JAVA 19 VS KOTLIN

ALEJANDRO GÓMEZ

Globant

@aldajo92

Globant

ALEJANDRO GÓMEZ

Sr Android Dev

Client: Disney park apps

Globant

PREVIOUSLY

Globant

Bytecode

Why Learn it?

- Full understanding from top to bottom
- Bytecode generation is fun and easy

May need to read bytecode someday. (pssssss is not read zeros and ones)

- Many libraries generates bytecode.

Globant

Bytecode

- `javac`
- `kotlinc`
- `javap`
- `javap -c <classfile>`
- `javap -p <classfile>`
- `javap -c -verbose <classfile>`

Globant

In terminal

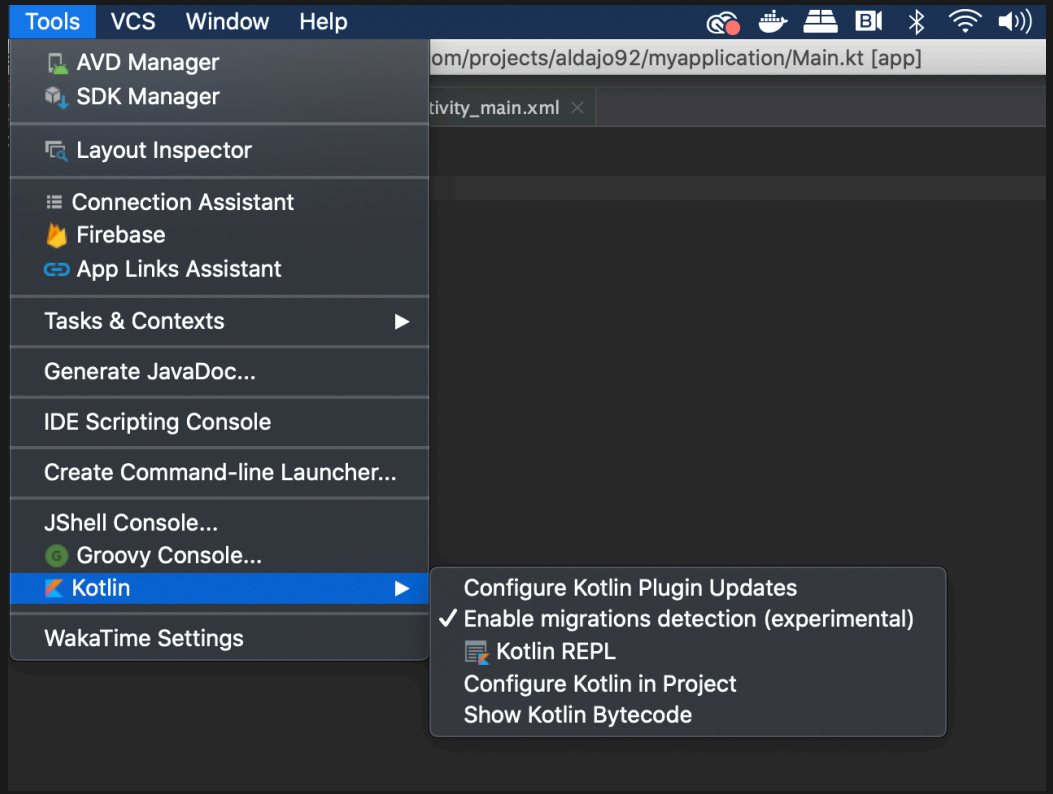
MainJava.java

```
$ javac MainJava.java
$ javap -c MainJava
$ javap -c -verbose MainJava
```

MainKotlin.kt

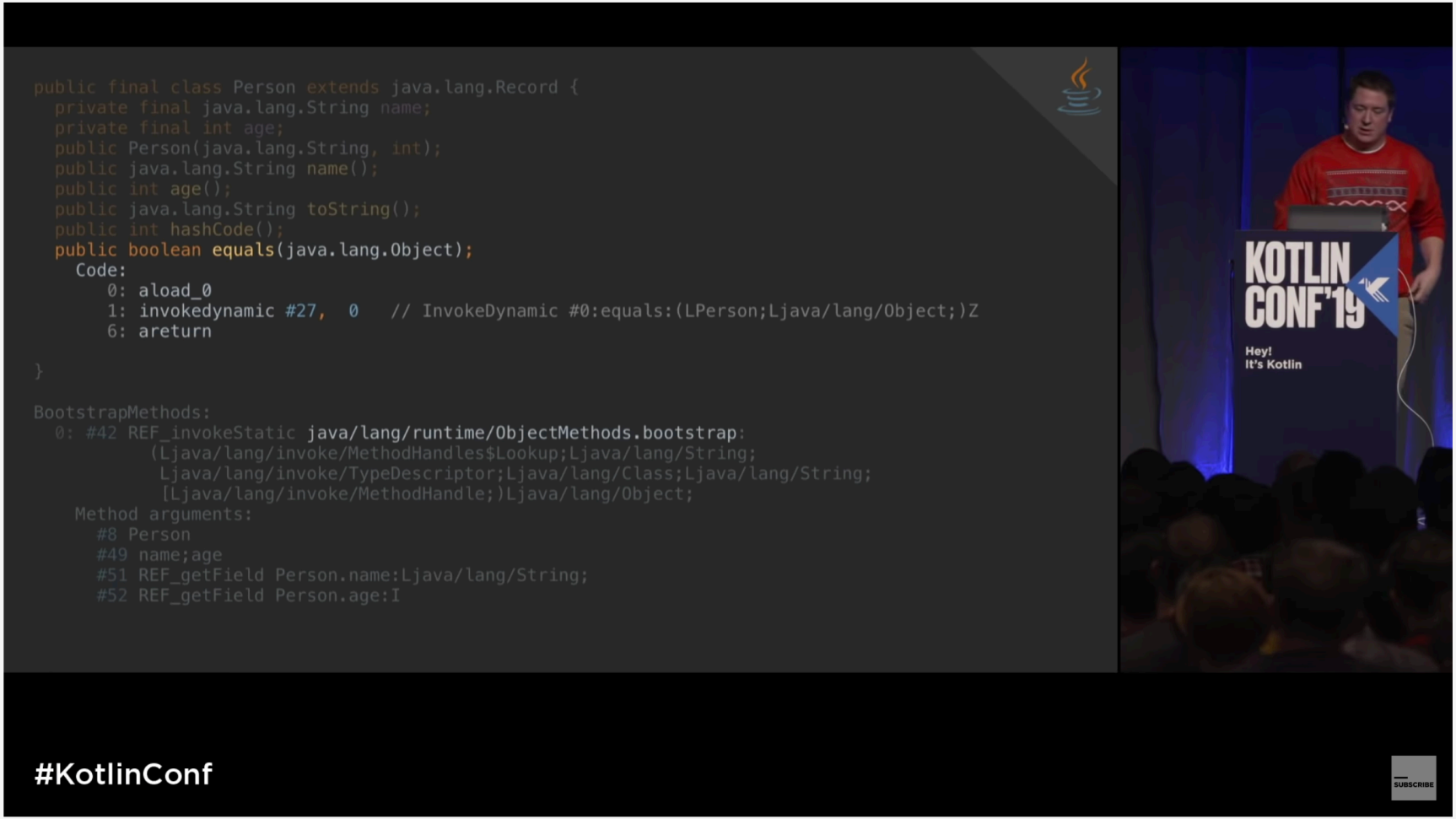
```
$ kotlinc MainKotlin.kt
$ javap -c MainKotlin
$ javap -c -verbose MainKotlin
```

Globant


Tools
VCS
Window
Help
AVD Manager
SDK Manager
Layout Inspector
Connection Assistant
Firebase
App Links Assistant
Tasks & Contexts
Generate JavaDoc...
IDE Scripting Console
Create Command-line Launcher...
JShell Console...
Groovy Console...
Kotlin
WakaTime Settings
om/projects/aldajo92/myapplication/Main.kt [app]
tivity_main.xml
Configure Kotlin Plugin Updates
Enable migrations detection (experimental)
Kotlin REPL
Configure Kotlin in Project
Show Kotlin Bytecode

Globant

Today…



#KotlinConf19 #Kotlin #JetBrains

KotlinConf 2019: What's New in Java 19: The end of Kotlin? by Jake Wharton

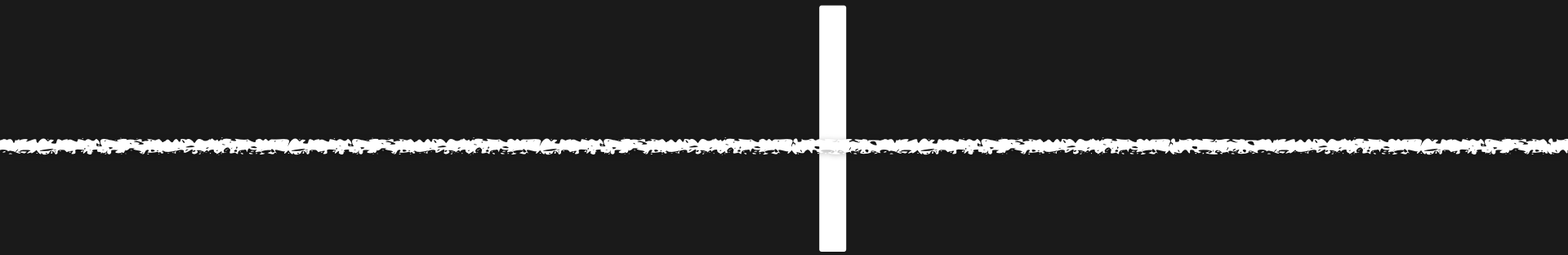
Globant

Java19 vs Kotlin

Java 1

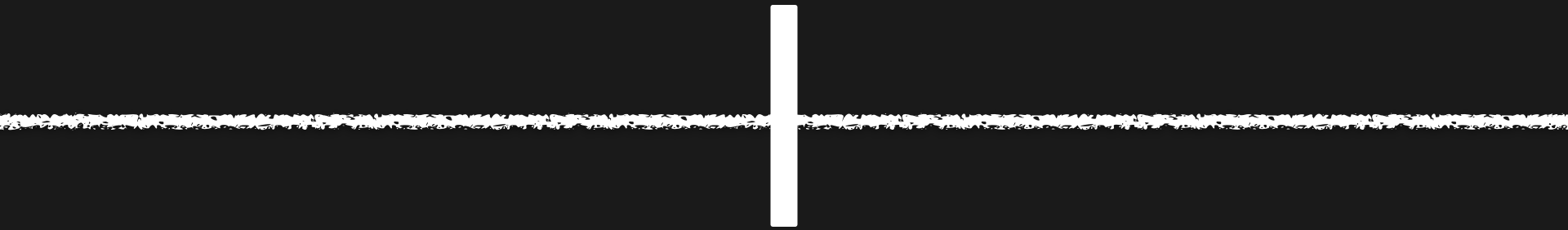
January 23, 1996

Java 5



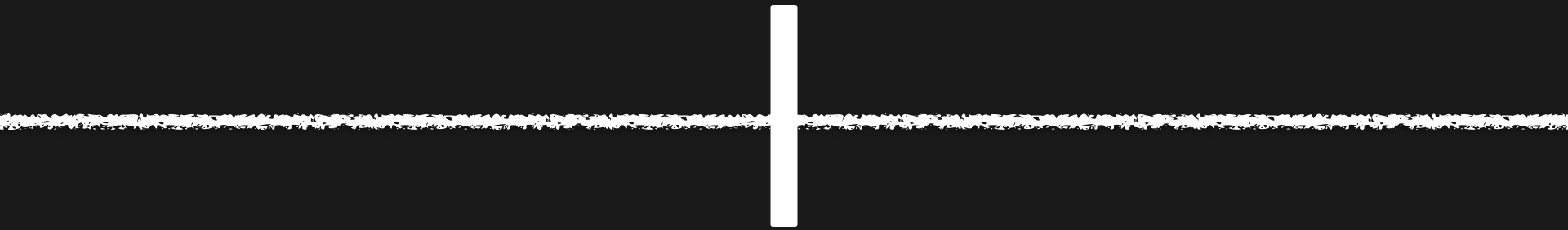
September 30, 2004

Java 7



July 28, 2011

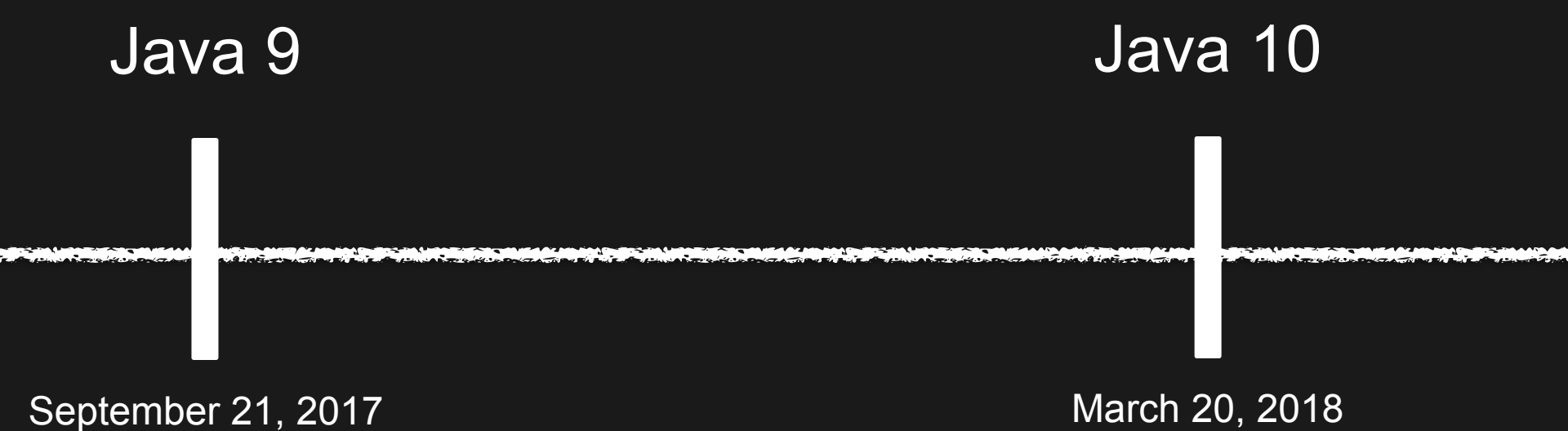
Java 8

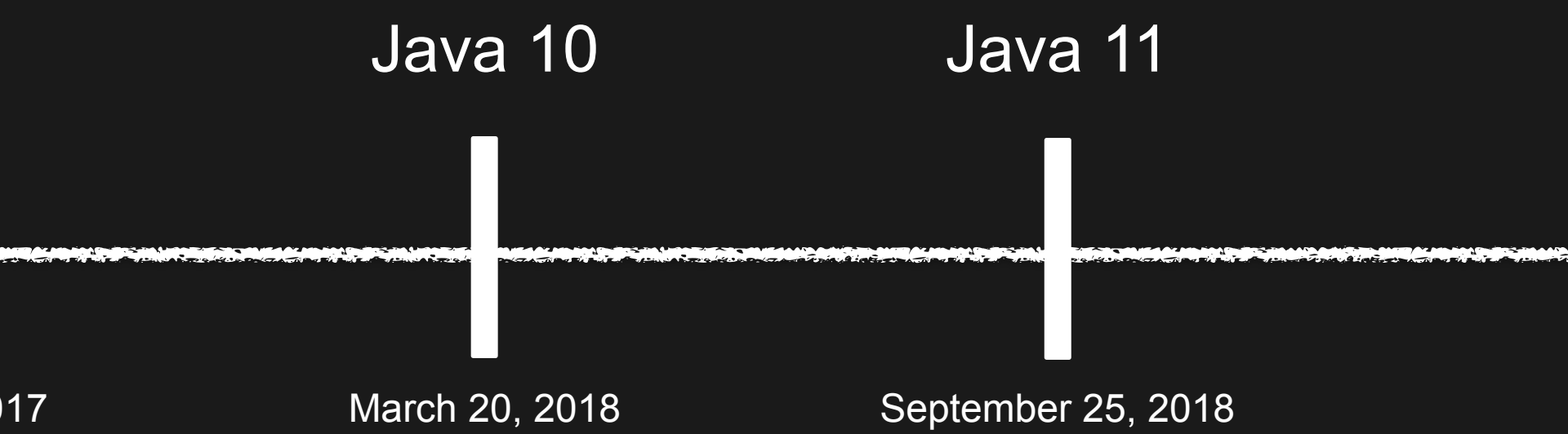


March 18, 2014

Java 9

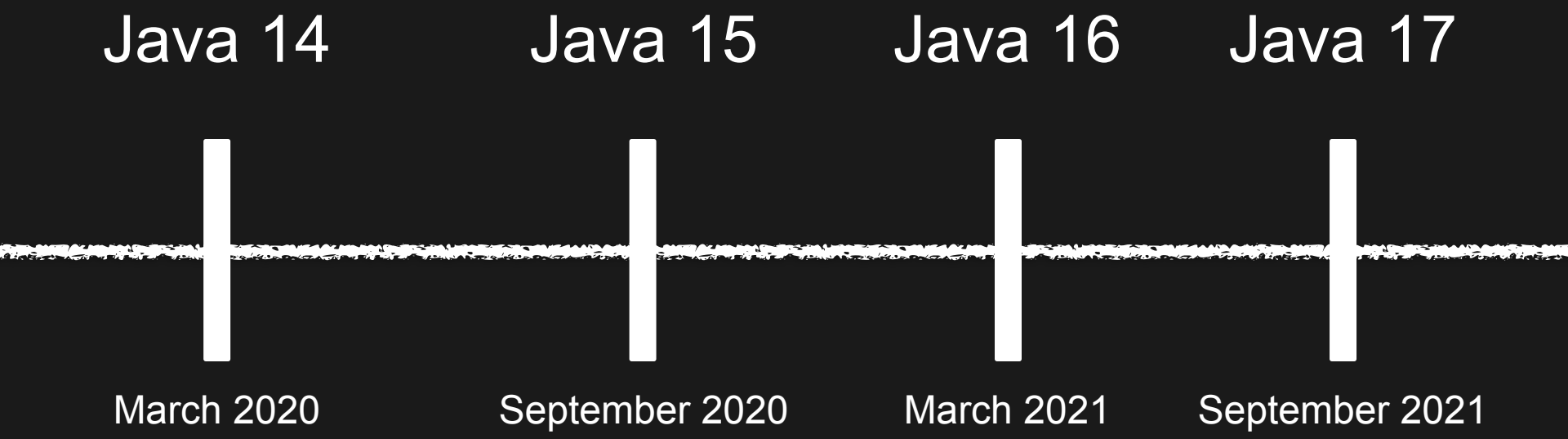
September 21, 2017

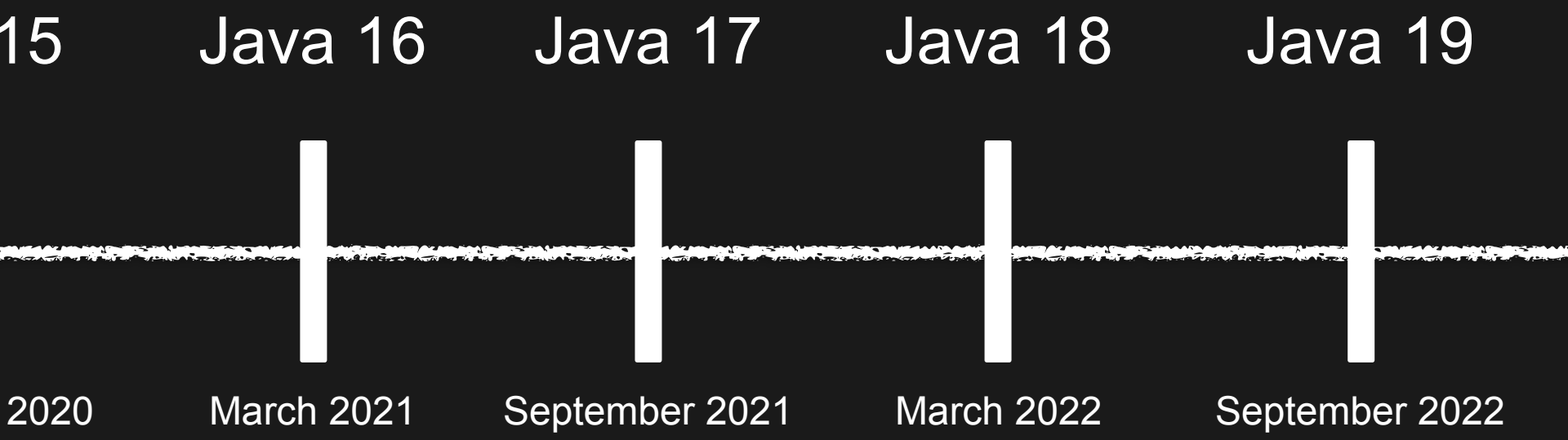

Java 9
Java 10
September 21, 2017
March 20, 2018


Java 10
Java 11
017
March 20, 2018
September 25, 2018


0
Java 11
Java 12
18
September 25, 2018
March 19, 2019
Sept


Java 12
Java 13
Java 14
Ja
March 19, 2019
September 17, 2019
March 2020
Septe


Java 14
Java 15
Java 16
Java 17
March 2020
September 2020
March 2021
September 2021


15
Java 16
Java 17
Java 18
Java 19
2020
March 2021
September 2021
March 2022
September 2022

Globant

Variable Type Inference

```kotlin
var count = 1
val users = mutableListOf<String>()

var default: String? = null
val tasks: Deque<Runnable> = ArrayDeque()
```

```java
var count = 1
final var users = new ArrayList<String>()

String default = null
Deque<Runnable> tasks = new ArrayDeque<>()
```

```kotlin
var count = 1
val users = mutableListOf<String>()

var default: String? = null
val tasks: Deque<Runnable> = ArrayDeque()

val run1: () -> Unit = { /* .. */ }
val run2 = { /* .. */ }
```

```java
var count = 1
final var users = new ArrayList<String>()

String default = null
Deque<Runnable> tasks = new ArrayDeque<>()

Runnable run1 = () -> { /* .. */ };
val run2 = new Runnable() {
    @Override public void run() { /* .. */ }
};
```

Globant

Local Functions

```kotlin
fun Graph.any(predicate: (Node) -> Boolean): Boolean {
    val seen = HashSet<Node>()

    fun Node.hasMatch(): Boolean {
        if (!seen.add(this)) return false // already seen
        if (predicate(this)) return true // match!
        return nodes.any { it.hasMatch() }
    }
    return root.hasMatch()
}
```

"Optimize generation of local functions"
youtrack.jetbrains.com/issue/KT-6336

```kotlin
fun Graph.any(predicate: (Node) -> Boolean): Boolean {
    val seen = HashSet<Node>()

    fun Node.hasMatch(): Boolean {
        if (!seen.add(this)) return false // already seen
        if (predicate(this)) return true // match!
        return nodes.any { it.hasMatch() }
    }
    return root.hasMatch()
}
```

```java
public static boolean any(Predicate<Node> predicate) {
     var seen = new HashSet<Node>();
     boolean hasMatch(Node node){
         if (!seen.add(node)) return false; // already seen
         if (predicate.test(node)) return true; // match!
         return node.getNodes().stream().anyMatch(n -> hasMatch(n));
     }
     return hasMatch(getRoot());
}
```

```java
public static boolean any(Predicate<Node> predicate) {
     var seen = new HashSet<Node>();
     boolean hasMatch(Node node){
         if (!seen.add(node)) return false; // already seen
         if (predicate.test(node)) return true; // match!
         return node.getNodes().stream().anyMatch(n -> hasMatch(n));
     }
     return hasMatch(getRoot());
}
```

Globant

Multiline String

```
fun main() {
  println("""
    |SELECT *
    |FROM users
    |WHERE name LIKE 'Jake %'
    |""".trimMargin())
}
```

```
SELECT *
FROM users
WHERE name LIKE 'Jake %'
```

```
fun main() {
  println("""
     !!!SELECT *
     !!!FROM users
     !!!WHERE name LIKE 'Jake %'
     !!!""".trimMargin(marginPrefix = "!!!"))
}
```

```
SELECT *
FROM users
WHERE name LIKE 'Jake %'
```

```kotlin
fun main() {
  println("""
    SELECT *
    FROM users
    WHERE name LIKE 'Jake %'
  """.trimIdent()
}
```

```
SELECT *
FROM users
WHERE name LIKE 'Jake %'
```

```java
public static void main(String[] args) {
  System.out.println("""
      SELECT *
      FROM users
      WHERE name LIKE 'Jake %'
      """);
}
```

```
SELECT *
FROM users
WHERE name LIKE 'Jake %'
```

```kotlin
fun main() {
  println("""
    SELECT *
    FROM users
    WHERE name LIKE 'Jake %'
  """.trimIdent()
}
```

```
SELECT *
FROM users
WHERE name LIKE 'Jake %'
```

```java
public static void main(String[] args) {
  System.out.println("""
      SELECT * \
      FROM users \
      WHERE name LIKE 'Jake %'
      """);
}
```

```
SELECT * FROM users WHERE name LIKE 'Jake %'
```

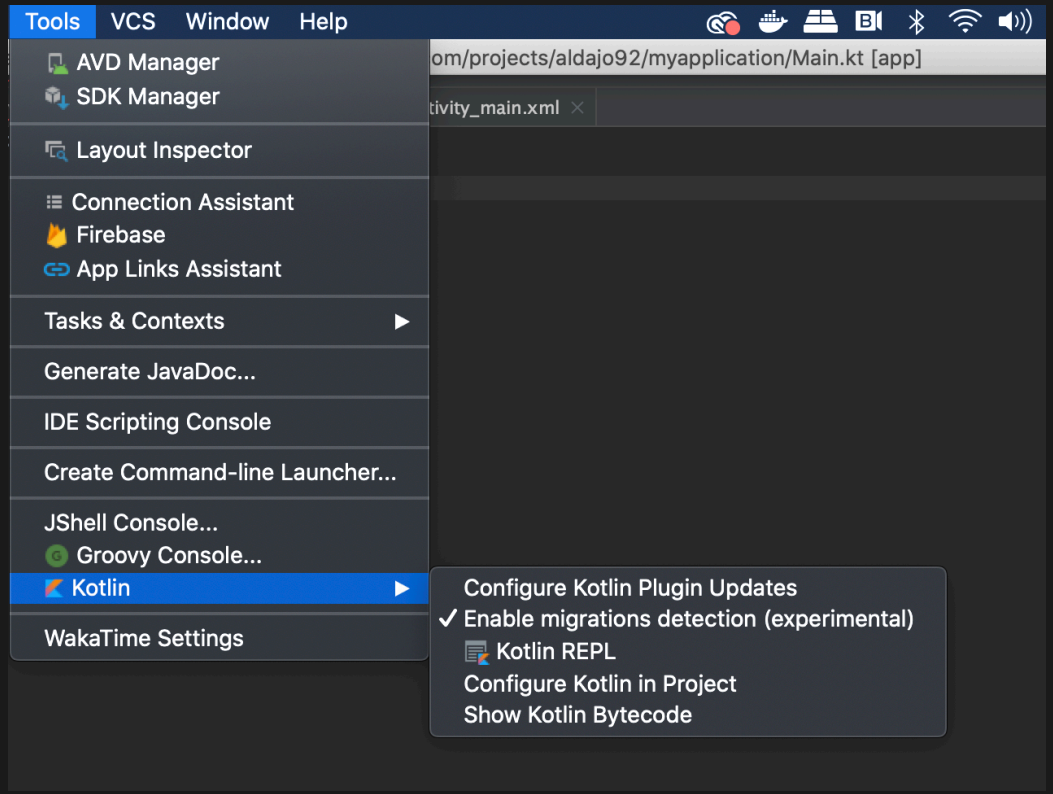
Globant

Value-Based Classes

```kotlin
data class Person(val name: String, val age: Int)
```

```java
record Person(String name, int age) { }
```

Globant


Tools
VCS
Window
Help
AVD Manager
SDK Manager
Layout Inspector
Connection Assistant
Firebase
App Links Assistant
Tasks & Contexts
Generate JavaDoc...
IDE Scripting Console
Create Command-line Launcher...
JShell Console...
Groovy Console...
Kotlin
WakaTime Settings
om/projects/aldajo92/myapplication/Main.kt [app]
tivity_main.xml
Configure Kotlin Plugin Updates
Enable migrations detection (experimental)
Kotlin REPL
Configure Kotlin in Project
Show Kotlin Bytecode

```
public final class Person {
   private final java.lang.String name; private final int age;
   public Person(java.lang.String, int); public java.lang.String
   getName(); public int getAge();
   public java.lang.String toString(); public int hashCode();
   public boolean equals(java.lang.Object);
   //..
}
```

```
public final class Person extends java.lang.Record {
   private final java.lang.String name;
   private final int age;
   public Person(java.lang.String, int);
   public java.lang.String name(); public int age();
   public java.lang.String toString(); public int hashCode();
   public boolean equals(java.lang.Object);
}
```

Globant

Sealed Hierarchies

```kotlin
sealed class Developer
data class Person(val name: String, val age: Int) : Developer()
data class Business(val name: String) : Developer()
```

```java
sealed class Developer { }
record Person(String name, int age) extends Developer { }
record Business(String name) extends Developer { }
```

```kotlin
sealed class Developer{
   abstract val name: String
}
data class Person(override val name: String, val age: Int) : Developer()
data class Business(override val name: String) : Developer()
```

```java
sealed interface Developer {
  String name();
}
record Person(@Override String name, int age) implements Developer { }
record Business(@Override String name) implements Developer { }
```

Globant

Type Matching

```kotlin
val o: Any = 1
if (o is Int) {
   println(o + 1)
}
```

```java
Object o = 1;
if (o instanceof Integer i) {
    System.out.println(i + 1);
}
```

```kotlin
val o: Any = 1
if (o is Int) {
   println(o + 1)
}
```

```java
Object o = 1;
if (o instanceof Integer) {
   Integer i = (Integer) o;
   System.out.println(i + 1);
}
```

Globant

The end of Kotlin?

```kotlin
data class Person(val name: String, val age: Int)

val alice = Person("Alice", 12)
val (name, age) = alice
val people = listOf(alice)
for ((name, age) in people) {
   // ..
}


val display = people.map { (name, age) -> "$name is $age years old" }
```

Java 10: Variable Type Inference

```
var count = 1
final var users = new ArrayList<String>()
```

Java 16/17: Local Methods

```
public static boolean any(Predicate<Node> predicate) {
  var seen = new HashSet<Node>();

  boolean hasMatch(Node node) {
    if (!seen.add(node)) return false; // already seen
    if (predicate.test(node)) return true; // match!
    return node.getNodes().stream().anyMatch(n -> hasMatch(n));
  }

  return hasMatch(getRoot());
}
```

Java 15: Multiline Strings

```
public static void main(String[] args) {
  System.out.println("""
      SELECT *
      FROM users
      WHERE name LIKE 'Jake %'
      """);
}
```

Java 15/16: Records

```
record Person(String name, int age) { }
```

Java 15/16: Sealed Hierarchies

```
sealed interface Developer { }
record Person(String name, int age) implements Developer { }
record Business(String name) implements Developer { }
```

Java 15/16: Type Matching

```
Object o = 1;
if (o instanceof Integer i) {
  System.out.println(i + 1);
}
```

Java 16/17: Pattern Matching

```
Download download = //...
var result = switch (download) {
  case App(var name, Person("Alice", _)) -> "Alice's app " + name
  case Movie(var title, Person("Alice", _)) -> "Alice's movie " + title
  case App(_), Movie(_) -> "Not by Alice"
};
```

Java 14: Expression Switch

```
Download download = //...
var result = switch (download) {
  case App(var name, Person("Alice", _)) -> "Alice's app " + name
  case Movie(var title, Person("Alice", _)) -> "Alice's movie " + title
  case App(_), Movie(_) -> "Not by Alice"
};
```

Java 16/17: Virtual Threads

```
public static void main(String... args) {
  Executor e = Executors.newWorkStealingPool();
  for (int count = 10; count >= 0; count--) {
    final var finalCount = count;
    e.execute(() -> {
      Thread.sleep(100 * finalCount);
      System.out.println(finalCount);
    });
  }
}
```

Globant

- New Java APIs and new VM improvements also help Kotlin.
- API, bytecode, and VM changes for new Java language features are also available for use by kotlinc
- No plans to tackle nullability in Java!
- Java in 3 years will also be competing with Kotlin in 3 years, not the Kotlin of today.
- Kotlin has an IDE to potentially evolve in ways that Java cannot Kotlin has first-class multiplatform.