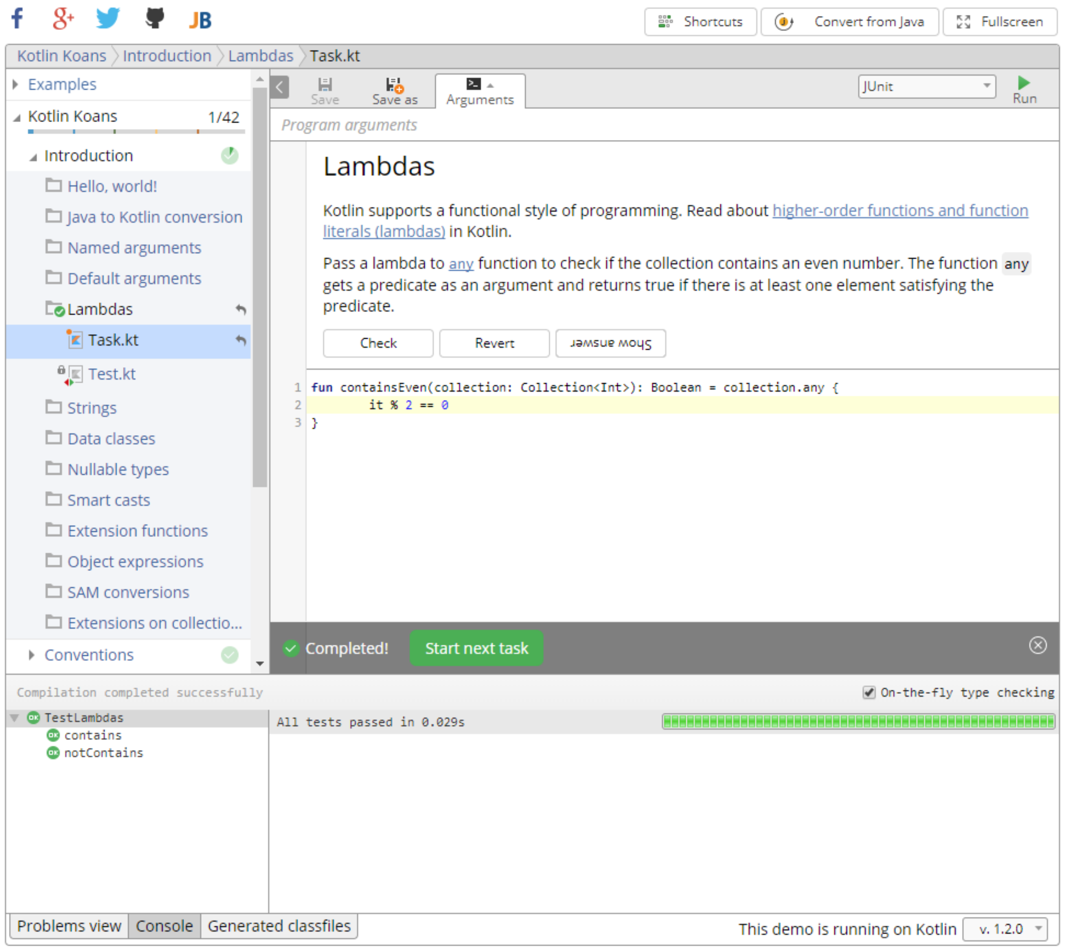
Coroutines

- Until now: callbacks, async tasks / threads, rx
- Motivation: async / await
    - compile "org.jetbrains.kotlinx:kotlinx-coroutines-core:0.19.2"
    - kotlin { experimental { coroutines 'enable' } }

Resources

Kotlin Koans

https://try.kotlinlang.org



Lambdas

Kotlin supports a functional style of programming. Read about higher-order functions and function literals (lambdas) in Kotlin.

Pass a lambda to any function to check if the collection contains an even number. The function `any` gets a predicate as an argument and returns true if there is at least one element satisfying the predicate.

```kotlin
fun containsEven(collection: Collection<Int>): Boolean = collection.any {
        it % 2 == 0
}
```

```
Compilation completed successfully
TestLambdas
  contains
  notContains
All tests passed in 0.029s
```

☑ On-the-fly type checking

This demo is running on Kotlin v. 1.2.0

Useful Resources

- Starting:
  - Kotlin refernce:
    https://kotlinlang.org/docs/reference/
  - Kotlin Koans:
    Onlince: https://try.kotlinlang.org GitHub:https://github.com/Kotlin/kotlin-koans
  - Introduction to Kotlin (Google Developer Days '17):
    https://www.youtube.com/watch?v=YbF8Q8LxAJs
- Coroutines
  - https://kotlinlang.org/docs/reference/coroutines.html
  - Svetlana Isakova - Kotlin coroutines (44 min)
    https://www.youtube.com/watch?v=nugOMl29K3k
- Additional
  - Danny Preussler - The power of Kotlin for your tests (40 min):
    https://www.youtube.com/watch?v=V3e9KRfGqsQ