KotlinConf

Understand Every Line of Your Codebase

Victoria Gonda
@TTGonda

Boris Farber
@BorisFarber

Speakers

Victoria

- Developer at Collective Idea
- Android and Rails

Boris

- Partner Engineer at Google
- Android Partnerships
- Android and Java Binary Analysis

@TTGonda
@BorisFarber

Plan

- Intro to Java & Kotlin build pipeline
- From Java to Kotlin
- From Kotlin to Java
- Summary

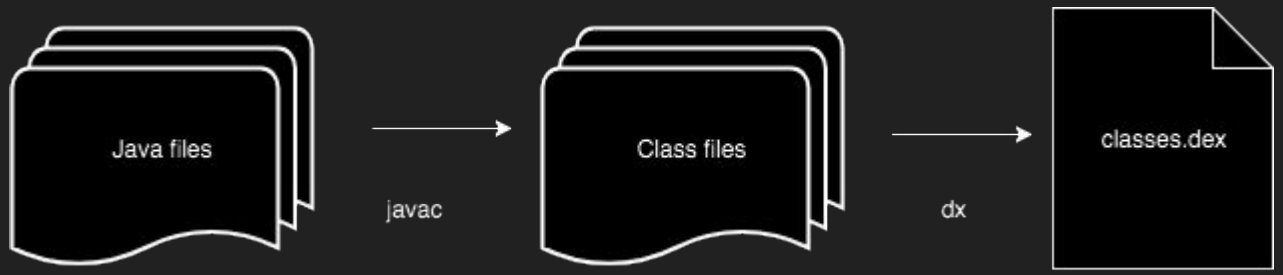
Goal

- Use your Java language skills to ramp up/work with Kotlin
- Understand Java/Kotlin language Interoperability

Java & Kotlin languages build pipeline

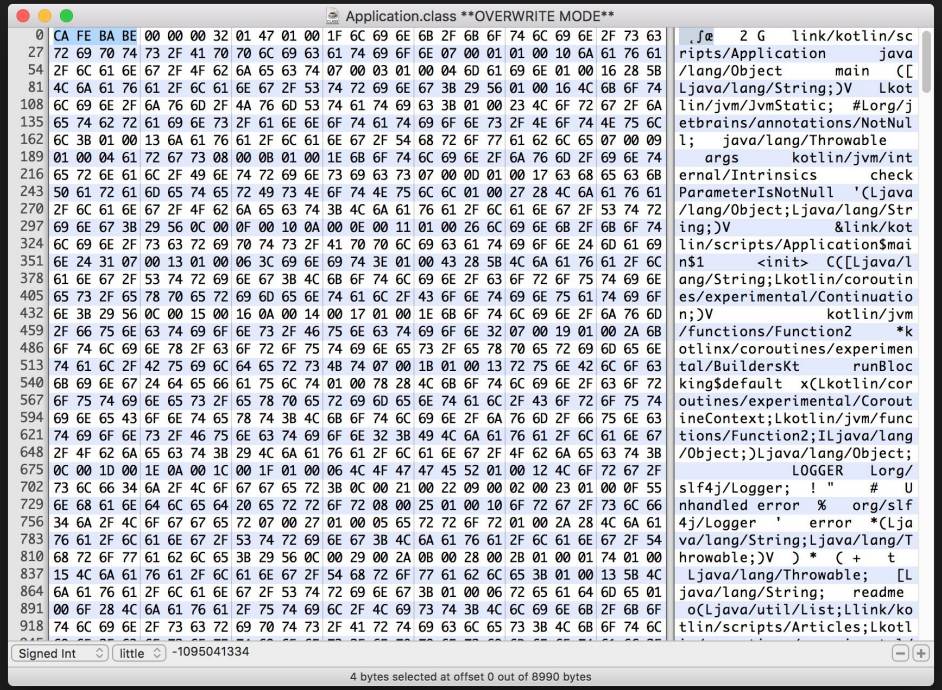
@TTGonda
@BorisFarber

Intro to Java Compilation



@TTGonda
@BorisFarber

Class file

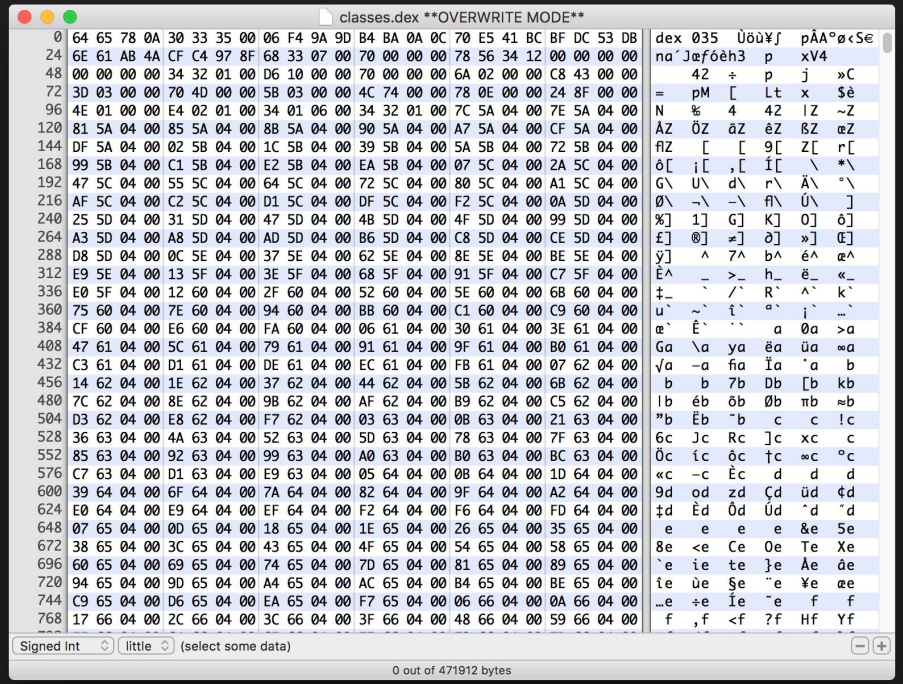


@TTGonda
@BorisFarber

Class Files

- Result of javac invocation
- Binary format
- Can be represented as a C struct with
  - Class name version and other details
  - Constant pool
  - Interfaces array
  - Fields array
  - Methods array
  - Attributes

Dex file



@TTGonda
@BorisFarber

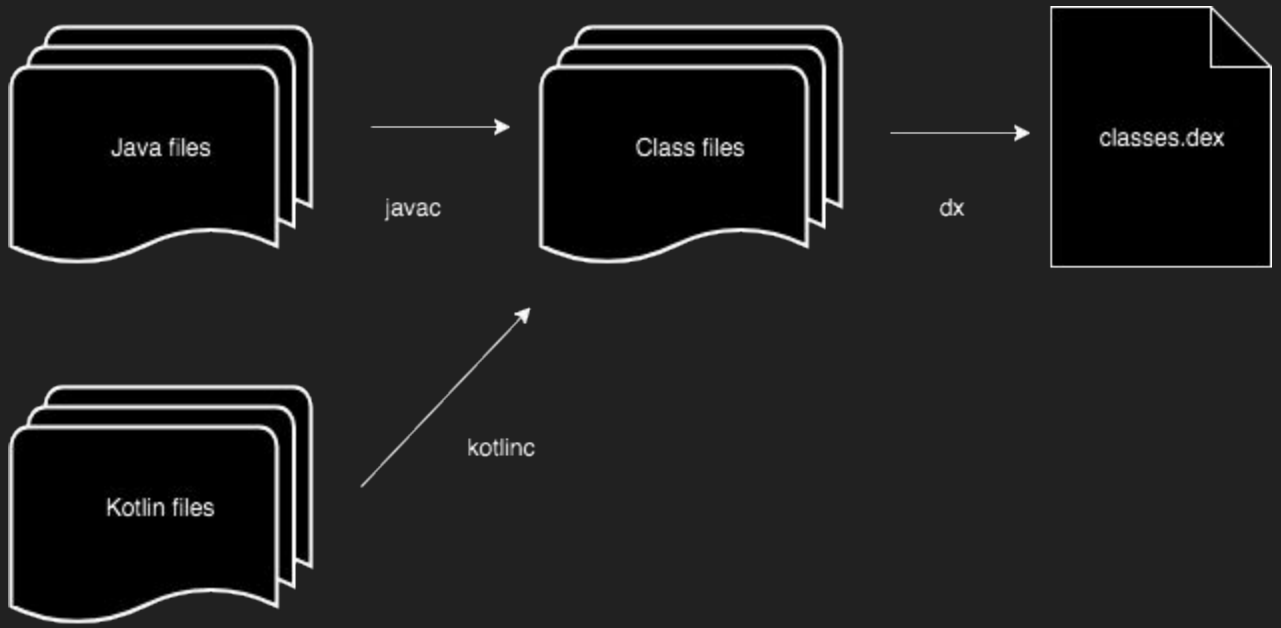
Dex Files

- Result of dx tool invocation
- Many classes → 1 classes.dex file
- Data from all classes is merged into tables (methods ...) less than 65K
- Different encoding method
- Different addressing mode (registers versus stack)

Dex & Multidex

- 16 bits field for all methods per classes dex
- 2^16 = 64k
- Includes both defined and used methods

Kotlin Fits In



How Kotlin Fits In

- Kotlin compiler (kotlinc) generates class files
- 1 kt file might have few class files
- 100% interoperable with Java language binary format (class)
- Can be added incrementally

Java Interoperability

- Java language interoperability is one of Kotlin language fundamentals
- Kotlin language syntax is very similar but not compatible to Java
- However we can use Java ecosystem (libraries and tools) with Kotlin as we were using with Java language

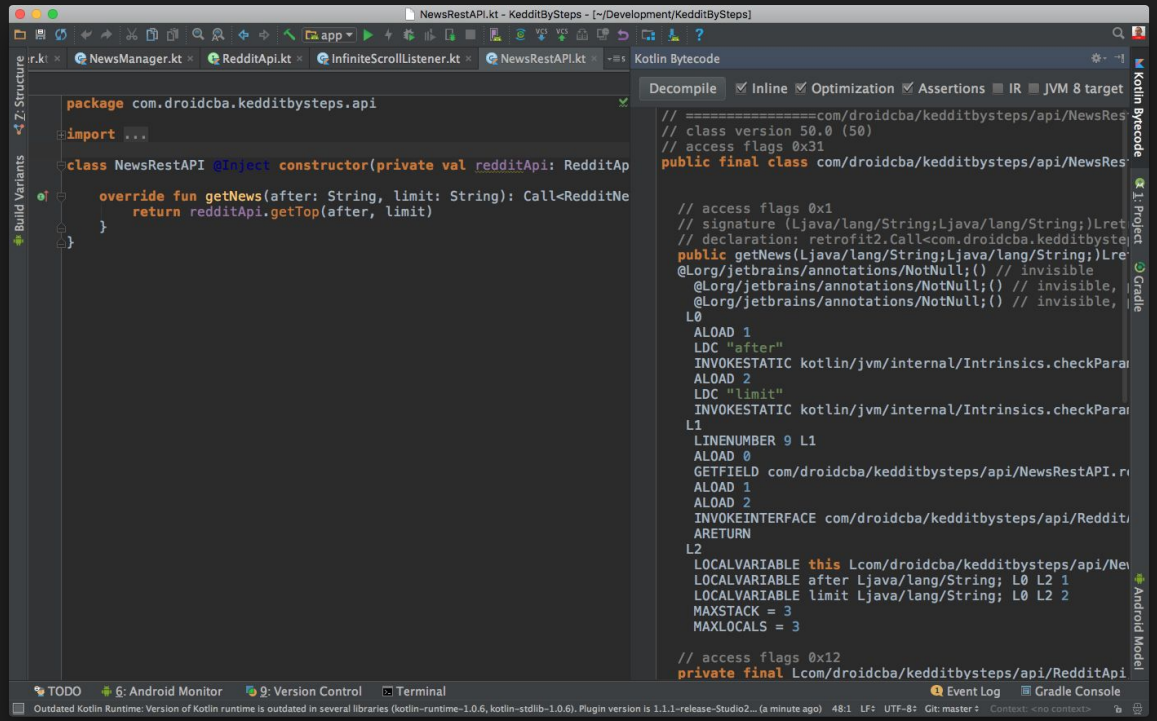
From Kotlin code to Java code

@TTGonda
@BorisFarber

Why

1. Curiousity
2. Understand how it works
3. Make tradeoff decisions
4. Assess performance
5. Assess apk size and method count

Android Studio and IntelliJ IDEA



@TTGonda
@BorisFarber

Wrap-Up

- Your Java language knowledge is valuable with Kotlin
- Pretty easy to start and get Kotlin code working
- Both Java and Kotlin generate class files
- Both Android DX tool & JVM know how to read class files
- There is a tooling to help you

@TTGonda
@BorisFarber

“We can use our prior knowledge of the Java programming language to help us understand, excel with the Kotlin programming language.”

Kotlin Classes

```
class User(val name: String)
```

Kotlin Classes

```
import kotlin.Metadata;
import kotlin.jvm.internal.Intrinsics;
import org.jetbrains.annotations.NotNull;

@Metadata(
   mv = {1, 1, 7},
   bv = {1, 0, 2},
   k = 1,
   d1 =
{"\u0000\u0012\n\u0002\u0018\u0002\n\u0002\u0010\u0000\n\u0000\n\u0002\u0010\u000e\n\u0002\b\u0004\u0018\u00002\u00020\u0001B\r\u0012\u0006\u0010\u0002\u001a\u00020\u0003¢\u0006\u0002\u0010\u0004R\u0011\u0010\u0002\u001a\u00020\u0003¢\u0006\b\n\u0000\u001a\u0004\b\u0005\u0010\u0006¨\u0006\u0007"},
   d2 = {"Lcom/victoriagonda/kotlinuncoveredexamples/User;", "", "name", "", "(Ljava/lang/String;)V", "getName",
"()Ljava/lang/String;", "production sources for module app"}
)
public final class User {
   @NotNull
   private final String name;

   @NotNull
   public final String getName() {
      return this.name;
   }

   public User(@NotNull String name) {
      Intrinsics.checkParameterIsNotNull(name, "name");
      super();
      this.name = name;
   }
}
```

@TTGonda
@BorisFarber

Kotlin Classes

```
public final class User {
  @NotNull
  private final String name;

  @NotNull
  public final String getName() {
     return this.name;
  }

  public User(@NotNull String name) {
     Intrinsics.checkParameterIsNotNull(name, "name");
     super();
     this.name = name;
  }
}
```

@TTGonda
@BorisFarber

Data Classes

```
data class User(val name: String)
```

```
public final class User {
  @NotNull
  public final String component1() {
     return this.name;
  }

  @NotNull
  public final User copy(@NotNull String name) {
     Intrinsics.checkParameterIsNotNull(name, "name");
     return new User(name);
  }

  // $FF: synthetic method
  // $FF: bridge method
  @NotNull
  public static User copy$default(User var0, String var1, int var2, Object var3) {
     if((var2 & 1) != 0) {
        var1 = var0.name;
     }

     return var0.copy(var1);
  }

  public String toString() {
     return "User(name=" + this.name + ")";
  }

  public int hashCode() {
     return this.name != null?this.name.hashCode():0;
  }

  public boolean equals(Object var1) {
     if(this != var1) {
        if(var1 instanceof User) {
           User var2 = (User)var1;
           if(Intrinsics.areEqual(this.name, var2.name)) {
              return true;
           }
        }

        return false;
     } else {
        return true;
     }
  }
}
```

@TTGonda
@BorisFarber

```
public final class User {
  // …

  @NotNull
  public final String component1() { /* … */ }

  @NotNull
  public final User copy(@NotNull String name) { /* … */ }

  // $FF: synthetic method
  // $FF: bridge method
  @NotNull
  public static User copy$default(User var0, String var1, int var2, Object var3) {
    /* … */
  }

  public String toString() { /* … */ }

  public int hashCode() { /* … */ }

  public boolean equals(Object var1) { /* … */ }
}
```

@TTGonda
@BorisFarber

```java
@NotNull
   public final User copy(@NotNull String name) {
      Intrinsics.checkParameterIsNotNull(name, "name");
      return new User(name);
   }

   // $FF: synthetic method
   // $FF: bridge method
   @NotNull
   public static User copy$default(User var0, String var1, int var2, Object var3) {
      if((var2 & 1) != 0) {
         var1 = var0.name;
      }

      return var0.copy(var1);
   }
```

@TTGonda
@BorisFarber

Null Checks

```
val middleName: String? = getMiddleName()
middleName?.length
```

Null Checks

```
String middleName = this.getMiddleName();
if(middleName != null) {
  middleName.length();
}
```

Null Checks

```
val middleName: String? = "M"
middleName?.length
```

Null Checks

```
String middleName = "M";
middleName.length();
```

Extension Functions

```
// StringExt.kt
fun String?.isEmpty(): Boolean {
 return this == null || this.length == 0
}
```

@TTGonda
@BorisFarber

Extension Functions

```
// StringExt.kt
fun String?.isEmpty(): Boolean {
 return this == null || this.length == 0
}

"hello".isEmpty() // false
"".isEmpty()      // true
```

@TTGonda
@BorisFarber

Extension Functions

```
public final class StringExtKt {
  public static final boolean isEmpty(@Nullable String $receiver) {
     return $receiver == null || $receiver.length() == 0;
  }
}
```

@TTGonda
@BorisFarber

Extension Functions

```
public final class StringExtKt {
  public static final boolean isEmpty(@Nullable String $receiver) {
     return $receiver == null || $receiver.length() == 0;
  }
}

StringExtKt.isEmpty("hello");  // false
StringExtKt.isEmpty("");  // true
```

@TTGonda
@BorisFarber

Lambdas (inline)

```
inline fun beforeAndAfter(
    start: String?,
    function: (string: String?) -> String
) {
 print("Before: $start")
 val after = function(start)
 print("After: $after")
}
```

@TTGonda
@BorisFarber

Lambdas (inline)

```
public final void beforeAndAfter(
    @Nullable String start, @NotNull Function1 function) {
  Intrinsics.checkParameterIsNotNull(function, "function");
  String after = "Before: " + start;
  System.out.print(after);

  after = (String)function.invoke(start);

  String var5 = "After: " + after;
  System.out.print(var5);
}
```

@TTGonda
@BorisFarber

Lambdas (inline)

```
fun example() {
 beforeAndAfter("hello", { string ->
   string + " world"
 })
}
```

Lambdas (inline)

```
public final void example() {
  String start$iv = "hello";
  String after$iv = "Before: " + start$iv;
  System.out.print(after$iv);
  after$iv = Intrinsics.stringPlus(start$iv, " world");
  String string = "After: " + after$iv;
  System.out.print(string);
}
```

@TTGonda
@BorisFarber

Lambdas (not inline)

```
fun beforeAndAfter(
    start: String,
    function: (string: String) -> String
) {
 print("Before: $start")
 val after = function(start)
 print("After: $after")
}
```

Lambdas (not inline)

```
public final void example() {
  this.beforeAndAfter("hello", (Function1)null.INSTANCE);
}
```

@TTGonda
@BorisFarber

```
final class com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1 extends kotlin/jvm/internal/Lambda  implements kotlin/jvm/functions/Function1  {
  // …

  // access flags 0x11
  public final invoke(Ljava/lang/String;)Ljava/lang/String;
  @Lorg/jetbrains/annotations/NotNull;() // invisible
    @Lorg/jetbrains/annotations/NotNull;() // invisible, parameter 0
   L0
    ALOAD 1
    LDC "string"
    INVOKESTATIC kotlin/jvm/internal/Intrinsics.checkParameterIsNotNull (Ljava/lang/Object;Ljava/lang/String;)V
   L1
    LINENUMBER 58 L1
    NEW java/lang/StringBuilder
    DUP
    INVOKESPECIAL java/lang/StringBuilder.<init> ()V
    ALOAD 1
    INVOKEVIRTUAL java/lang/StringBuilder.append (Ljava/lang/String;)Ljava/lang/StringBuilder;
    LDC " world"
    INVOKEVIRTUAL java/lang/StringBuilder.append (Ljava/lang/String;)Ljava/lang/StringBuilder;
    INVOKEVIRTUAL java/lang/StringBuilder.toString ()Ljava/lang/String;
   L2
    ARETURN
   L3
    LOCALVARIABLE this Lcom/victoriagonda/kotlinuncoveredexamples/Lambda$example$1; L0 L3 0
    LOCALVARIABLE string Ljava/lang/String; L0 L3 1
    MAXSTACK = 2
    MAXLOCALS = 2

  // access flags 0x19
  public final static Lcom/victoriagonda/kotlinuncoveredexamples/Lambda$example$1; INSTANCE

  // access flags 0x8
  static <clinit>()V
    NEW com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1
    DUP
    INVOKESPECIAL com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1.<init> ()V
    PUTSTATIC com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1.INSTANCE : Lcom/victoriagonda/kotlinuncoveredexamples/Lambda$example$1;
    RETURN
    MAXSTACK = 2
    MAXLOCALS = 0

// …
}
```

@TTGonda
@BorisFarber

```
final class com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1
    extends kotlin/jvm/internal/Lambda
    implements kotlin/jvm/functions/Function1  {
  // …

  // access flags 0x11
  public final invoke(Ljava/lang/String;)Ljava/lang/String;
  @Lorg/jetbrains/annotations/NotNull;() // invisible
    @Lorg/jetbrains/annotations/NotNull;() // invisible, parameter 0
   L0
    ALOAD 1
    LDC "string"
    INVOKESTATIC kotlin/jvm/internal/Intrinsics
        .checkParameterIsNotNull (Ljava/lang/Object;Ljava/lang/String;)V
   L1
    LINENUMBER 58 L1
    NEW java/lang/StringBuilder
    DUP
    INVOKESPECIAL java/lang/StringBuilder.<init> ()V
    ALOAD 1
    INVOKEVIRTUAL java/lang/StringBuilder
        .append (Ljava/lang/String;)Ljava/lang/StringBuilder;
    LDC " world"
```

@TTGonda
@BorisFarber

```
final class com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1
    extends kotlin/jvm/internal/Lambda
    implements kotlin/jvm/functions/Function1  {
  // …

  // access flags 0x11
  public final invoke(Ljava/lang/String;)Ljava/lang/String;
  @Lorg/jetbrains/annotations/NotNull;() // invisible
    @Lorg/jetbrains/annotations/NotNull;() // invisible, parameter 0
   L0
    ALOAD 1
    LDC "string"
    INVOKESTATIC kotlin/jvm/internal/Intrinsics
        .checkParameterIsNotNull (Ljava/lang/Object;Ljava/lang/String;)V
   L1
    LINENUMBER 58 L1
    NEW java/lang/StringBuilder
    DUP
    INVOKESPECIAL java/lang/StringBuilder.<init> ()V
    ALOAD 1
    INVOKEVIRTUAL java/lang/StringBuilder
        .append (Ljava/lang/String;)Ljava/lang/StringBuilder;
    LDC " world"
```

@TTGonda
@BorisFarber

```
    INVOKESTATIC kotlin/jvm/internal/Intrinsics
        .checkParameterIsNotNull (Ljava/lang/Object;Ljava/lang/String;)V
   L1
    LINENUMBER 58 L1
    NEW java/lang/StringBuilder
    DUP
    INVOKESPECIAL java/lang/StringBuilder.<init> ()V
    ALOAD 1
    INVOKEVIRTUAL java/lang/StringBuilder
        .append (Ljava/lang/String;)Ljava/lang/StringBuilder;
    LDC " world"
    INVOKEVIRTUAL java/lang/StringBuilder
        .append (Ljava/lang/String;)Ljava/lang/StringBuilder;
    INVOKEVIRTUAL java/lang/StringBuilder.toString ()Ljava/lang/String;
   L2
    ARETURN
   L3
    LOCALVARIABLE this
        Lcom/victoriagonda/kotlinuncoveredexamples/Lambda$example$1; L0 L3 0
    LOCALVARIABLE string Ljava/lang/String; L0 L3 1
    MAXSTACK = 2
    MAXLOCALS = 2

// access flags 0x19
public final static
```

@TTGonda
@BorisFarber

```
Lcom/victoriagonda/kotlinuncoveredexamples/Lambda$example$1; L0 L3 0
    LOCALVARIABLE string Ljava/lang/String; L0 L3 1
    MAXSTACK = 2
    MAXLOCALS = 2

  // access flags 0x19
  public final static
      Lcom/victoriagonda/kotlinuncoveredexamples/Lambda$example$1; INSTANCE

  // access flags 0x8
  static <clinit>()V
    NEW com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1
    DUP
    INVOKESPECIAL
        com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1.<init> ()V
    PUTSTATIC
        com/victoriagonda/kotlinuncoveredexamples/Lambda$example$1.INSTANCE :
        Lcom/victoriagonda/kotlinuncoveredexamples/Lambda$example$1;
    RETURN
    MAXSTACK = 2
    MAXLOCALS = 0

// …
}
```

@TTGonda
@BorisFarber

Kotlin inside your APK

- 5-6K method count
- 1 MB library size

References

- https://proandroiddev.com/exploring-the-class-side-of-kotlin-d4af0d1065e2
- https://collectiveidea.com/blog/archives/2017/05/16/kotlin-uncovered-part-1

Takeaways

- Kotlin language and Java language are highly interoperable, both using .class and .dex files
- These can be inspected to better understand the size, performance, and implementation
- Viewing the equivalent Java from the Kotlin bytecode can be a tool to grasp the specifics of Kotlin language features

Thank You

Victoria Gonda
@TTGonda
victoriagonda.com
Boris Farber
@BorisFarber