

# KOTLIN TRAINING

# TODAY'S AGENDA

- ▣ Introduction to Kotlin
- ▣ Java Vs Kotlin
- ▣ Kotlin Basics – Basic Syntax, Idioms, Kotlin By Example, Coding Conventions
- ▣ Kotlin Concepts - Basic types, Type checks and casts
- ▣ Kotlin Concepts - Control flow - Conditions and loops, Returns and jumps Exceptions
- ▣ Kotlin Concepts – Classes & Objects – Classes, Inheritance, Properties

# Introduction to Kotlin

The next big thing in the Java world

# What is Kotlin?

- ▣ A (fairly) new statically-typed language from JetBrains (creators of IntelliJ Idea, ReSharper, PyCharm and other IDEs and IDE extensions)
- ▣ Tries to fix many of Java's shortcomings
- ▣ Compiles to JVM bytecode, JavaScript (!) and Kotlin native (no VM)
- ▣ Created with focus on Java interoperability – Kotlin and Java classes can be used together in a project (but compilation is much faster than Scala's)

# What is Kotlin?

- ▣ Works seamlessly with IntelliJ Idea, just a bit less seamlessly with Eclipse
- ▣ The Java interoperability is real – even complex applications, relying on annotation processing
- ▣ It's gaining more and more traction and is being adopted increasingly often, especially in the recent year
- ▣ Has very good support from JetBrains, an established company, who use Kotlin to develop their own products

# Google and Android

At Google I/O 2017, Google announces that Kotlin would receive first-class support for Android development



# What does Kotlin feature?

- ▣ Null safety (drastically limiting the number of NPEs);
- ▣ If, try-catch, when expressions;
- ▣ Extension functions;
- ▣ Var and val keywords;
- ▣ Inline functions;
- ▣ Named function parameters;
- ▣ Multi-value return functions;
- ▣ Semi-colons are optional

# More things...

- ▣ Functional programming constructs (like higher order functions)
- ▣ Smart casting
- ▣ Async/await
- ▣ **Data classes**
- ▣ Properties (C#- like)
- ▣ No checked exceptions
- ▣ Operator overloading
- ▣ ... this is just scratching the surface



# Let's code something!

# Conclusion

- ▣ Kotlin is low-cost: can be developed in the same IDE, in the same project as Java code
- ▣ Did I mention first-class Spring support?
- ▣ It's on track to become one of the leading JVM languages
- ▣ It's best suited to work with IntelliJ idea; it can be adapter to Eclipse, albeit with some shortcomings
- ▣ It's cool; allows bragging to friends about using the latest technologies