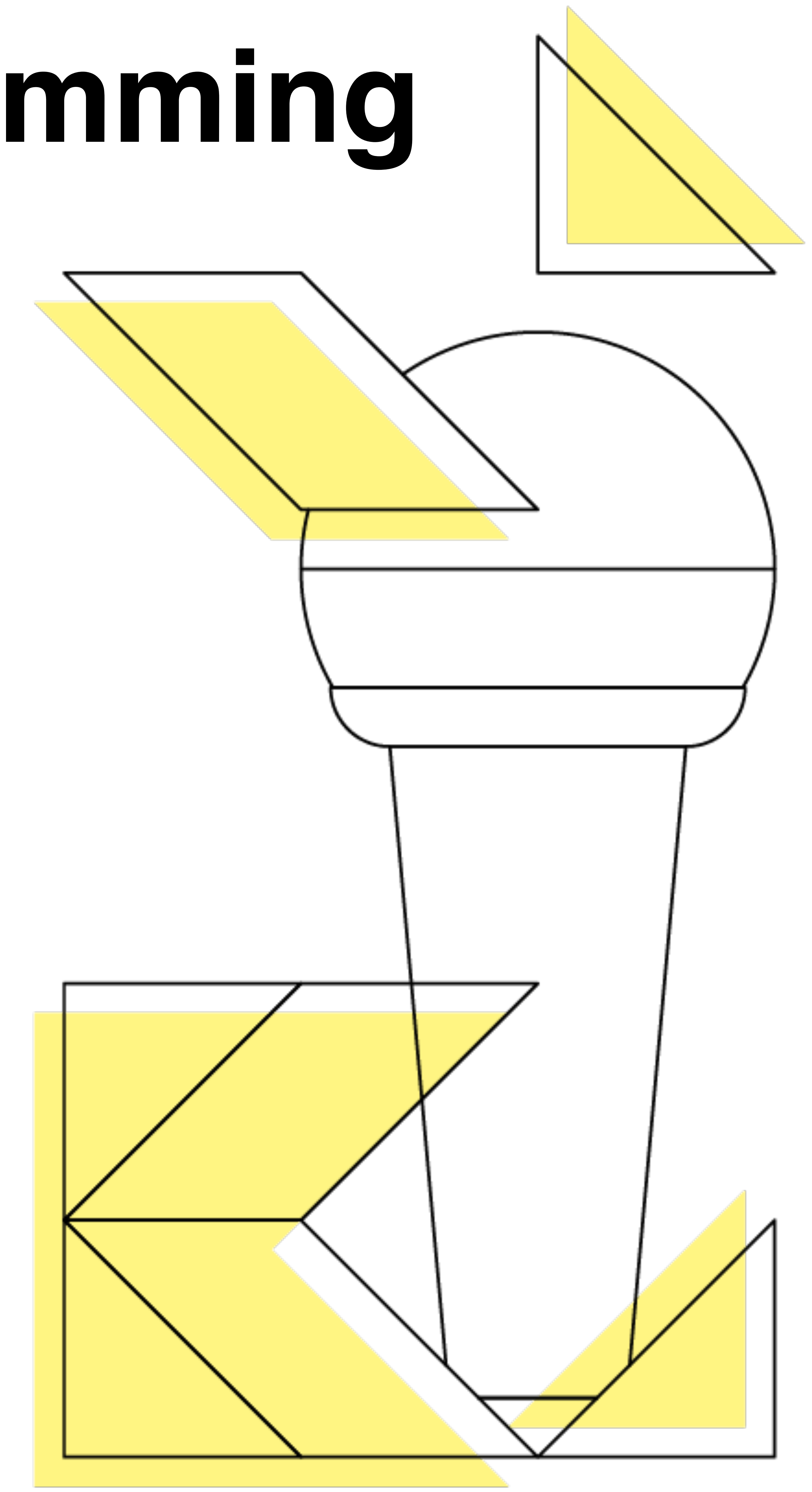
Kotlin High Performance Programming

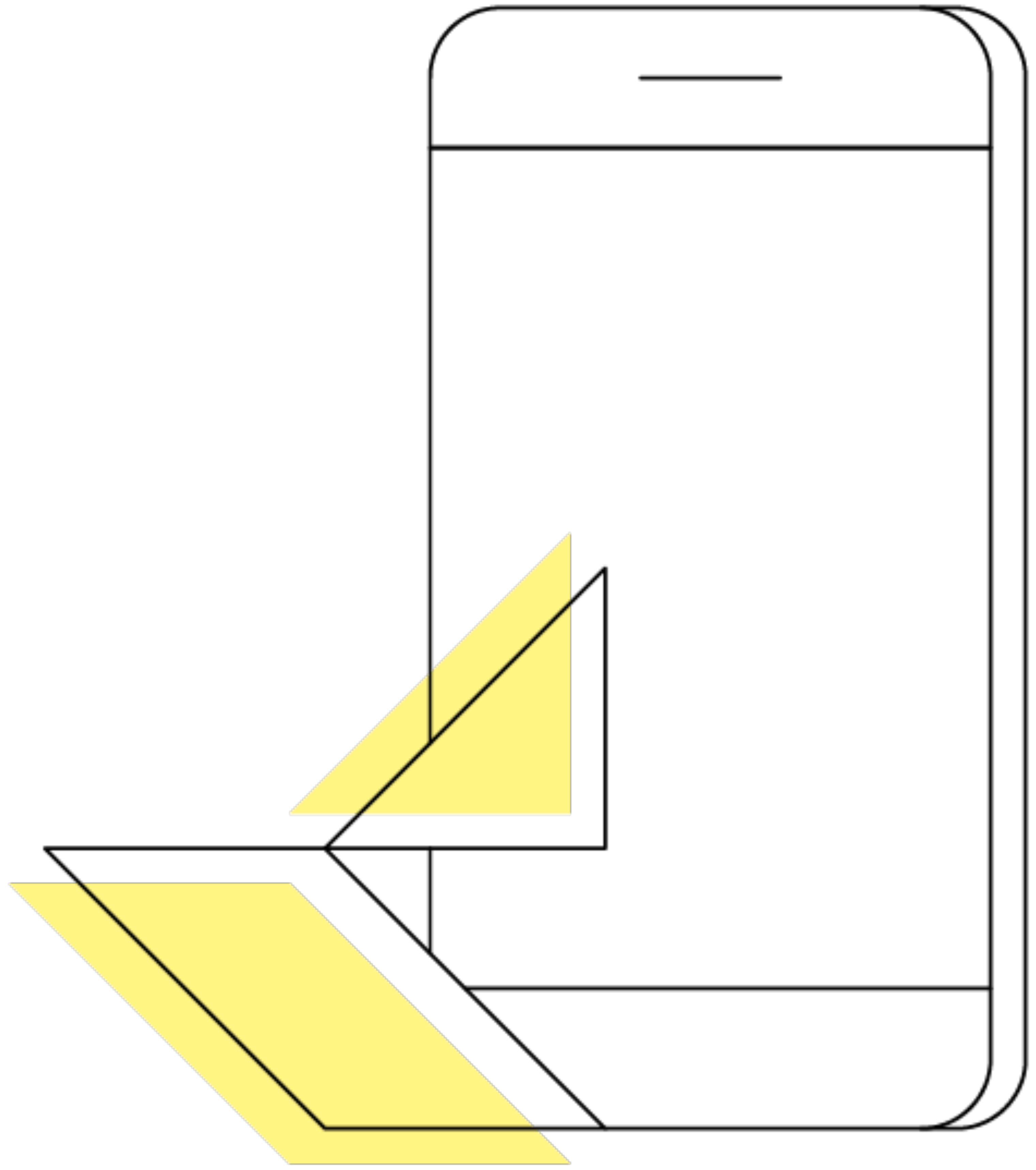
朱涛
Android Developer, Momo Inc.



History

Java 1.0 in 1996

Kotlin 1.0 in 2016



Kotlin, a better Java?

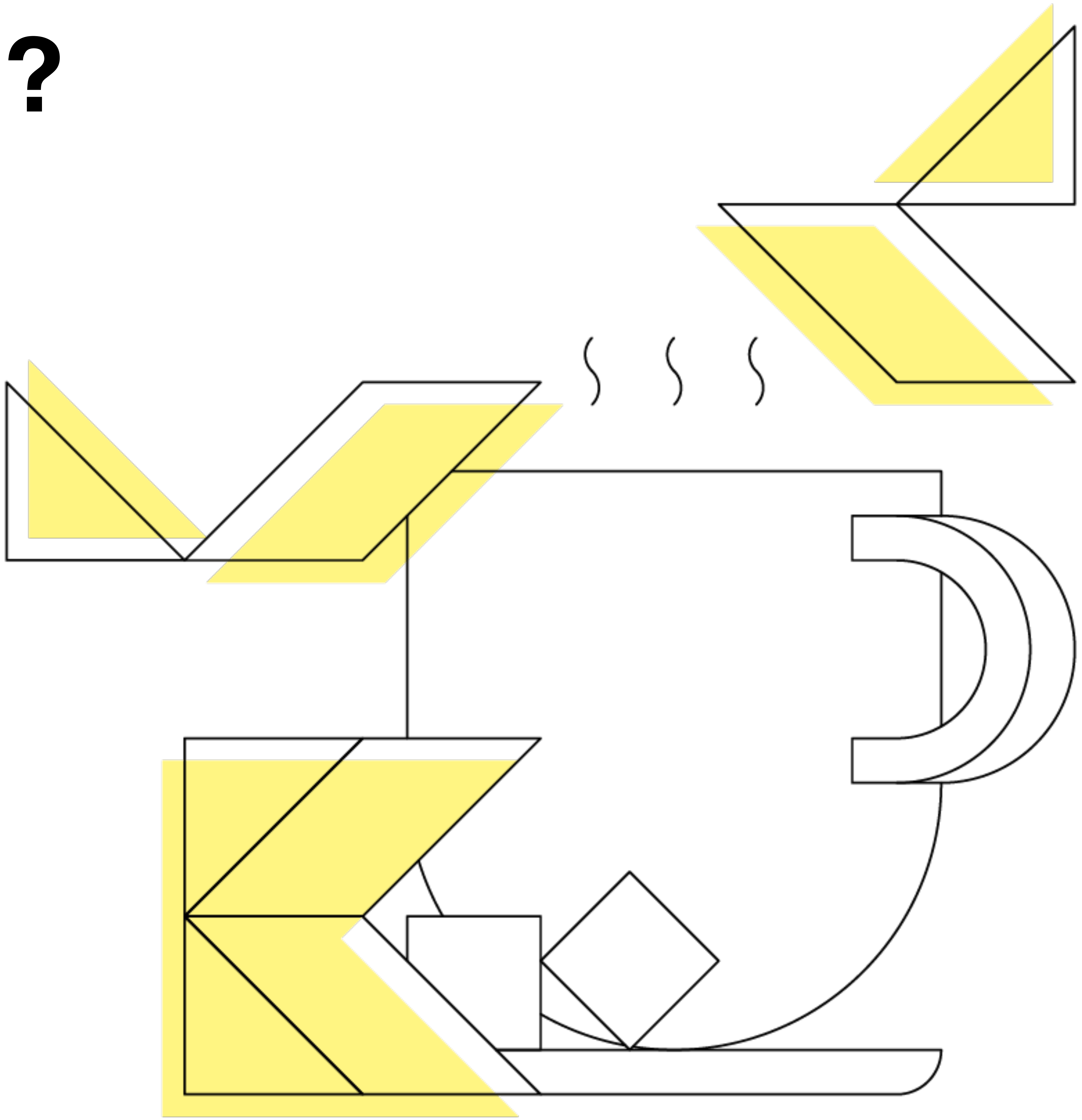
Safer

More Concise

More Productive

Interoperable

Tool-friendly

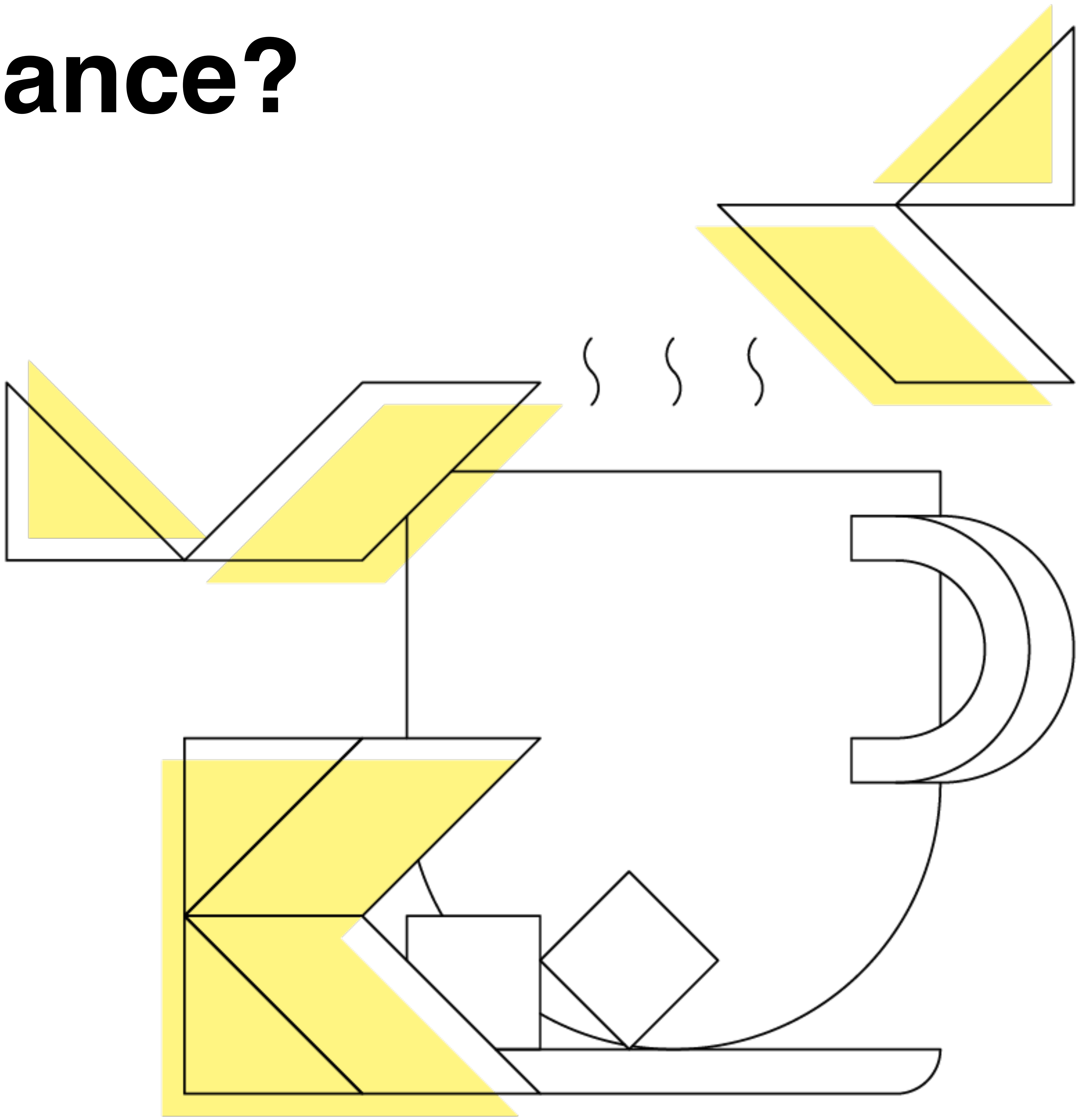


What about performance?

CPU

Memory

FPS



Tools

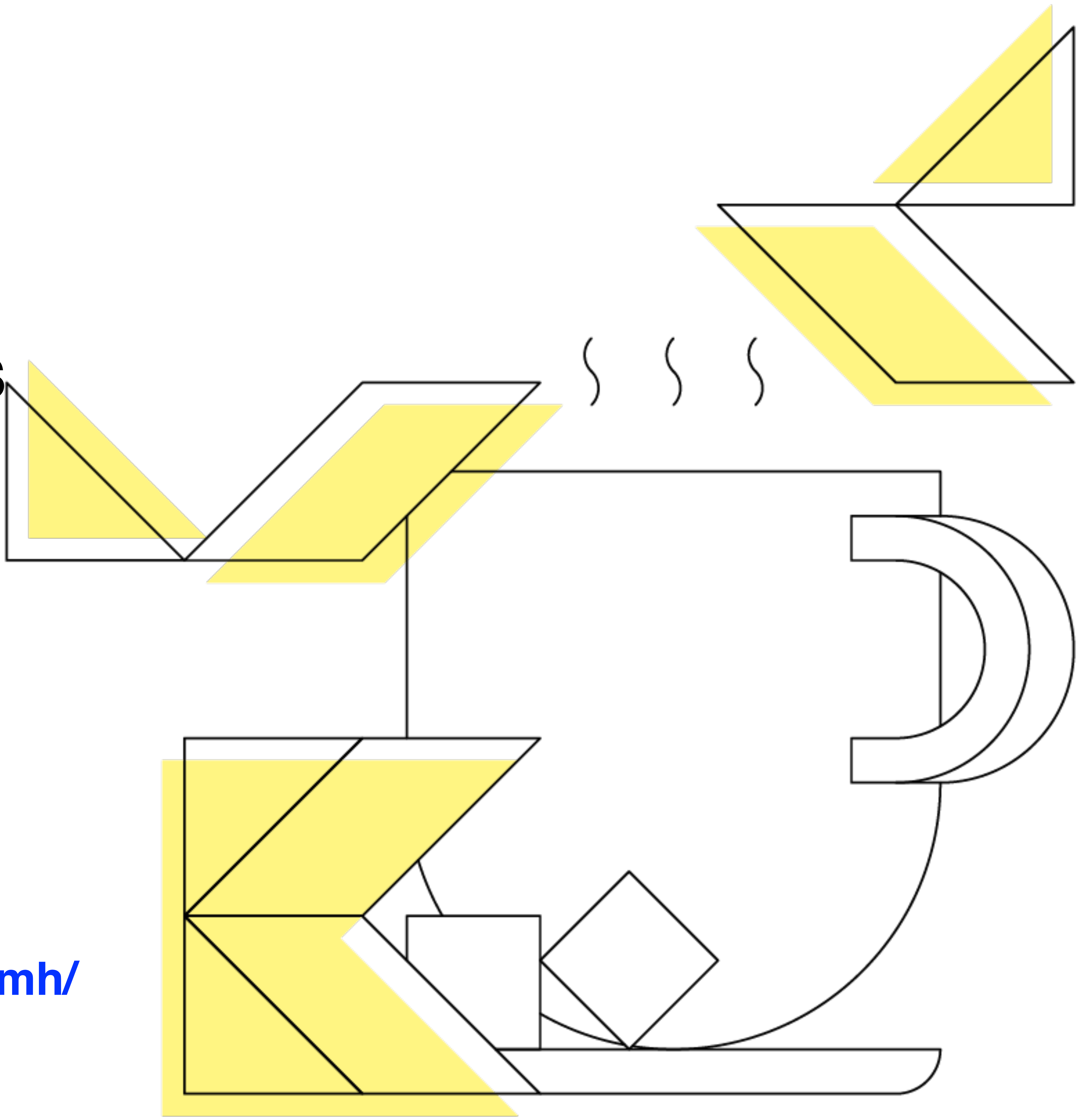
Show Kotlin Bytecode tools

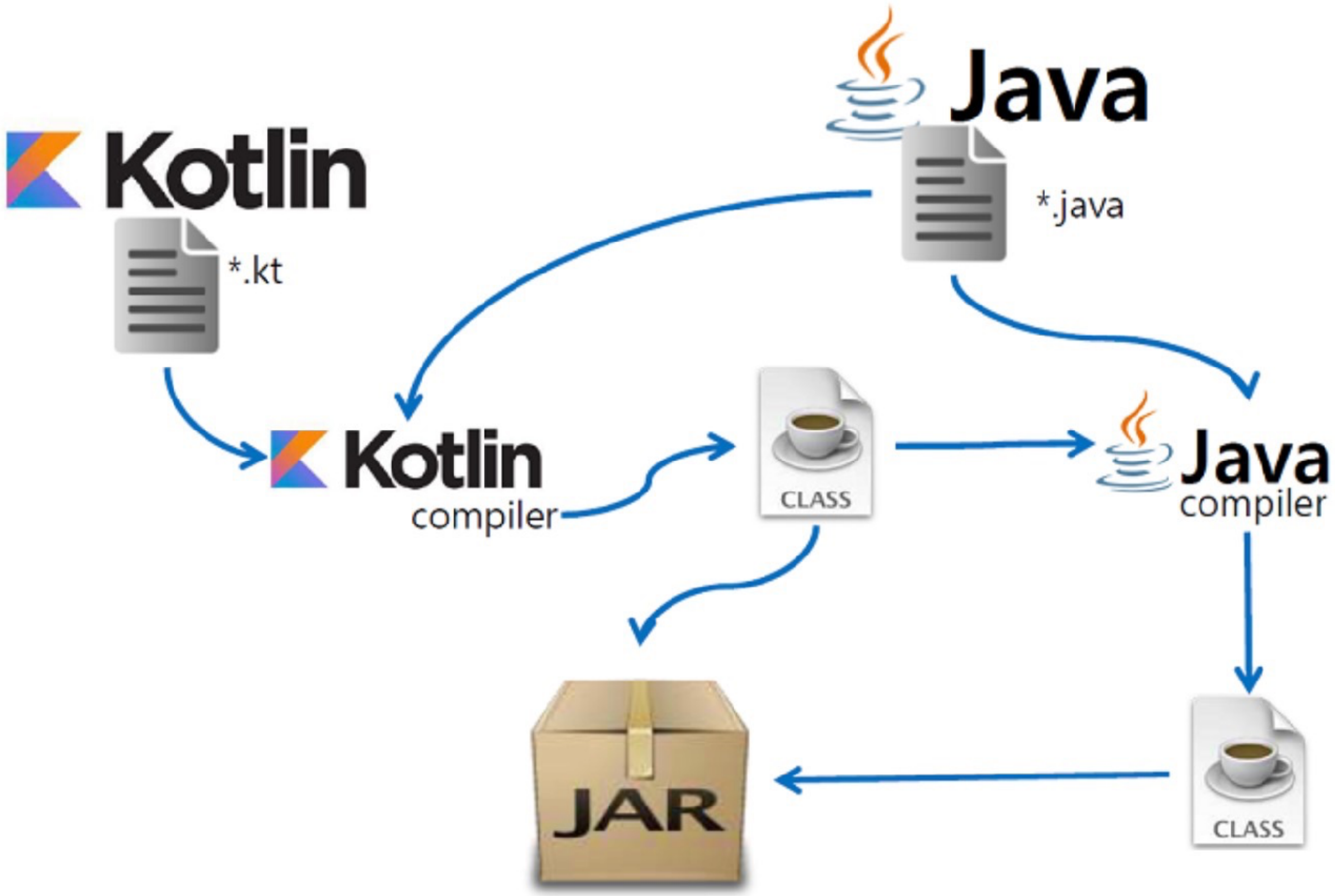
Benchmarks

https://github.com/Kotlin/kotlin-benchmarks

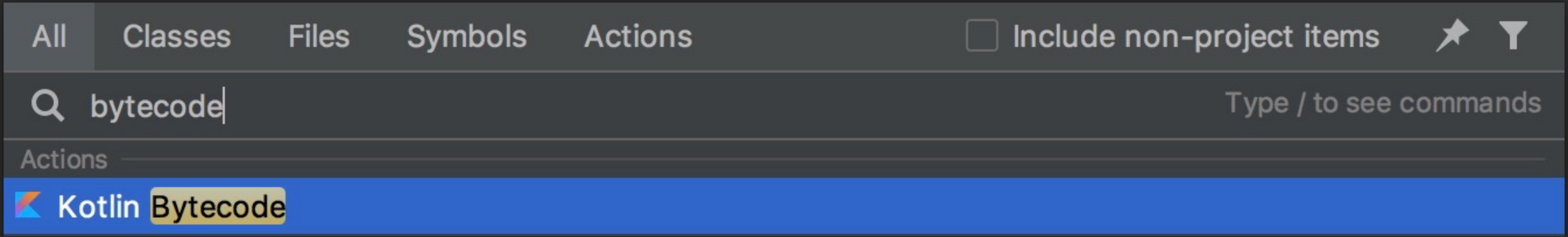
https://developer.android.com/kotlin

https://openjdk.java.net/projects/code-tools/jmh/




Kotlin
*.kt
Java
*.java
Kotlin compiler
CLASS
Java compiler
CLASS
JAR

Show Kotlin Bytecode tools



bytecode

Kotlin Bytecode

Benchmarks

```
apply plugin: 'kotlin-kapt'

…

dependencies {
    implementation 'org.openjdk.jmh:jmh-core:1.21'
    kapt 'org.openjdk.jmh:jmh-generator-annprocess:1.21'
}
```

StringBuilder

Java world…

```java
public static void stringAdd() {
    String string = "";
    for(int i=0;i<10000;i++){
        string += "hello";
    }
}

public static void stringBuilder() {
    StringBuilder builder = new StringBuilder();
    for(int i=0;i<10000;i++){
        builder.append("hello");
    }
}
```

```
public static void stringAdd() {
    String string = "";
    for(int i=0;i<10000;i++){
        string += "hello"; ①
    }
}

public static void stringBuilder() {
    StringBuilder builder = new StringBuilder();
    for(int i=0;i<10000;i++){
        builder.append("hello"); ②
    }
}
```

```
public sta[illegible]oid [illegible]ngAdd() {
    String [illegible];
    for(int i[illegible]000;i++){
        str[illegible]"hello"; ①
    }
}

public static void stringBuilder() {
    StringBuilder builder = new StringBuilder();
    for(int i=0;i<10000;i++){
        builder.append("hello"); ②
    }
}
```

Kotlin world?

```kotlin
fun stringAdd() {
    var string = ""
    for (i in 0..9_999) {
        string += " kotlin"
    }
}
```

```
    SIPUSH 9999
    …
    NEW java/lang/StringBuilder
    DUP
    INVOKESPECIAL java/lang/StringBuilder.<init> ()V
    SWAP
    INVOKEVIRTUAL java/lang/StringBuilder.append (Ljava/
lang/String;)Ljava/lang/StringBuilder;
    LDC " kotlin"
    …
    GOTO L2
```

```
SIPUSH 9999
…
NEW java/lang/StringBuilder ①
DUP
INVOKESPECIAL java/lang/StringBuilder.<init> ②
SWAP
INVOKEVIRTUAL java/lang/StringBuilder.append
(Ljava/lang/String;)Ljava/lang/StringBuilder; ③
LDC " kotlin"
…
GOTO L2
```

```
SIPUSH 9999
…
NEW java/lang/StringBuilder ①
DUP
INVOKESPECIAL java/lang/StringBuilder.<init> ②
SWAP
INVOKEVIRTUAL java[illegible]g/St[illegible]gBuilder.append ③
(Ljava/lang/String;[illegible]ang/StringBuilder;
LDC " kotlin"
…
GOTO L2
```

```kotlin
fun stringBuilder() {
    val stringBuilder = StringBuilder()
    for (i in 0..9_999) {
        stringBuilder.append(" kotlin")
    }
}
```

```
NEW java/lang/StringBuilder
DUP
INVOKESPECIAL java/lang/StringBuilder.<init> ()V
…
SIPUSH 9999
ISTORE 2
…
LDC " kotlin"
INVOKEVIRTUAL java/lang/StringBuilder.append (Ljava/lang/
String;)Ljava/lang/StringBuilder;
…
GOTO L2
```

```
NEW java/lang/StringBuilder ①
DUP
INVOKESPECIAL java/lang/StringBuilder.<init>②
…
SIPUSH 9999
ISTORE 2
…
LDC " kotlin"
INVOKEVIRTUAL java/lang/StringBuilder.append ③
(Ljava/lang/String;)Ljava/lang/StringBuilder;
…
GOTO L2
```

```
NEW java/lang/StringBuilder ①
DUP
INVOKESPECIAL java/lang/StringBuilder.<init>②
…
SIPUSH 9999
ISTORE 2
…
LDC " kotlin"                                    ③
INVOKEVIRTUAL java/lang/StringBuilder.append
(Ljava/lang/String;)Ljava/lang/StringBuilder;
…
GOTO L2
```

1. Use StringBuilder in loop

1. Use StringBuilder in loop

— Java coding experience is helpful.

Primitive

float

Float

int

Integer

double

Double

byte

short

Short

Byte

char

Character

Primitive

Float

int Int float Float

Integer

double Double

Double

byte Byte short

Byte Short

Short

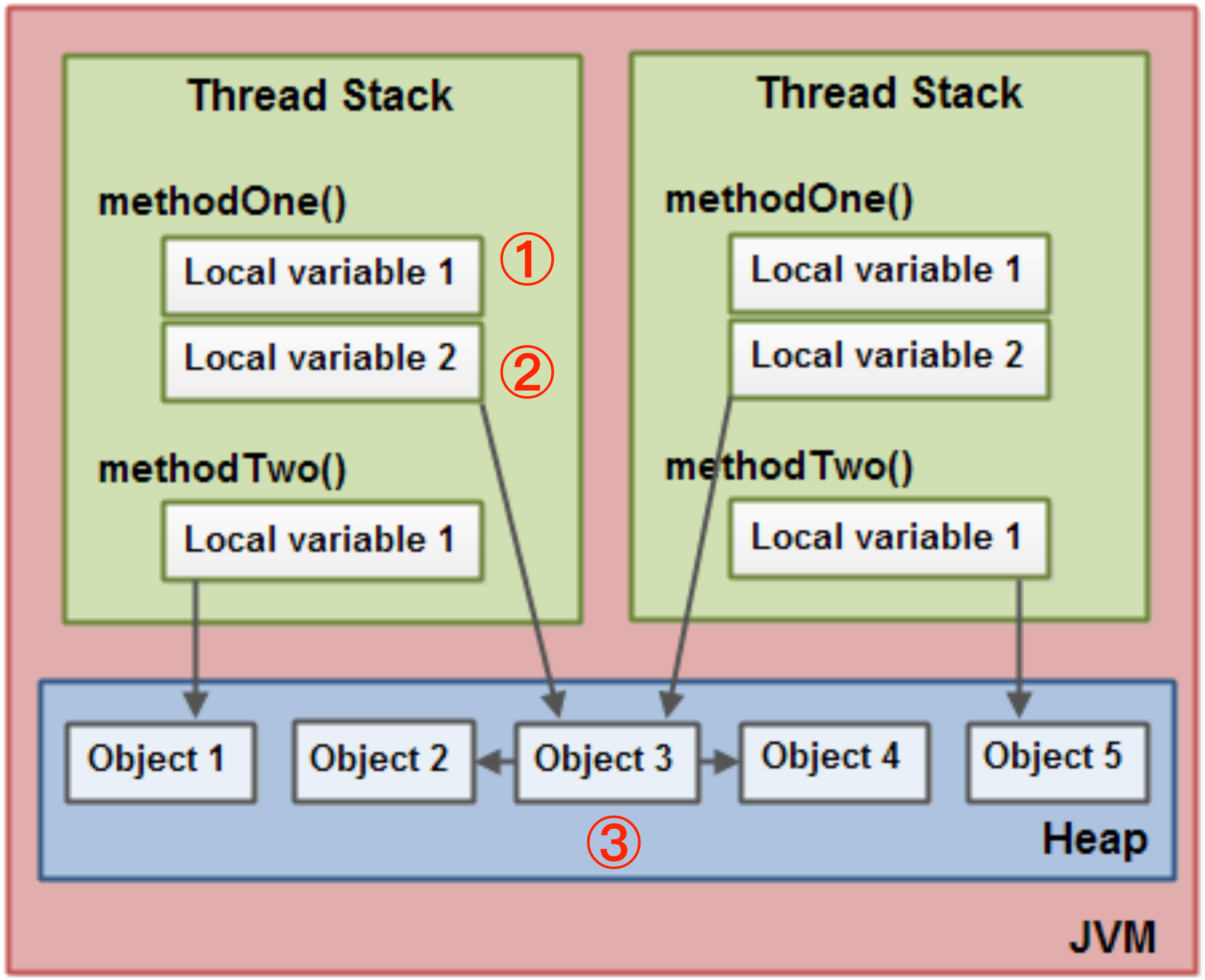
char Char

Character

Primitive

Primitive or Wrapper

Primitive	Wrapper
int, float…	Integer, Float…
Better Performance	Object Oriented, null instead of 0
Stack	Heap


Thread Stack
methodOne()
Local variable 1
①
Local variable 2
②
methodTwo()
Local variable 1
Thread Stack
methodOne()
Local variable 1
Local variable 2
methodTwo()
Local variable 1
Object 1
Object 2
Object 3
Object 4
Object 5
③
Heap
JVM

```kotlin
fun funInt() {
    val count: Int = 1
    var nullCount: Int? = null
    nullCount = 1
}
```

```
fun funInt() {
    val count: Int = 1
    var nullCount: Int? = null
    nullCount = 1
}

public static final void funInt() {
   int count = 1;
   Integer nullCount = null;
   nullCount = 1;
}
```

Array

Java	Kotlin
int / Integer	Int
int[] / Integer[]	??

Java	Kotlin
int / Integer	Int
int[] / Integer[]	IntArray / Array<Int>

```kotlin
fun primitiveObject() {
    var array = arrayOf(1, 2, 3, 4, 5)
}
```

```
fun primitiveObject() {
    var array = arrayOf(1, 2, 3, 4, 5)
}

public static final void primitiveObject() {
   Integer[] array = new Integer[]{1, 2, 3, 4,
5};
}
```

```
fun primitiveObject() {
    var array = arrayOf(1, 2, 3, 4, 5)
}

public static final void primitiveObject() {
    Integer[] array = new Integer[]{1, 2, 3, 4,
5};
}
```

```
fun primitiveObject() {                  Array<Int>
    var array = arrayOf(1, 2, 3, 4, 5)
}

public static final void primitiveObject() {
   Integer[] array = new Integer[]{1, 2, 3, 4,
5};
}
```

```
ANEWARRAY java/lang/Integer
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
```

```
ANEWARRAY java/lang/Integer
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf
(I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
INVOKESTATIC java/lang/Integer.valueOf (I)Ljava/lang/Integer;
…
```

Boxing

```kotlin
fun primitive() {
    val array = intArrayOf(1, 2, 3, 4, 5)
}
```

```kotlin
fun primitive() {
    val array = intArrayOf(1, 2, 3, 4, 5)
}
```

```java
public static final void primitive() {
   int[] var10000 = new int[]{1, 2, 3, 4, 5};
}
```

```kotlin
fun primitive() {
    val array = intArrayOf(1, 2, 3, 4, 5)
}
```

```java
public static final void primitive() {
   int[] var10000 = new int[]{1, 2, 3, 4, 5};
}
```

```
fun primitive() {
    val array = intArrayOf(1, 2, 3, 4, 5)
}                                   IntArray


public static final void primitive() {
    int[] var10000 = new int[]{1, 2, 3, 4, 5};
}
```

```
NEWARRAY T_INT
DUP
ICONST_0
ICONST_1
IASTORE
DUP
ICONST_1
ICONST_2
IASTORE
DUP
ICONST_2
ICONST_3
IASTORE
DUP
ICONST_3
ICONST_4
…
```

```
NEWARRAY T_INT
DUP
ICONST_0
ICONST_1
IASTORE
ICONST_1
ICONST_2
IASTORE
ICONST_2
ICONST_3
IASTORE
ICONST_3
ICONST_4
…
```

No Boxing

2. Try to use primitive type

2. Try to use primitive type
3. Use primitive array, instead of Array<T>

2. Try to use primitive type
3. Use primitive array, instead of Array<T>

— We should be more careful in Kotlin.

Inline functions

```kotlin
inline fun log(message: String) {
    Log.i(TAG, message)
}

fun main() {
    log("Hello")
    log("world")
}
```

```
inline fun log(message: String) {
    Log.i(TAG, message)
}
fun main() {
    log("Hello")
    log("world")
}
public static final void main(…) {
   Log.i(TAG, “Hello");
   Log.i(TAG, "world");
}
```

inlining works best for functions with parameters of functional types inlining works best for functions with parameters of functional types

How?

```
inline fun repeat(times: Int, action: (Int) -> Unit) {
    for (index in 0 until times) {
        action(index)
    }
}
```

```kotlin
inline fun repeat(…, action: (Int) -> Unit) {
    for (index in 0 until times) {
        action(index)
    }
}
```

How inline works?

```kotlin
fun noInlineRepeat(times: Int, action: (Int) -> Unit) {
    for (index in 0 until times) {
        action(index)
    }
}
```

```kotlin
fun main() {
    repeat(100_000_000) {
        count = it
    }
    noInlineRepeat(100_000_000) {
        count = it
    }
}
```

```
public static final void main(@NotNull String[] args) {
    int times = 100000000;
    int index = 0;
    for(int i = times; index < i; ++index) {
        count = index;
    }

    Function1 lambda = new MyInlineKt$lambda();
    noInlineRepeat(100000000, lambda);
}
```

```
public static final void main(@NotNull String[] args) {
    int times = 100000000;
    int index = 0;
    for(int i = times; index < i; ++index) {
        count = index;
    }

    Function1 lambda = new MyInlineKt$lambda();
    noInlineRepeat(100000000, lambda);
}
```

```java
public static final void main(@NotNull String[] args) {
    int times = 100000000;
    int index = 0;
    for(int i = times; index < i; ++index) {
        count = index;
    }

    Function1 lambda = new MyInlineKt$lambda();
    noInlineRepeat(100000000, lambda);
}
```

```java
public static final void noInlineRepeat(int times,
Function1 action) {
    Intrinsics.checkParameterIsNotNull(action, "action");
    int index = 0;
    for(int i = times; index < i; ++index) {
        action.invoke(index);
    }
}
```

```java
public static final void noInlineRepeat(int times,
Function1 action) {
    Intrinsics.checkParameterIsNotNull(action, "action");
    int index = 0;
    for(int i = times; index < i; ++index) {
        action.invoke(index);
    }
}
```

```
public static final void main(@NotNull String[] args) {
    int times = 100000000;
    int index = 0;
    for(int i = times; index < i; ++index) {
        count = index;
    }

    Function1 lambda = new MyInlineKt$lambda();
    noInlineRepeat(100000000, lambda);
}
```

```
public static class MyInlineKt$lambda implements Function1 {
    @Override
    public Object invoke(Object o) {
        return null;
    }
}

public interface Function1<in P1, out R> : Function<R> {
    /** Invokes the function with the specified argument.*/
    public operator fun invoke(p1: P1): R
}
```

```
… class MyInlineKt$lambda implements Function1 {
    @Override
    public Object invoke(Object o) {
        return null;
    }
}

… interface Function1<in P1, out R> : Function<R> {
    /** Invokes the function with the specified argument.*/
    public operator fun invoke(p1: P1): R
}
```

A new class

```
… class MyInlineKt$lambda implements Function1 {
    @Override
    public Object invoke(Object o) {
        return null;
    }
}

… interface Function1<in P1, out R> : Function<R> {
    /** Invokes the function with the specified argument.*/
    public operator fun invoke(p1: P1): R
}
```

```
public static final void main(@NotNull String[] args) {
    int times = 100000000;
    int index = 0;
    for(int i = times; index < i; ++index) {
        count = index;
    }
                                  New object
    Function1 lambda = new MyInlineKt$lambda();
    noInlineRepeat(100000000, lambda);
}
          Function call
```

	inline	noInline
New class	No	Yes
New object	No	Yes
Function call	No	Yes

Benchmark

```kotlin
@Benchmark
fun testInline() {
    repeat(100_000_000) {
        count = it
    }
}

@Benchmark
fun testNonInline() {
    noInlineRepeat(100_000_000) {
        count = it
    }
}
```

```
Benchmark                  Score           Error
testInline           3314916.252 ±     55640.758
testNonInline      250525218.556 ±   4708169.758
```

4. Use inline with higher-order function

4. Use inline with higher-order function

— Kotlin can be better.

More…

@JvmField is helpful

Don’t abuse null safe

Don’t abuse lateinit

Immutability is preferred

Manage top-level constant and function

More…

Be careful with closure

Be careful with companion objects

Be careful with forEach on range

Consider using inline classes

And more…

Take advantage of lazy delegation

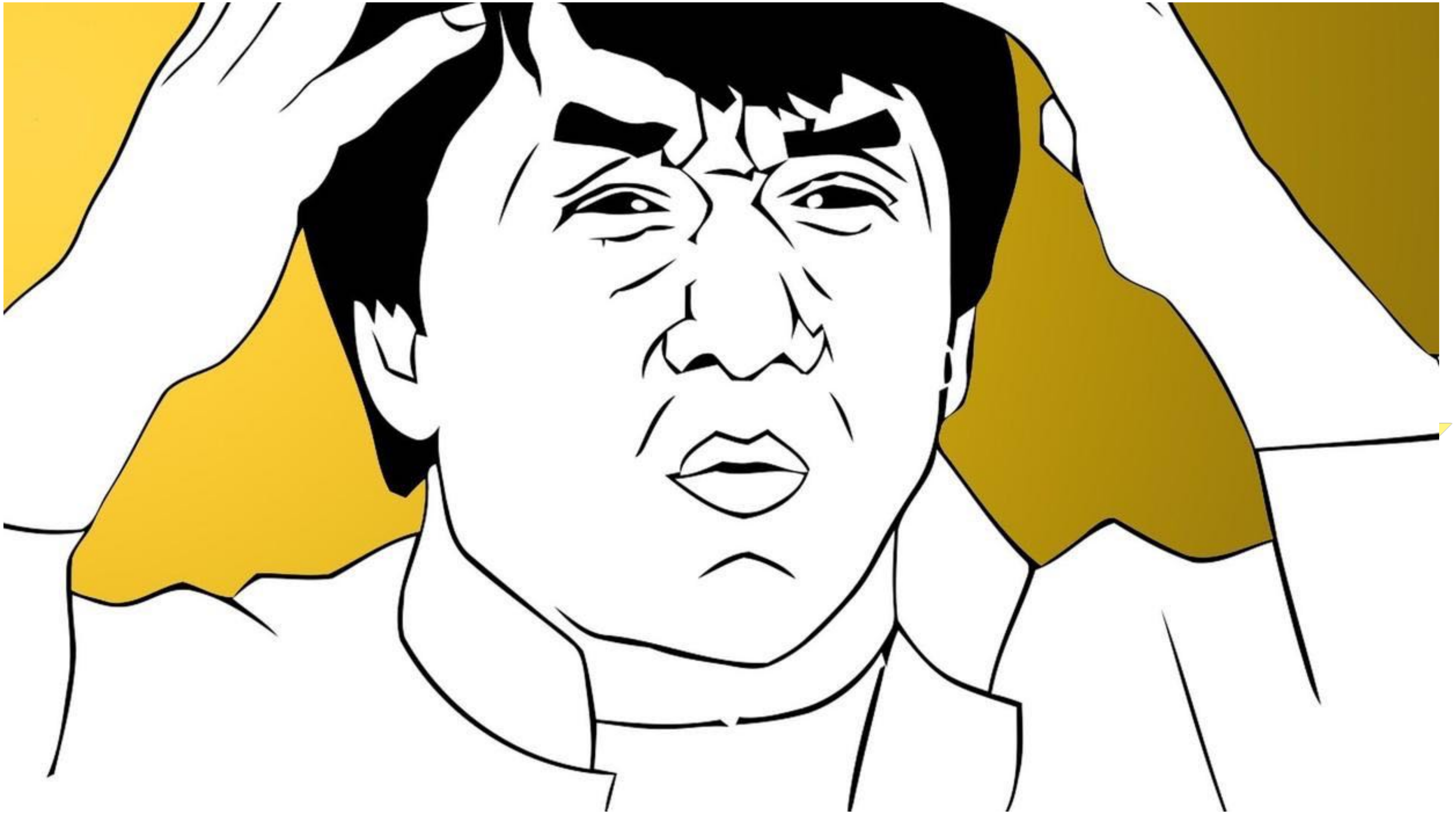
Consider using LazyThreadSafetyMode explicitly

Consider using Backing Property to optimize access

Be aware of Kotlin Collection under the hood

Prefer Sequence for big collections

And…



Solution?

Solution?

Static Code Analysis Tools

With the power of Android Lint.

```
/**
 * Created by zhu.tao on 2019-08-19.
 */
fun test(array: Array<Int>) {

}
```

With the power of Android Lint.

```
/**
 * Created by zhu.tao on 2019-08-19.
 */
fun test(array: Array<Int>) {

}
```

Use primitive array, instead of Array<T> more... (⌘F1)

With the power of Android Lint.

```
/**
 * Created by zhu.tao on 2019-08-19.
 */
fun test(array: IntArray) {

}
```

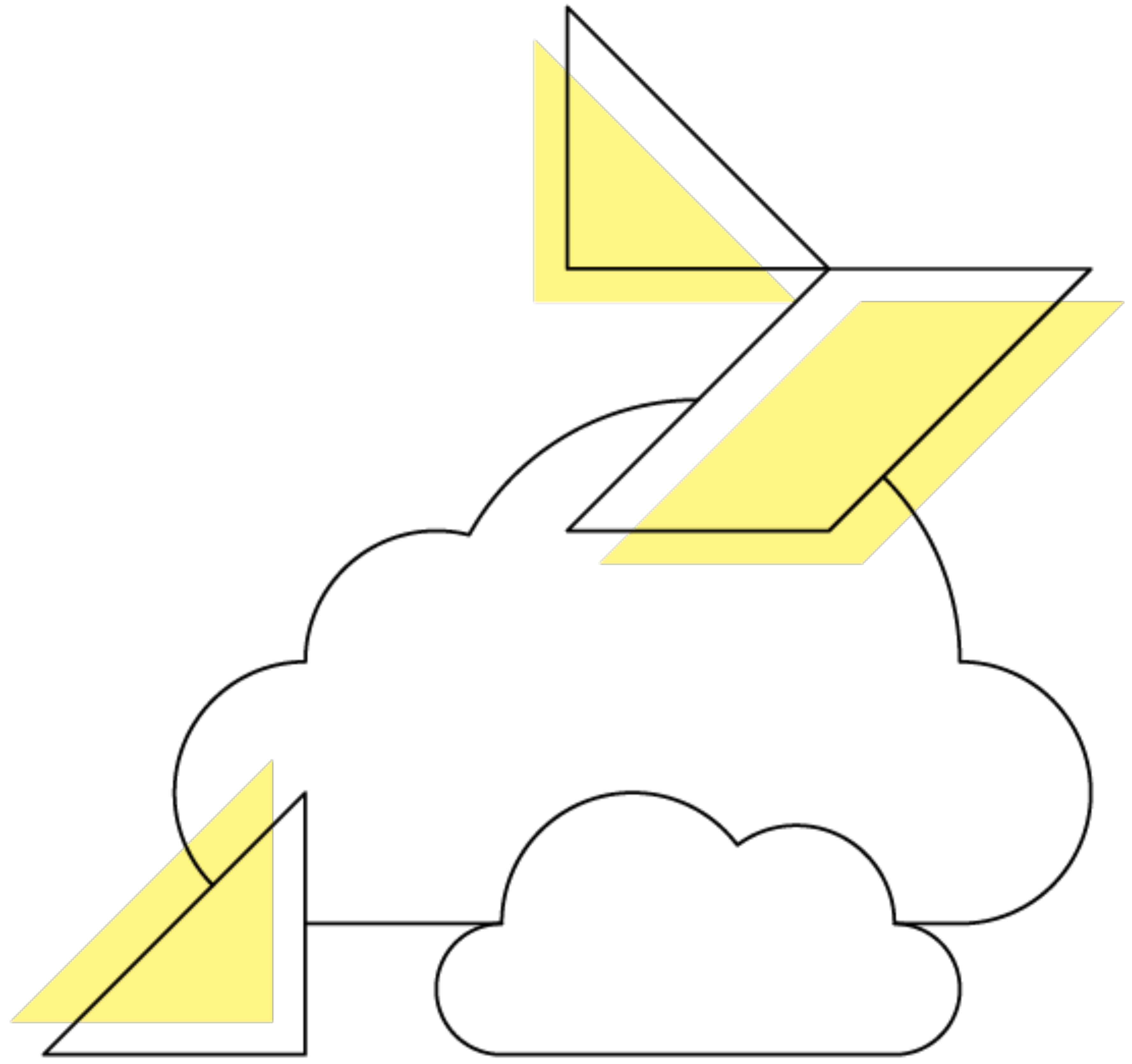
Reference

https://kotlinlang.org/docs/reference/

https://developer.android.com/studio/write/lint

https://groups.google.com/forum/#!forum/lint-dev

https://openjdk.java.net/projects/code-tools/jmh/



Thank you

Wechat: GrabSky