

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

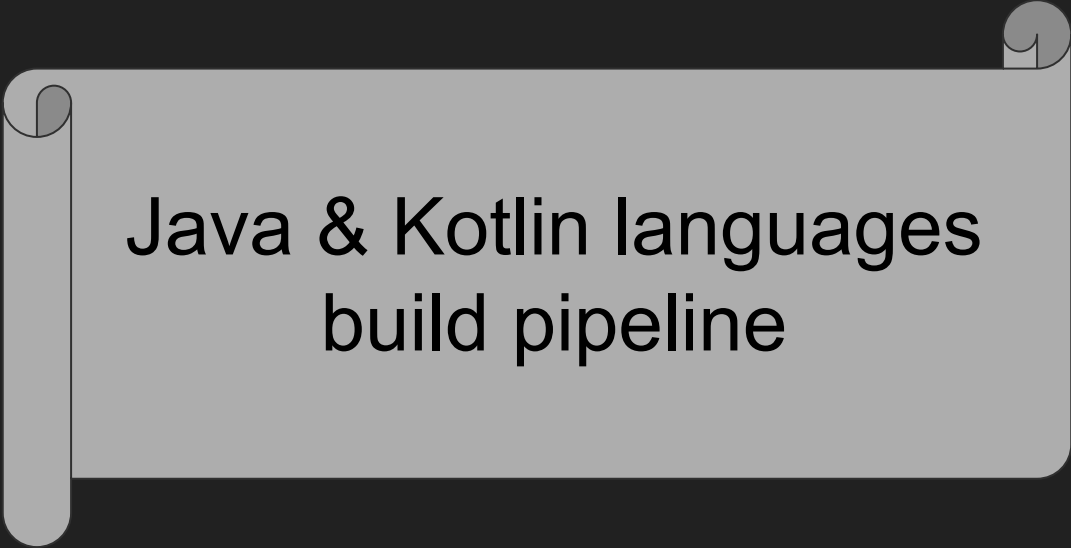
- Intro to Java & Kotlin build pipeline
- From Java to Kotlin
- From Kotlin to Java
- Summary

Goal

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

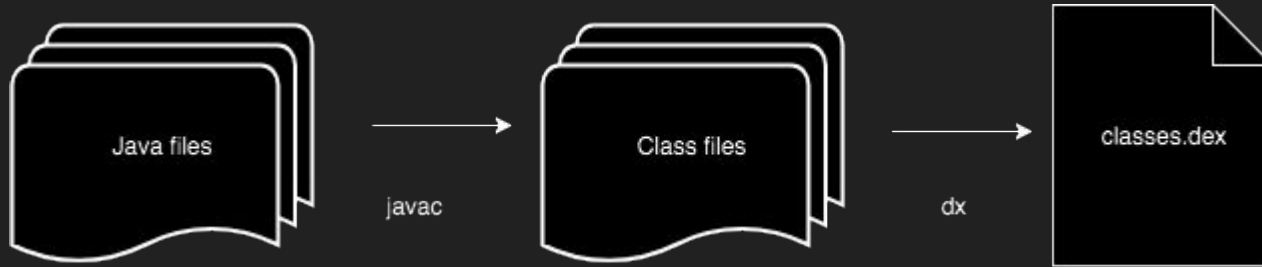
@TTGonda

@BorisFarber

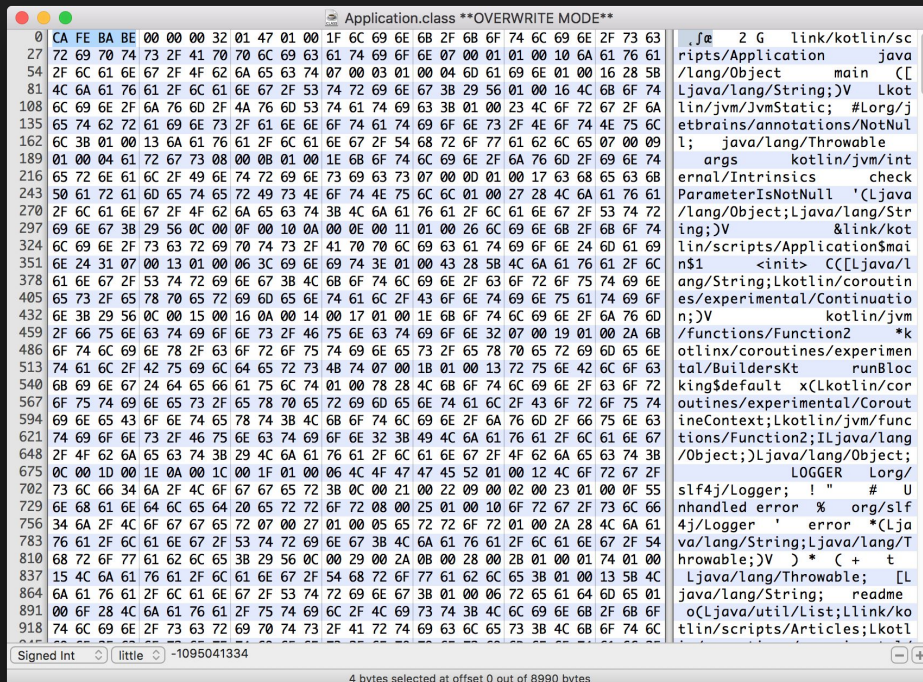


Java & Kotlin languages build pipeline

Intro to Java Compilation



Class file



```
Application.class **OVERWRITE MODE**
00 CA FE BA BE 00 00 00 32 01 47 01 00 1F 6C 69 6E 68 2F 6B 6F 74 6C 69 6E 2F 73 63
27 72 69 70 74 73 2F 41 70 70 6C 69 63 61 74 69 6F 6E 07 00 01 01 00 10 6A 61 76 61
54 2F 6C 61 6E 67 2F 4F 62 6A 65 63 74 07 00 03 01 00 04 6D 61 69 6E 01 00 16 28 5B
81 4C 6A 61 76 61 2F 6C 61 6E 67 2F 53 74 72 69 6E 67 3B 29 56 01 00 16 4C 6B 6F 74
108 6C 69 6E 2F 6A 76 6D 2F 4A 76 6D 53 74 61 74 69 63 3B 01 00 23 4C 6F 72 67 2F 6A
135 65 74 62 72 61 69 6E 73 2F 61 6E 6E 6F 74 61 74 69 6F 6E 73 2F 4E 6F 74 4E 75 6C
162 6C 3B 01 00 13 6A 61 76 61 2F 6C 61 6E 67 2F 54 68 72 6F 77 61 62 6C 65 07 00 09
189 01 00 04 61 72 67 73 08 00 0B 01 00 1E 68 6F 74 6C 69 6E 2F 6A 76 6D 2F 69 6E 74
216 65 72 6E 61 6C 2F 49 6E 74 72 69 6E 73 69 63 73 07 00 0D 01 00 17 63 68 65 63 68
243 50 61 72 61 6D 65 74 65 72 49 73 4E 6F 74 4E 75 6C 6C 01 00 27 28 4C 6A 61 76 61
270 2F 6C 61 6E 67 2F 4F 62 6A 65 63 74 3B 4C 6A 61 76 61 2F 6C 61 6E 67 2F 53 74 72
297 69 6E 67 3B 29 56 0C 00 0F 00 10 0A 00 0E 00 11 01 00 26 6C 69 6E 68 2F 68 6F 74
324 6C 69 6E 2F 73 63 72 69 70 74 73 2F 41 70 70 6C 69 63 61 74 69 6F 6E 24 6D 61 69
351 6E 24 31 07 00 13 01 00 06 3C 69 6E 69 74 3E 01 00 43 28 5B 4C 6A 61 76 61 2F 6C
378 61 6E 67 2F 53 74 72 69 6E 67 3B 4C 68 6F 74 6C 69 6E 2F 63 6F 72 6F 75 74 69 6E
405 65 73 2F 65 78 70 65 72 69 6D 65 6E 74 61 6C 2F 43 6F 6E 74 69 6E 75 61 74 69 6F
432 6E 3B 29 56 0C 00 15 00 16 0A 00 14 00 17 01 00 1E 68 6F 74 6C 69 6E 2F 6A 76 6D
459 2F 6E 75 6E 63 74 69 6F 6E 73 2F 46 75 6E 63 74 69 6F 6E 32 07 00 19 01 00 2A 6B
486 6F 74 6C 69 6E 78 2F 63 6F 72 6F 75 74 69 6E 65 73 2F 65 78 70 65 72 69 6D 65 6E
513 74 61 6C 2F 42 75 69 6C 64 65 72 73 48 74 07 00 1B 01 00 13 72 75 6E 42 6C 6F 63
540 68 69 6E 67 24 64 65 66 61 75 6C 74 01 00 78 28 4C 68 6F 74 6C 69 6E 2F 63 6F 72
567 6F 75 74 69 6E 65 73 2F 65 78 70 65 72 69 6D 65 6E 74 61 6C 2F 43 6F 72 6F 75 74
594 69 6E 65 43 6F 6E 74 65 78 74 3B 4C 68 6F 74 6C 69 6E 2F 6A 76 6D 2F 66 75 6E 63
621 74 69 6F 6E 73 2F 46 75 6E 63 74 69 6F 6E 32 3B 49 4C 6A 61 76 61 2F 6C 61 6E 67
648 2F 4F 62 6A 65 63 74 3B 29 4C 6A 61 76 61 2F 6C 61 6E 67 2F 4F 62 6A 65 63 74 3B
675 0C 00 1D 00 1E 0A 00 1C 00 1F 01 00 06 4C 4F 47 47 45 52 01 00 12 4C 6F 72 67 2F
702 73 6C 66 34 6A 2F 4C 6F 67 67 65 72 3B 0C 00 21 00 22 09 00 02 00 23 01 00 0F 55
729 6E 68 61 6E 64 6C 65 64 20 65 72 72 6F 72 08 00 25 01 00 10 6F 72 67 2F 73 6C 66
756 34 6A 2F 4C 6F 67 65 72 07 00 27 01 00 05 65 72 72 6F 72 01 00 2A 28 4C 6A 61
783 76 61 2F 6C 61 6E 67 2F 53 74 72 69 6E 67 3B 4C 6A 61 76 61 2F 6C 61 6E 67 2F 54
810 68 72 6F 77 61 62 6C 65 3B 29 56 0C 00 29 00 2A 0B 00 28 00 2B 01 00 01 74 01 00
837 15 4C 6A 61 76 61 2F 6C 61 6E 67 2F 54 68 72 6F 77 61 62 6C 65 3B 01 00 13 5B 4C
864 6A 61 76 61 2F 6C 61 6E 67 2F 53 74 72 69 6E 67 3B 01 00 06 72 65 61 64 6D 65 01
891 00 6F 28 4C 6A 61 76 61 2F 75 74 69 6C 2F 4C 69 73 74 3B 4C 6C 69 6E 68 2F 6B 6F
918 74 6C 69 6E 2F 73 63 72 69 70 74 73 2F 41 72 74 69 63 6C 65 73 3B 4C 6B 6F 74 6C
```

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

```
classes.dex **OVERWRITE MODE**
0 64 65 78 0A 30 33 35 00 06 F4 9A 9D B4 BA 0A 0C 70 E5 41 BC BF DC 53 DB
24 6E 61 AB 4A CF C4 97 8F 68 33 07 00 70 00 00 00 78 56 34 12 00 00 00 00
48 00 00 00 00 34 32 01 00 D6 10 00 00 70 00 00 00 6A 02 00 00 C8 43 00 00
72 3D 03 00 00 70 4D 00 00 5B 03 00 00 4C 74 00 00 78 0E 00 00 24 8F 00 00
96 4E 01 00 00 E4 02 01 00 34 01 06 00 34 32 01 00 7C 5A 04 00 7E 5A 04 00
120 81 5A 04 00 85 5A 04 00 8B 5A 04 00 90 5A 04 00 A7 5A 04 00 CF 5A 04 00
144 DF 5A 04 00 02 5B 04 00 1C 5B 04 00 39 5B 04 00 5A 5B 04 00 72 5B 04 00
168 99 5B 04 00 C1 5B 04 00 E2 5B 04 00 EA 5B 04 00 07 5C 04 00 2A 5C 04 00
192 47 5C 04 00 55 5C 04 00 64 5C 04 00 72 5C 04 00 80 5C 04 00 A1 5C 04 00
216 AF 5C 04 00 C2 5C 04 00 D1 5C 04 00 DF 5C 04 00 F2 5C 04 00 0A 5D 04 00
240 25 5D 04 00 31 5D 04 00 47 5D 04 00 4B 5D 04 00 4F 5D 04 00 99 5D 04 00
264 A3 5D 04 00 A8 5D 04 00 AD 5D 04 00 B6 5D 04 00 C8 5D 04 00 CE 5D 04 00
288 D8 5D 04 00 0C 5E 04 00 37 5E 04 00 62 5E 04 00 8E 5E 04 00 BE 5E 04 00
312 E9 5E 04 00 13 5F 04 00 3E 5F 04 00 68 5F 04 00 91 5F 04 00 C7 5F 04 00
336 E0 5F 04 00 12 60 04 00 2F 60 04 00 52 60 04 00 5E 60 04 00 6B 60 04 00
360 75 60 04 00 7E 60 04 00 94 60 04 00 BB 60 04 00 C1 60 04 00 C9 60 04 00
384 CF 60 04 00 E6 60 04 00 FA 60 04 00 06 61 04 00 30 61 04 00 3E 61 04 00
408 47 61 04 00 5C 61 04 00 79 61 04 00 91 61 04 00 9F 61 04 00 B0 61 04 00
432 C3 61 04 00 D1 61 04 00 DE 61 04 00 EC 61 04 00 FB 61 04 00 07 62 04 00
456 14 62 04 00 1E 62 04 00 37 62 04 00 44 62 04 00 5B 62 04 00 6B 62 04 00
480 7C 62 04 00 8E 62 04 00 9B 62 04 00 AF 62 04 00 B9 62 04 00 C5 62 04 00
504 D3 62 04 00 E8 62 04 00 F7 62 04 00 03 63 04 00 0B 63 04 00 21 63 04 00
528 36 63 04 00 4A 63 04 00 52 63 04 00 5D 63 04 00 78 63 04 00 7F 63 04 00
552 85 63 04 00 92 63 04 00 99 63 04 00 A0 63 04 00 B0 63 04 00 BC 63 04 00
576 C7 63 04 00 D1 63 04 00 E9 63 04 00 05 64 04 00 0B 64 04 00 1D 64 04 00
600 39 64 04 00 6F 64 04 00 7A 64 04 00 82 64 04 00 9F 64 04 00 A2 64 04 00
624 E0 64 04 00 E9 64 04 00 EF 64 04 00 F2 64 04 00 F6 64 04 00 FD 64 04 00
648 07 65 04 00 0D 65 04 00 18 65 04 00 1E 65 04 00 26 65 04 00 35 65 04 00
672 38 65 04 00 3C 65 04 00 43 65 04 00 4F 65 04 00 54 65 04 00 58 65 04 00
696 60 65 04 00 69 65 04 00 74 65 04 00 7D 65 04 00 81 65 04 00 89 65 04 00
720 94 65 04 00 9D 65 04 00 A4 65 04 00 AC 65 04 00 B4 65 04 00 BE 65 04 00
744 C9 65 04 00 D6 65 04 00 EA 65 04 00 F7 65 04 00 06 66 04 00 0A 66 04 00
768 17 66 04 00 2C 66 04 00 3C 66 04 00 3F 66 04 00 48 66 04 00 59 66 04 00

dex 035 Üöüÿf pÅA°ø<Se
na'Jæfóeh3 p xV4
42 ÷ p j »C
= pM [ Lt x $è
N ¼ 4 42 lZ ~Z
ÄZ ÖZ äZ êZ ßZ æZ
flZ [ [ 9[ Z[ r[
ô[ i[ ,[ Í[ \ *\\
G\ U\ d\ r\ Å\ °\
Ø\ -\ -\ fl\ Ú\ ]
%[ 1] G] K] 0] ô]
£[ 0] ¢[ ð] »] [E]
ÿ] ^ 7^ b^ é^ æ^
Ê^ _ >_ h_ ë_ «_
±_ ´ /´ R´ ^´ k´
u´ ~´ î´ º´ j´ ´
æ´ Ê´ ´´ a´ 0a´ >a
Ga \a ya éa úa »a
√a -a fia Ía ´a b
b b 7b Db [b kb
lb éb öb øb tb »b
*b Êb ´b c c !c
6c Jc Rc ]c xc c
0c ic oc tc »c °c
«c -c Êc d d d
9d od zd Çd üd ¢d
¿d Éd Ōd Úd ´d ´d
e e e e &e 5e
8e <e Ce Oe Te Xe
`e ie te }e Ae de
ie ue $e `e ¥e æe
...e +e Ie `e f f
f ,f <f ?f Hf Yf
```

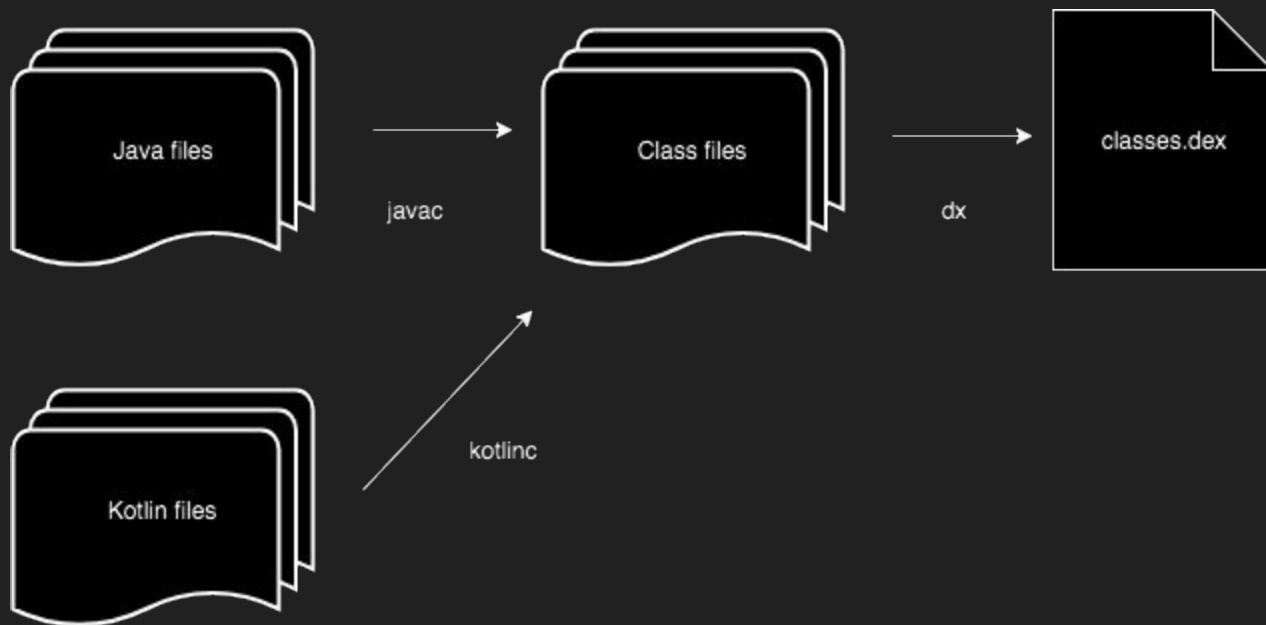
Dex Files

- 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

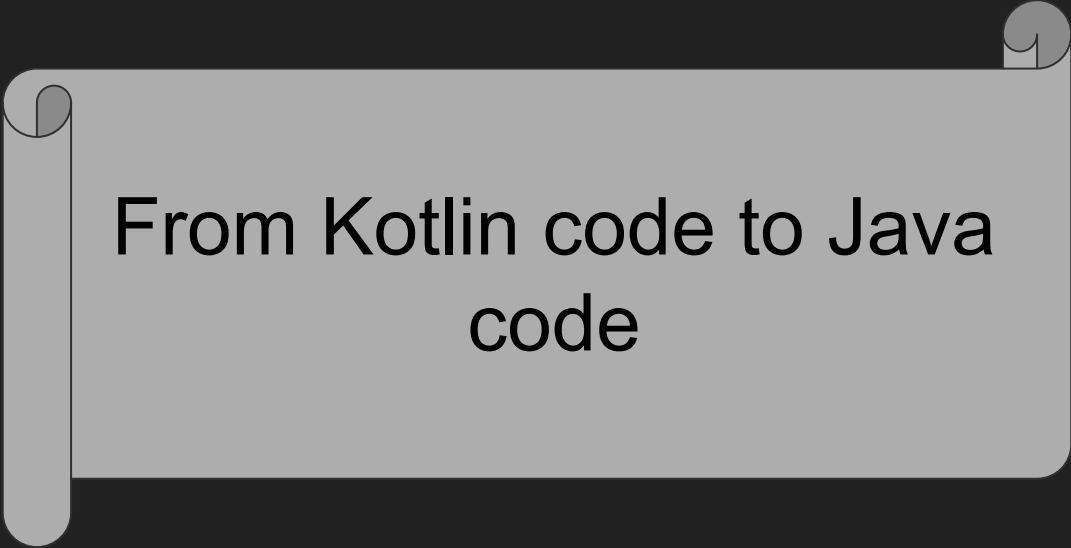
- Kotlin compiler (kotlinc) generates class files
- 1 kt file might have few class files
- 100% interoperable with Java language binary format (class)
- Can be added incrementally

@TTGonda

@BorisFarber

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

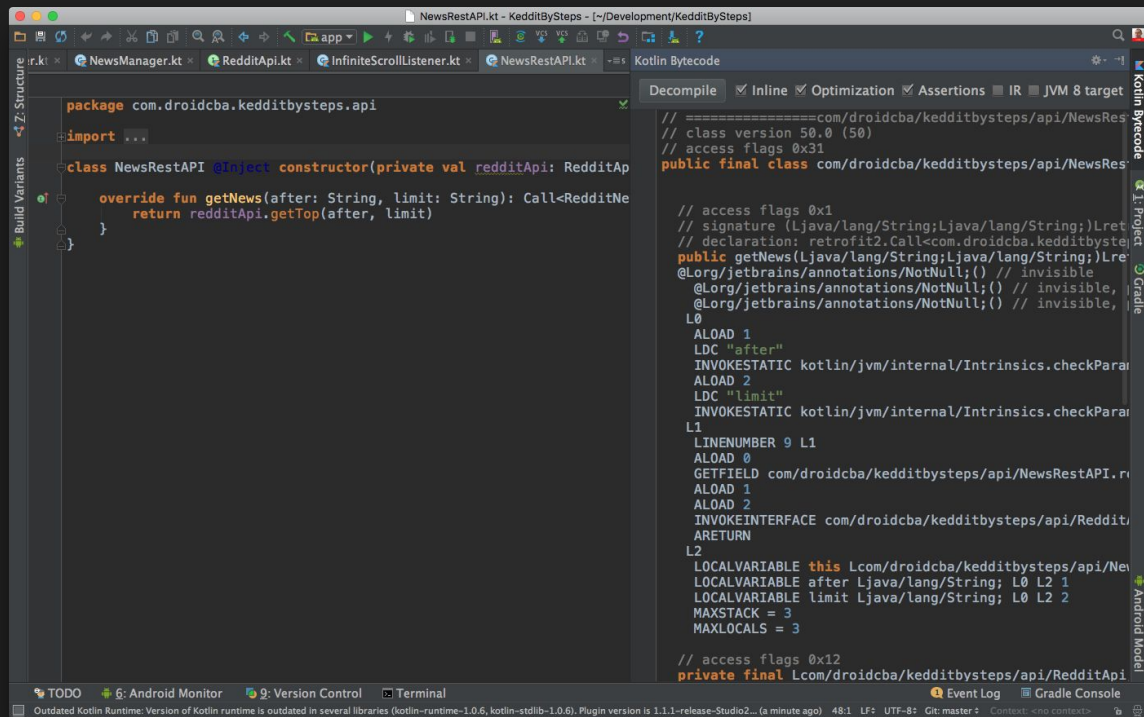


From Kotlin code to Java code

Why

1. Curiosity
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.”

@TTGonda

@BorisFarber

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\u00012\u0006\u0010\u0002\u001a\u00020\u0003\u0006\u0002\u0010\u0004R\u0011\u0010\u0002\u001a\u00020\u0003\u0006\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.checkNotNull(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

```
@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
}
```

Extension Functions

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

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

Extension Functions

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

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
```

Lambdas (inline)

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

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);  
}
```

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);  
}
```

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);  
}
```

```

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.checkNotNull (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>

@TTGonda

@BorisFarber

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