USE TECH

Java-Swift interop vs Kotlin-Swift export. Путь к сингулярности

Анна Жаркова
Lead Mobile developer

Обо мне

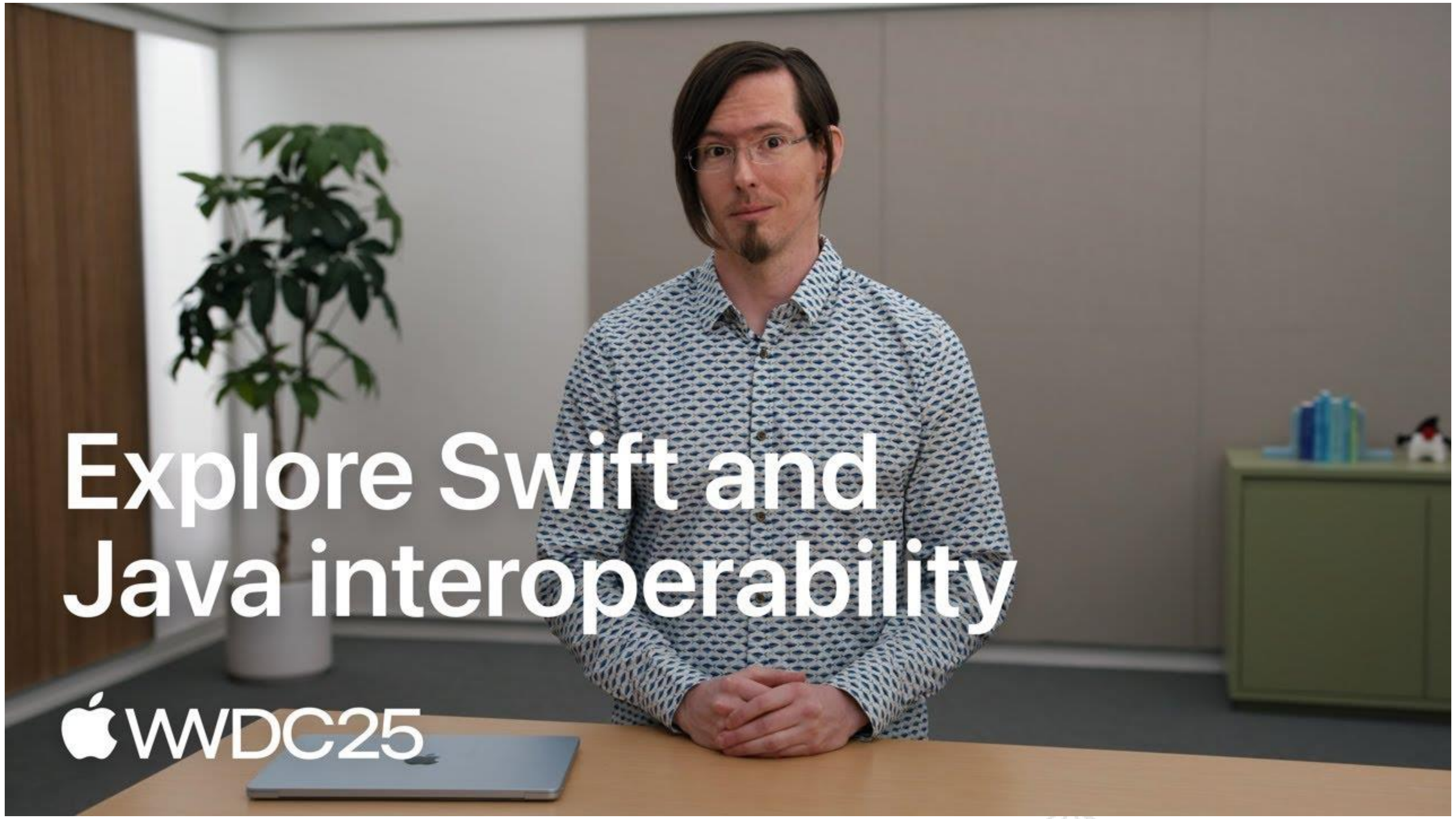
- В мобильной разработке с 2013
- Ведущий мобильный разработчик в Usetech
- Нативная разработка под iOS и Android (Swift/Objective-C, Kotlin/Java) кросс-платформа (Xamarin, Kotlin multiplatform)
- Ментор, управляю командой направления
- Спикер на конференциях AppsConf, Mobius, TechTrain, DroidCon (2022)
- Преподаватель в Otus (iOS Pro и базовый)
- Автор статей по мобильной разработке (SwiftUI, iOS, KMM)
- Автор книги «Kotlin Multiplatform на практике»

Обсудим:

- Swift-Java, основная концепция
- Нюансы компиляции Swift-Java/Kotlin
- Из Java в Swift
- Из Swift в Java
- Из Swift в Android
- Kotlin-Swift Export

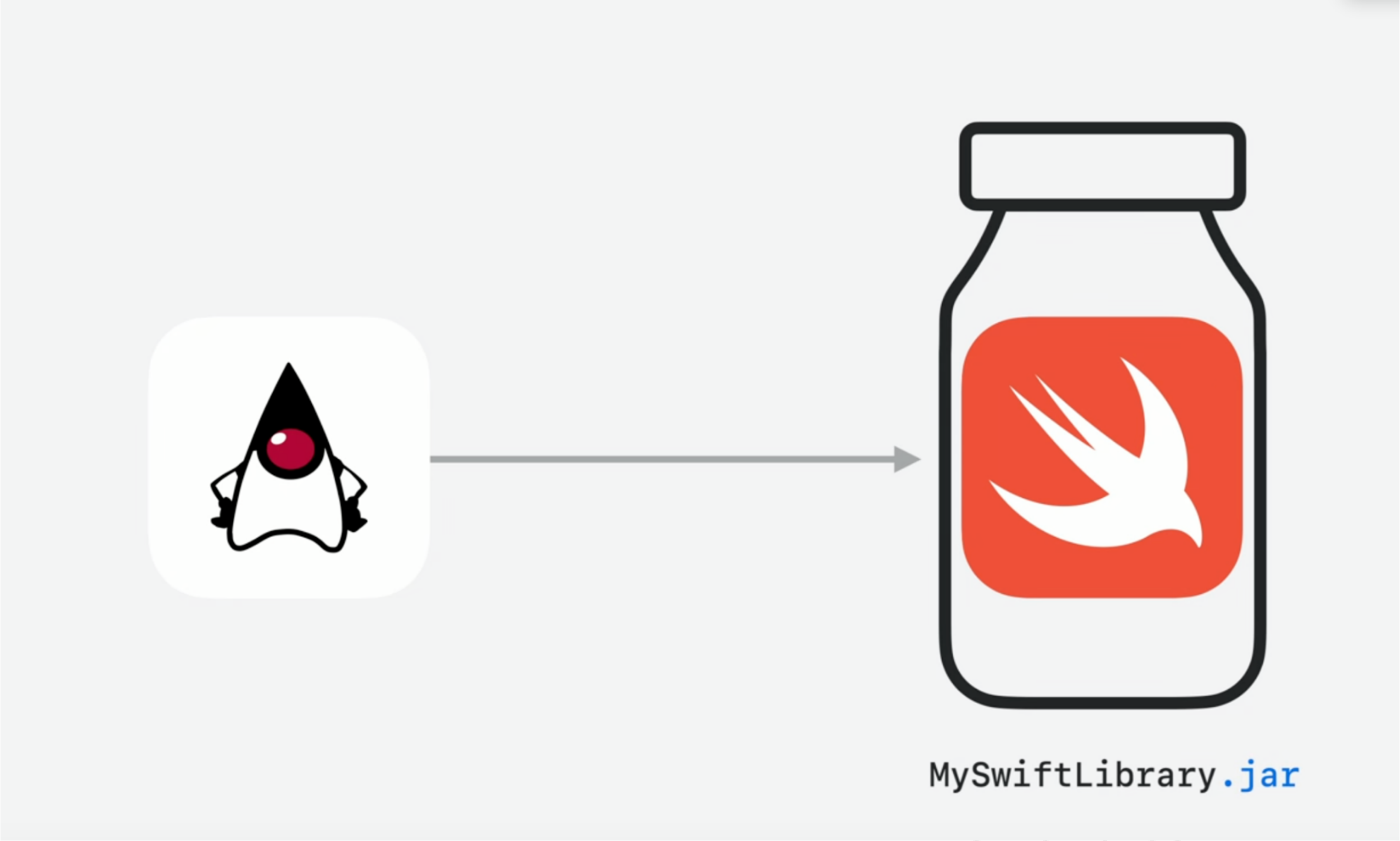
Swift-Java Interoperability. WWDC 2025

https://github.com/swiftlang/swift-java



Swift-Java Interoperability. WWDC 2025

https://github.com/swiftlang/swift-java



Swift-Java Interoperability. WWDC 2025

Swift Package

- **JavaKit** – Swift libraries and macros

Java Library

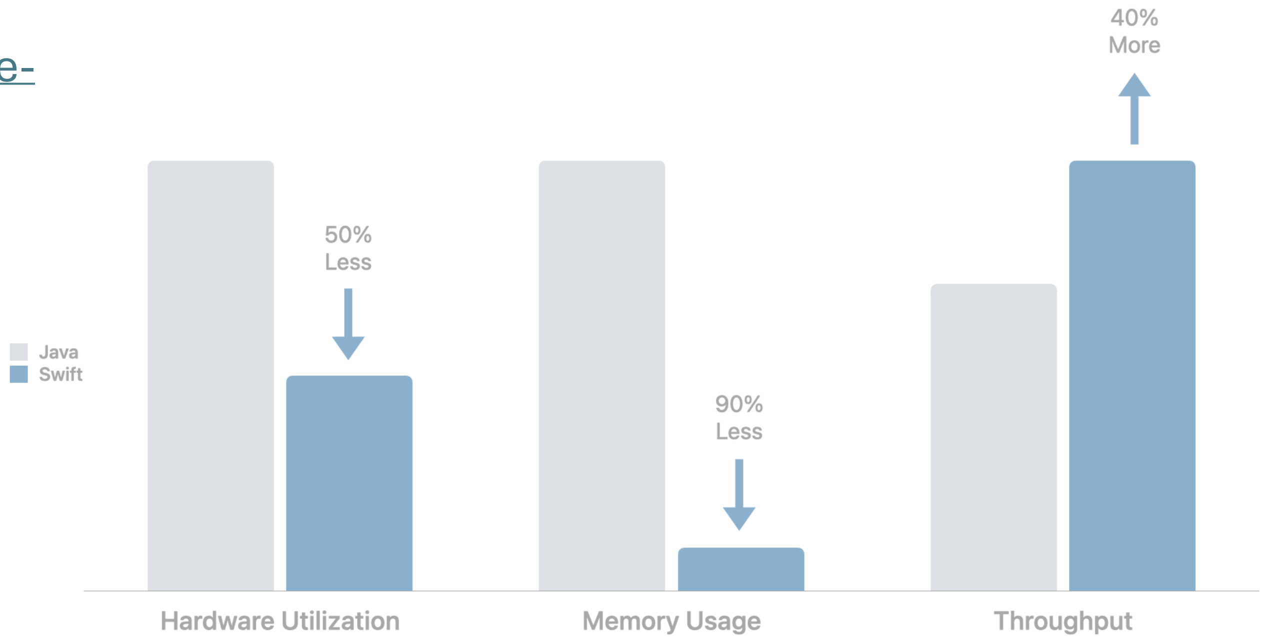
- **SwiftKit** – Java library for efficient Swift interoperability

Зачем

Миграция приложений с Java на Swift:

- Password Monitoring

https://www.swift.org/blog/swift-at-apple-migrating-the-password-monitoring-service-from-java/

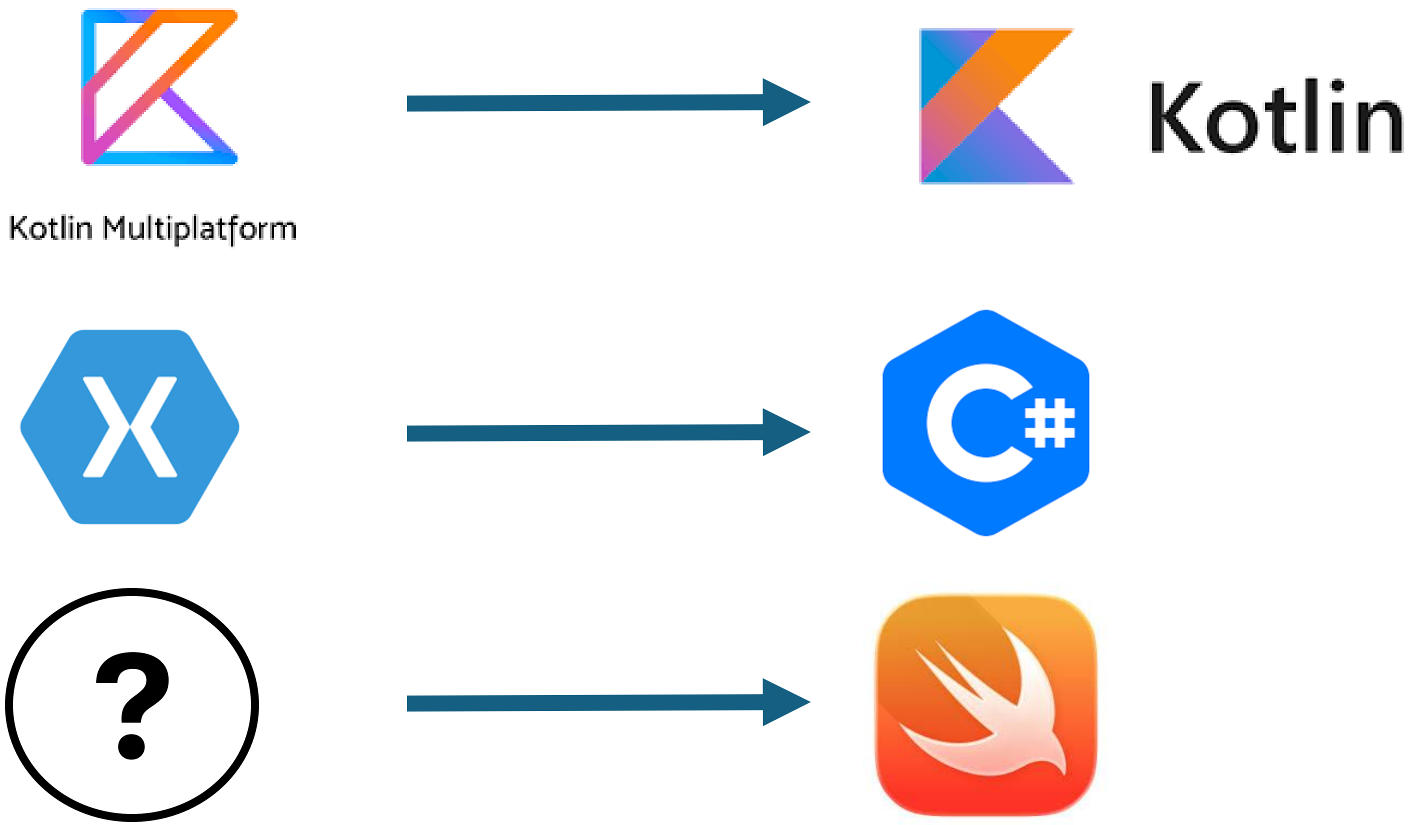


USETECH

Зачем

- Общая кодовая база без кросс-платформы
- Работы по сближению Kotlin и Swift

Кросс-платформа для натива

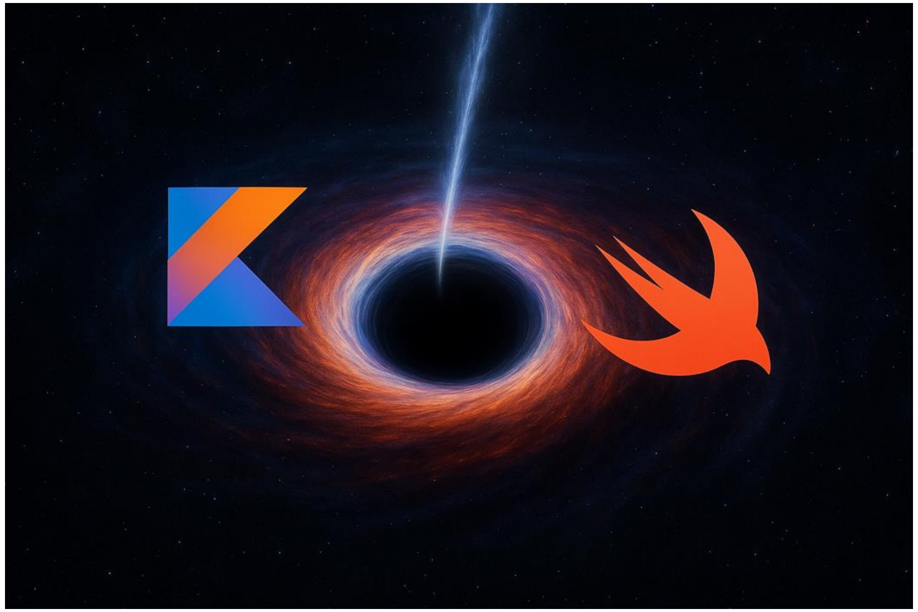


Kotlin Multiplatform

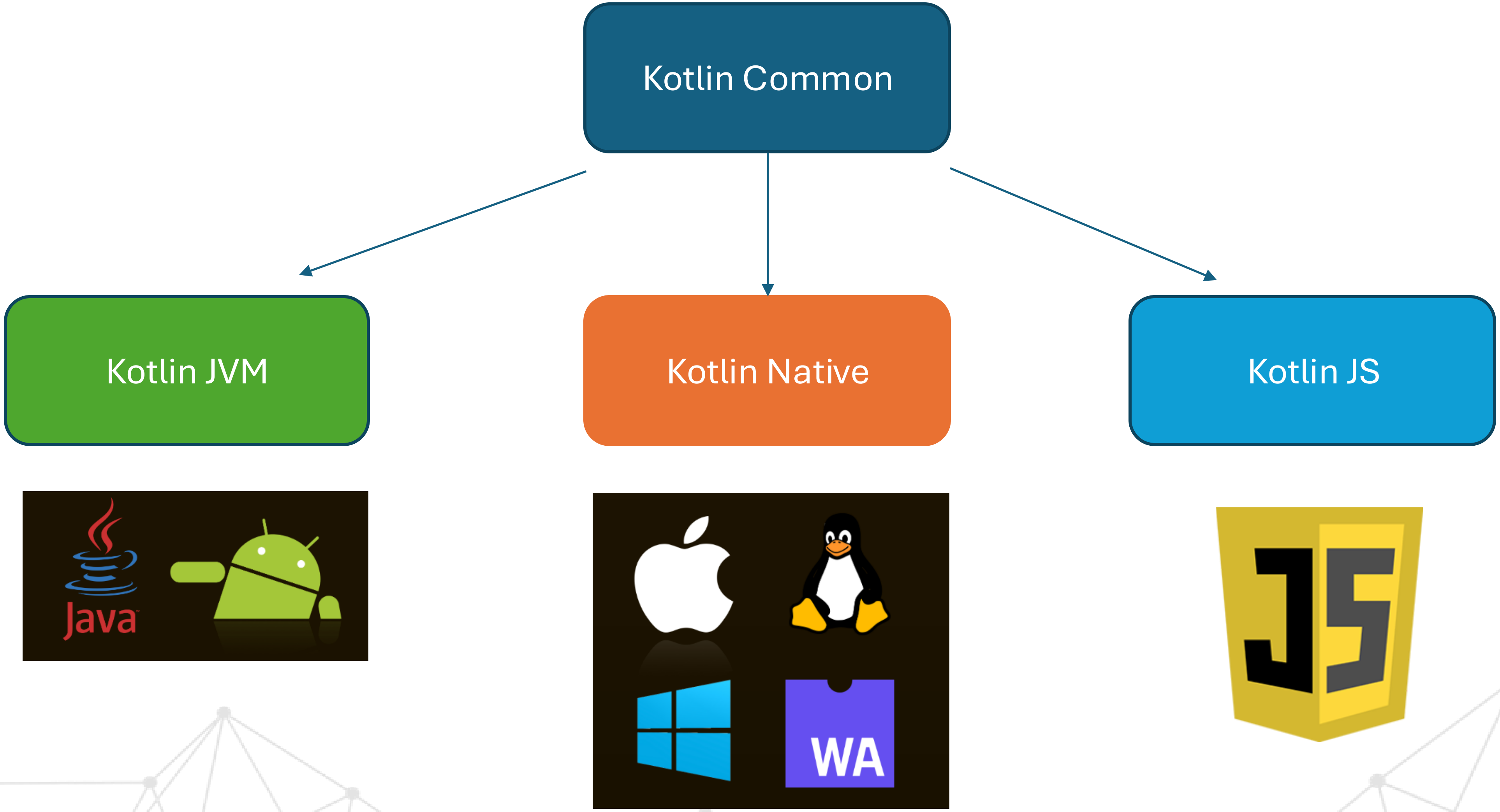
Kotlin

?

Как связаны Swift и Kotlin в KMP (и не только)



Kotlin для каждой платформы



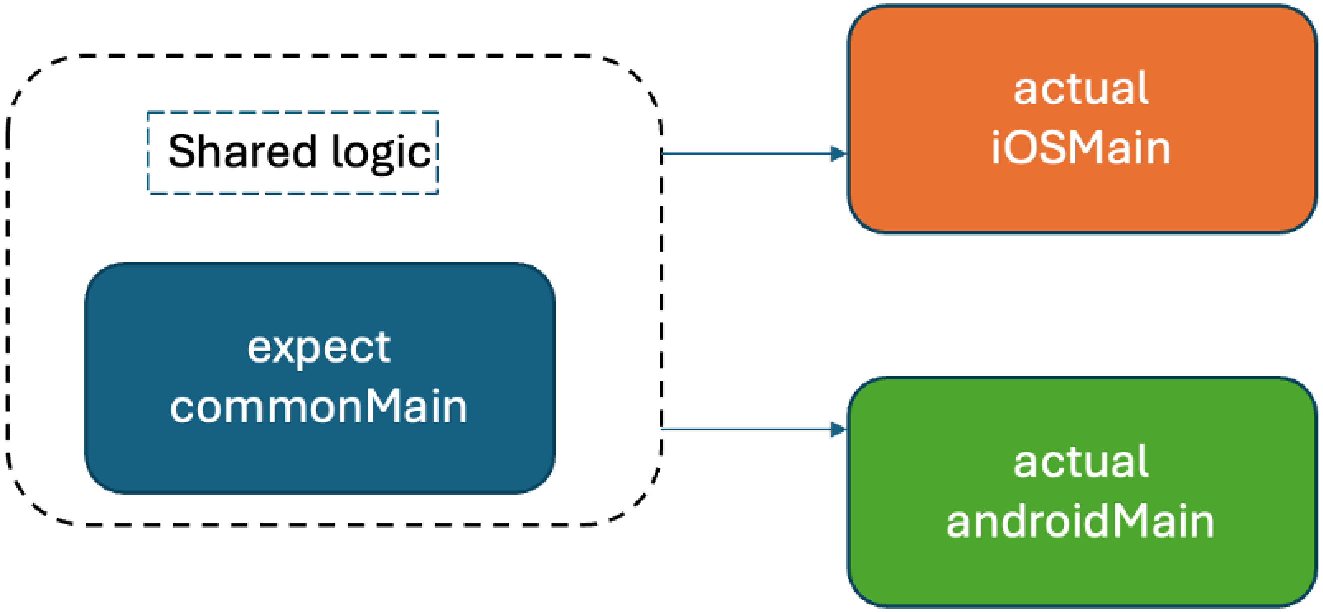
Kotlin Common

Kotlin JVM

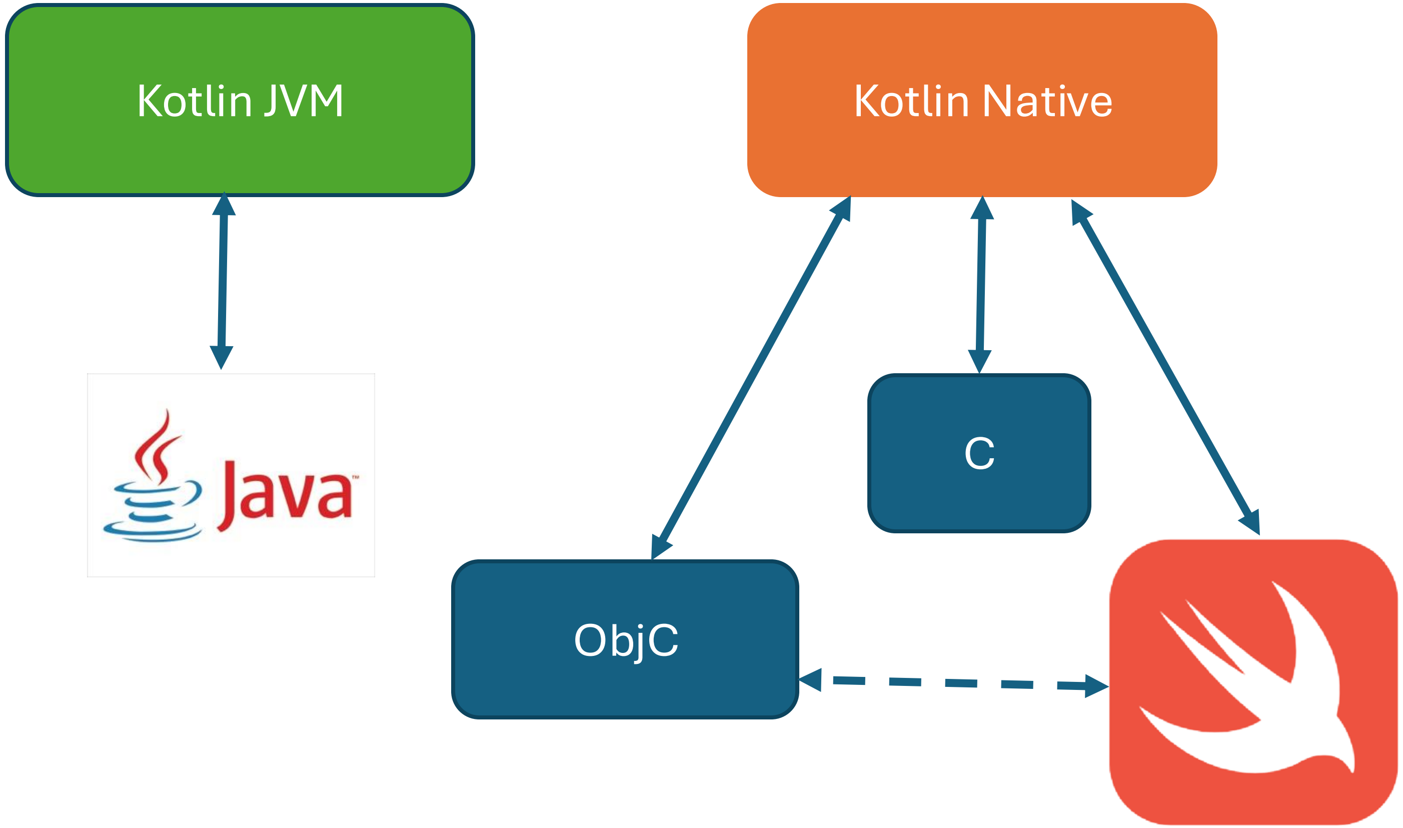
Kotlin Native

Kotlin JS

Kotlin для каждой платформы

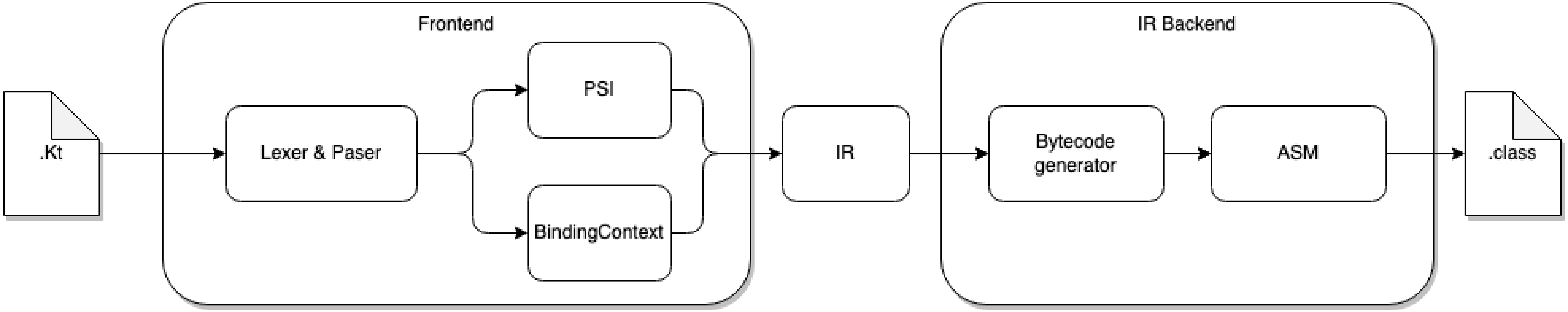


Interoperability



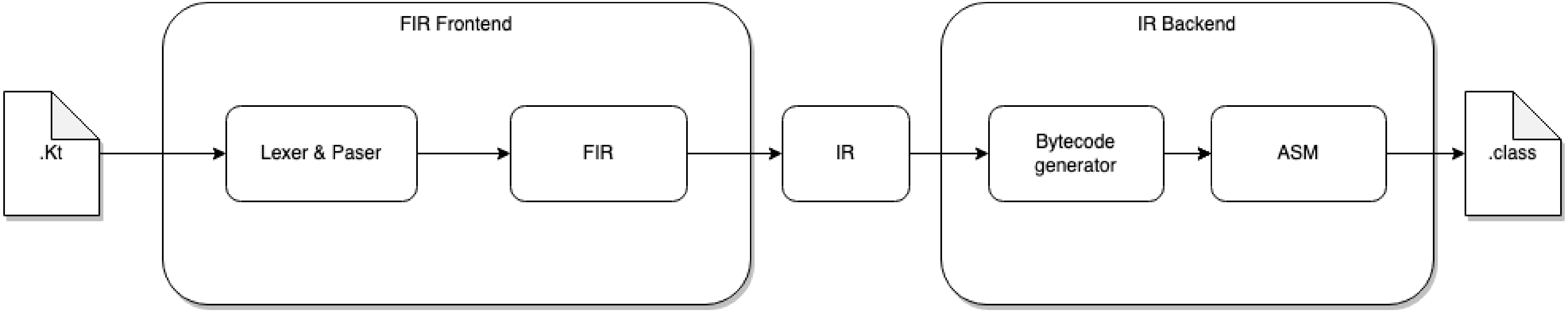
Компиляция.K1

IR backend по умолчанию с Kotlin 1.5.0



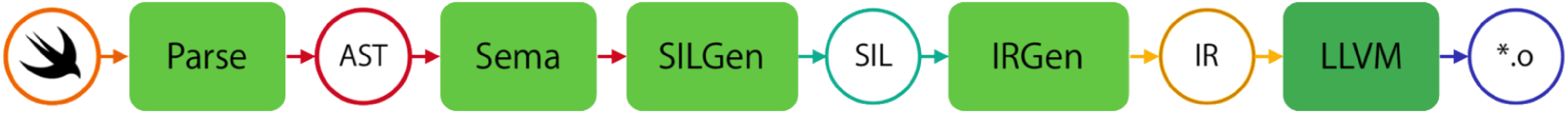
Компиляция FIR&IR.K2

FIR совмещает PSI и Binding Condext

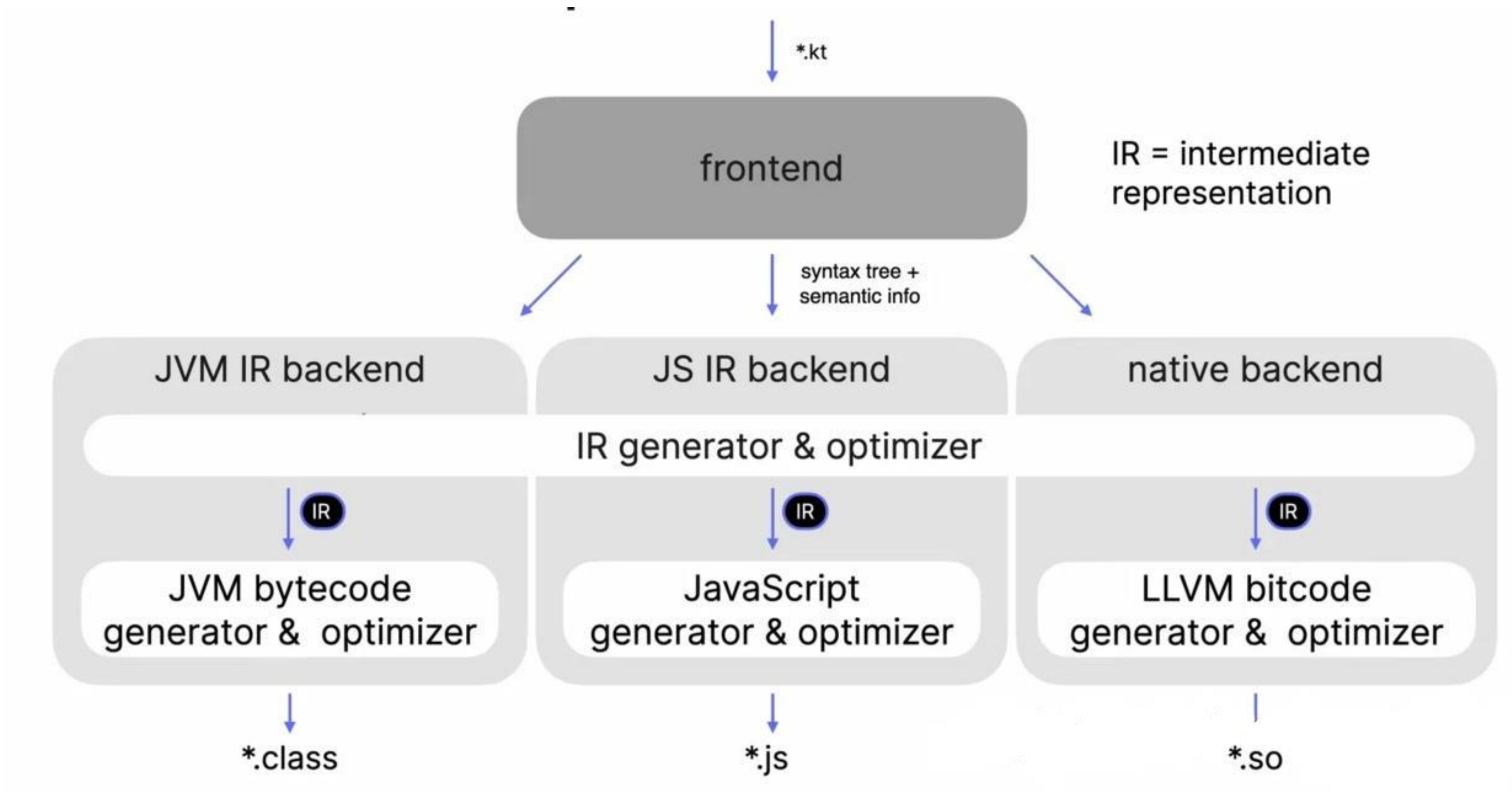


Swift. Компиляция

Swift Intermediate Language

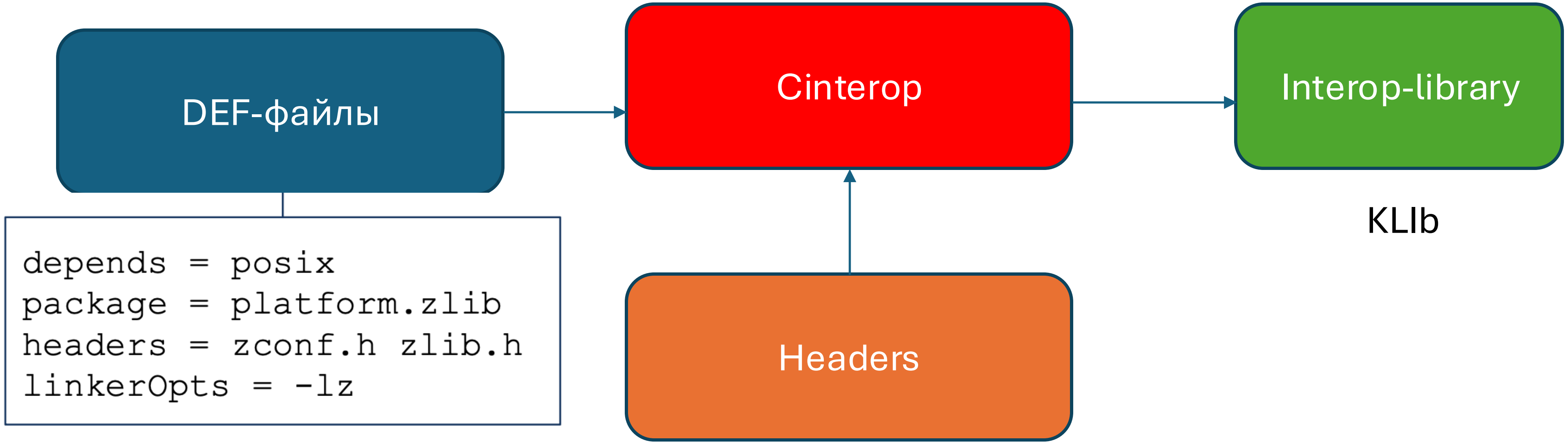


Компиляция IR



Kotlin/Native C-Interop

Параметры C-Interop в def-файле



DEF-файлы

Cinterop

Interop-library

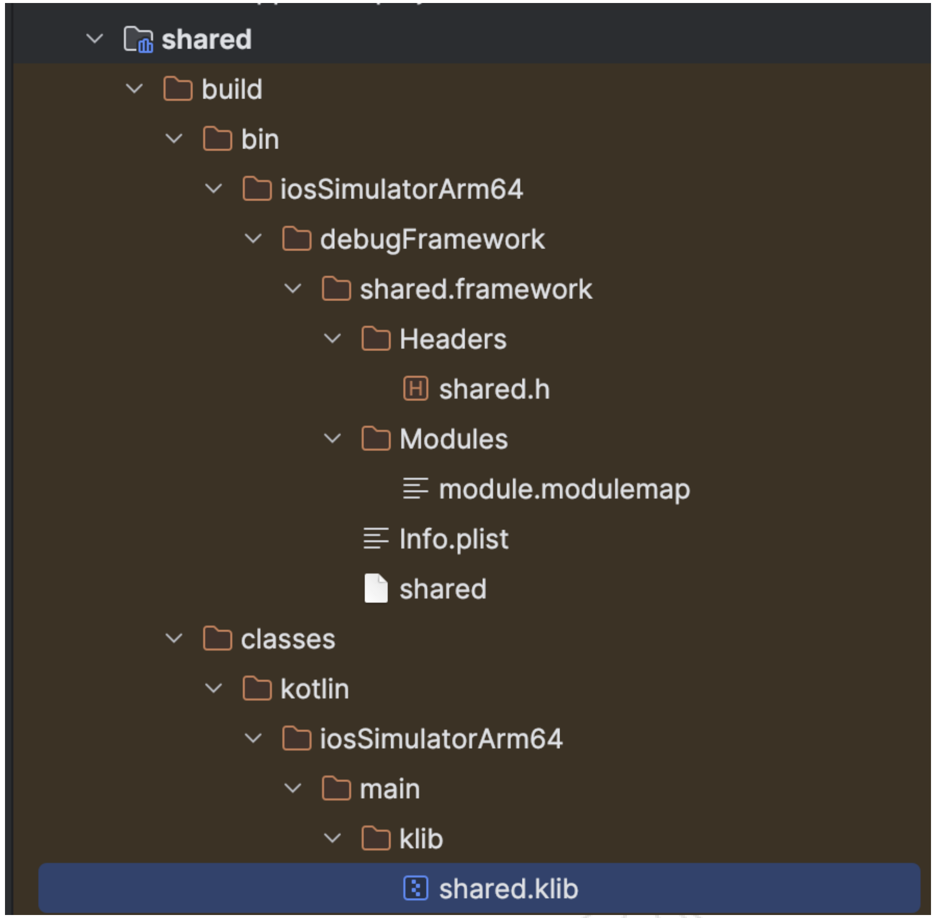
Headers

KLIb

```
depends = posix
package = platform.zlib
headers = zconf.h zlib.h
linkerOpts = -lz
```

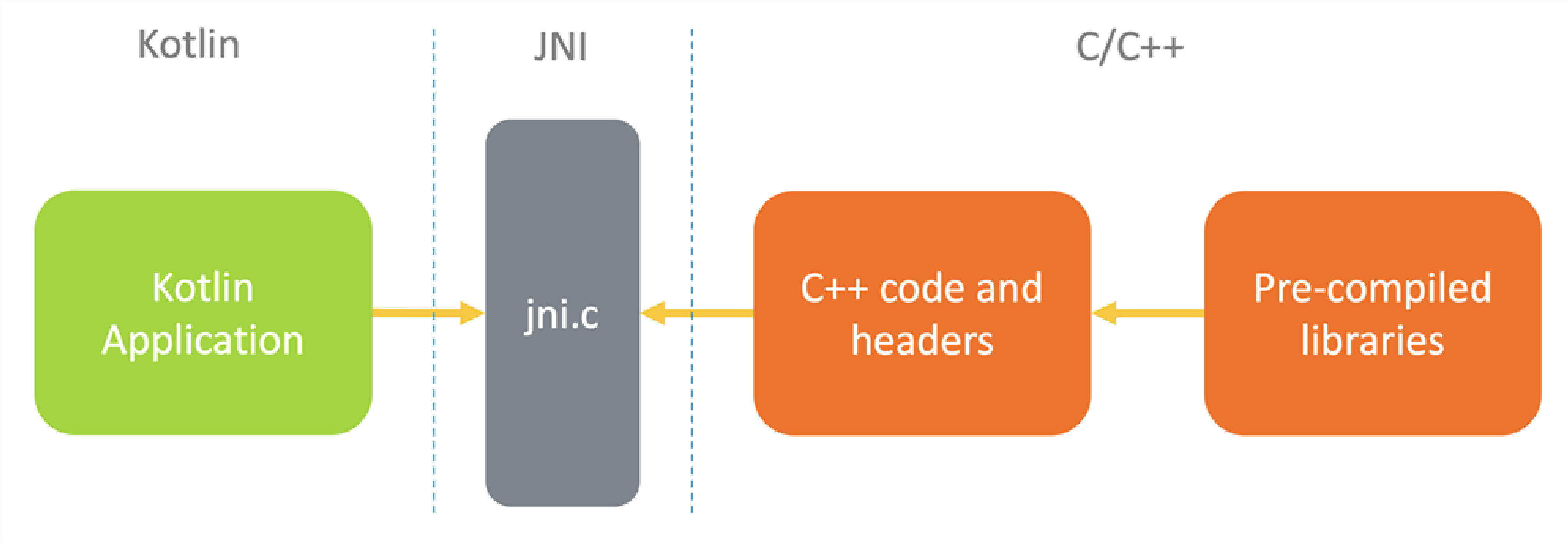
Kotlin/Native C-Interop

Objective-C + C



```
shared
  build
    bin
      iosSimulatorArm64
        debugFramework
          shared.framework
            Headers
              shared.h
            Modules
              module.modulemap
            Info.plist
            shared
    classes
      kotlin
        iosSimulatorArm64
          main
            klib
              shared.klib
```

Kotlin–C++. JNI


Kotlin
JNI
C/C++
Kotlin Application
jni.c
C++ code and headers
Pre-compiled libraries

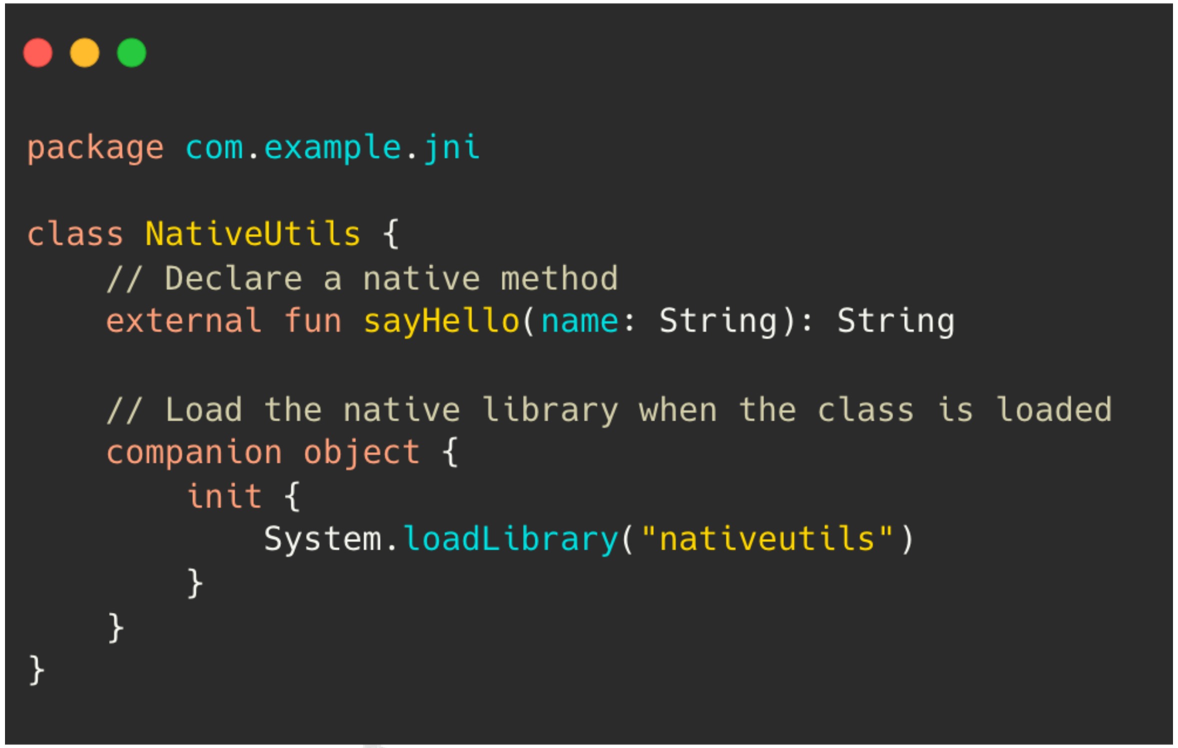
Kotlin–C++. JNI

Android NDK:

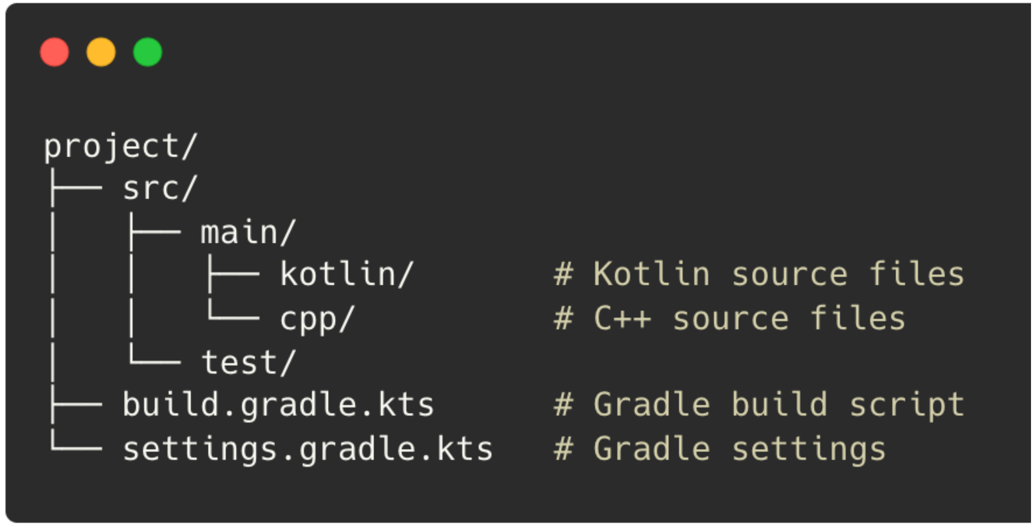
- OpenGL
- Готовые SDK на C++
- Библиотеки криптографические
- Защищенные SDK

Kotlin–C++. JNI

Пример подключения JNI



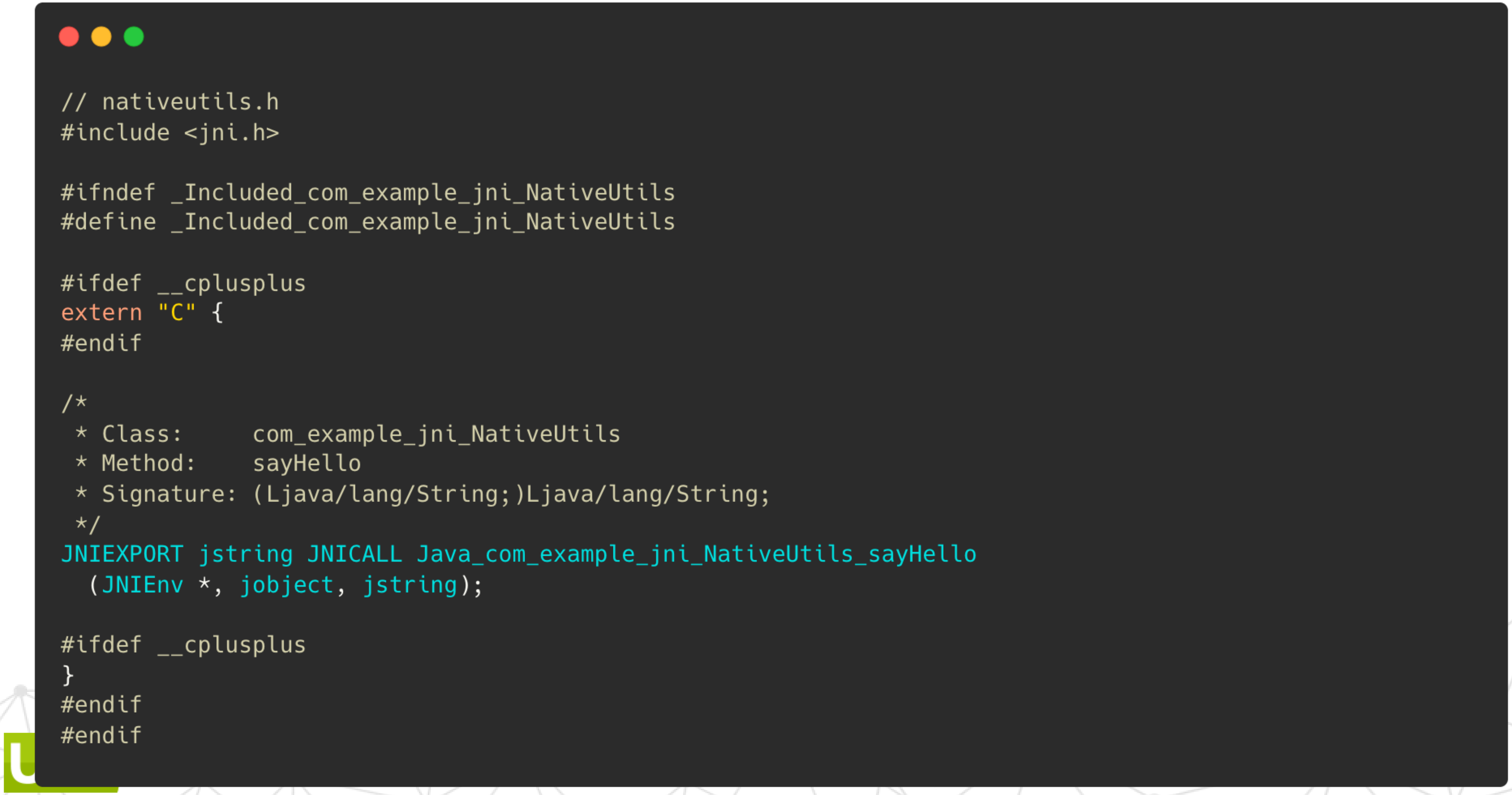
```kotlin
package com.example.jni

class NativeUtils {
    // Declare a native method
    external fun sayHello(name: String): String

    // Load the native library when the class is loaded
    companion object {
        init {
            System.loadLibrary("nativeutils")
        }
    }
}
```



```
project/
├── src/
│   ├── main/
│   │   ├── kotlin/       # Kotlin source files
│   │   └── cpp/          # C++ source files
│   └── test/
├── build.gradle.kts      # Gradle build script
└── settings.gradle.kts   # Gradle settings
```

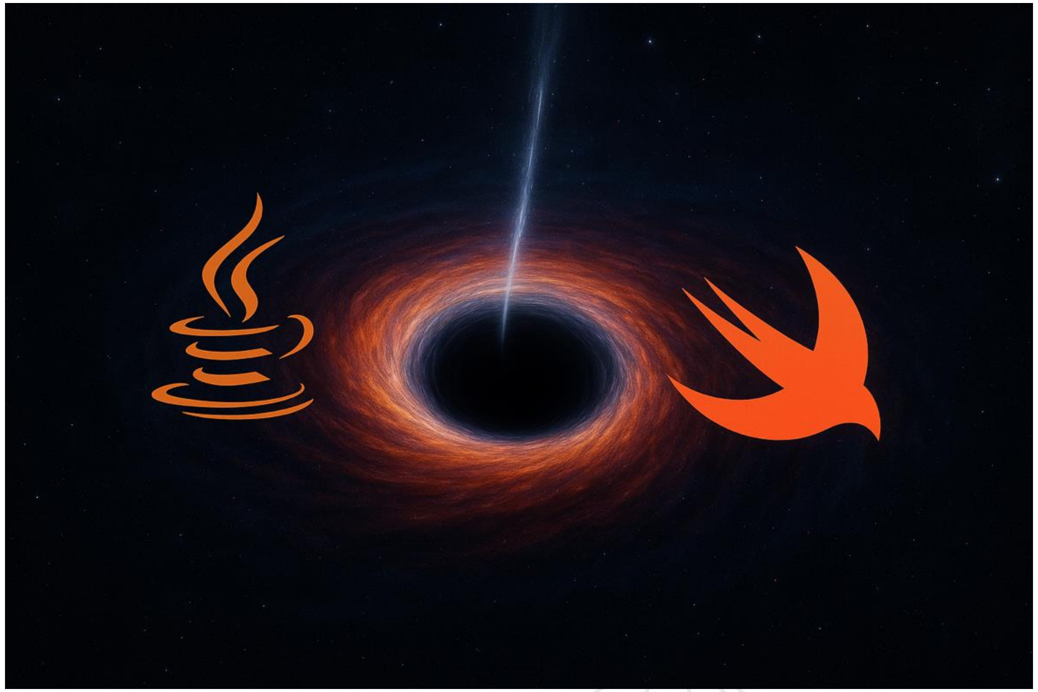
Kotlin–C++. JNI

Пример кода JNI



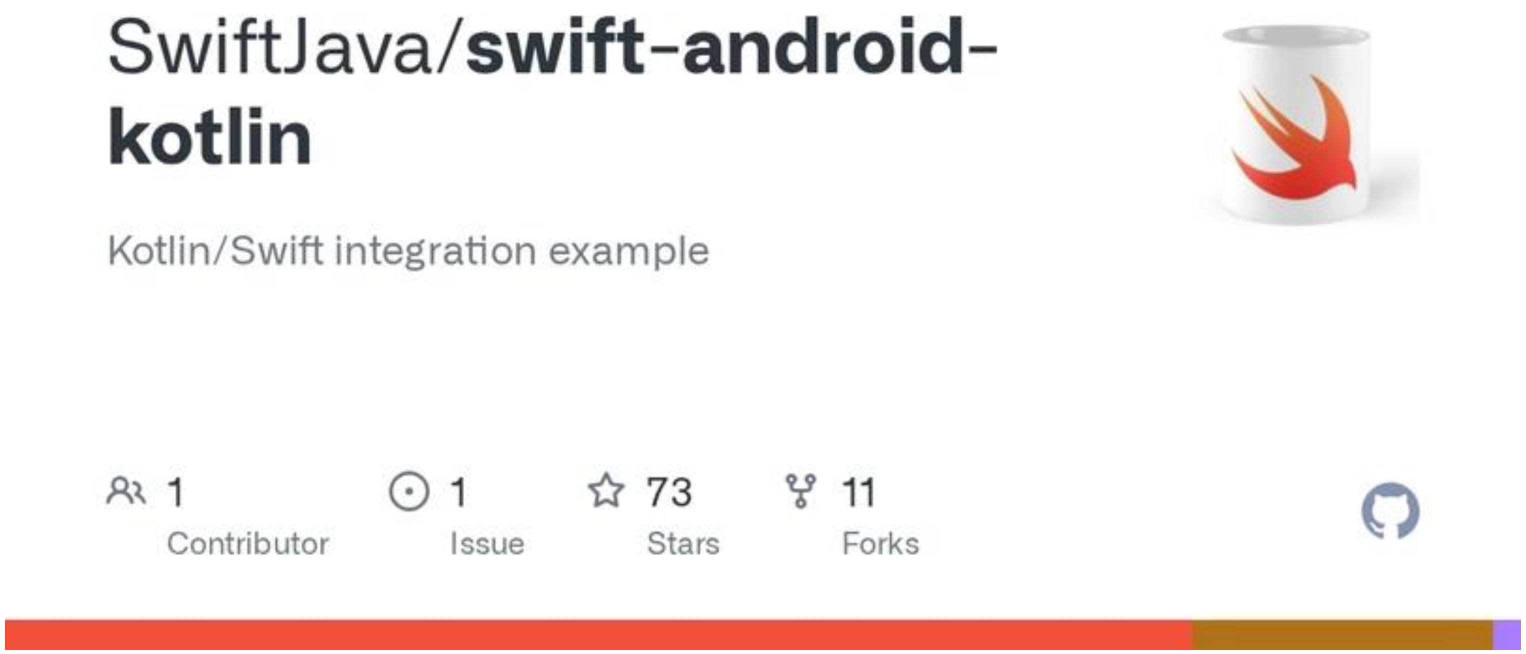
```
// nativeutils.h
#include <jni.h>

#ifndef _Included_com_example_jni_NativeUtils
#define _Included_com_example_jni_NativeUtils

#ifdef __cplusplus
extern "C" {
#endif

/*
 * Class:     com_example_jni_NativeUtils
 * Method:    sayHello
 * Signature: (Ljava/lang/String;)Ljava/lang/String;
 */
JNIEXPORT jstring JNICALL Java_com_example_jni_NativeUtils_sayHello
  (JNIEnv *, jobject, jstring);

#ifdef __cplusplus
}
#endif
#endif
```

А теперь посмотрим на Swift-Java



Взаимодействие Kotlin/Java - Swift

Идея не нова
(и даже не Apple)



SwiftJava/**swift-android-kotlin**

Kotlin/Swift integration example

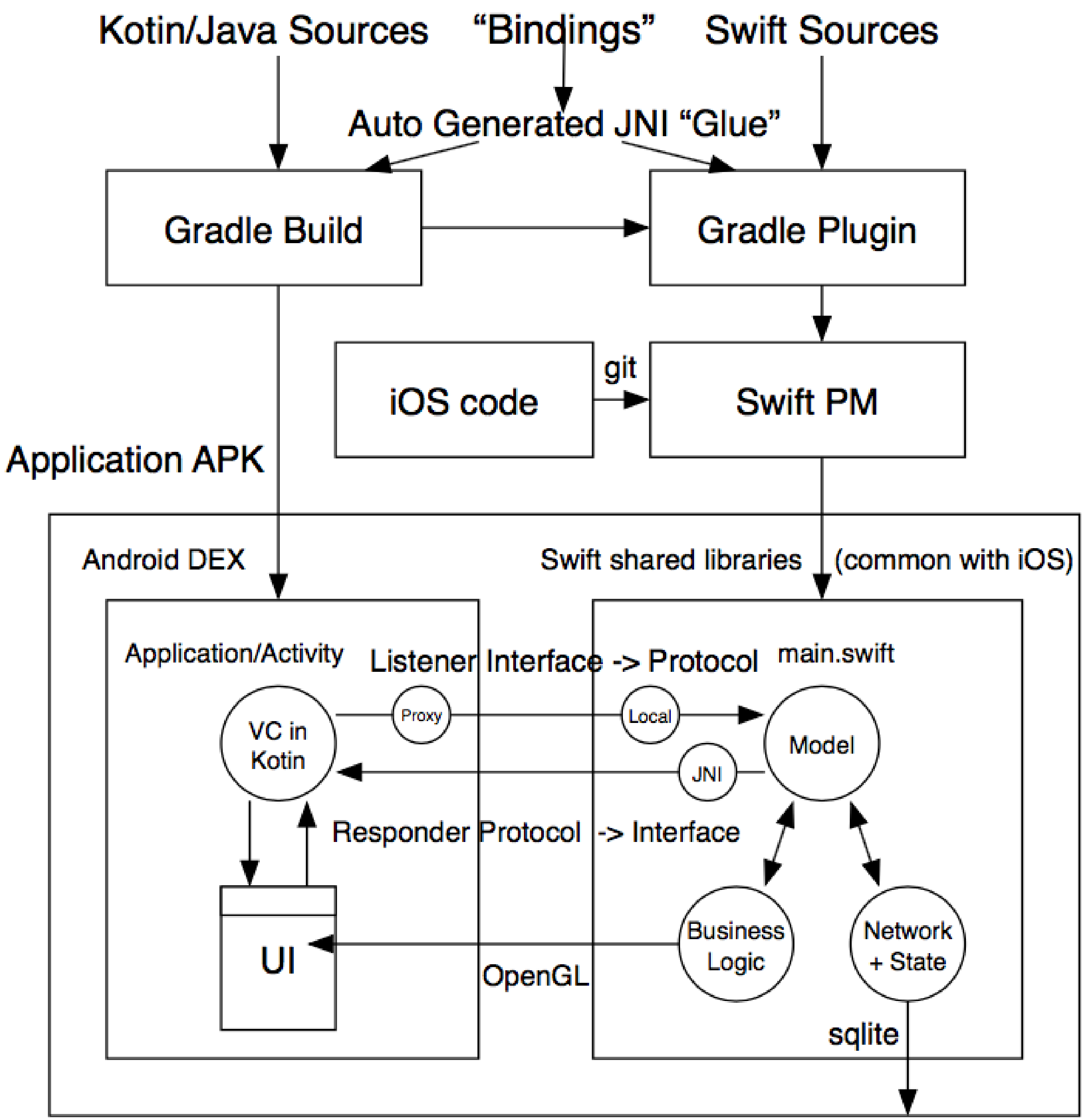
1 Contributor

1 Issue

73 Stars

11 Forks

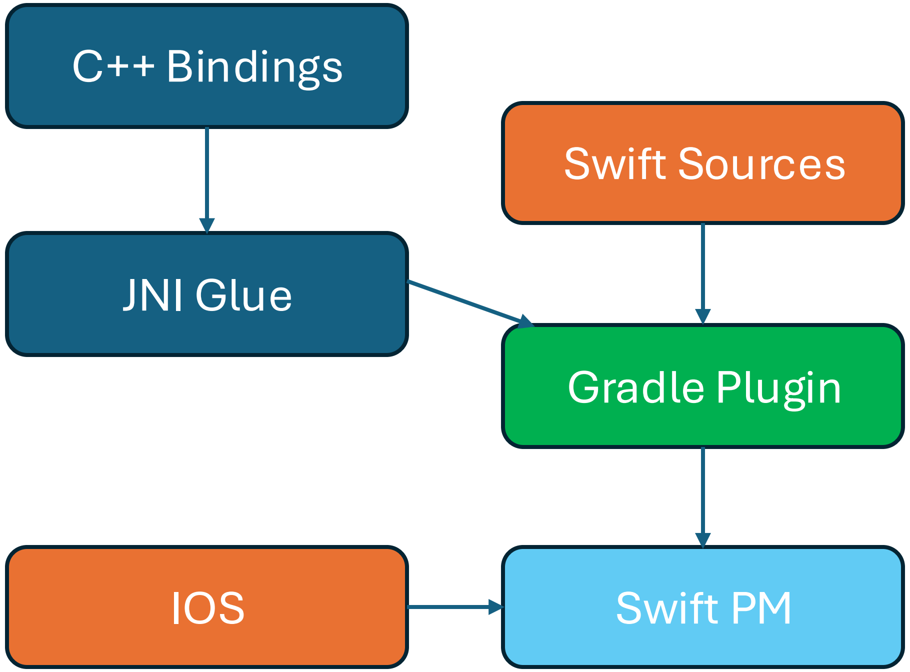
Взаимодействие Kotlin/Java - Swift



Взаимодействие Kotlin/Java - Swift

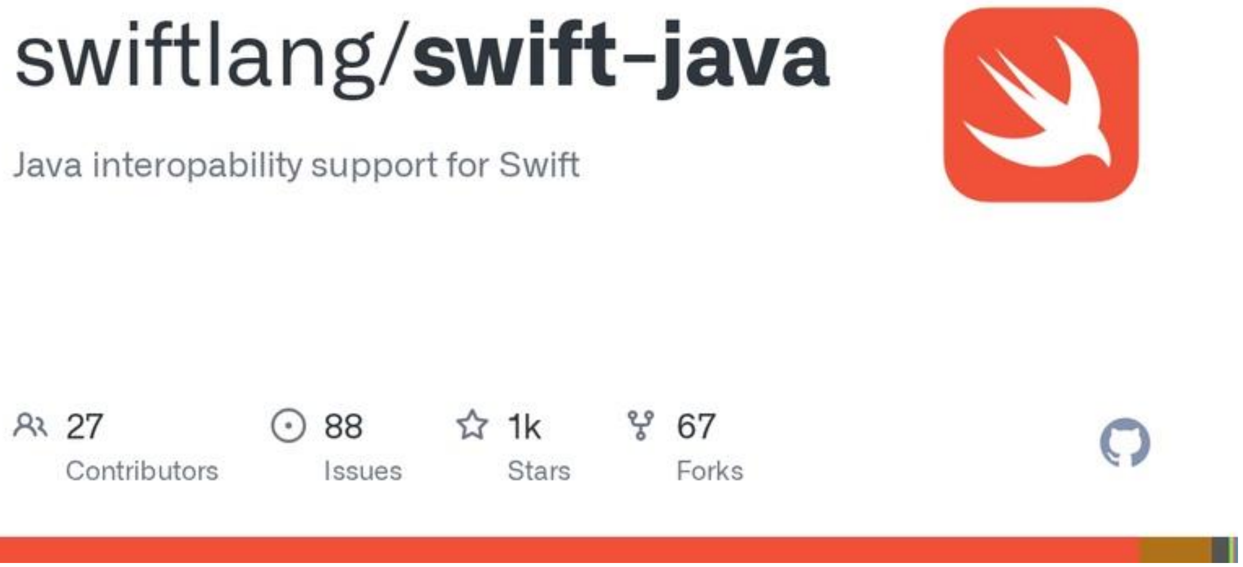
C++ -> JNI -> Swift

Java -> JNI -> Swift



Swift-Java

- https://github.com/swiftlang/swift-java/



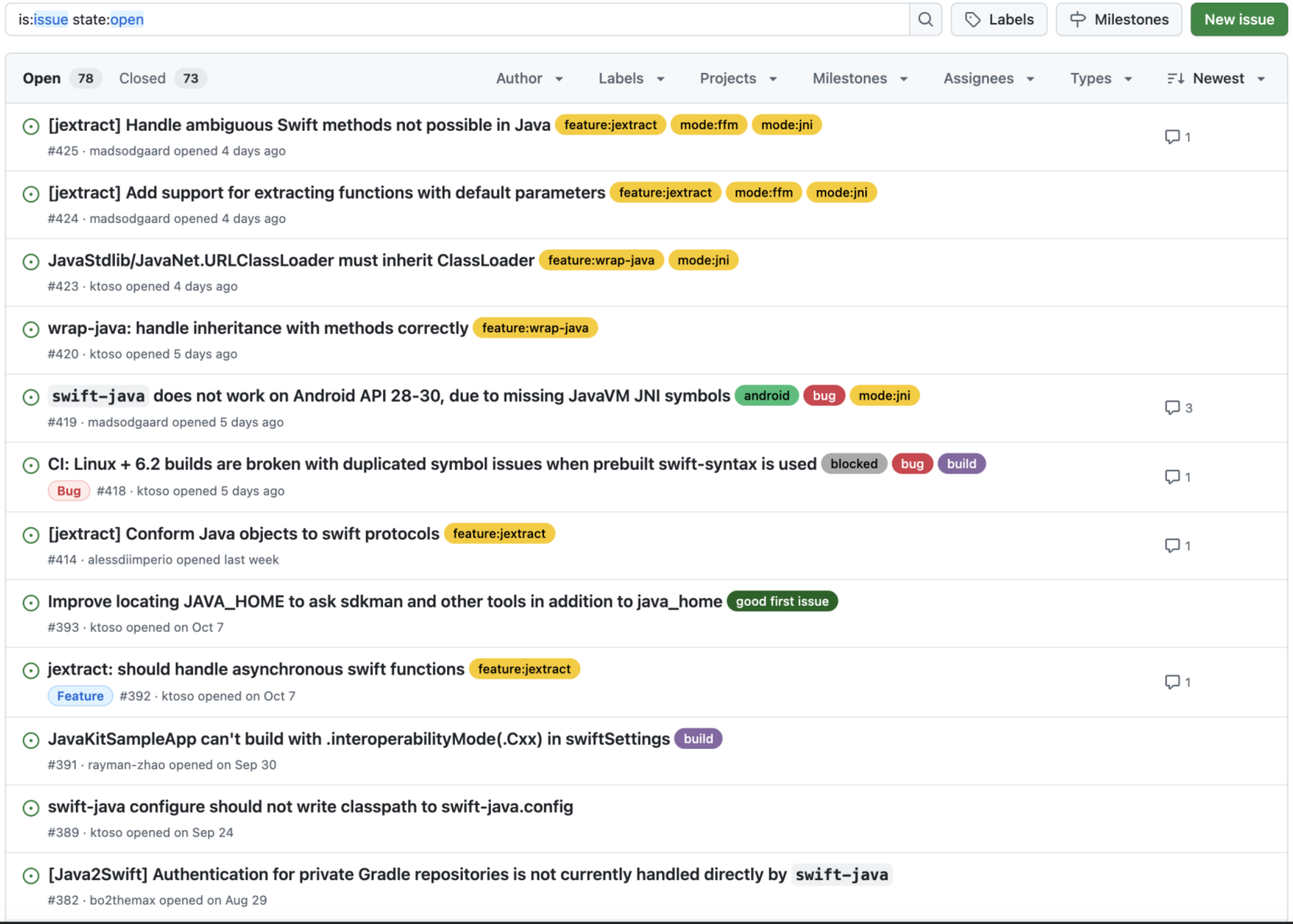
swiftlang/**swift-java**

Java interopability support for Swift

Generates hard-to-get-right code.

Генерирует код, который сложно написать правильно.

Проблемы Swift-Java



[jextract] Handle ambiguous Swift methods not possible in Java feature:jextract mode:ffm mode:jni
#425 · madsodgaard opened 4 days ago · 1

[jextract] Add support for extracting functions with default parameters feature:jextract mode:ffm mode:jni
#424 · madsodgaard opened 4 days ago

JavaStdlib/JavaNet.URLClassLoader must inherit ClassLoader feature:wrap-java mode:jni
#423 · ktoso opened 4 days ago

wrap-java: handle inheritance with methods correctly feature:wrap-java
#420 · ktoso opened 5 days ago

`swift-java` does not work on Android API 28-30, due to missing JavaVM JNI symbols android bug mode:jni
#419 · madsodgaard opened 5 days ago · 3

CI: Linux + 6.2 builds are broken with duplicated symbol issues when prebuilt swift-syntax is used blocked bug build
Bug #418 · ktoso opened 5 days ago · 1

[jextract] Conform Java objects to swift protocols feature:jextract
#414 · alessdiimperio opened last week · 1

Improve locating JAVA_HOME to ask sdkman and other tools in addition to java_home good first issue
#393 · ktoso opened on Oct 7

jextract: should handle asynchronous swift functions feature:jextract
Feature #392 · ktoso opened on Oct 7 · 1

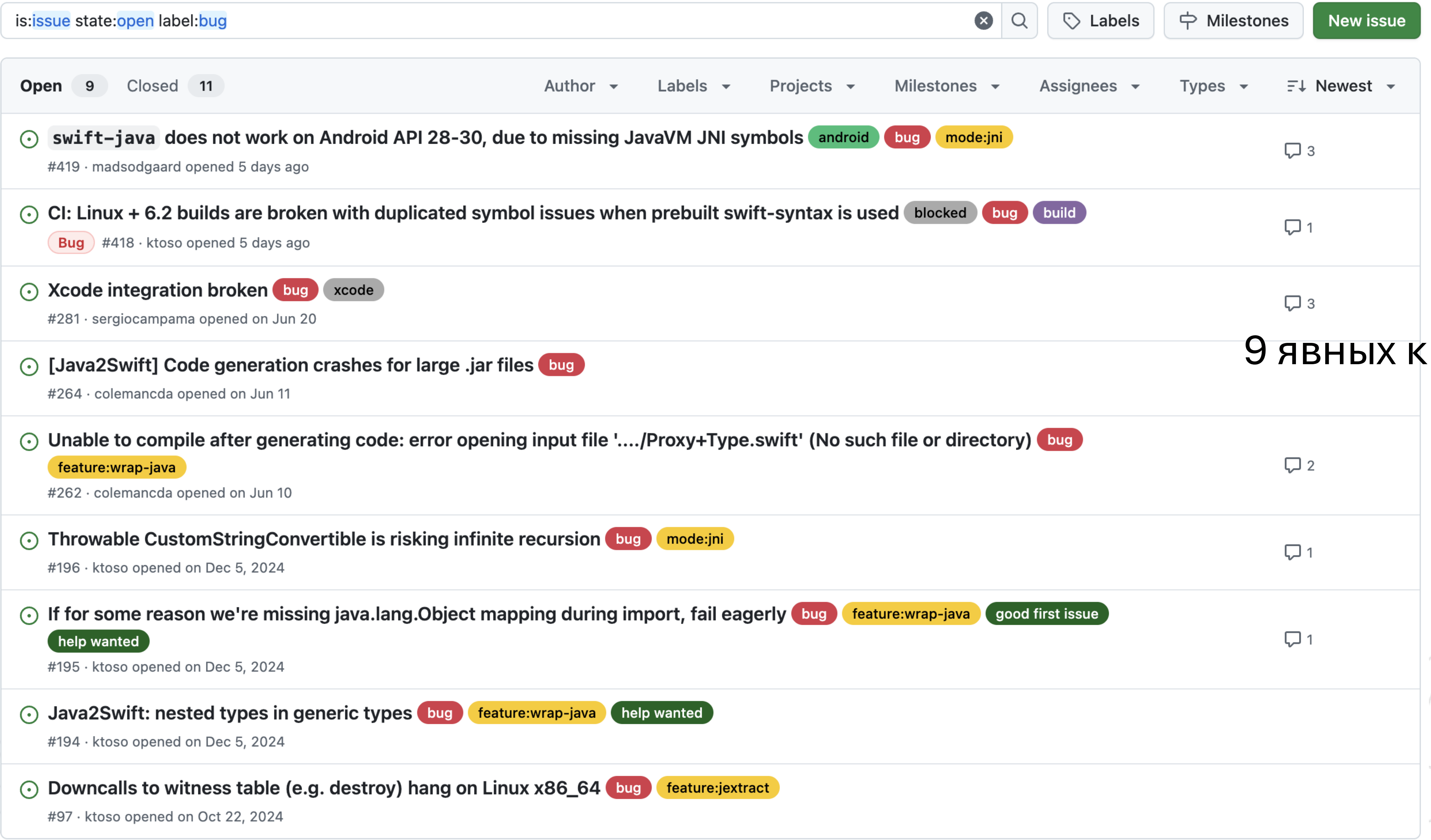
JavaKitSampleApp can't build with .interoperabilityMode(.Cxx) in swiftSettings build
#391 · rayman-zhao opened on Sep 30

swift-java configure should not write classpath to swift-java.config
#389 · ktoso opened on Sep 24

[Java2Swift] Authentication for private Gradle repositories is not currently handled directly by `swift-java`
#382 · bo2themax opened on Aug 29

78 открытых тикетов

Проблемы Swift-Java



is:issue state:open label:bug

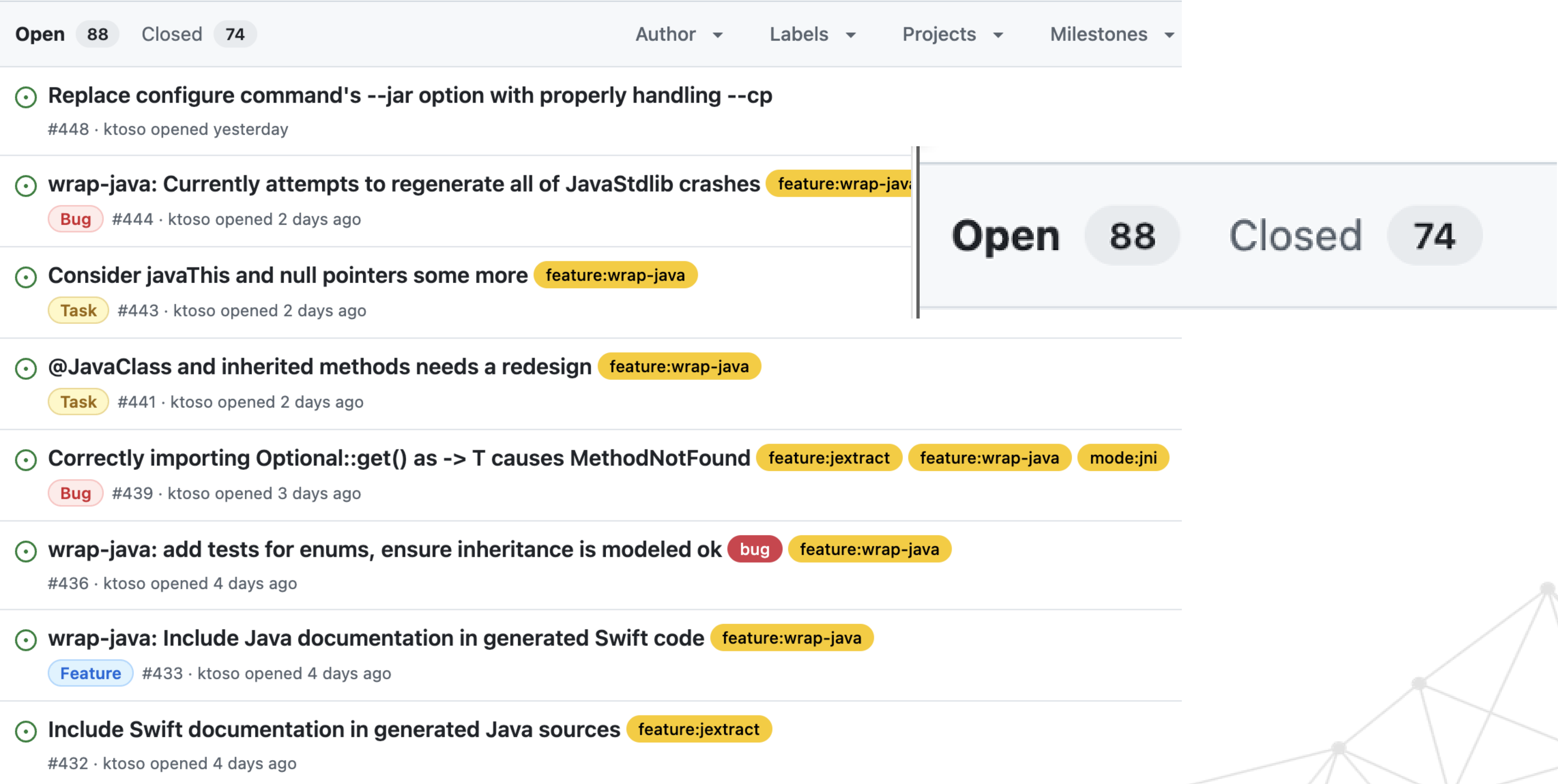
Open 9 Closed 11

- `swift-java` does not work on Android API 28-30, due to missing JavaVM JNI symbols — android · bug · mode:jni — #419 · madsodgaard opened 5 days ago — 3
- CI: Linux + 6.2 builds are broken with duplicated symbol issues when prebuilt swift-syntax is used — blocked · bug · build — Bug #418 · ktoso opened 5 days ago — 1
- Xcode integration broken — bug · xcode — #281 · sergiocampama opened on Jun 20 — 3
- [Java2Swift] Code generation crashes for large .jar files — bug — #264 · colemancda opened on Jun 11
- Unable to compile after generating code: error opening input file '..../Proxy+Type.swift' (No such file or directory) — bug · feature:wrap-java — #262 · colemancda opened on Jun 10 — 2
- Throwable CustomStringConvertible is risking infinite recursion — bug · mode:jni — #196 · ktoso opened on Dec 5, 2024 — 1
- If for some reason we're missing java.lang.Object mapping during import, fail eagerly — bug · feature:wrap-java · good first issue · help wanted — #195 · ktoso opened on Dec 5, 2024 — 1
- Java2Swift: nested types in generic types — bug · feature:wrap-java · help wanted — #194 · ktoso opened on Dec 5, 2024
- Downcalls to witness table (e.g. destroy) hang on Linux x86_64 — bug · feature:jextract — #97 · ktoso opened on Oct 22, 2024

9 явных критикалов

Проблемы Swift-Java

Пока писался доклад, количество открытых багов выросло 78 –>88



Open 88 Closed 74 Author ▾ Labels ▾ Projects ▾ Milestones ▾

○ **Replace configure command's --jar option with properly handling --cp**
#448 · ktoso opened yesterday

○ **wrap-java: Currently attempts to regenerate all of JavaStdlib crashes** feature:wrap-java
Bug #444 · ktoso opened 2 days ago

○ **Consider javaThis and null pointers some more** feature:wrap-java
Task #443 · ktoso opened 2 days ago

○ **@JavaClass and inherited methods needs a redesign** feature:wrap-java
Task #441 · ktoso opened 2 days ago

○ **Correctly importing Optional::get() as -> T causes MethodNotFound** feature:jextract feature:wrap-java mode:jni
Bug #439 · ktoso opened 3 days ago

○ **wrap-java: add tests for enums, ensure inheritance is modeled ok** bug feature:wrap-java
#436 · ktoso opened 4 days ago

○ **wrap-java: Include Java documentation in generated Swift code** feature:wrap-java
Feature #433 · ktoso opened 4 days ago

○ **Include Swift documentation in generated Java sources** feature:jextract
#432 · ktoso opened 4 days ago

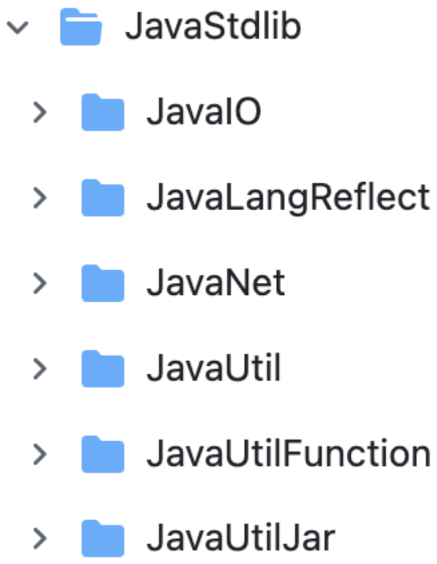
Open 88 Closed 74

Swift-Java

- Экспериментальный статус – не готово для полноценного использования
- C Java лучше, чем с Kotlin
- Работает только с переведенными решениями
- Зависимости только те, которые содержать поддерживаемые библиотеки

Swift-Java.Решения (пример)

- Java.Net – сеть
- Java.Concurrency – многопоточность
- JavaLangReflect - десериализация



JavaStdlib

JavaIO

JavaLangReflect

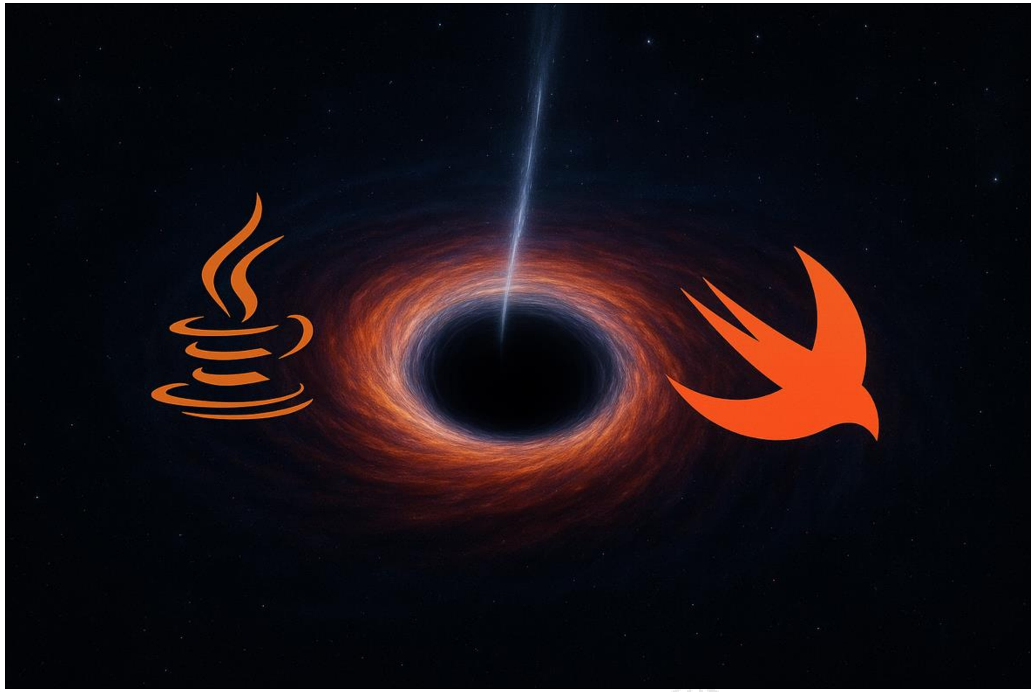
JavaNet

JavaUtil

JavaUtilFunction

JavaUtilJar

Java -> Swift попробуем в деле



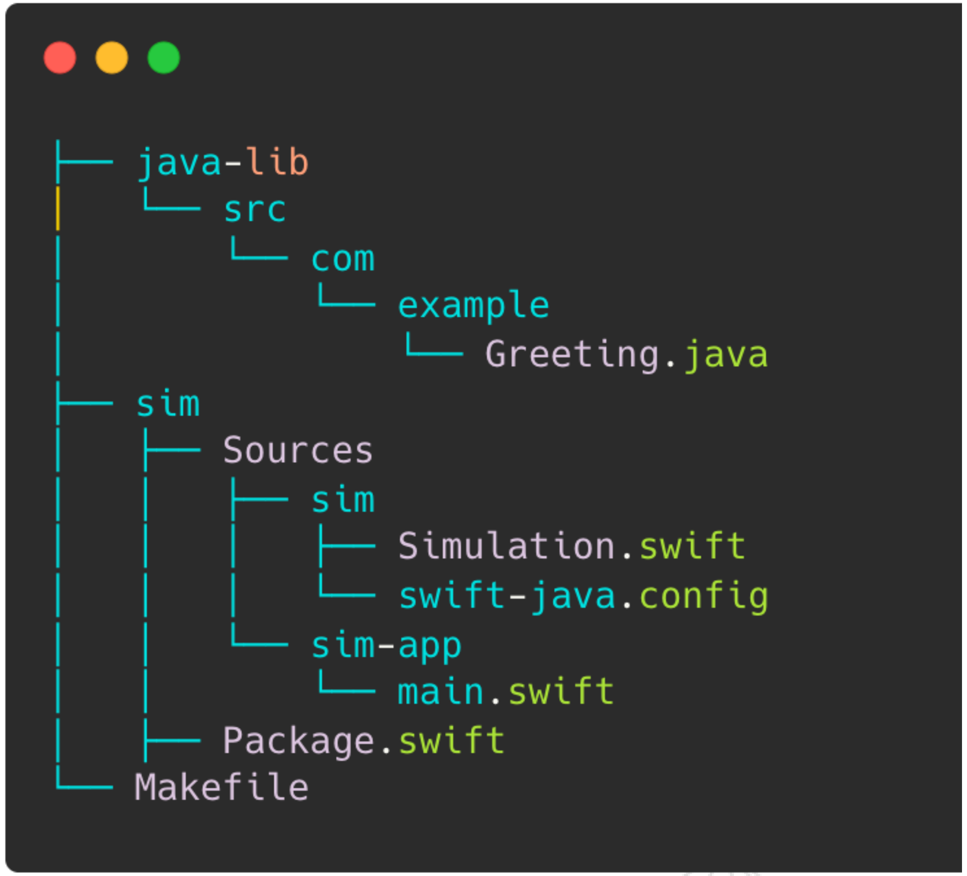
Настроим проект

Иерархия базового решения

Здесь java

Здесь конфигуратор и вызов

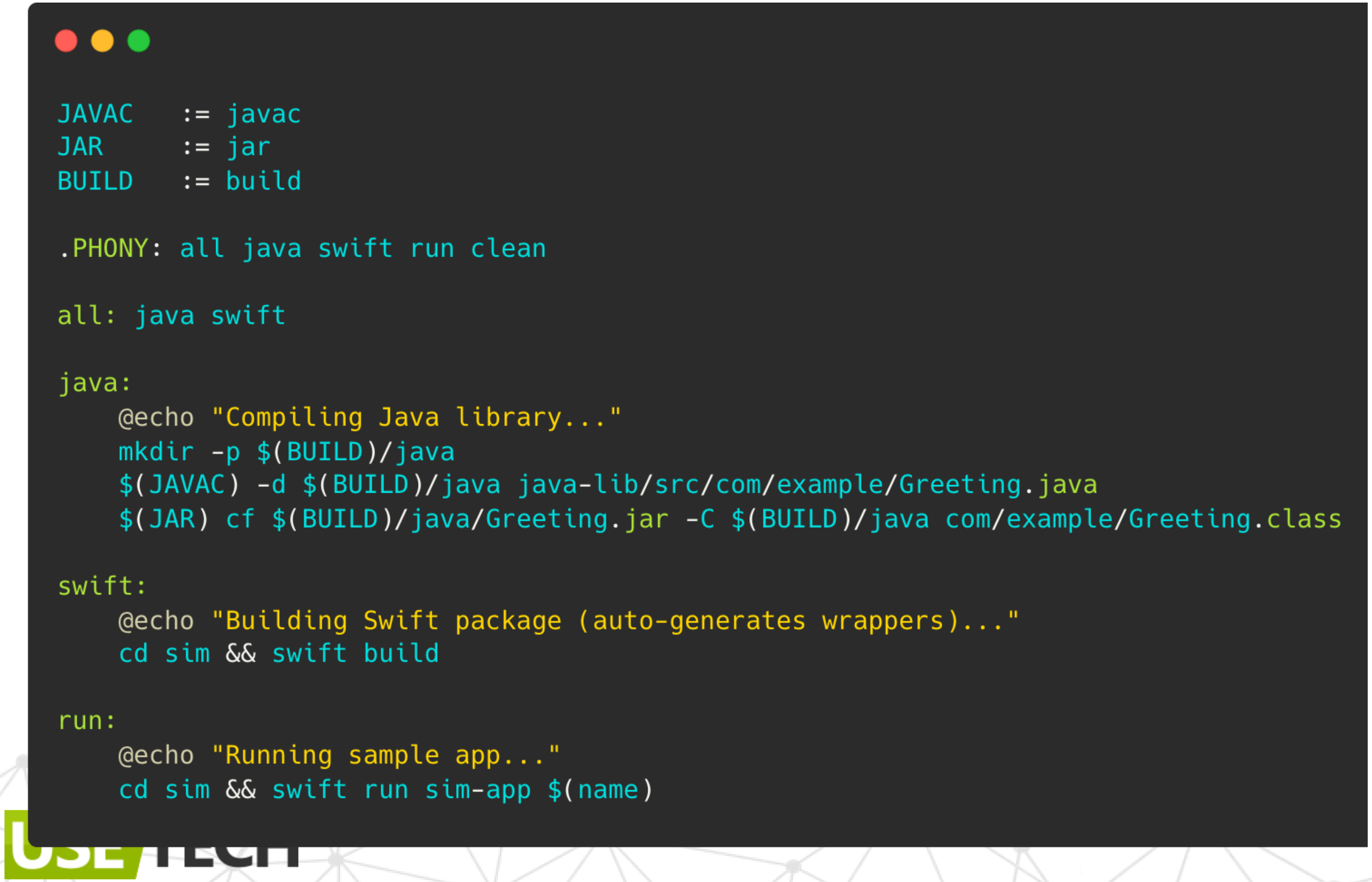
Здесь executable target



```
├── java-lib
│   └── src
│       └── com
│           └── example
│               └── Greeting.java
├── sim
│   ├── Sources
│   │   ├── sim
│   │   │   ├── Simulation.swift
│   │   │   └── swift-java.config
│   │   └── sim-app
│   │       └── main.swift
│   ├── Package.swift
└── Makefile
```

Настроим проект. Скрипт запуска

Скрипт для запуска – необязательно

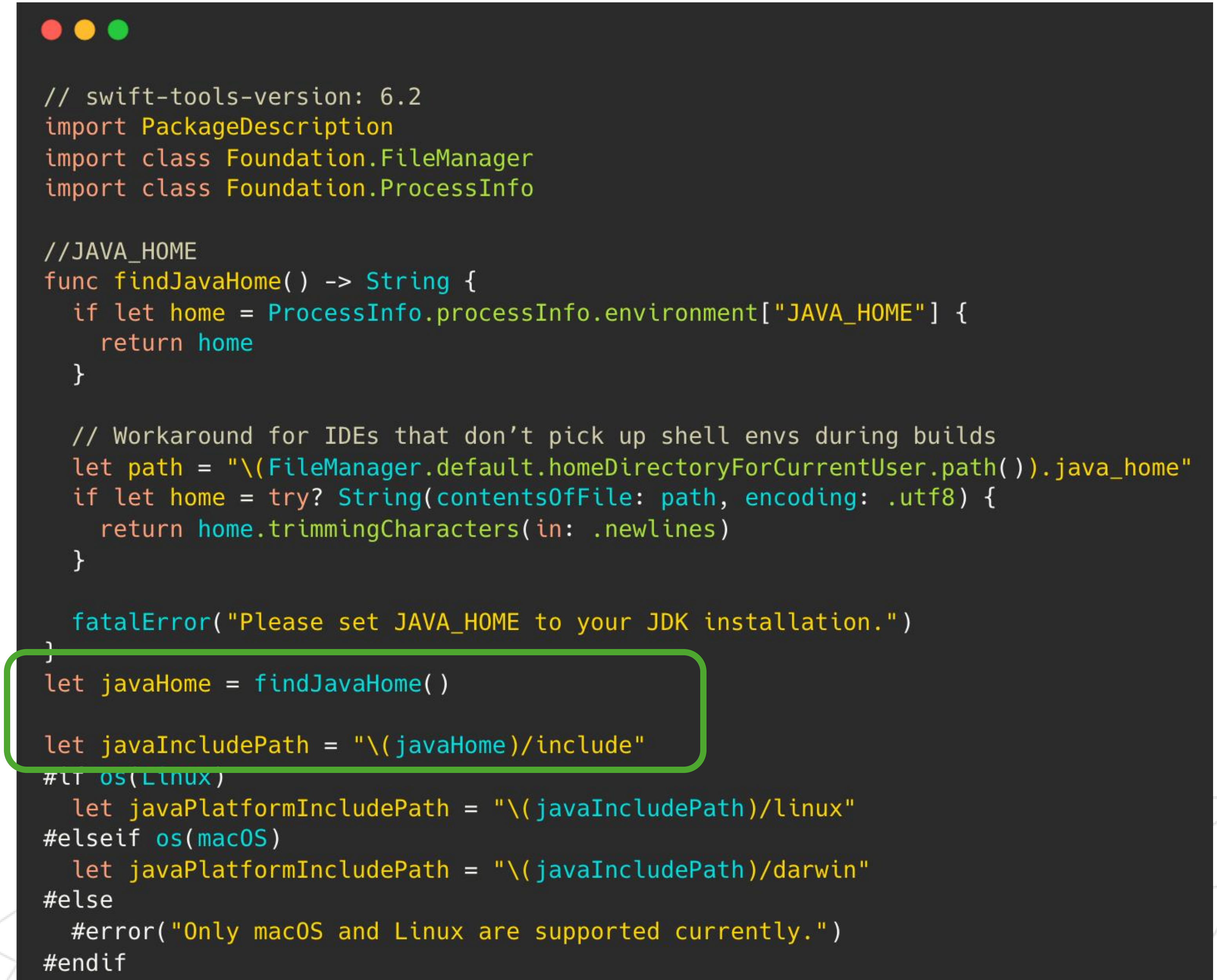


```
JAVAC   := javac
JAR     := jar
BUILD   := build

.PHONY: all java swift run clean

all: java swift

java:
    @echo "Compiling Java library..."
    mkdir -p $(BUILD)/java
    $(JAVAC) -d $(BUILD)/java java-lib/src/com/example/Greeting.java
    $(JAR) cf $(BUILD)/java/Greeting.jar -C $(BUILD)/java com/example/Greeting.class

swift:
    @echo "Building Swift package (auto-generates wrappers)..."
    cd sim && swift build

run:
    @echo "Running sample app..."
    cd sim && swift run sim-app $(name)
```

Можно сразу jar

Настроим проект. Package

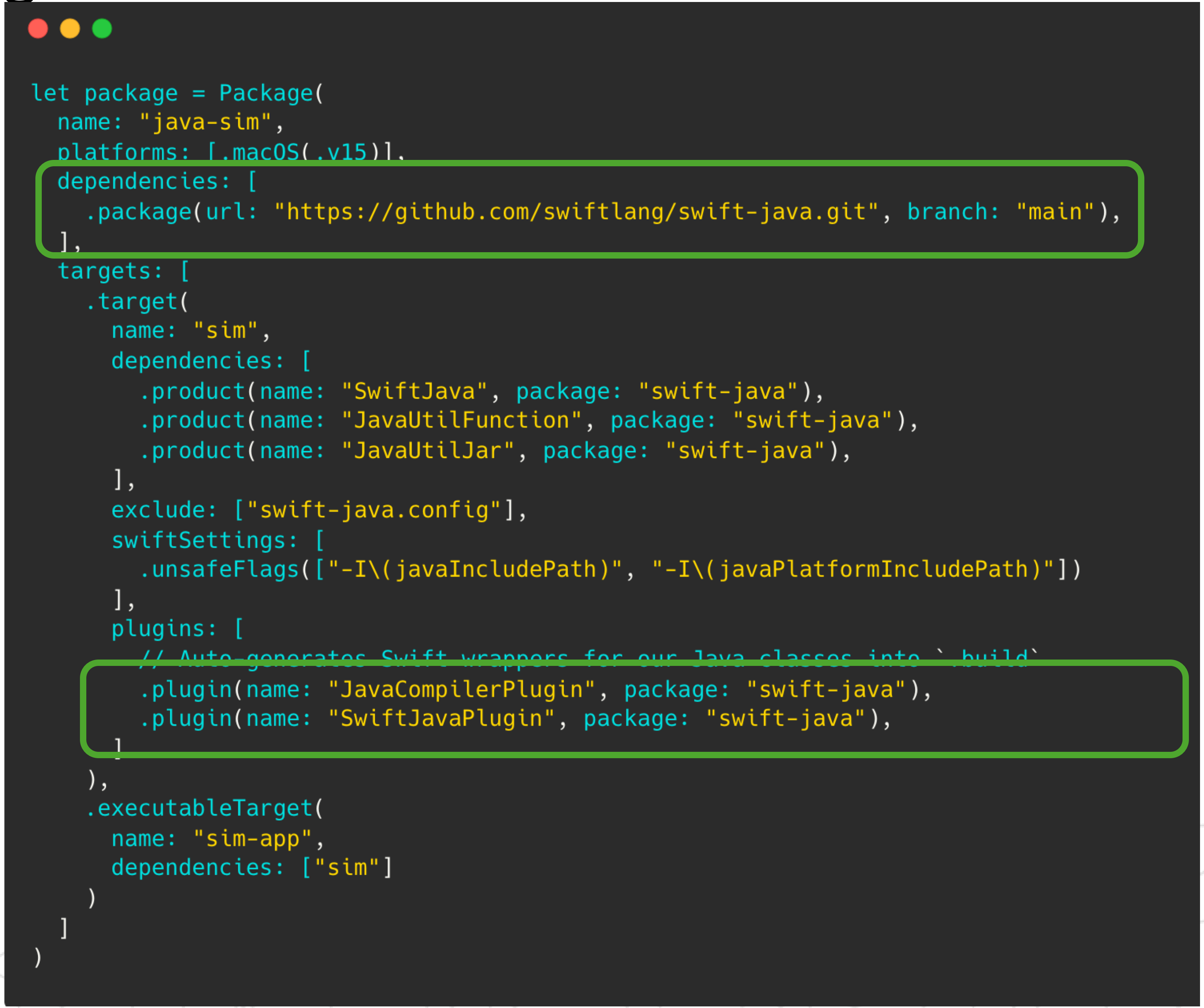
Получение пути к Java SDK



```swift
// swift-tools-version: 6.2
import PackageDescription
import class Foundation.FileManager
import class Foundation.ProcessInfo

//JAVA_HOME
func findJavaHome() -> String {
  if let home = ProcessInfo.processInfo.environment["JAVA_HOME"] {
    return home
  }

  // Workaround for IDEs that don’t pick up shell envs during builds
  let path = "\(FileManager.default.homeDirectoryForCurrentUser.path()).java_home"
  if let home = try? String(contentsOfFile: path, encoding: .utf8) {
    return home.trimmingCharacters(in: .newlines)
  }

  fatalError("Please set JAVA_HOME to your JDK installation.")
}
let javaHome = findJavaHome()

let javaIncludePath = "\(javaHome)/include"
#if os(Linux)
  let javaPlatformIncludePath = "\(javaIncludePath)/linux"
#elseif os(macOS)
  let javaPlatformIncludePath = "\(javaIncludePath)/darwin"
#else
  #error("Only macOS and Linux are supported currently.")
#endif
```

Настроим проект. Package

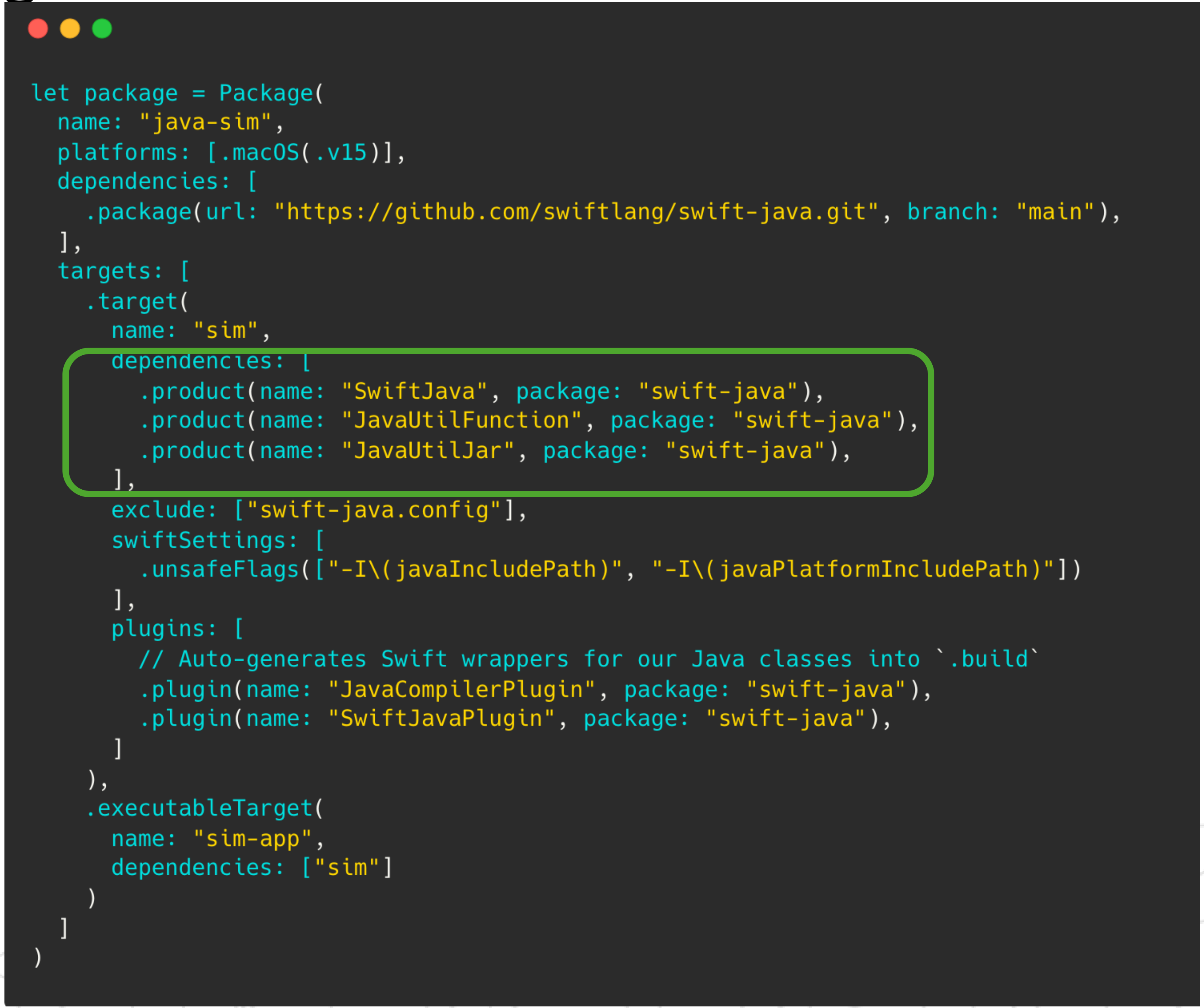
Подключаем плагин SwiftJava



```swift
let package = Package(
  name: "java-sim",
  platforms: [.macOS(.v15)],
  dependencies: [
    .package(url: "https://github.com/swiftlang/swift-java.git", branch: "main"),
  ],
  targets: [
    .target(
      name: "sim",
      dependencies: [
        .product(name: "SwiftJava", package: "swift-java"),
        .product(name: "JavaUtilFunction", package: "swift-java"),
        .product(name: "JavaUtilJar", package: "swift-java"),
      ],
      exclude: ["swift-java.config"],
      swiftSettings: [
        .unsafeFlags(["-I\(javaIncludePath)", "-I\(javaPlatformIncludePath)"])
      ],
      plugins: [
        // Auto-generates Swift wrappers for our Java classes into `.build`
        .plugin(name: "JavaCompilerPlugin", package: "swift-java"),
        .plugin(name: "SwiftJavaPlugin", package: "swift-java"),
      ]
    ),
    .executableTarget(
      name: "sim-app",
      dependencies: ["sim"]
    )
  ]
)
```

Настроим проект. Package

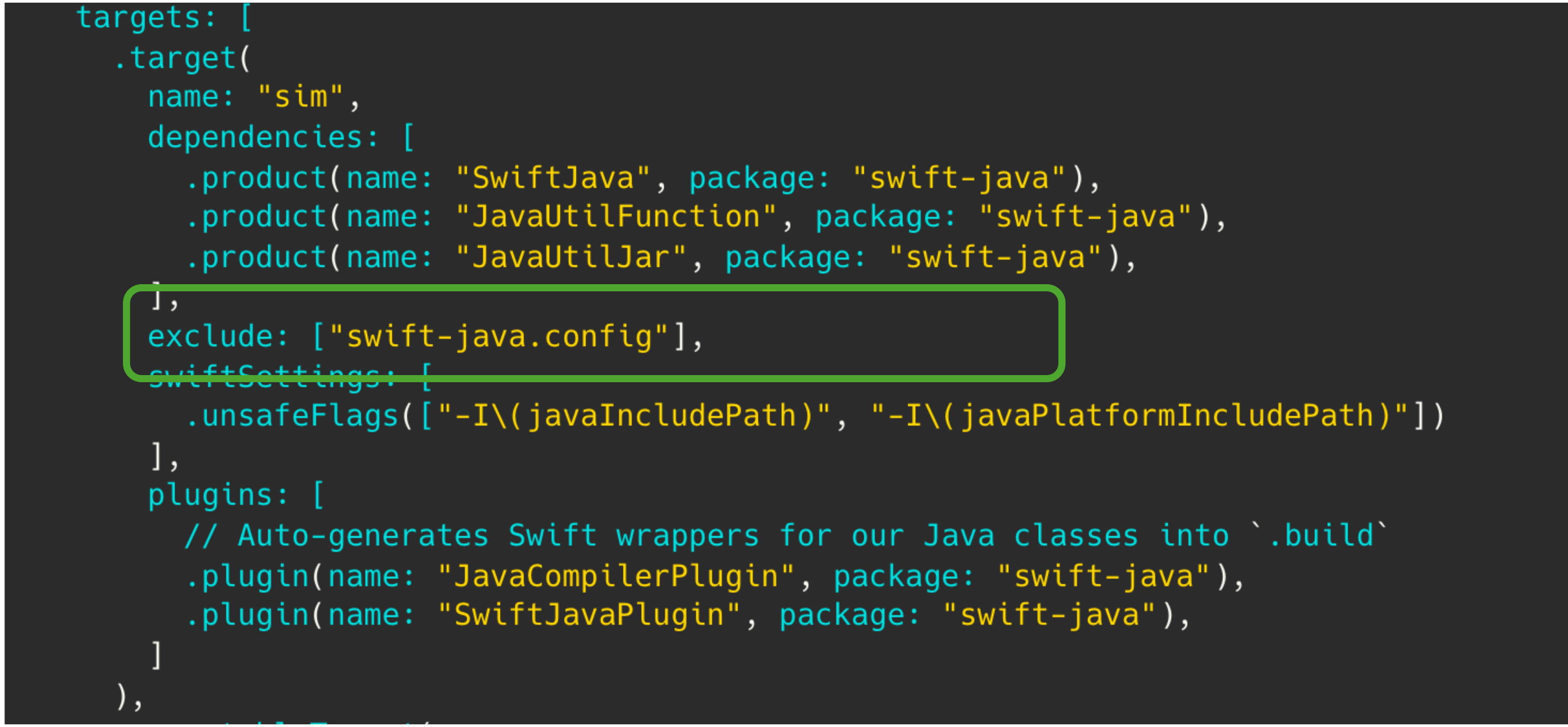
Подключаем полезные зависимости



```
let package = Package(
  name: "java-sim",
  platforms: [.macOS(.v15)],
  dependencies: [
    .package(url: "https://github.com/swiftlang/swift-java.git", branch: "main"),
  ],
  targets: [
    .target(
      name: "sim",
      dependencies: [
        .product(name: "SwiftJava", package: "swift-java"),
        .product(name: "JavaUtilFunction", package: "swift-java"),
        .product(name: "JavaUtilJar", package: "swift-java"),
      ],
      exclude: ["swift-java.config"],
      swiftSettings: [
        .unsafeFlags(["-I\(javaIncludePath)", "-I\(javaPlatformIncludePath)"])
      ],
      plugins: [
        // Auto-generates Swift wrappers for our Java classes into `.build`
        .plugin(name: "JavaCompilerPlugin", package: "swift-java"),
        .plugin(name: "SwiftJavaPlugin", package: "swift-java"),
      ]
    ),
    .executableTarget(
      name: "sim-app",
      dependencies: ["sim"]
    )
  ]
)
```

Настроим проект. Package

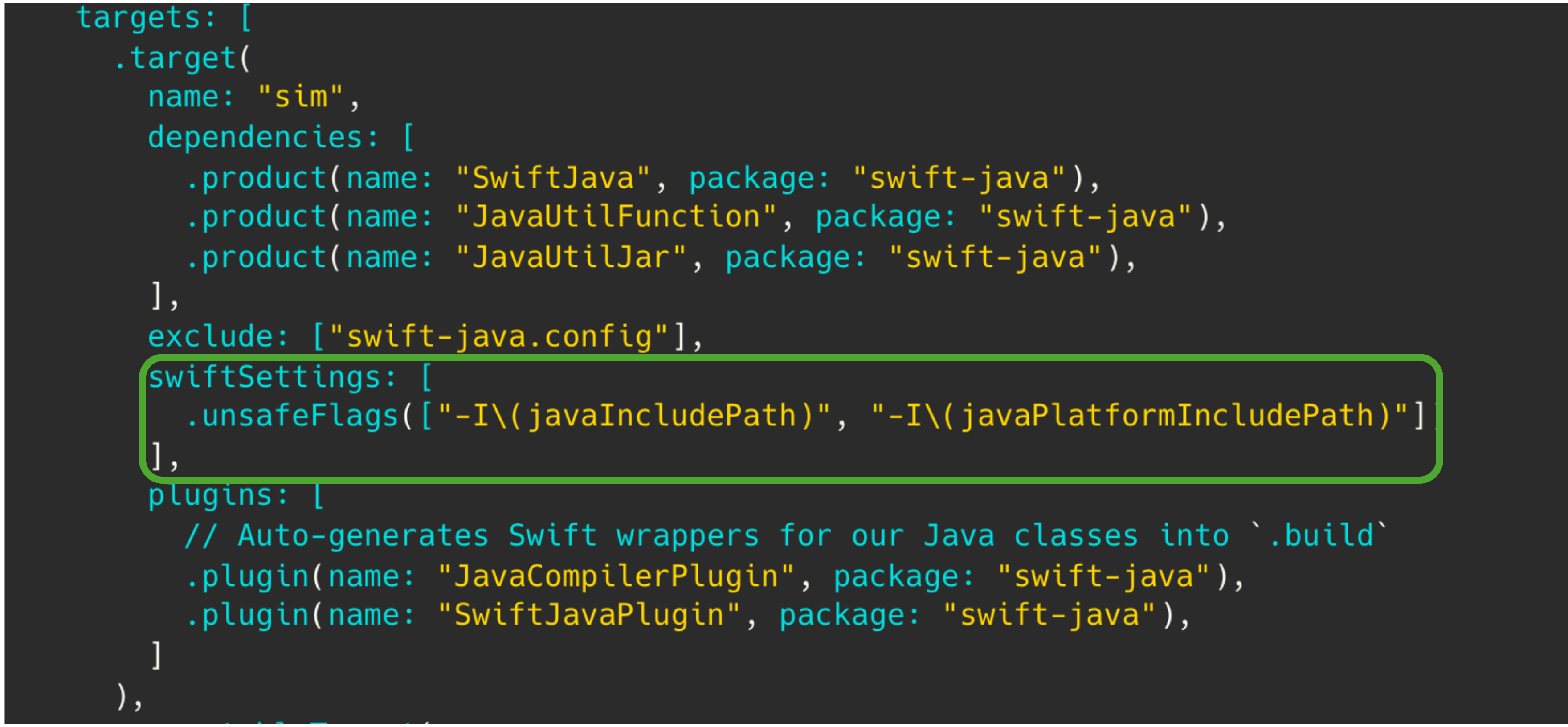
Подключаем конфиг



```
    targets: [
      .target(
        name: "sim",
        dependencies: [
          .product(name: "SwiftJava", package: "swift-java"),
          .product(name: "JavaUtilFunction", package: "swift-java"),
          .product(name: "JavaUtilJar", package: "swift-java"),
        ],
        exclude: ["swift-java.config"],
        swiftSettings: [
          .unsafeFlags(["-I\(javaIncludePath)", "-I\(javaPlatformIncludePath)"])
        ],
        plugins: [
          // Auto-generates Swift wrappers for our Java classes into `.build`
          .plugin(name: "JavaCompilerPlugin", package: "swift-java"),
          .plugin(name: "SwiftJavaPlugin", package: "swift-java"),
        ]
      ),
```

Настроим проект. Package

Подключаем пути к Java

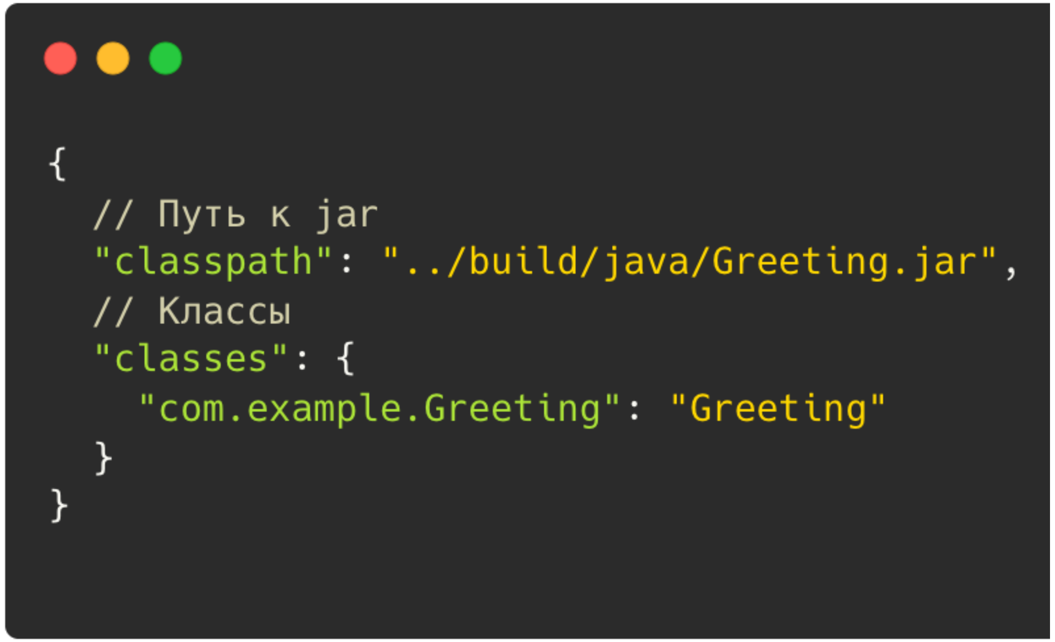


```
  targets: [
    .target(
      name: "sim",
      dependencies: [
        .product(name: "SwiftJava", package: "swift-java"),
        .product(name: "JavaUtilFunction", package: "swift-java"),
        .product(name: "JavaUtilJar", package: "swift-java"),
      ],
      exclude: ["swift-java.config"],
      swiftSettings: [
        .unsafeFlags(["-I\(javaIncludePath)", "-I\(javaPlatformIncludePath)"])
      ],
      plugins: [
        // Auto-generates Swift wrappers for our Java classes into `.build`
        .plugin(name: "JavaCompilerPlugin", package: "swift-java"),
        .plugin(name: "SwiftJavaPlugin", package: "swift-java"),
      ]
    ),
```

Настроим проект. Конфиг

Swift-Java.config

Пути ко всем файлам



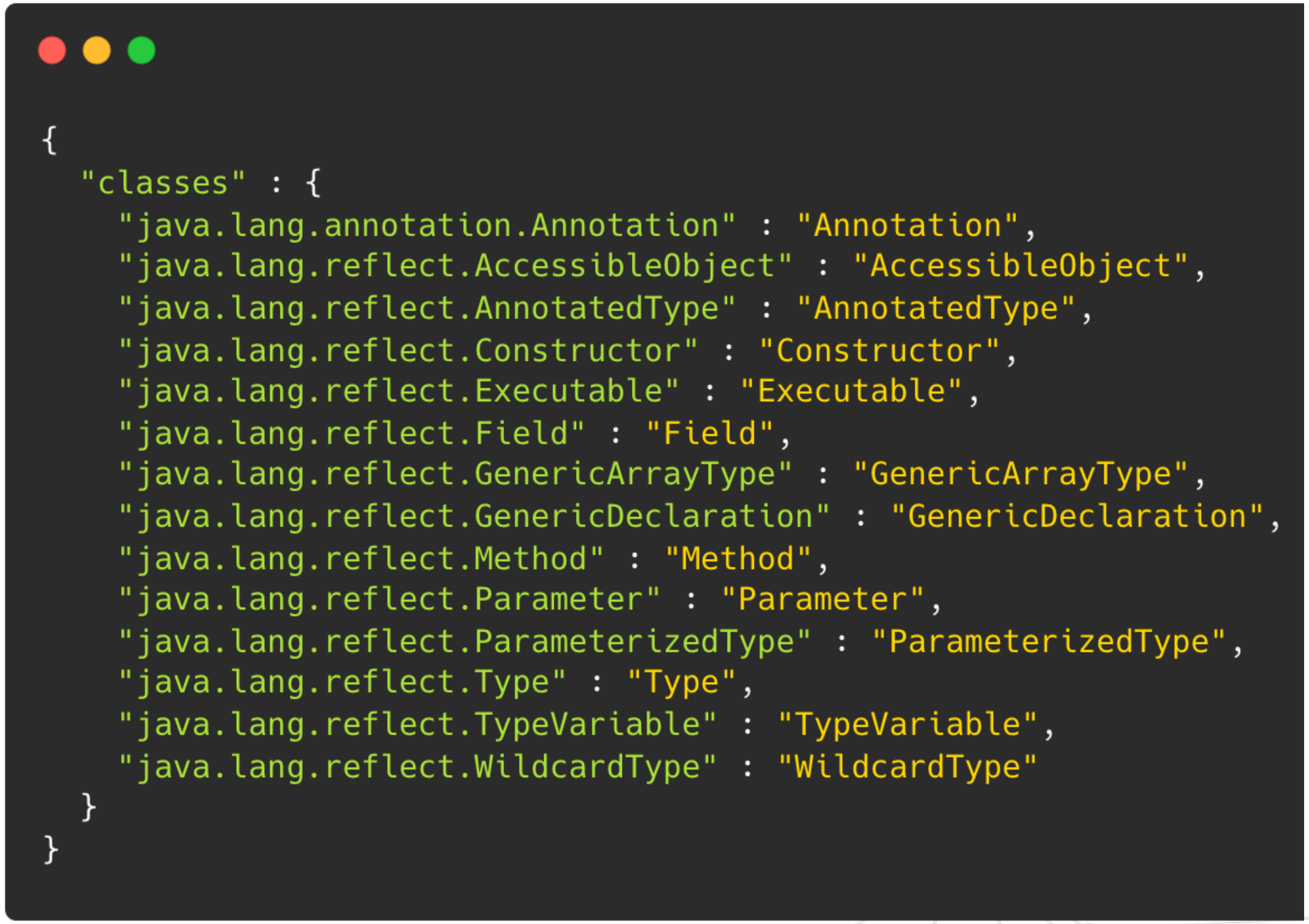
```
{
  // Путь к jar
  "classpath": "../build/java/Greeting.jar",
  // Классы
  "classes": {
    "com.example.Greeting": "Greeting"
  }
}
```

Настроим проект. Конфиг

Swift-Java.config

Пути ко всем файлам

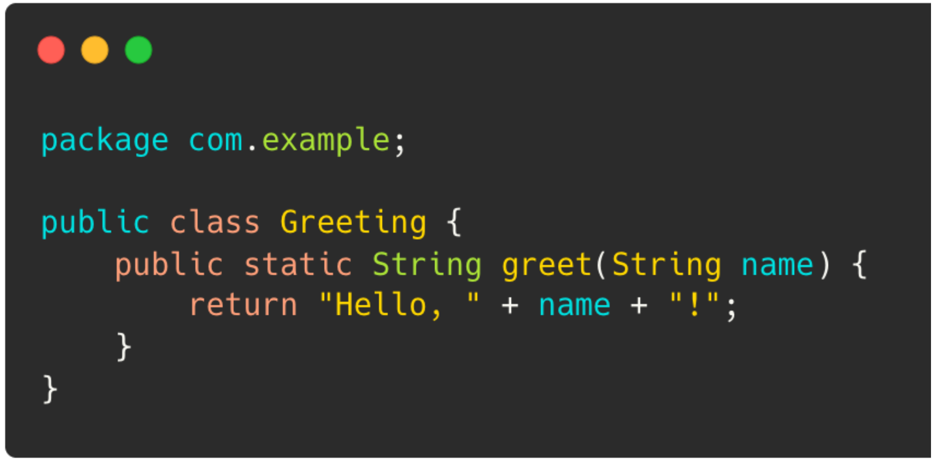
(пример Sources/JavaStdlib/JavaLangReflect/swift-java.config)



```
{
  "classes" : {
    "java.lang.annotation.Annotation" : "Annotation",
    "java.lang.reflect.AccessibleObject" : "AccessibleObject",
    "java.lang.reflect.AnnotatedType" : "AnnotatedType",
    "java.lang.reflect.Constructor" : "Constructor",
    "java.lang.reflect.Executable" : "Executable",
    "java.lang.reflect.Field" : "Field",
    "java.lang.reflect.GenericArrayType" : "GenericArrayType",
    "java.lang.reflect.GenericDeclaration" : "GenericDeclaration",
    "java.lang.reflect.Method" : "Method",
    "java.lang.reflect.Parameter" : "Parameter",
    "java.lang.reflect.ParameterizedType" : "ParameterizedType",
    "java.lang.reflect.Type" : "Type",
    "java.lang.reflect.TypeVariable" : "TypeVariable",
    "java.lang.reflect.WildcardType" : "WildcardType"
  }
}
```

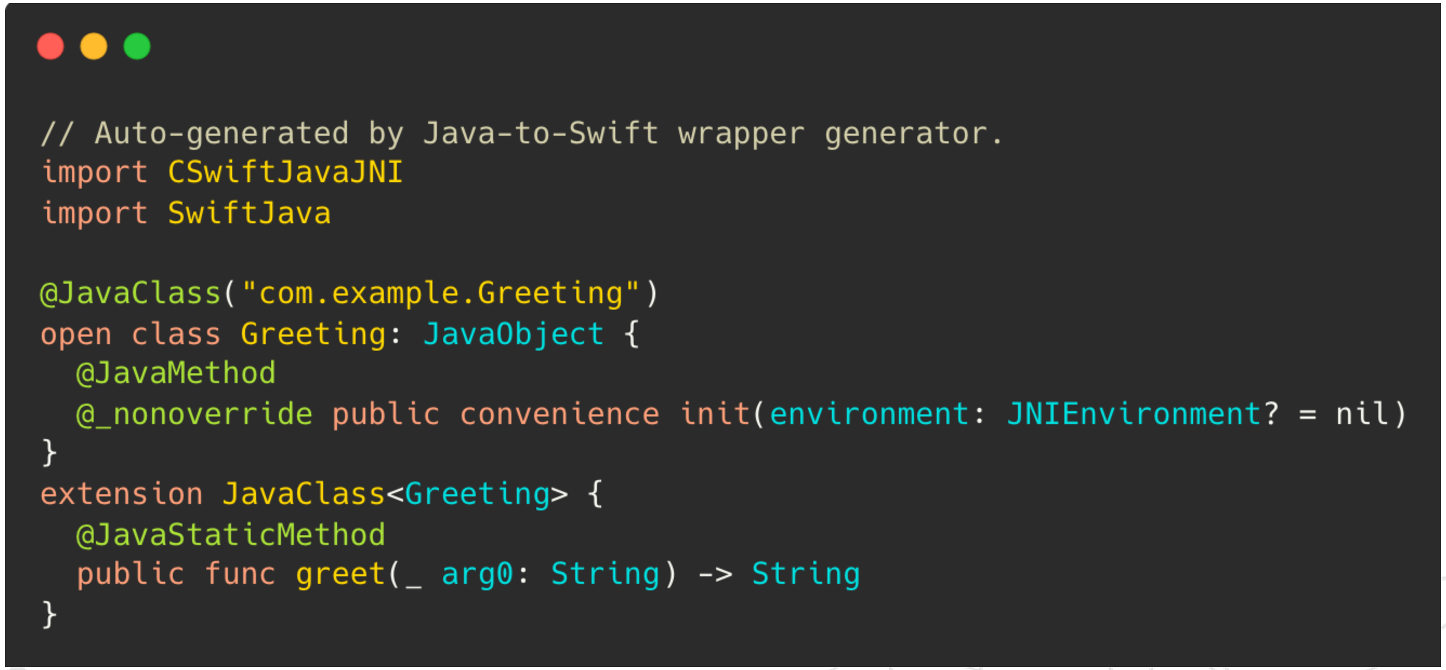
Настроим проект. Проверяем, запускаем

Было



```
package com.example;

public class Greeting {
    public static String greet(String name) {
        return "Hello, " + name + "!";
    }
}
```

Стало

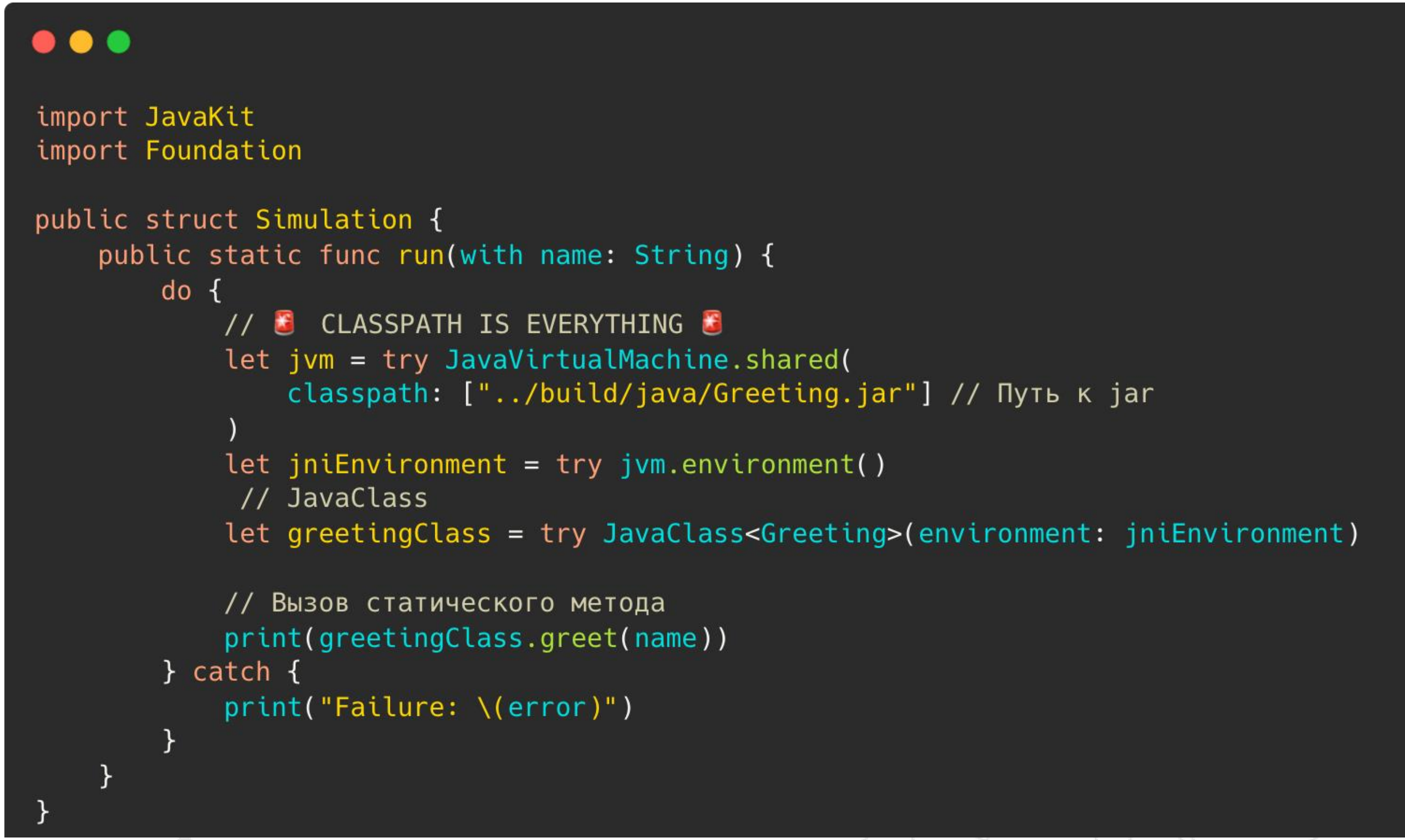


```
// Auto-generated by Java-to-Swift wrapper generator.
import CSwiftJavaJNI
import SwiftJava

@JavaClass("com.example.Greeting")
open class Greeting: JavaObject {
  @JavaMethod
  @_nonoverride public convenience init(environment: JNIEnvironment? = nil)
}
extension JavaClass<Greeting> {
  @JavaStaticMethod
  public func greet(_ arg0: String) -> String
}
```

Настроим проект. Проверяем, запускаем

Входная точка – статический метод

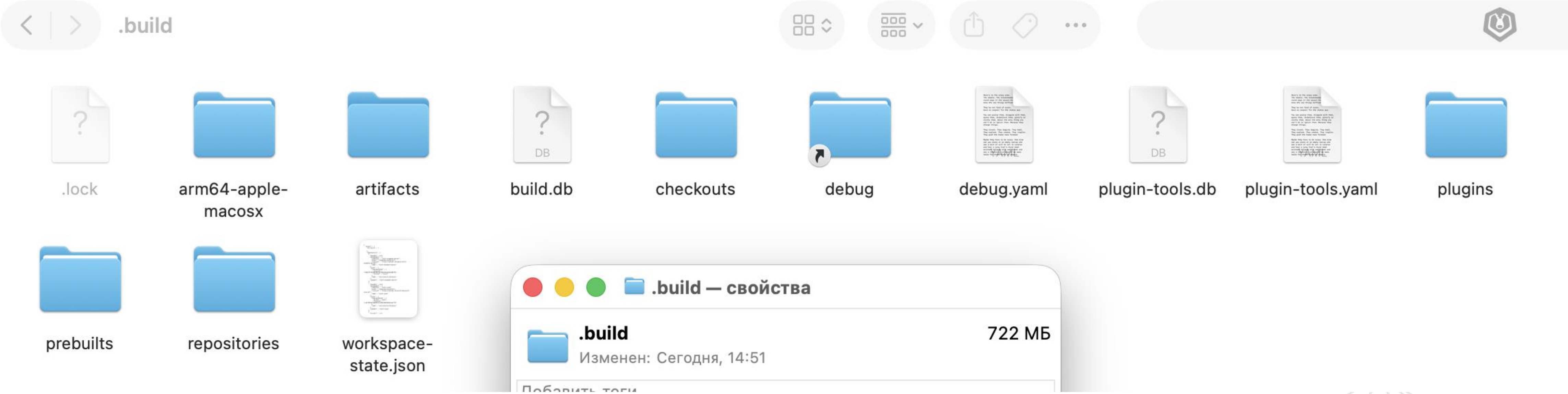
make – swift run



```swift
import JavaKit
import Foundation

public struct Simulation {
    public static func run(with name: String) {
        do {
            // 🚨 CLASSPATH IS EVERYTHING 🚨
            let jvm = try JavaVirtualMachine.shared(
                classpath: ["../build/java/Greeting.jar"] // Путь к jar
            )
            let jniEnvironment = try jvm.environment()
             // JavaClass
            let greetingClass = try JavaClass<Greeting>(environment: jniEnvironment)

            // Вызов статического метода
            print(greetingClass.greet(name))
        } catch {
            print("Failure: \(error)")
        }
    }
}
```

Настроим проект. Проверяем, запускаем

.build – здесь все



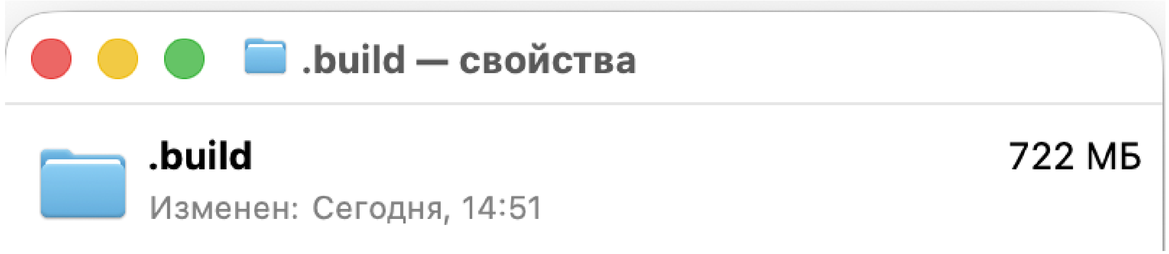
.build 722 МБ

Изменен: Сегодня, 14:51

Добавить теги

Настроим проект. Проверяем, запускаем

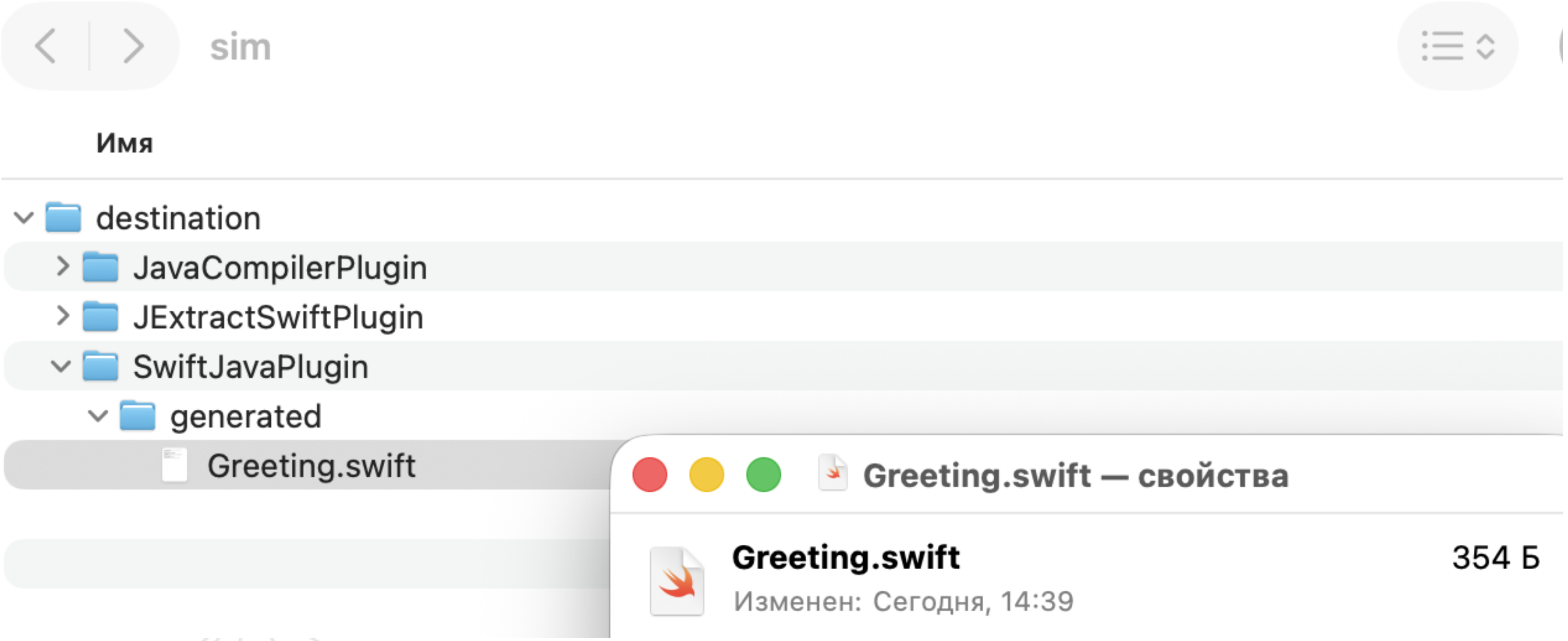
.build – здесь все



.build — свойства

.build 722 МБ

Изменен: Сегодня, 14:51



sim

Имя

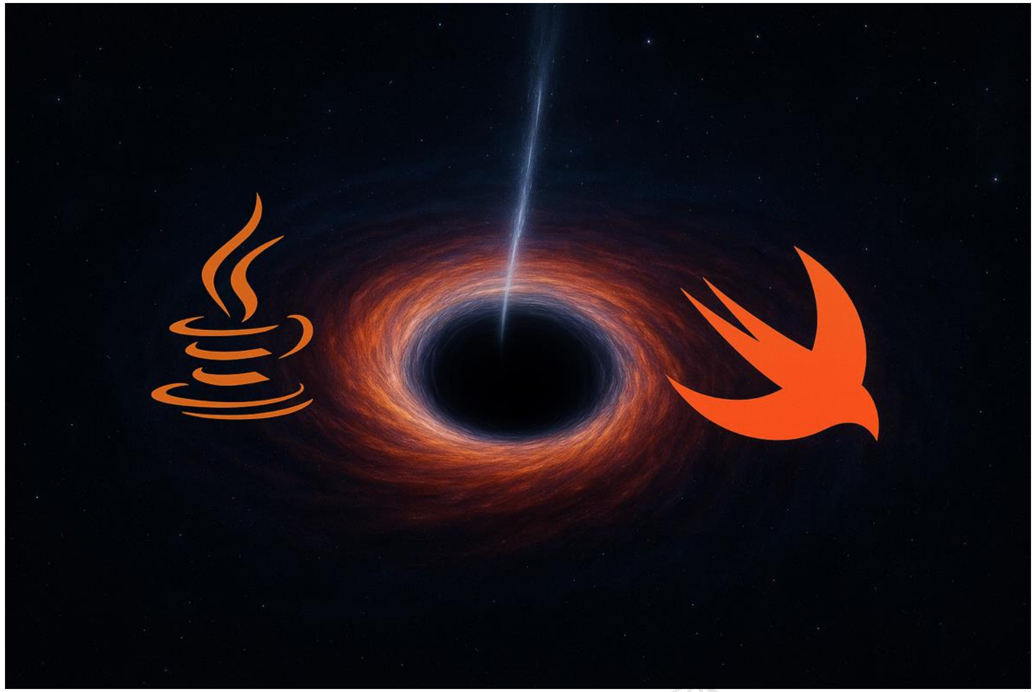
- destination
  - JavaCompilerPlugin
  - JExtractSwiftPlugin
  - SwiftJavaPlugin
    - generated
      - Greeting.swift

Greeting.swift — свойства

Greeting.swift 354 Б

Изменен: Сегодня, 14:39

Пример серьезнее
Сетевой клиент

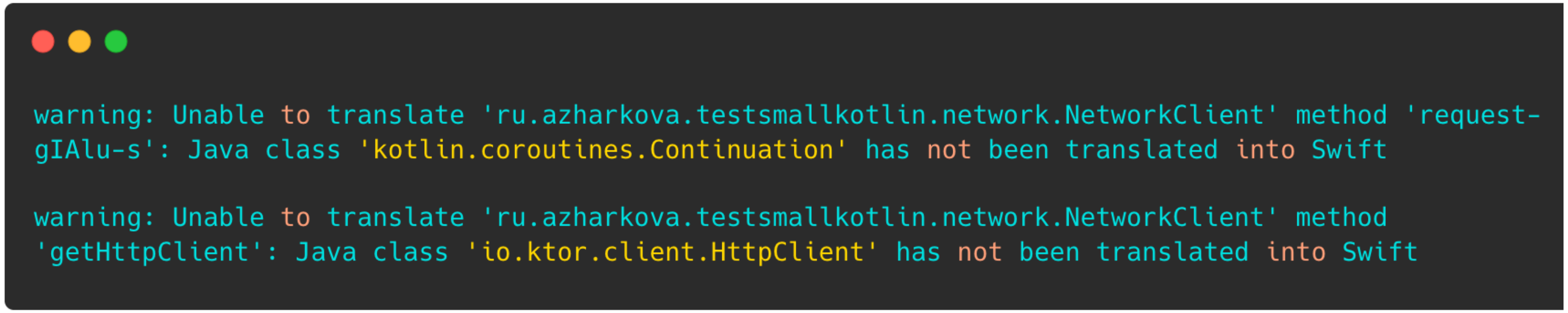


Сетевой клиент

- Ktor (неудачная)
- OkHttp (неудачная версия)
- Pure Java - рабочая

Ktor и Kotlin

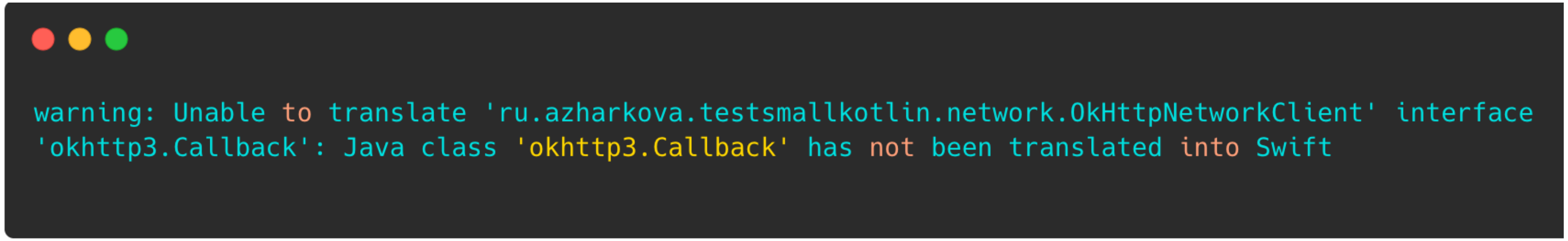
- Не все методы и свойства известны, не все транслируются
- Корутины работать не будут



```
warning: Unable to translate 'ru.azharkova.testsmallkotlin.network.NetworkClient' method 'request-
gIAlu-s': Java class 'kotlin.coroutines.Continuation' has not been translated into Swift

warning: Unable to translate 'ru.azharkova.testsmallkotlin.network.NetworkClient' method
'getHttpClient': Java class 'io.ktor.client.HttpClient' has not been translated into Swift
```

OkHttp

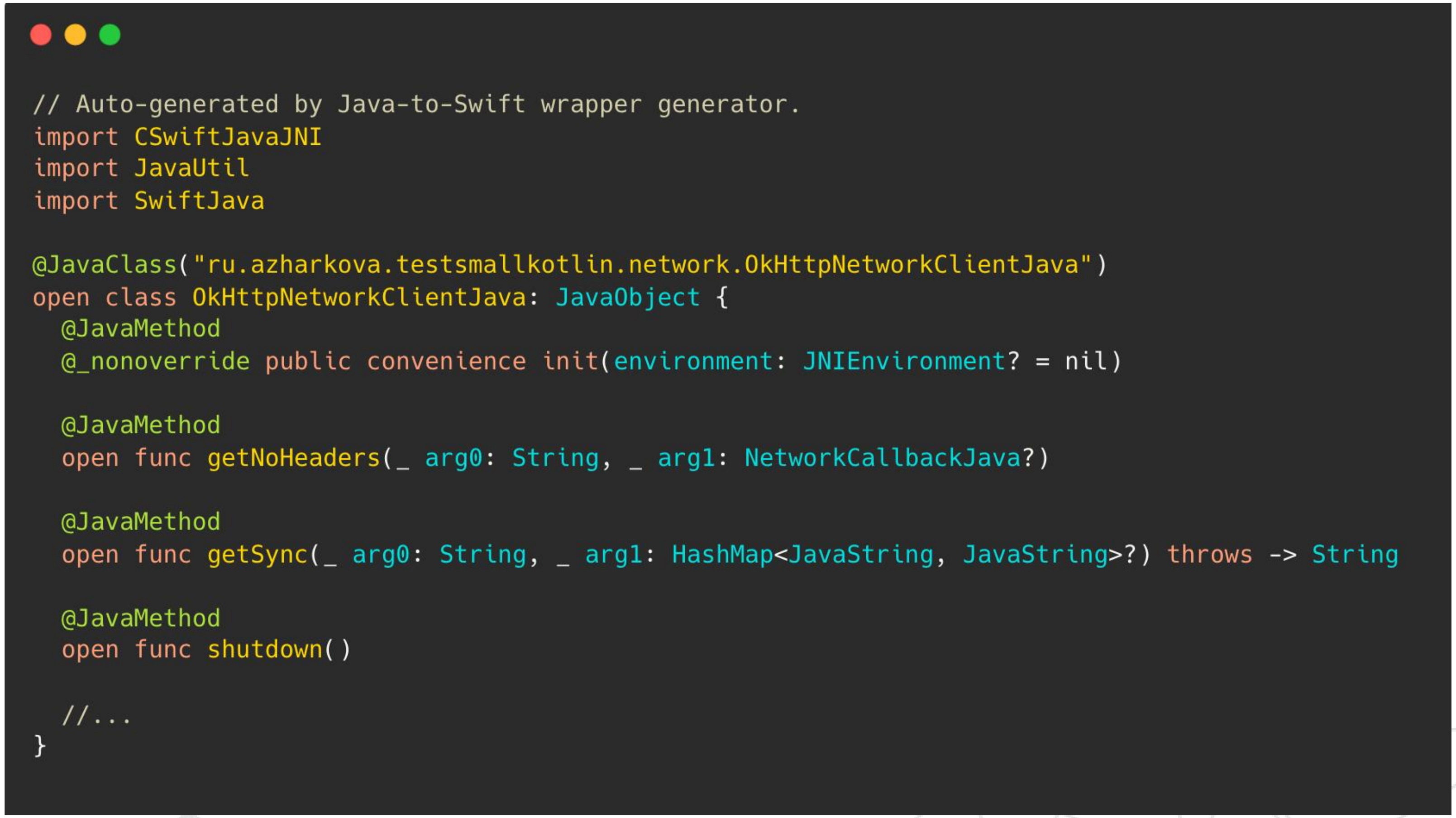
- Не все методы и свойства известны, не все транслируются



```
warning: Unable to translate 'ru.azharkova.testsmallkotlin.network.OkHttpNetworkClient' interface
'okhttp3.Callback': Java class 'okhttp3.Callback' has not been translated into Swift
```

OkHttp

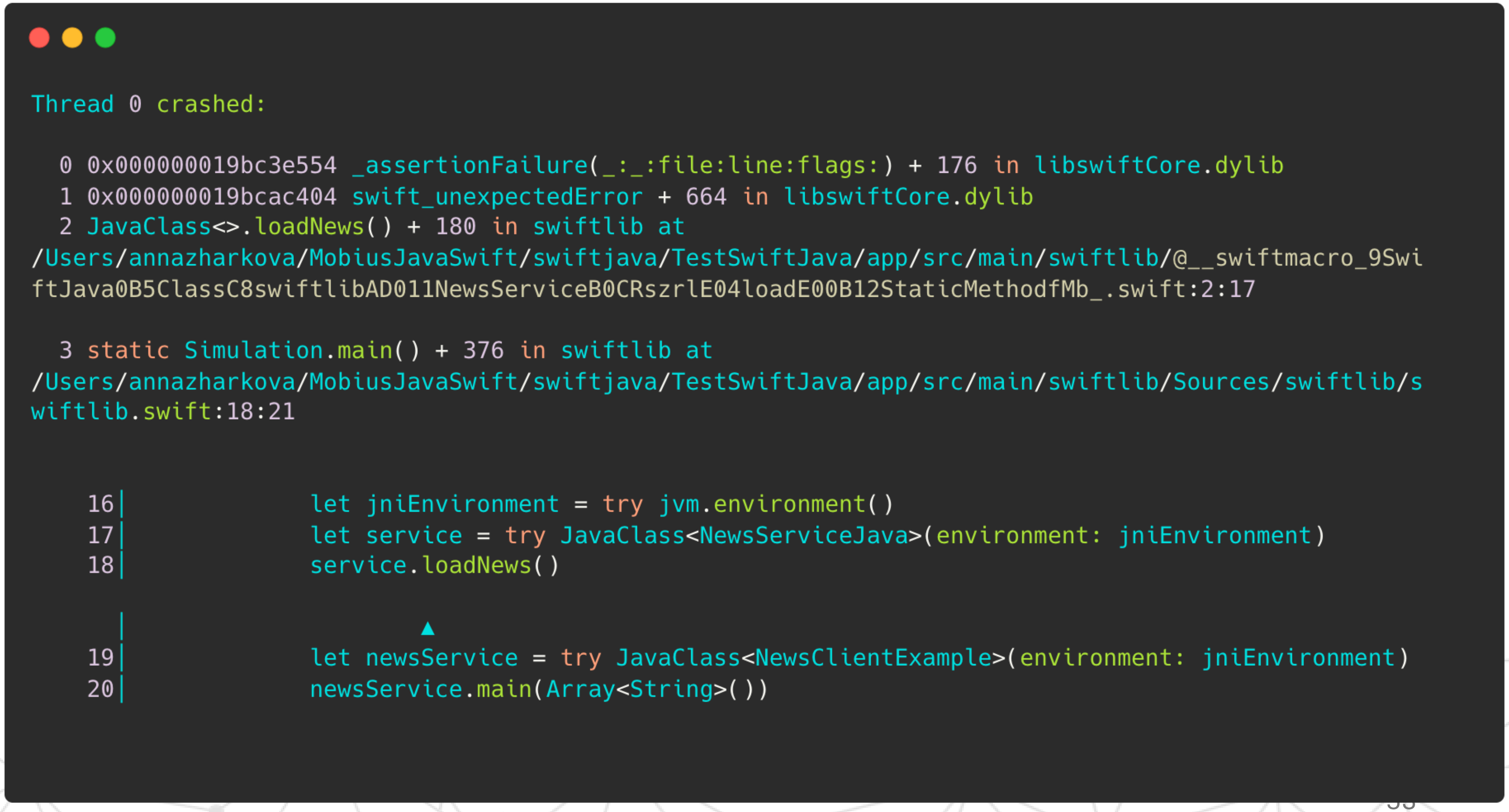
- Файл сгенерирован, но методы не отработают



```
// Auto-generated by Java-to-Swift wrapper generator.
import CSwiftJavaJNI
import JavaUtil
import SwiftJava

@JavaClass("ru.azharkova.testsmallkotlin.network.OkHttpNetworkClientJava")
open class OkHttpNetworkClientJava: JavaObject {
  @JavaMethod
  @_nonoverride public convenience init(environment: JNIEnvironment? = nil)

  @JavaMethod
  open func getNoHeaders(_ arg0: String, _ arg1: NetworkCallbackJava?)

  @JavaMethod
  open func getSync(_ arg0: String, _ arg1: HashMap<JavaString, JavaString>?) throws -> String

  @JavaMethod
  open func shutdown()

  //...
}
```

OkHttp

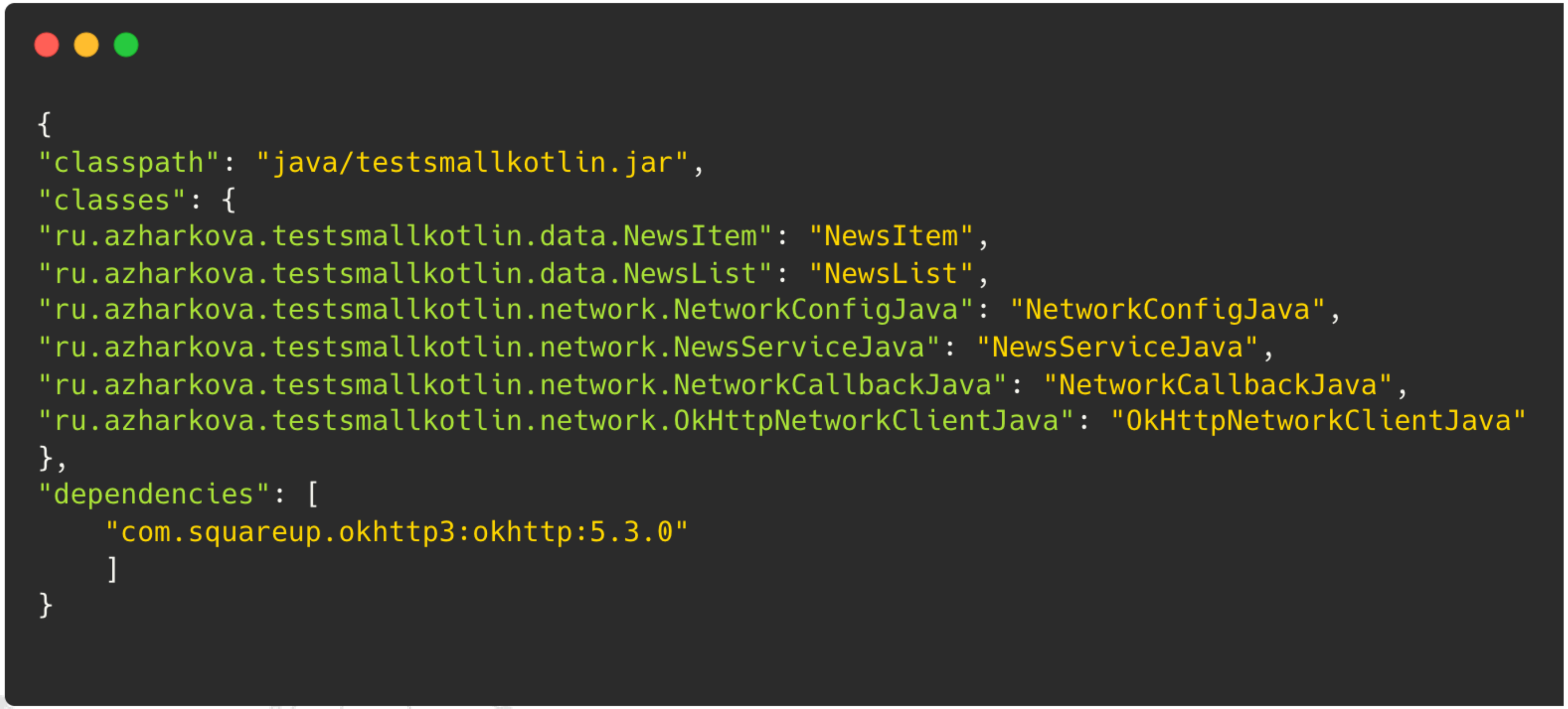
- Kotlin concurrency крашится при вызове из стандартного потока (конфликт)



```
Thread 0 crashed:

  0 0x000000019bc3e554 _assertionFailure(_:_:file:line:flags:) + 176 in libswiftCore.dylib
  1 0x000000019bcac404 swift_unexpectedError + 664 in libswiftCore.dylib
  2 JavaClass<>.loadNews() + 180 in swiftlib at
/Users/annazharkova/MobiusJavaSwift/swiftjava/TestSwiftJava/app/src/main/swiftlib/@__swiftmacro_9Swi
ftJava0B5ClassC8swiftlibAD011NewsServiceB0CRszrlE04loadE00B12StaticMethodfMb_.swift:2:17

  3 static Simulation.main() + 376 in swiftlib at
/Users/annazharkova/MobiusJavaSwift/swiftjava/TestSwiftJava/app/src/main/swiftlib/Sources/swiftlib/s
wiftlib.swift:18:21


    16|             let jniEnvironment = try jvm.environment()
    17|             let service = try JavaClass<NewsServiceJava>(environment: jniEnvironment)
    18|             service.loadNews()
      |                      ▲
    19|             let newsService = try JavaClass<NewsClientExample>(environment: jniEnvironment)
    20|             newsService.main(Array<String>())
```

Настраиваем зависимости

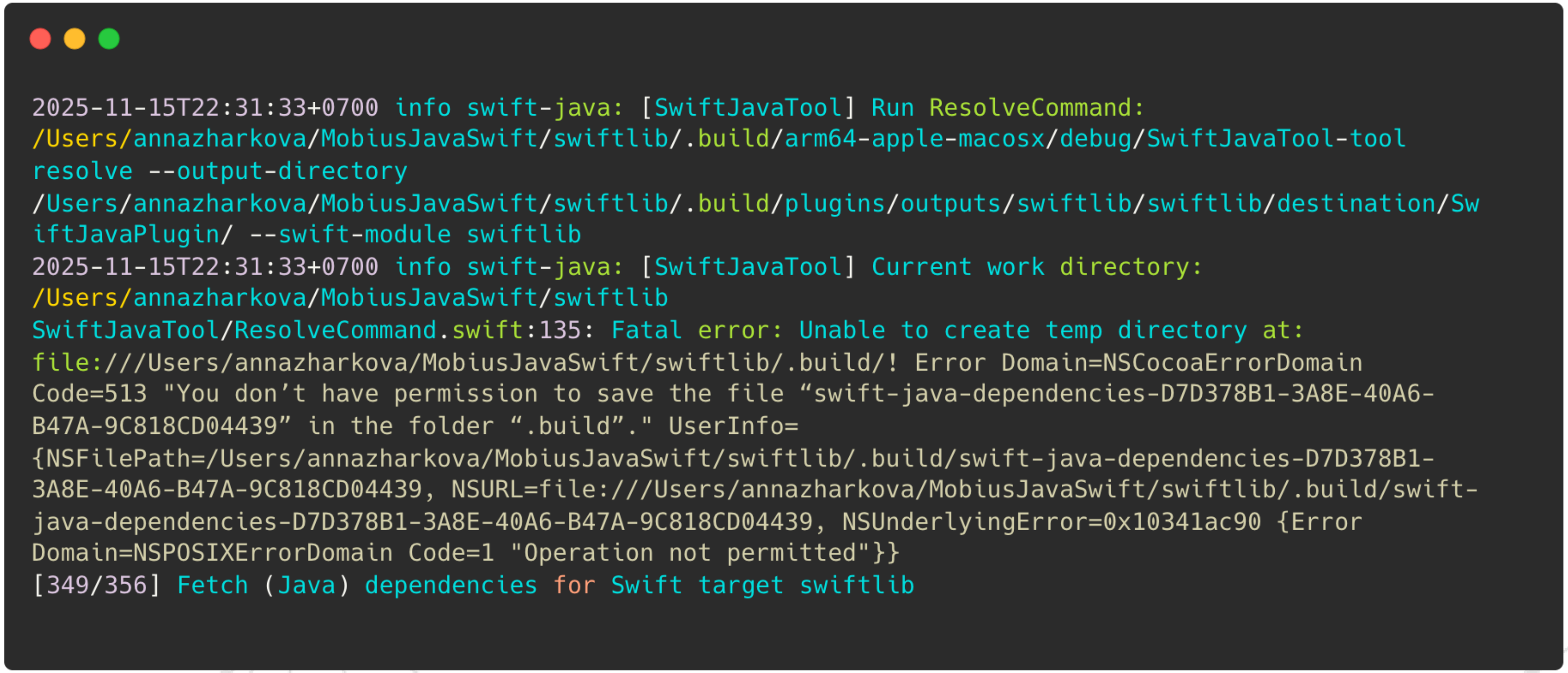
swift-java.config



```
{
"classpath": "java/testsmallkotlin.jar",
"classes": {
"ru.azharkova.testsmallkotlin.data.NewsItem": "NewsItem",
"ru.azharkova.testsmallkotlin.data.NewsList": "NewsList",
"ru.azharkova.testsmallkotlin.network.NetworkConfigJava": "NetworkConfigJava",
"ru.azharkova.testsmallkotlin.network.NewsServiceJava": "NewsServiceJava",
"ru.azharkova.testsmallkotlin.network.NetworkCallbackJava": "NetworkCallbackJava",
"ru.azharkova.testsmallkotlin.network.OkHttpNetworkClientJava": "OkHttpNetworkClientJava"
},
"dependencies": [
    "com.squareup.okhttp3:okhttp:5.3.0"
    ]
}
```

Запускаем стандартно

swift run -> падаем на скачивании зависимостей

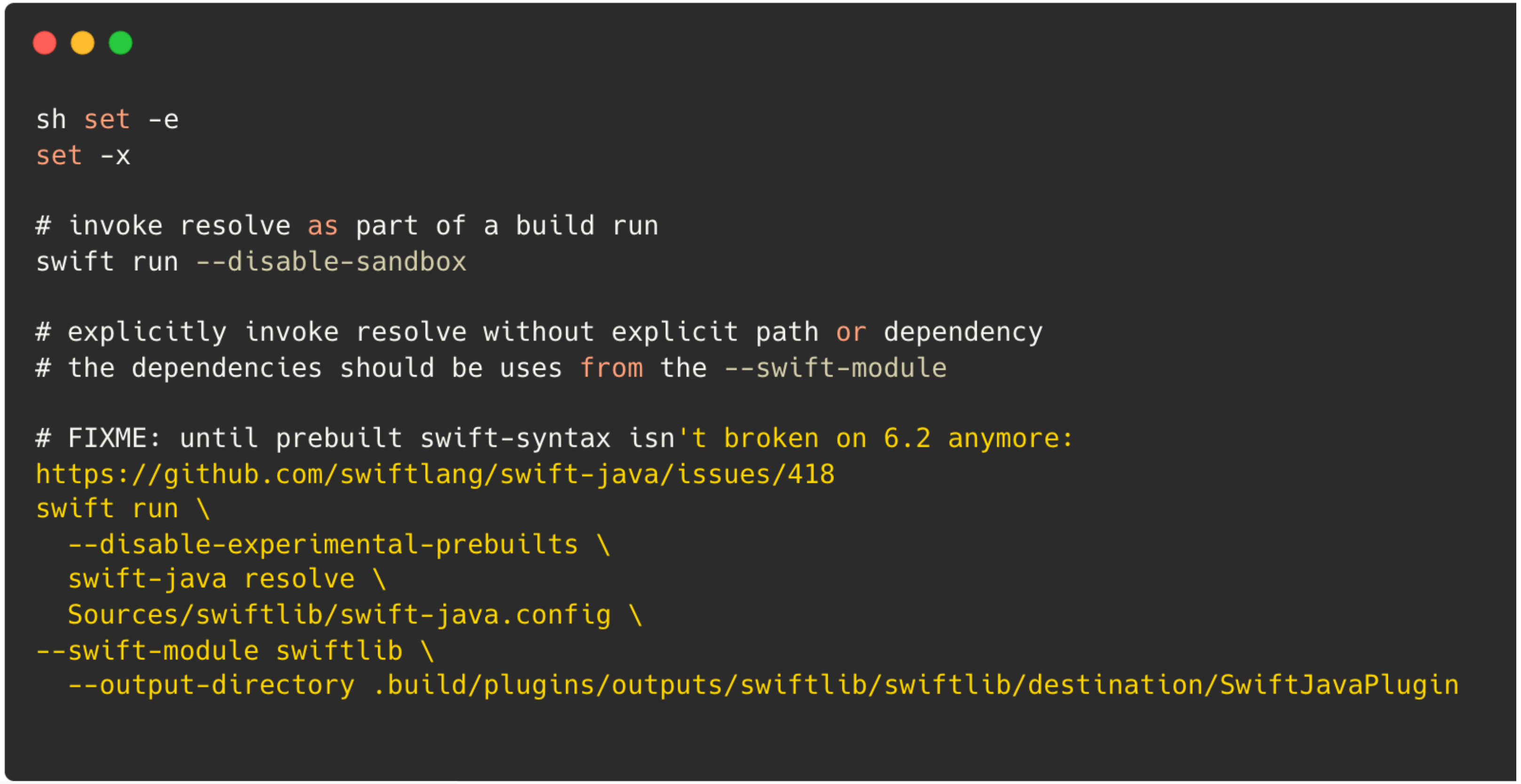


```
2025-11-15T22:31:33+0700 info swift-java: [SwiftJavaTool] Run ResolveCommand:
/Users/annazharkova/MobiusJavaSwift/swiftlib/.build/arm64-apple-macosx/debug/SwiftJavaTool-tool
resolve --output-directory
/Users/annazharkova/MobiusJavaSwift/swiftlib/.build/plugins/outputs/swiftlib/swiftlib/destination/Sw
iftJavaPlugin/ --swift-module swiftlib
2025-11-15T22:31:33+0700 info swift-java: [SwiftJavaTool] Current work directory:
/Users/annazharkova/MobiusJavaSwift/swiftlib
SwiftJavaTool/ResolveCommand.swift:135: Fatal error: Unable to create temp directory at:
file:///Users/annazharkova/MobiusJavaSwift/swiftlib/.build/! Error Domain=NSCocoaErrorDomain
Code=513 "You don’t have permission to save the file “swift-java-dependencies-D7D378B1-3A8E-40A6-
B47A-9C818CD04439” in the folder “.build”." UserInfo=
{NSFilePath=/Users/annazharkova/MobiusJavaSwift/swiftlib/.build/swift-java-dependencies-D7D378B1-
3A8E-40A6-B47A-9C818CD04439, NSURL=file:///Users/annazharkova/MobiusJavaSwift/swiftlib/.build/swift-
java-dependencies-D7D378B1-3A8E-40A6-B47A-9C818CD04439, NSUnderlyingError=0x10341ac90 {Error
Domain=NSPOSIXErrorDomain Code=1 "Operation not permitted"}}
[349/356] Fetch (Java) dependencies for Swift target swiftlib
```

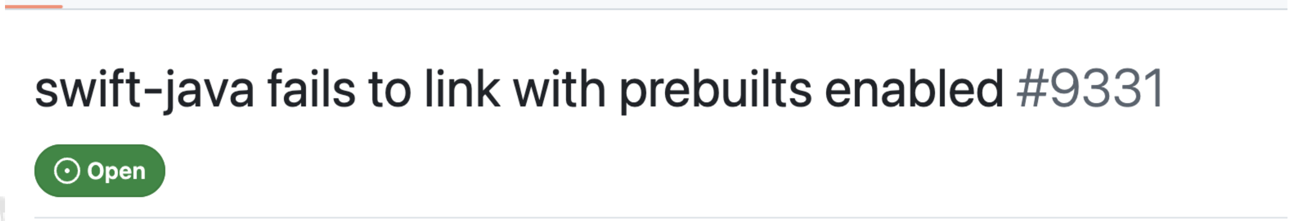
Запускаем через скрипт

Известный баг

Фиксим пребилты



```
sh set -e
set -x

# invoke resolve as part of a build run
swift run --disable-sandbox

# explicitly invoke resolve without explicit path or dependency
# the dependencies should be uses from the --swift-module

# FIXME: until prebuilt swift-syntax isn't broken on 6.2 anymore:
https://github.com/swiftlang/swift-java/issues/418
swift run \
  --disable-experimental-prebuilts \
  swift-java resolve \
  Sources/swiftlib/swift-java.config \
--swift-module swiftlib \
  --output-directory .build/plugins/outputs/swiftlib/swiftlib/destination/SwiftJavaPlugin
```

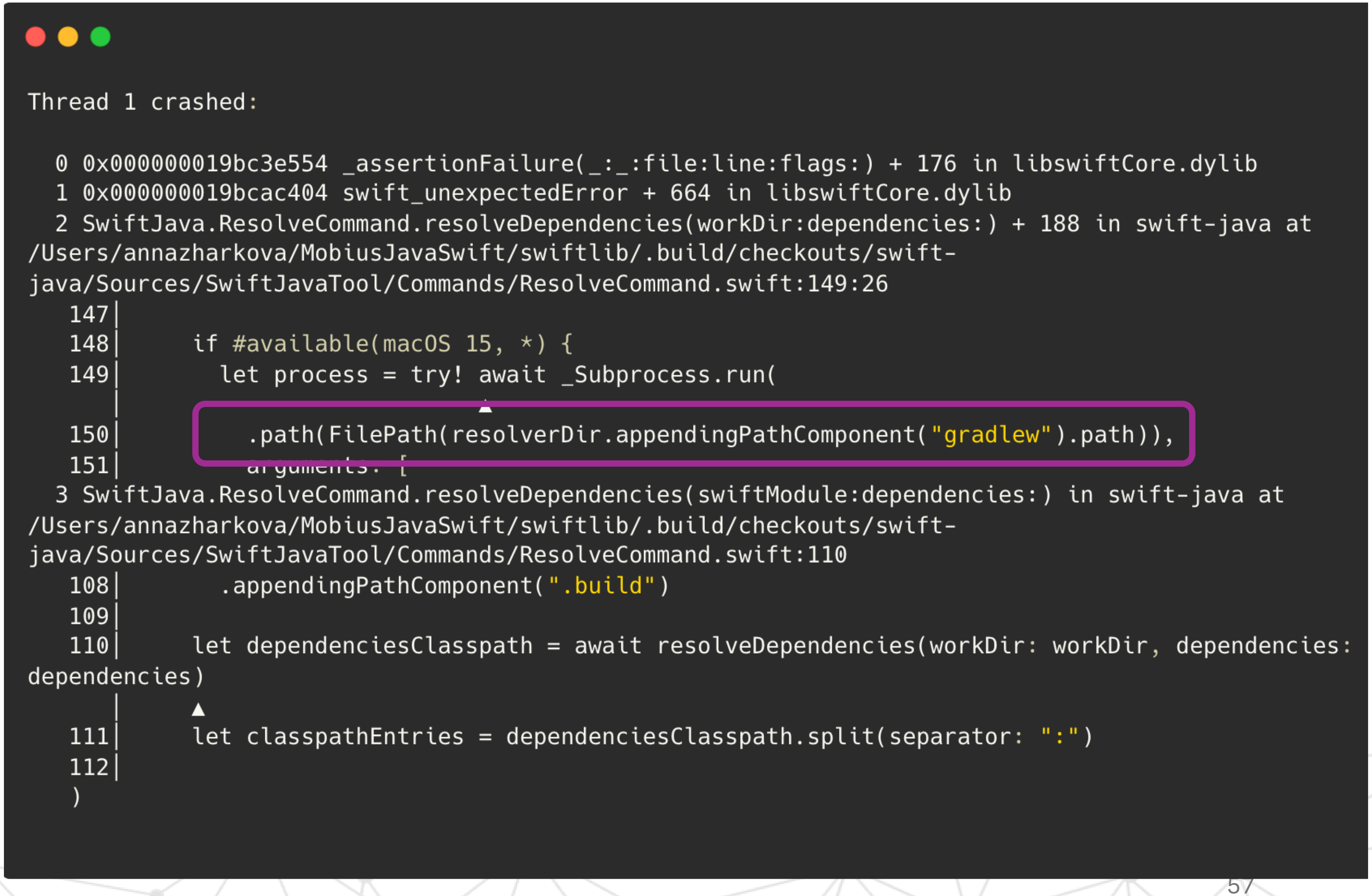


swift-java fails to link with prebuilts enabled #9331

Запускаем через скрипт

Crash!

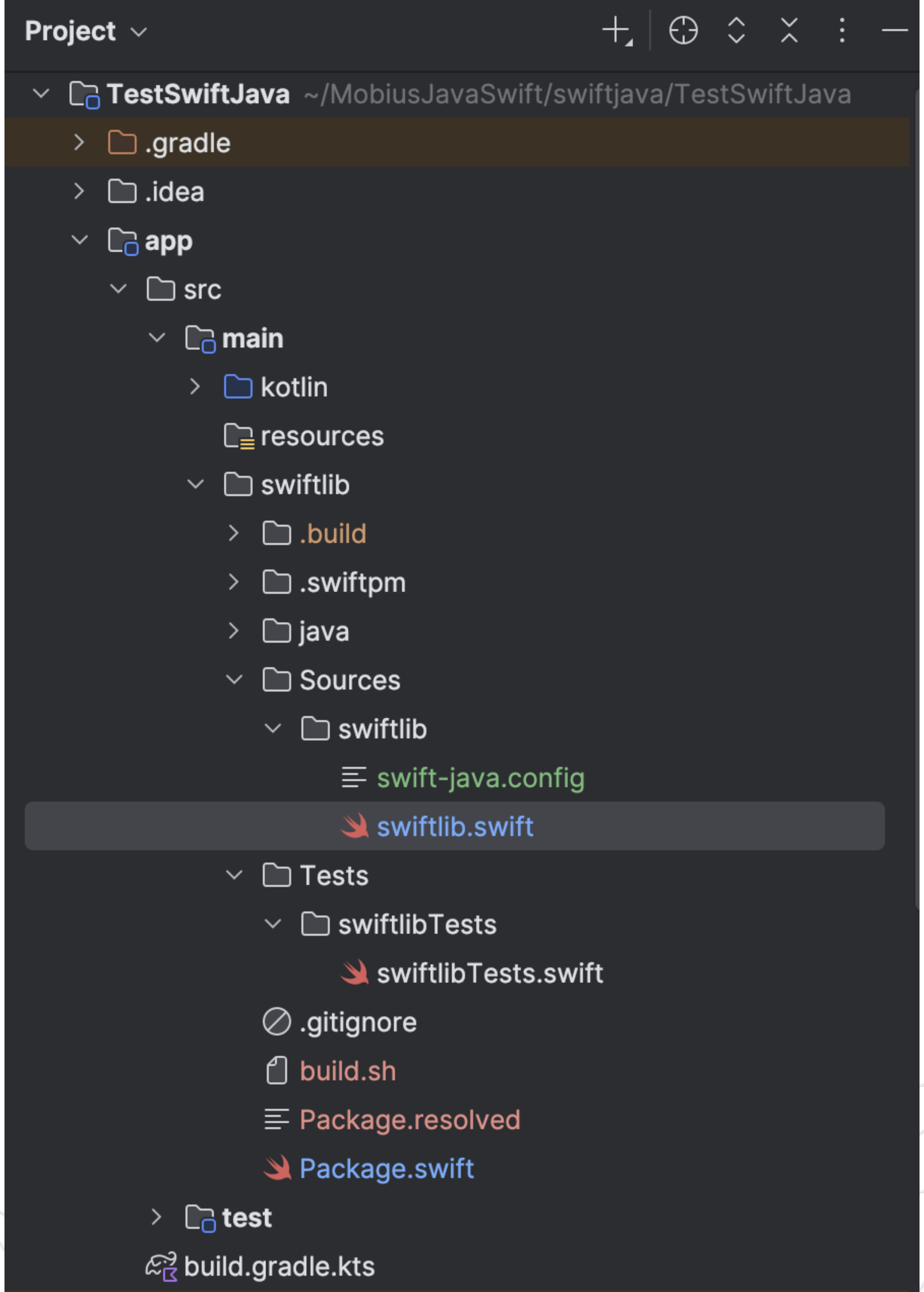
Не нашли gradlew



```
Thread 1 crashed:

  0 0x000000019bc3e554 _assertionFailure(_:_:file:line:flags:) + 176 in libswiftCore.dylib
  1 0x000000019bcac404 swift_unexpectedError + 664 in libswiftCore.dylib
  2 SwiftJava.ResolveCommand.resolveDependencies(workDir:dependencies:) + 188 in swift-java at
/Users/annazharkova/MobiusJavaSwift/swiftlib/.build/checkouts/swift-
java/Sources/SwiftJavaTool/Commands/ResolveCommand.swift:149:26
    147│
    148│      if #available(macOS 15, *) {
    149│        let process = try! await _Subprocess.run(
       │                           ▲
    150│          .path(FilePath(resolverDir.appendingPathComponent("gradlew").path)),
    151│          arguments: [
  3 SwiftJava.ResolveCommand.resolveDependencies(swiftModule:dependencies:) in swift-java at
/Users/annazharkova/MobiusJavaSwift/swiftlib/.build/checkouts/swift-
java/Sources/SwiftJavaTool/Commands/ResolveCommand.swift:110
    108│        .appendingPathComponent(".build")
    109│
    110│      let dependenciesClasspath = await resolveDependencies(workDir: workDir, dependencies:
dependencies)
       │      ▲
    111│      let classpathEntries = dependenciesClasspath.split(separator: ":")
    112│
    )
```

Gradle в структуру!!!

Если нужны зависимости gradle, то кладем Package swift вовнутрь Gradle проекта

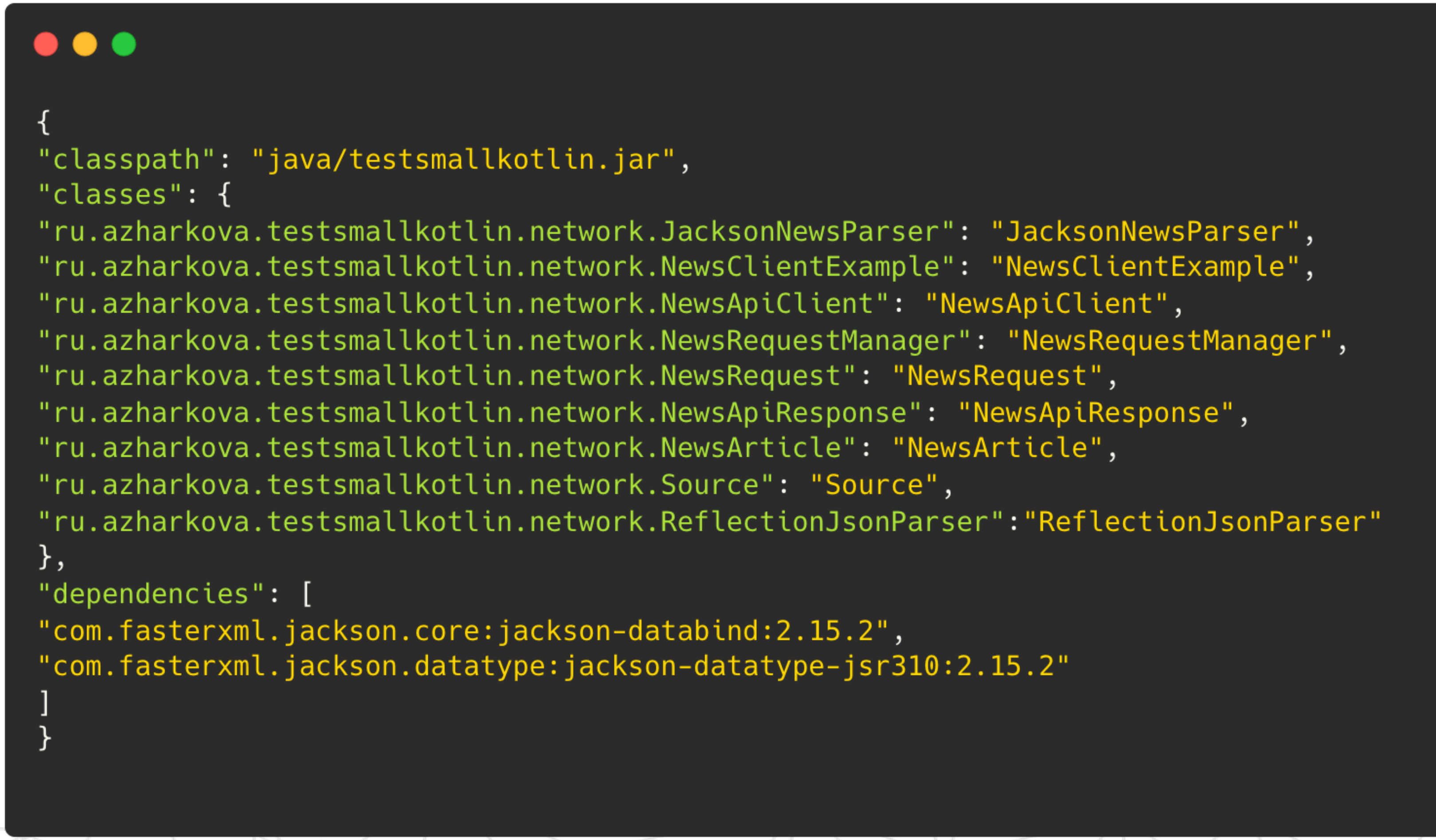


Project

```
TestSwiftJava ~/MobiusJavaSwift/swiftjava/TestSwiftJava
  .gradle
  .idea
  app
    src
      main
        kotlin
        resources
        swiftlib
          .build
          .swiftpm
          java
          Sources
            swiftlib
              swift-java.config
              swiftlib.swift
          Tests
            swiftlibTests
              swiftlibTests.swift
          .gitignore
          build.sh
          Package.resolved
          Package.swift
      test
    build.gradle.kts
```

Pure Java вариант

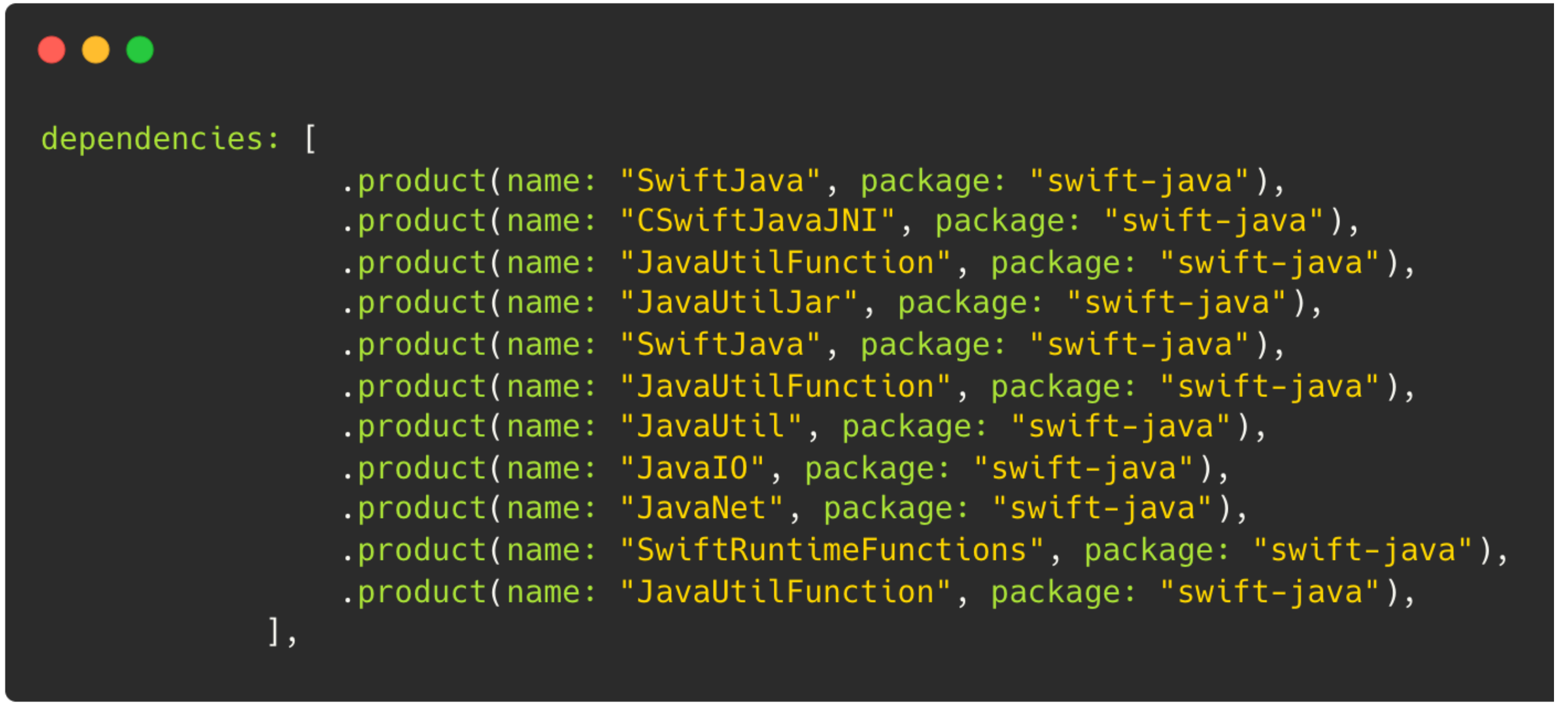
- Многопоточность на Java.Concurrency
- Сеть на Java.Net
- Парсер на рефлексии



```
{
"classpath": "java/testsmallkotlin.jar",
"classes": {
"ru.azharkova.testsmallkotlin.network.JacksonNewsParser": "JacksonNewsParser",
"ru.azharkova.testsmallkotlin.network.NewsClientExample": "NewsClientExample",
"ru.azharkova.testsmallkotlin.network.NewsApiClient": "NewsApiClient",
"ru.azharkova.testsmallkotlin.network.NewsRequestManager": "NewsRequestManager",
"ru.azharkova.testsmallkotlin.network.NewsRequest": "NewsRequest",
"ru.azharkova.testsmallkotlin.network.NewsApiResponse": "NewsApiResponse",
"ru.azharkova.testsmallkotlin.network.NewsArticle": "NewsArticle",
"ru.azharkova.testsmallkotlin.network.Source": "Source",
"ru.azharkova.testsmallkotlin.network.ReflectionJsonParser":"ReflectionJsonParser"
},
"dependencies": [
"com.fasterxml.jackson.core:jackson-databind:2.15.2",
"com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.15.2"
]
}
```

Pure Java вариант

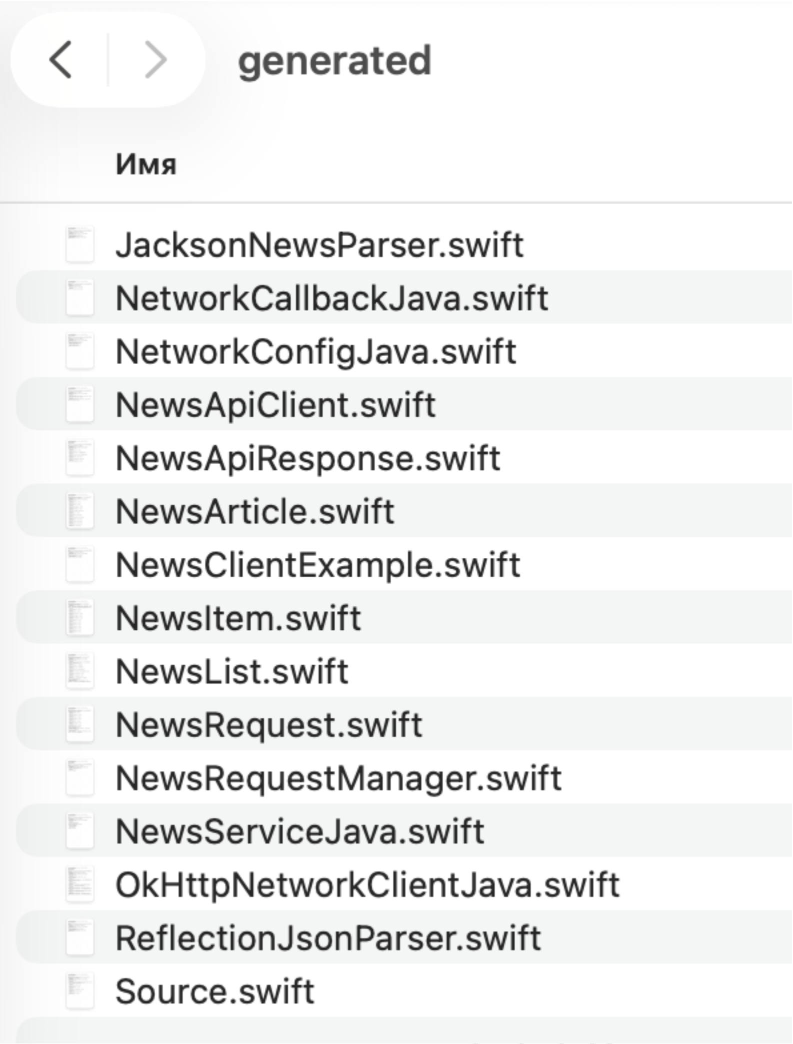
- Многопоточность на Java.Concurrency
- Сеть на Java.Net
- Парсер на рефлексии



```
dependencies: [
                .product(name: "SwiftJava", package: "swift-java"),
                .product(name: "CSwiftJavaJNI", package: "swift-java"),
                .product(name: "JavaUtilFunction", package: "swift-java"),
                .product(name: "JavaUtilJar", package: "swift-java"),
                .product(name: "SwiftJava", package: "swift-java"),
                .product(name: "JavaUtilFunction", package: "swift-java"),
                .product(name: "JavaUtil", package: "swift-java"),
                .product(name: "JavaIO", package: "swift-java"),
                .product(name: "JavaNet", package: "swift-java"),
                .product(name: "SwiftRuntimeFunctions", package: "swift-java"),
                .product(name: "JavaUtilFunction", package: "swift-java"),
            ],
```

Pure Java вариант

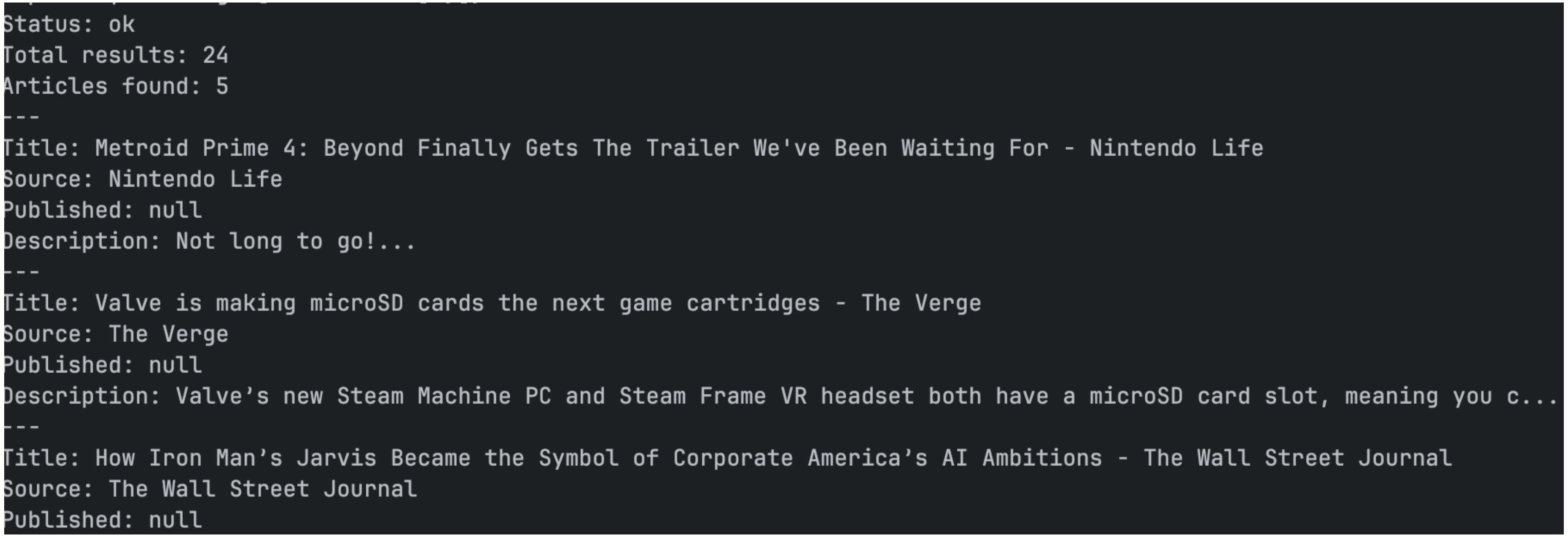
- Готовые данные



JacksonNewsParser.swift
NetworkCallbackJava.swift
NetworkConfigJava.swift
NewsApiClient.swift
NewsApiResponse.swift
NewsArticle.swift
NewsClientExample.swift
NewsItem.swift
NewsList.swift
NewsRequest.swift
NewsRequestManager.swift
NewsServiceJava.swift
OkHttpNetworkClientJava.swift
ReflectionJsonParser.swift
Source.swift

Pure Java вариант

- Результат

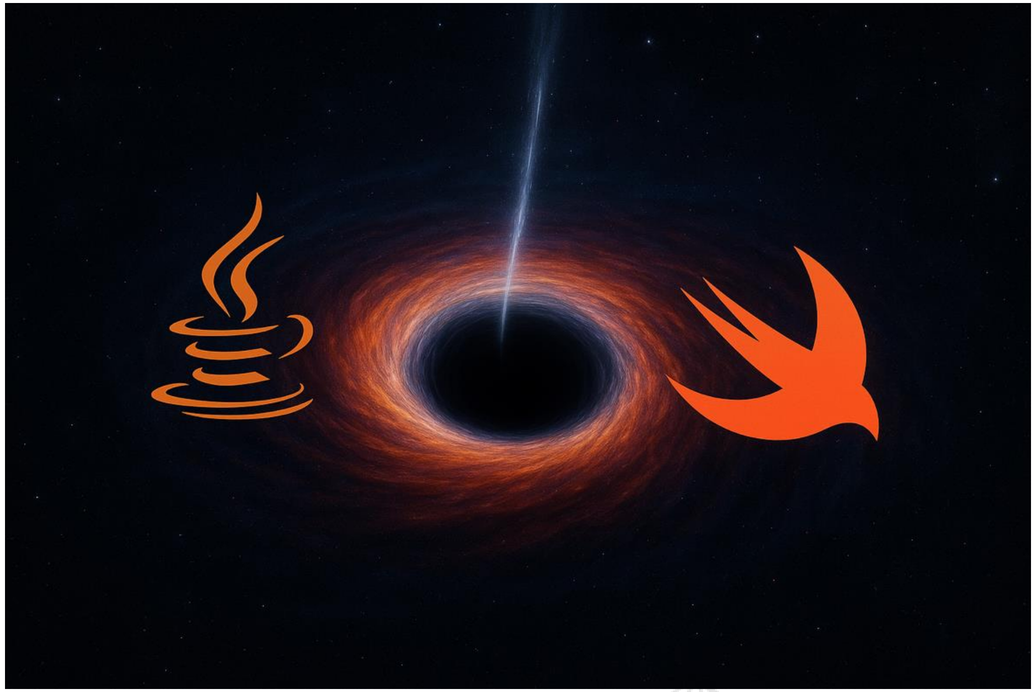


```
Status: ok
Total results: 24
Articles found: 5
---
Title: Metroid Prime 4: Beyond Finally Gets The Trailer We've Been Waiting For - Nintendo Life
Source: Nintendo Life
Published: null
Description: Not long to go!...
---
Title: Valve is making microSD cards the next game cartridges - The Verge
Source: The Verge
Published: null
Description: Valve’s new Steam Machine PC and Steam Frame VR headset both have a microSD card slot, meaning you c...
---
Title: How Iron Man’s Jarvis Became the Symbol of Corporate America’s AI Ambitions - The Wall Street Journal
Source: The Wall Street Journal
Published: null
```

Summary

- Портируются не все данные
- Лучше всего и стабильнее с pure Java решениями
- Kotlin – не все

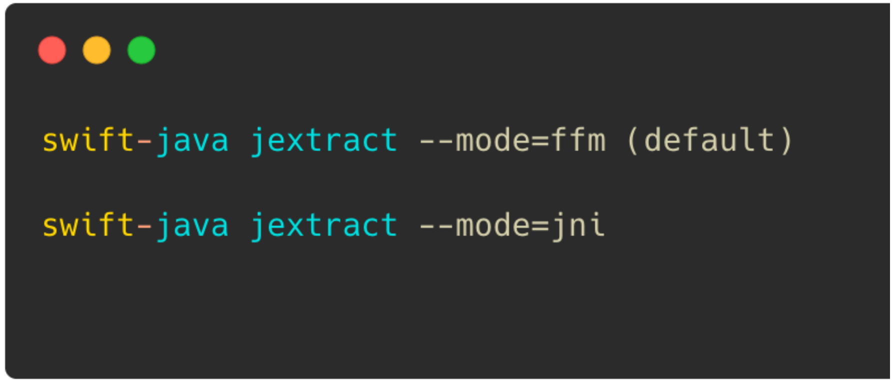
Теперь Swift -> Java



Swift -> Java

Jextract – Java код из Swift

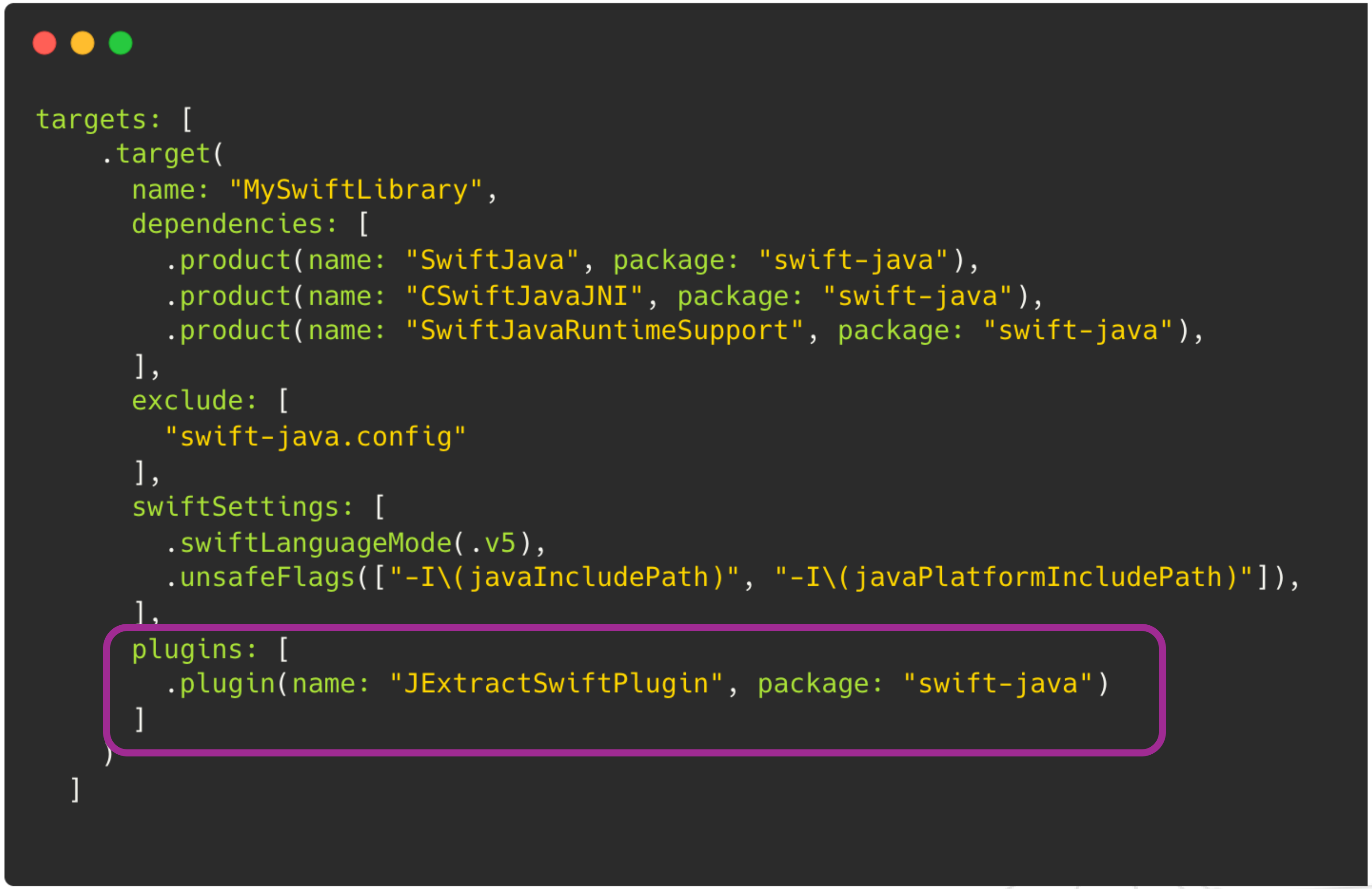
- ffm (Swift 6.2+, JDK 25)
- jni (Swift 6.2+, JDK 7+)



```
swift-java jextract --mode=ffm (default)

swift-java jextract --mode=jni
```

Swift -> Java

Подготовка проекта



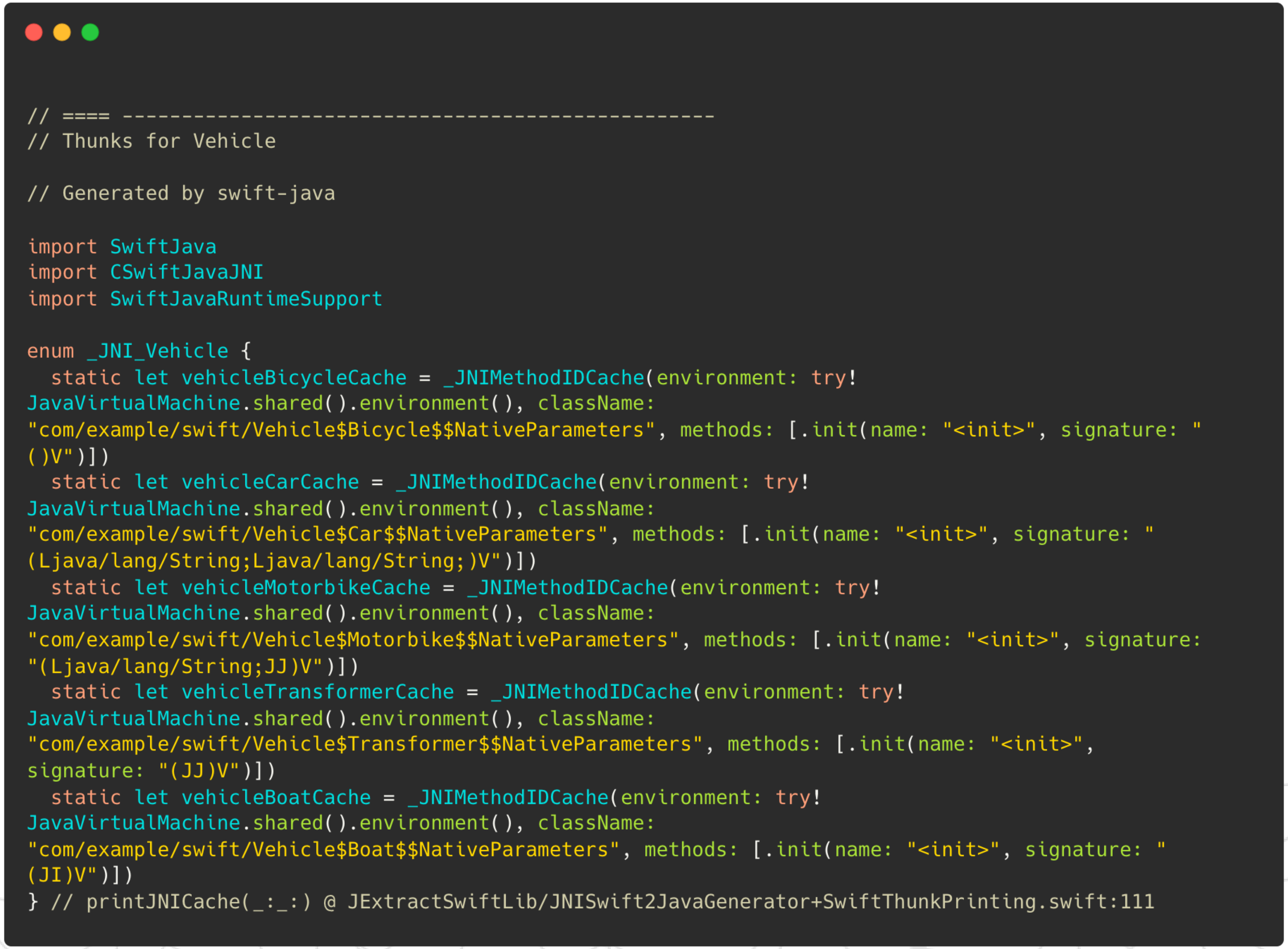
```
targets: [
    .target(
      name: "MySwiftLibrary",
      dependencies: [
        .product(name: "SwiftJava", package: "swift-java"),
        .product(name: "CSwiftJavaJNI", package: "swift-java"),
        .product(name: "SwiftJavaRuntimeSupport", package: "swift-java"),
      ],
      exclude: [
        "swift-java.config"
      ],
      swiftSettings: [
        .swiftLanguageMode(.v5),
        .unsafeFlags(["-I\(javaIncludePath)", "-I\(javaPlatformIncludePath)"]),
      ],
      plugins: [
        .plugin(name: "JExtractSwiftPlugin", package: "swift-java")
      ]
    )
  ]
```

Swift -> Java.Результат генерации

<name>+swiftjava.swift

> 360 строк

Биндинги



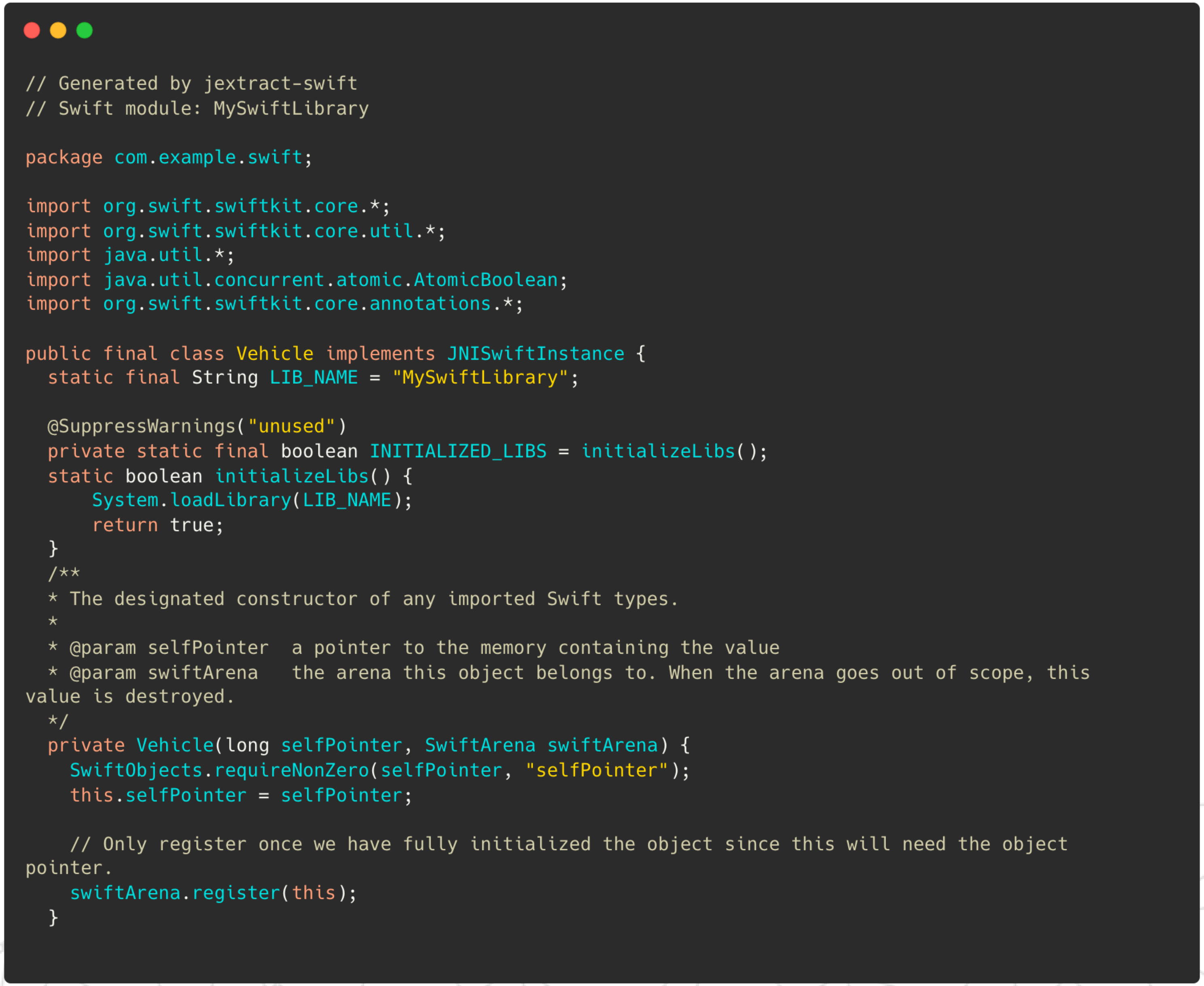
```
// ==== ---------------------------------------------------
// Thunks for Vehicle

// Generated by swift-java

import SwiftJava
import CSwiftJavaJNI
import SwiftJavaRuntimeSupport

enum _JNI_Vehicle {
  static let vehicleBicycleCache = _JNIMethodIDCache(environment: try!
JavaVirtualMachine.shared().environment(), className:
"com/example/swift/Vehicle$Bicycle$$NativeParameters", methods: [.init(name: "<init>", signature: "
()V")])
  static let vehicleCarCache = _JNIMethodIDCache(environment: try!
JavaVirtualMachine.shared().environment(), className:
"com/example/swift/Vehicle$Car$$NativeParameters", methods: [.init(name: "<init>", signature: "
(Ljava/lang/String;Ljava/lang/String;)V")])
  static let vehicleMotorbikeCache = _JNIMethodIDCache(environment: try!
JavaVirtualMachine.shared().environment(), className:
"com/example/swift/Vehicle$Motorbike$$NativeParameters", methods: [.init(name: "<init>", signature:
"(Ljava/lang/String;JJ)V")])
  static let vehicleTransformerCache = _JNIMethodIDCache(environment: try!
JavaVirtualMachine.shared().environment(), className:
"com/example/swift/Vehicle$Transformer$$NativeParameters", methods: [.init(name: "<init>",
signature: "(JJ)V")])
  static let vehicleBoatCache = _JNIMethodIDCache(environment: try!
JavaVirtualMachine.shared().environment(), className:
"com/example/swift/Vehicle$Boat$$NativeParameters", methods: [.init(name: "<init>", signature: "
(JI)V")])
} // printJNICache(_:_:) @ JExtractSwiftLib/JNISwift2JavaGenerator+SwiftThunkPrinting.swift:111
```

Swift -> Java.Результат генерации

Много строк

Биндинг Swift -> Swift

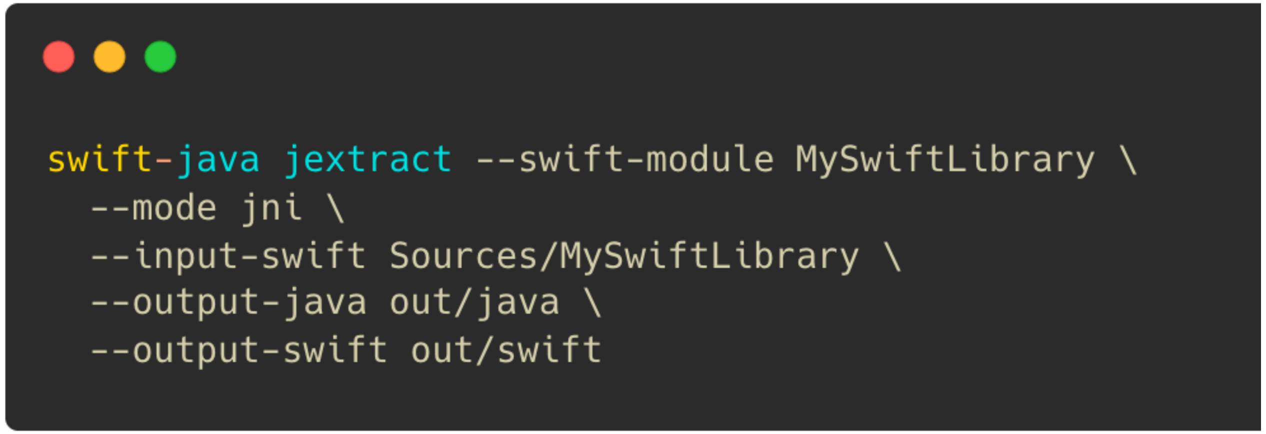
JNISwiftInstance



```java
// Generated by jextract-swift
// Swift module: MySwiftLibrary

package com.example.swift;

import org.swift.swiftkit.core.*;
import org.swift.swiftkit.core.util.*;
import java.util.*;
import java.util.concurrent.atomic.AtomicBoolean;
import org.swift.swiftkit.core.annotations.*;

public final class Vehicle implements JNISwiftInstance {
  static final String LIB_NAME = "MySwiftLibrary";

  @SuppressWarnings("unused")
  private static final boolean INITIALIZED_LIBS = initializeLibs();
  static boolean initializeLibs() {
      System.loadLibrary(LIB_NAME);
      return true;
  }
  /**
  * The designated constructor of any imported Swift types.
  *
  * @param selfPointer  a pointer to the memory containing the value
  * @param swiftArena   the arena this object belongs to. When the arena goes out of scope, this
value is destroyed.
  */
  private Vehicle(long selfPointer, SwiftArena swiftArena) {
    SwiftObjects.requireNonZero(selfPointer, "selfPointer");
    this.selfPointer = selfPointer;

    // Only register once we have fully initialized the object since this will need the object
pointer.
    swiftArena.register(this);
  }
```

Swift -> Java. Запуск вручную

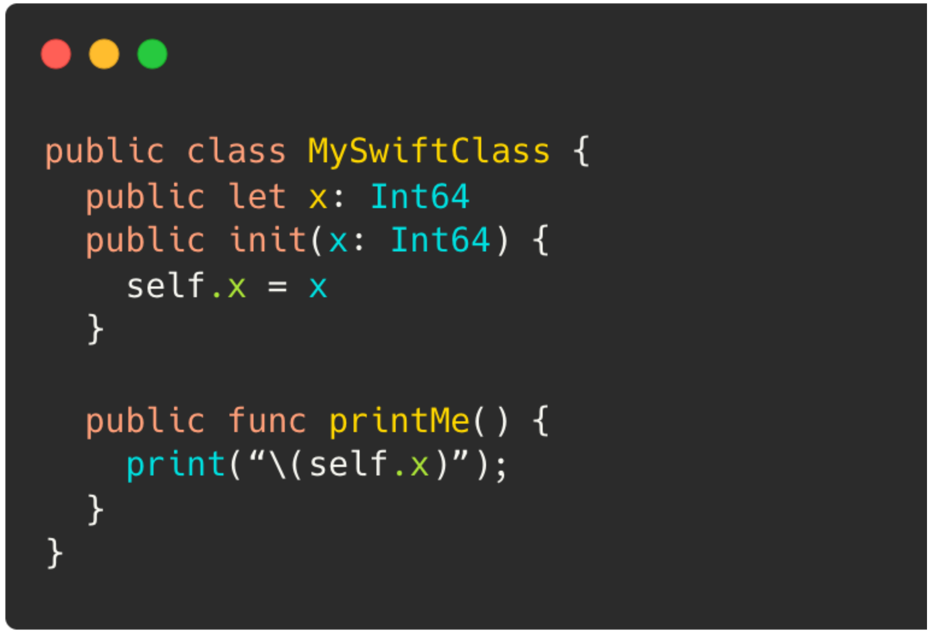
Указываем mode jni, входные файлы и путь для сохранения



```
swift-java jextract --swift-module MySwiftLibrary \
  --mode jni \
  --input-swift Sources/MySwiftLibrary \
  --output-java out/java \
  --output-swift out/swift
```

Swift -> Java. Запуск вручную

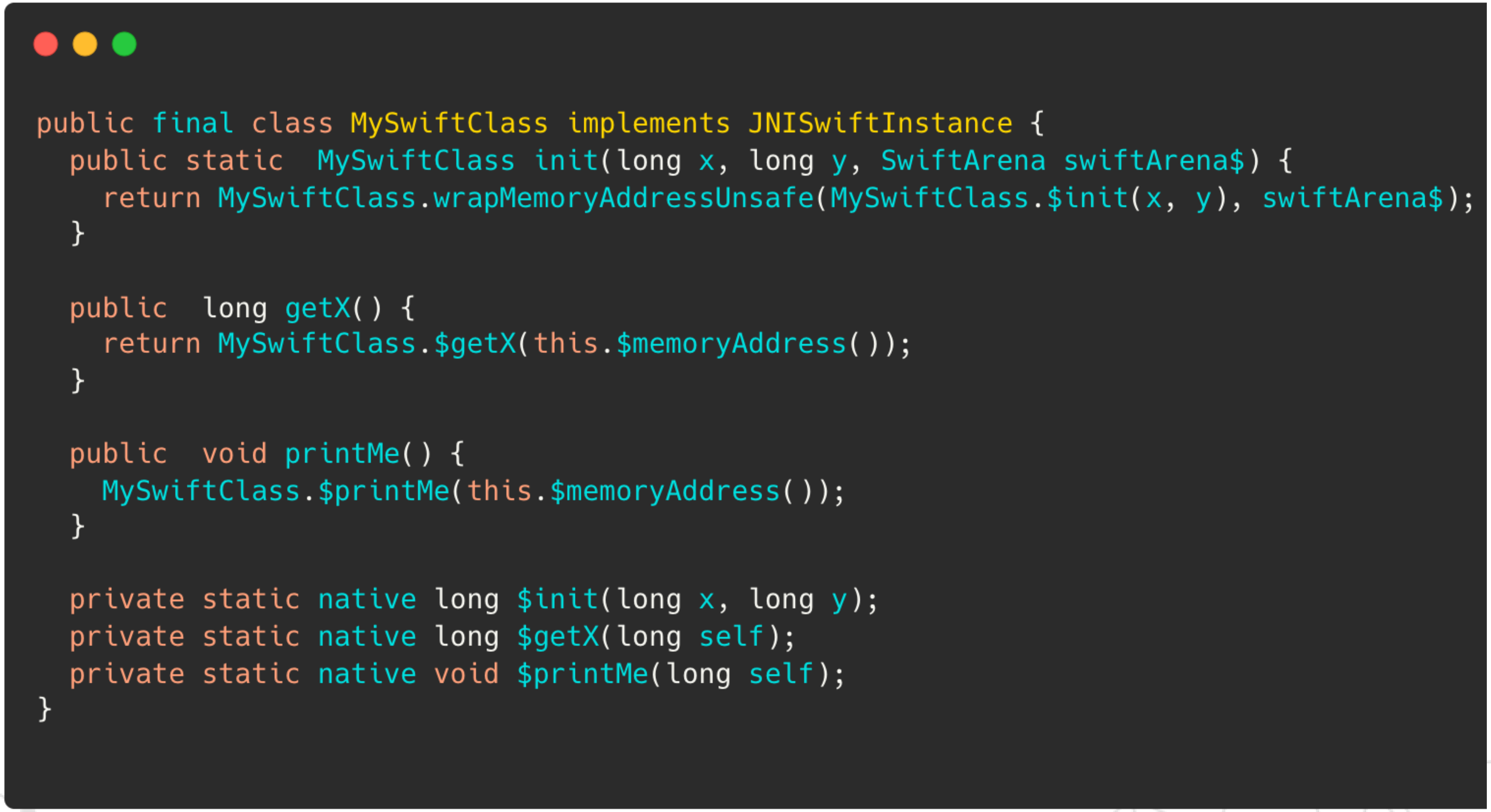
Было



```
public class MySwiftClass {
  public let x: Int64
  public init(x: Int64) {
    self.x = x
  }

  public func printMe() {
    print(“\(self.x)”);
  }
}
```

Swift -> Java. Запуск вручную

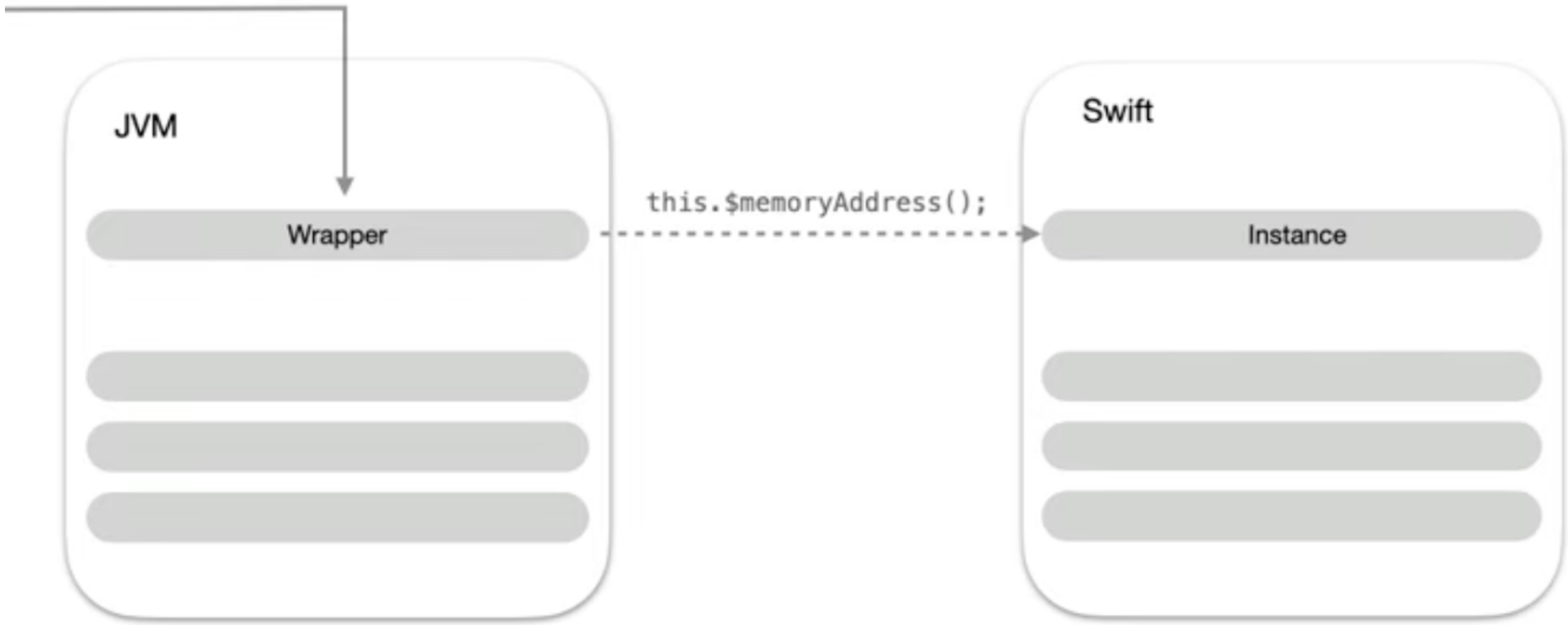
Стало



```
public final class MySwiftClass implements JNISwiftInstance {
  public static  MySwiftClass init(long x, long y, SwiftArena swiftArena$) {
    return MySwiftClass.wrapMemoryAddressUnsafe(MySwiftClass.$init(x, y), swiftArena$);
  }

  public  long getX() {
    return MySwiftClass.$getX(this.$memoryAddress());
  }

  public  void printMe() {
    MySwiftClass.$printMe(this.$memoryAddress());
  }

  private static native long $init(long x, long y);
  private static native long $getX(long self);
  private static native void $printMe(long self);
}
```

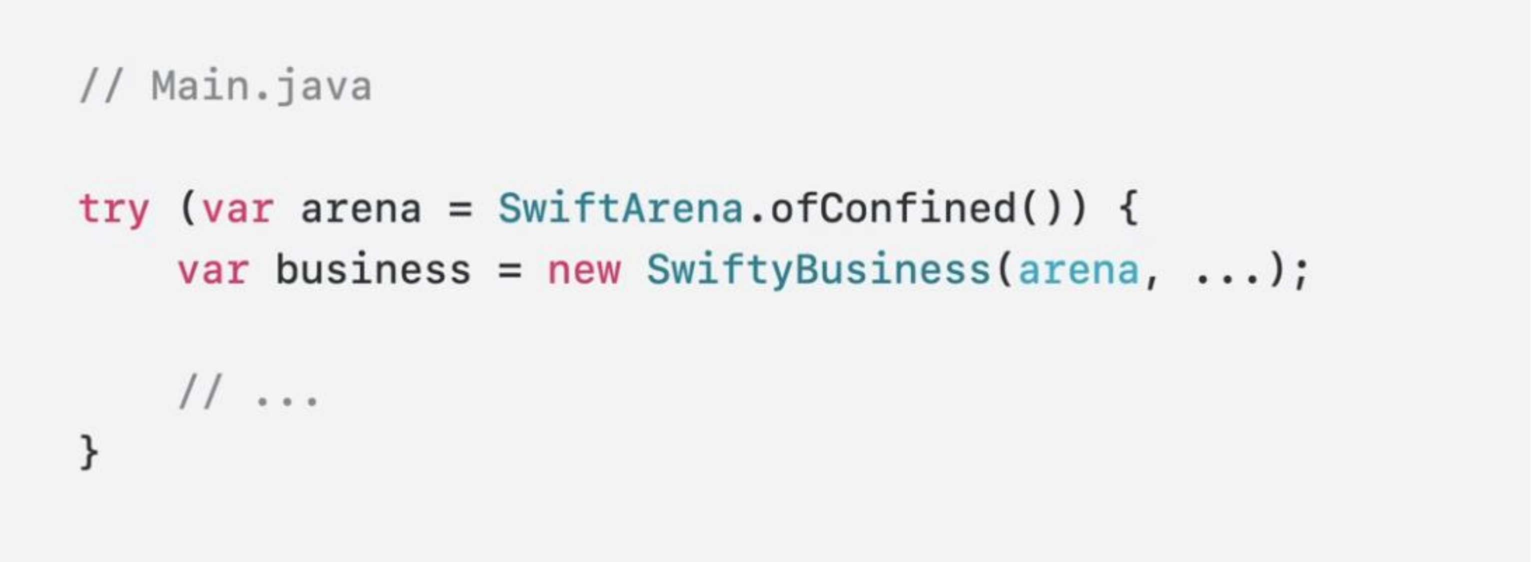
Управление памятью

SwiftArena



~~Quake 3.~~ Swift Arena

SwiftArena



```
// Main.java

try (var arena = SwiftArena.ofConfined()) {
    var business = new SwiftyBusiness(arena, ...);

    // ...
}
```

~~Quake 3.~~ Swift Arena

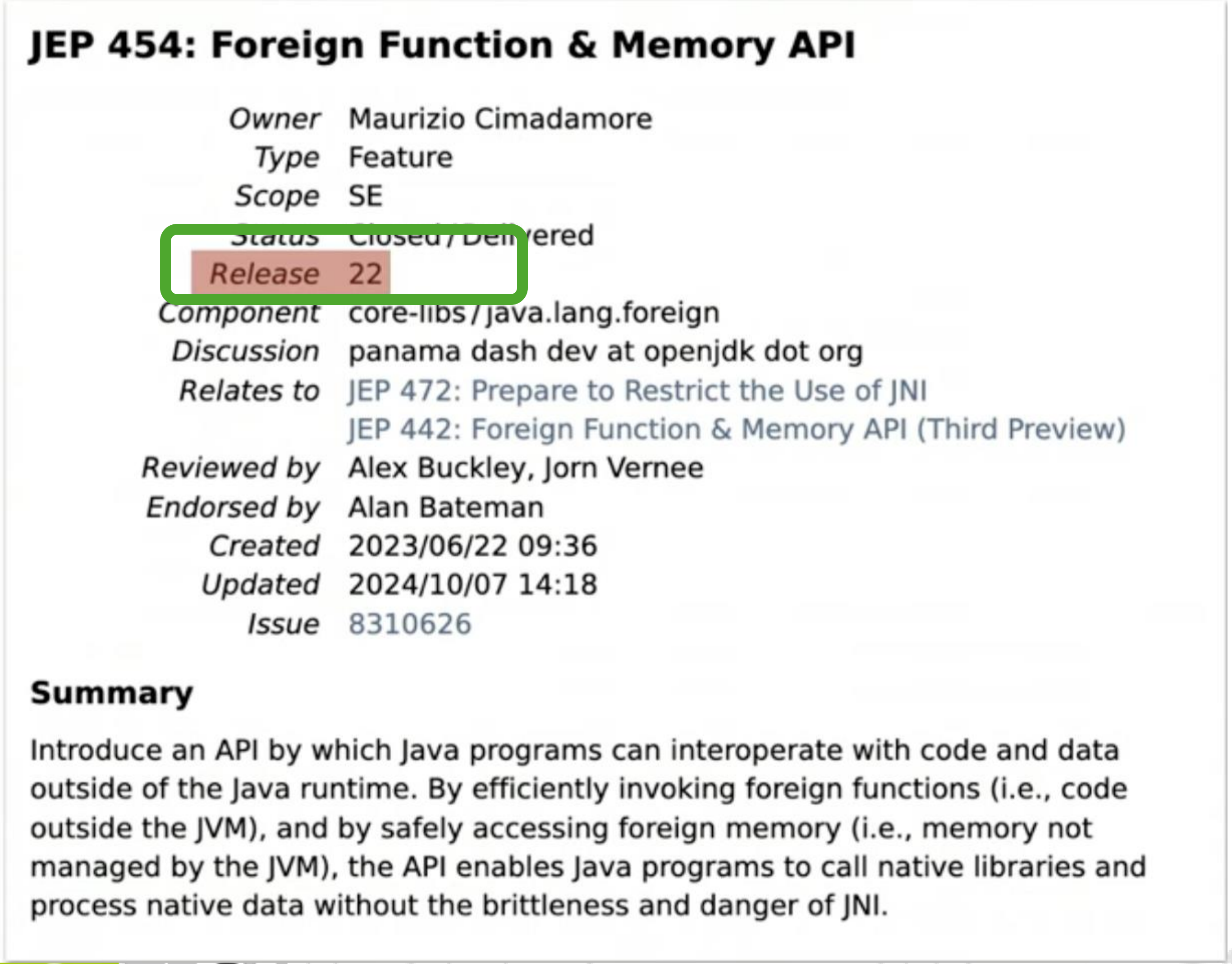
SwiftArena - это тип для гарантированного освобождения памяти от Java-обёртки:

- SwiftArena.ofConfined(): используется с try-with-resource для освобождения всех в scope
- SwiftArena.ofAuto(): освобождает экземпляры со сборщиком мусора

Java Foreign Functions and Memory API

- Улучшенный контроль над нативной памятью
- Более безопасная альтернатива JNI
- Лучшая производительность и безопасность

JNI vs FFM



JEP 454: Foreign Function & Memory API

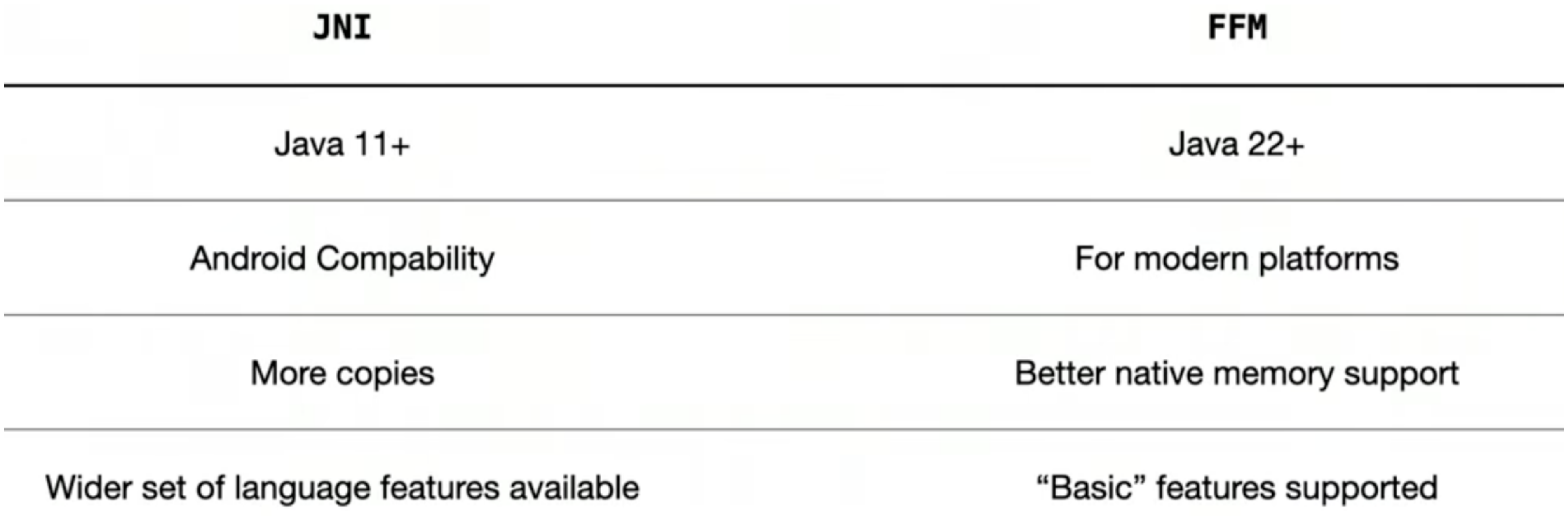
Owner	Maurizio Cimadamore
Type	Feature
Scope	SE
Status	Closed / Delivered
Release	22
Component	core-libs / java.lang.foreign
Discussion	panama dash dev at openjdk dot org
Relates to	JEP 472: Prepare to Restrict the Use of JNI JEP 442: Foreign Function & Memory API (Third Preview)
Reviewed by	Alex Buckley, Jorn Vernee
Endorsed by	Alan Bateman
Created	2023/06/22 09:36
Updated	2024/10/07 14:18
Issue	8310626

Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI.

FFM с Android не работает

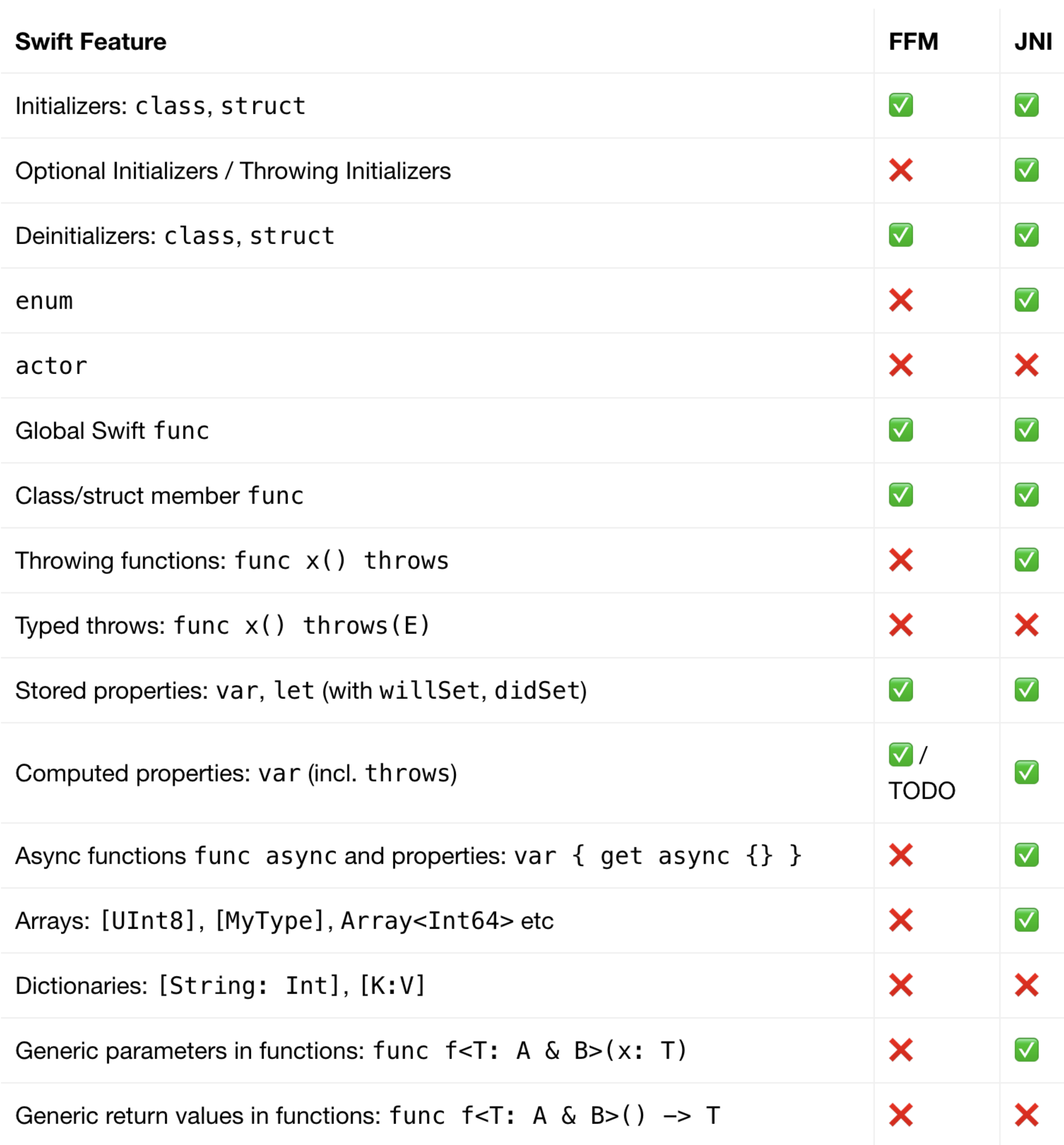
JNI vs FFM



JNI	FFM
Java 11+	Java 22+
Android Compability	For modern platforms
More copies	Better native memory support
Wider set of language features available	“Basic” features supported

JExtract. Ограничения

Jextract
https://swiftpackageindex.com/swiftlang/swift-java/main/documentation/swiftjavadocumentation/supportedfeatures



Swift Feature	FFM	JNI
Initializers: `class`, `struct`	✅	✅
Optional Initializers / Throwing Initializers	❌	✅
Deinitializers: `class`, `struct`	✅	✅
`enum`	❌	✅
`actor`	❌	❌
Global Swift `func`	✅	✅
Class/struct member `func`	✅	✅
Throwing functions: `func x() throws`	❌	✅
Typed throws: `func x() throws(E)`	❌	❌
Stored properties: `var`, `let` (with `willSet`, `didSet`)	✅	✅
Computed properties: `var` (incl. `throws`)	✅ / TODO	✅
Async functions `func async` and properties: `var { get async {} }`	❌	✅
Arrays: `[UInt8]`, `[MyType]`, `Array<Int64>` etc	❌	✅
Dictionaries: `[String: Int]`, `[K:V]`	❌	❌
Generic parameters in functions: `func f<T: A & B>(x: T)`	❌	✅
Generic return values in functions: `func f<T: A & B>() -> T`	❌	❌

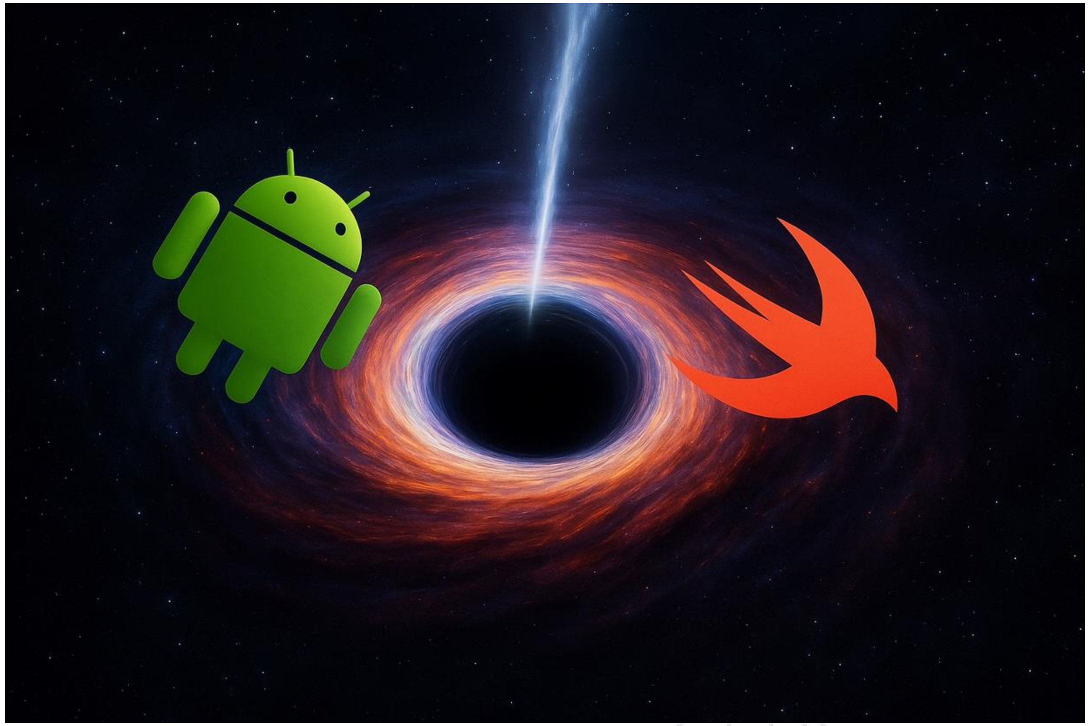
JExtract. С чем проблемы

- Коллекции
- Исключения
- Enum
- Actor
- Generics
- Async/ await (поддерживаются только для JNI)

Roadmap

- Распределение и упаковка
- Транзитивные зависимости
- Реализации протоколов на Java
- Многопоточность в Swift

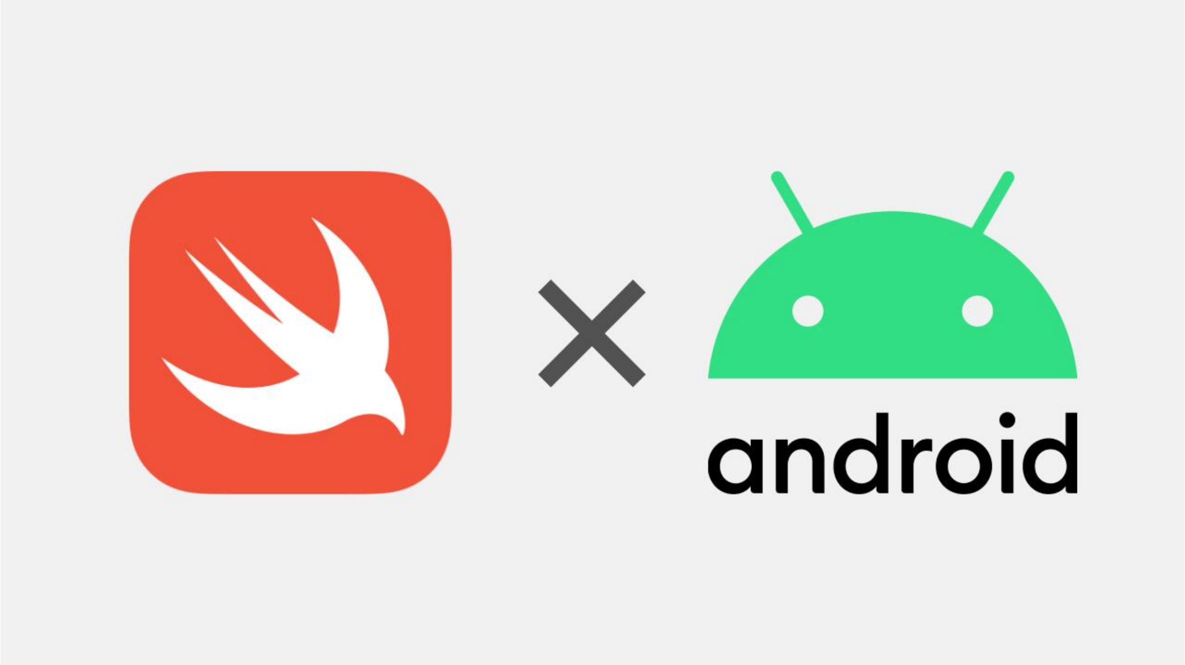
Swift-> Java для Android



Swift Android SDK

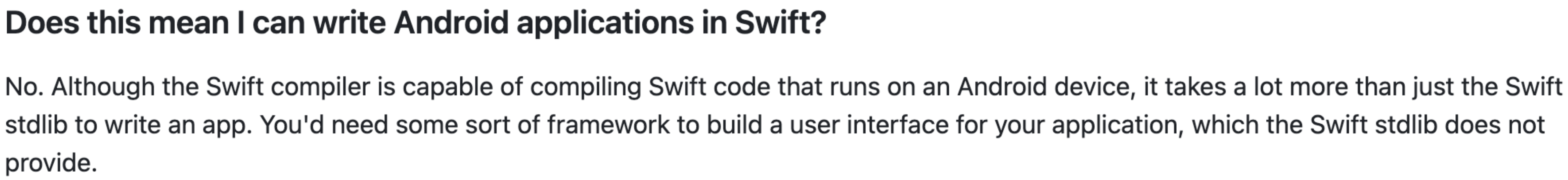
Октябрь 2025

https://www.swift.org/documentation/articles/swift-sdk-for-android-getting-started.html



Swift Android SDK

Написать все приложение Android на Swift пока не получится

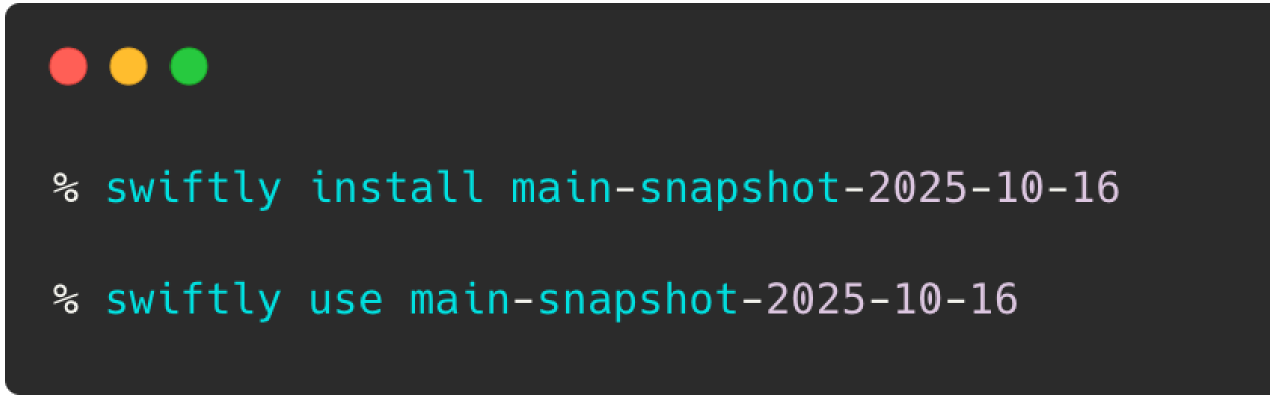


Does this mean I can write Android applications in Swift?

No. Although the Swift compiler is capable of compiling Swift code that runs on an Android device, it takes a lot more than just the Swift stdlib to write an app. You'd need some sort of framework to build a user interface for your application, which the Swift stdlib does not provide.

Что понадобится

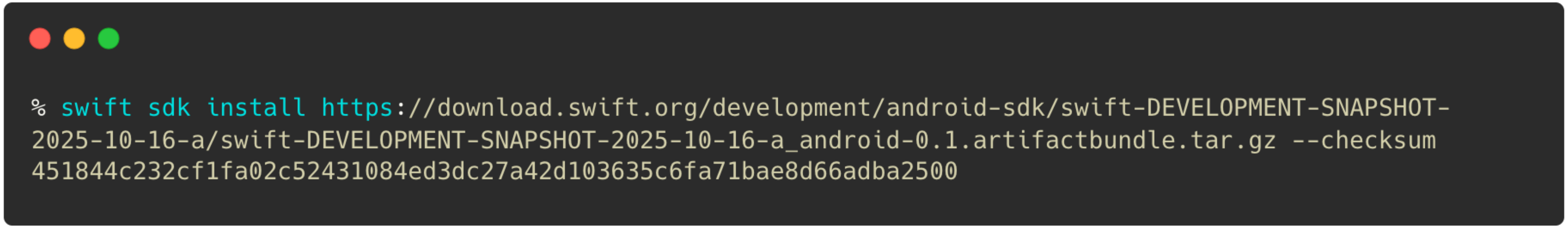
- swiftly
- Swift Toolchain
- Android Swift SDK
- Android NDK



```
% swiftly install main-snapshot-2025-10-16

% swiftly use main-snapshot-2025-10-16
```

Что понадобится

- Swift Toolchain
- Android Swift SDK
- Android NDK

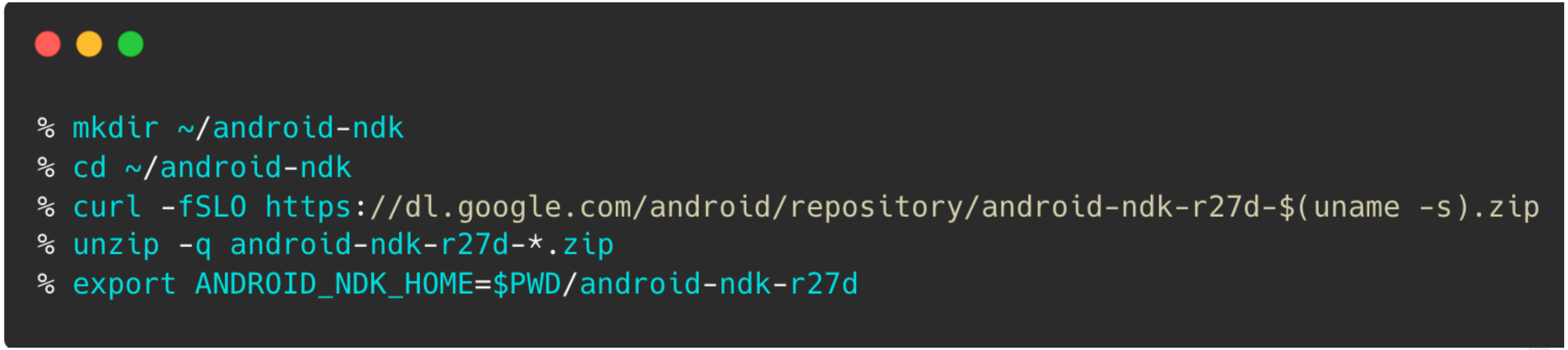


```
% swift sdk install https://download.swift.org/development/android-sdk/swift-DEVELOPMENT-SNAPSHOT-
2025-10-16-a/swift-DEVELOPMENT-SNAPSHOT-2025-10-16-a_android-0.1.artifactbundle.tar.gz --checksum
451844c232cf1fa02c52431084ed3dc27a42d103635c6fa71bae8d66adba2500
```

Что понадобится

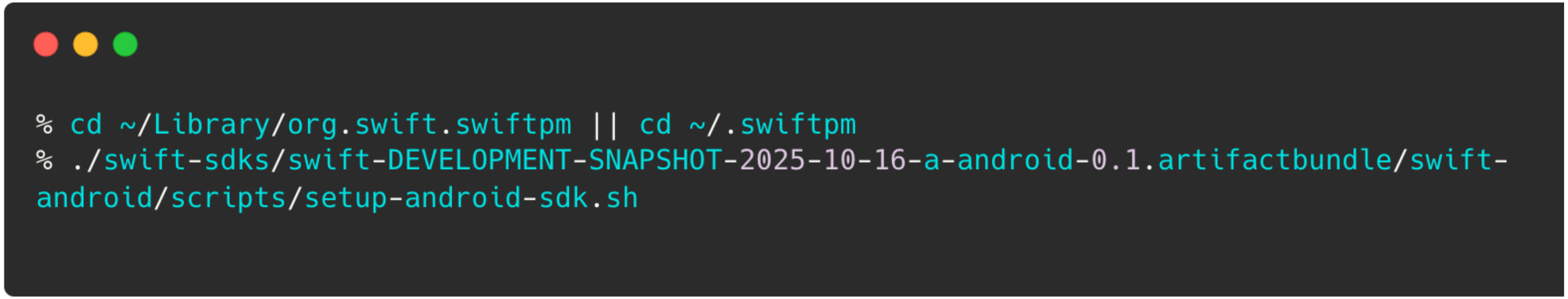
- Swift Toolchain
- Android Swift SDK
- Android NDK

Если NDK у вас уже есть, просто установите ANDROID_NDK_HOME



```
% mkdir ~/android-ndk
% cd ~/android-ndk
% curl -fSLO https://dl.google.com/android/repository/android-ndk-r27d-$(uname -s).zip
% unzip -q android-ndk-r27d-*.zip
% export ANDROID_NDK_HOME=$PWD/android-ndk-r27d
```

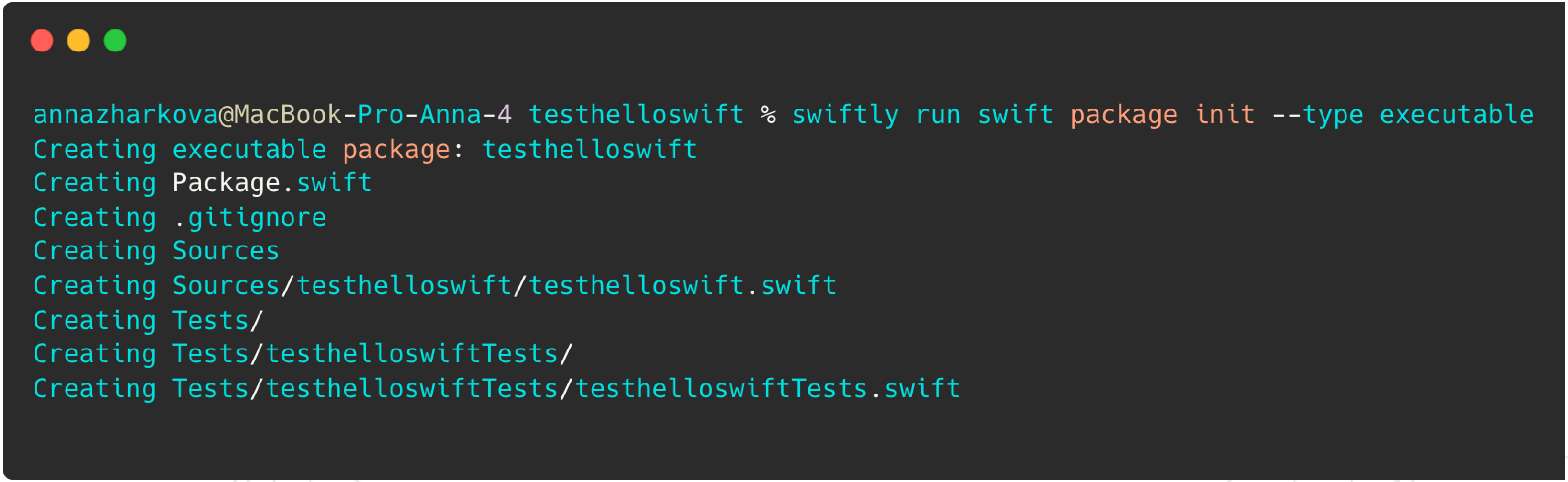
Важно! Линкуем NDK с SwiftPM



```
% cd ~/Library/org.swift.swiftpm || cd ~/.swiftpm
% ./swift-sdks/swift-DEVELOPMENT-SNAPSHOT-2025-10-16-a-android-0.1.artifactbundle/swift-android/scripts/setup-android-sdk.sh
```

Тестовый пример на Swift Android

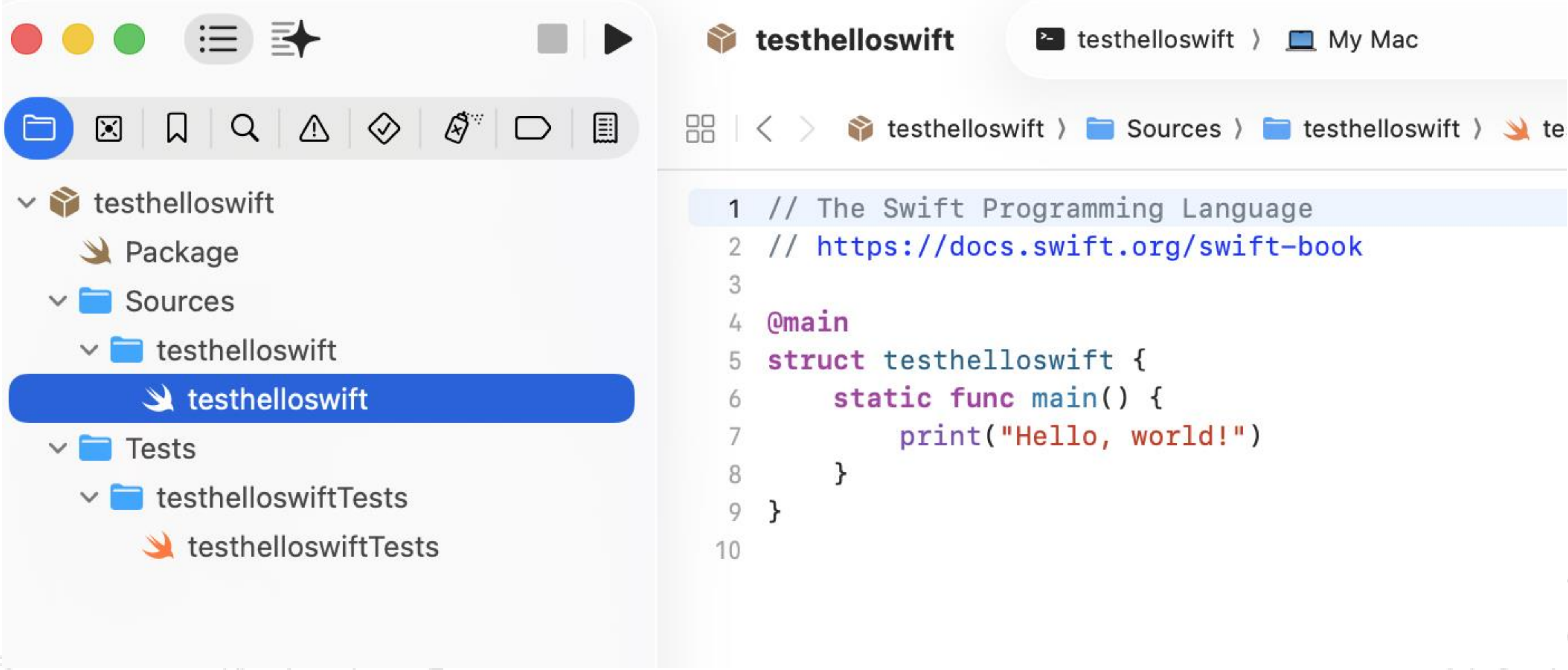
Создаем пакет с помощью swiftly



```
annazharkova@MacBook-Pro-Anna-4 testhelloswift % swiftly run swift package init --type executable
Creating executable package: testhelloswift
Creating Package.swift
Creating .gitignore
Creating Sources
Creating Sources/testhelloswift/testhelloswift.swift
Creating Tests/
Creating Tests/testhelloswiftTests/
Creating Tests/testhelloswiftTests/testhelloswiftTests.swift
```

Тестовый пример на Swift Android

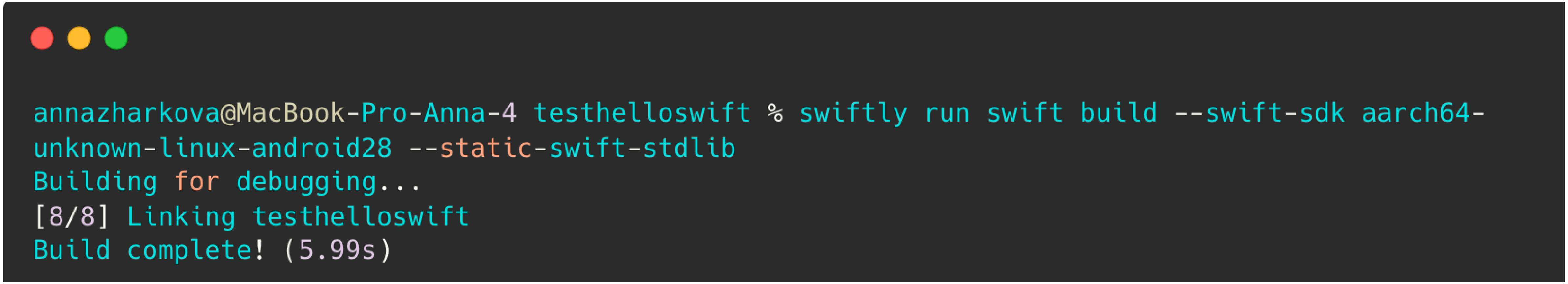
Создаем пакет с помощью swiftly



```swift
// The Swift Programming Language
// https://docs.swift.org/swift-book

@main
struct testhelloswift {
    static func main() {
        print("Hello, world!")
    }
}

```

Тестовый пример на Swift Android

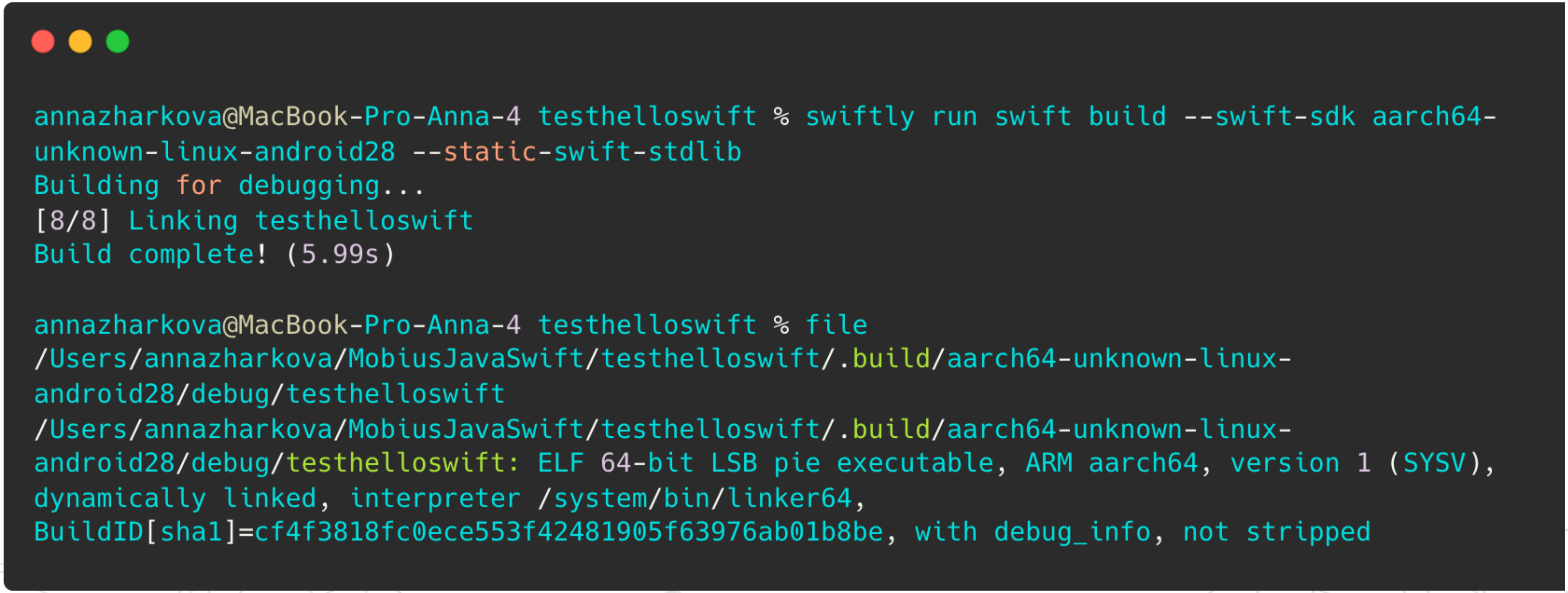
Вызываем сборку под нужную архитектуру aarch64



```
annazharkova@MacBook-Pro-Anna-4 testhelloswift % swiftly run swift build --swift-sdk aarch64-
unknown-linux-android28 --static-swift-stdlib
Building for debugging...
[8/8] Linking testhelloswift
Build complete! (5.99s)
```

Тестовый пример на Swift Android

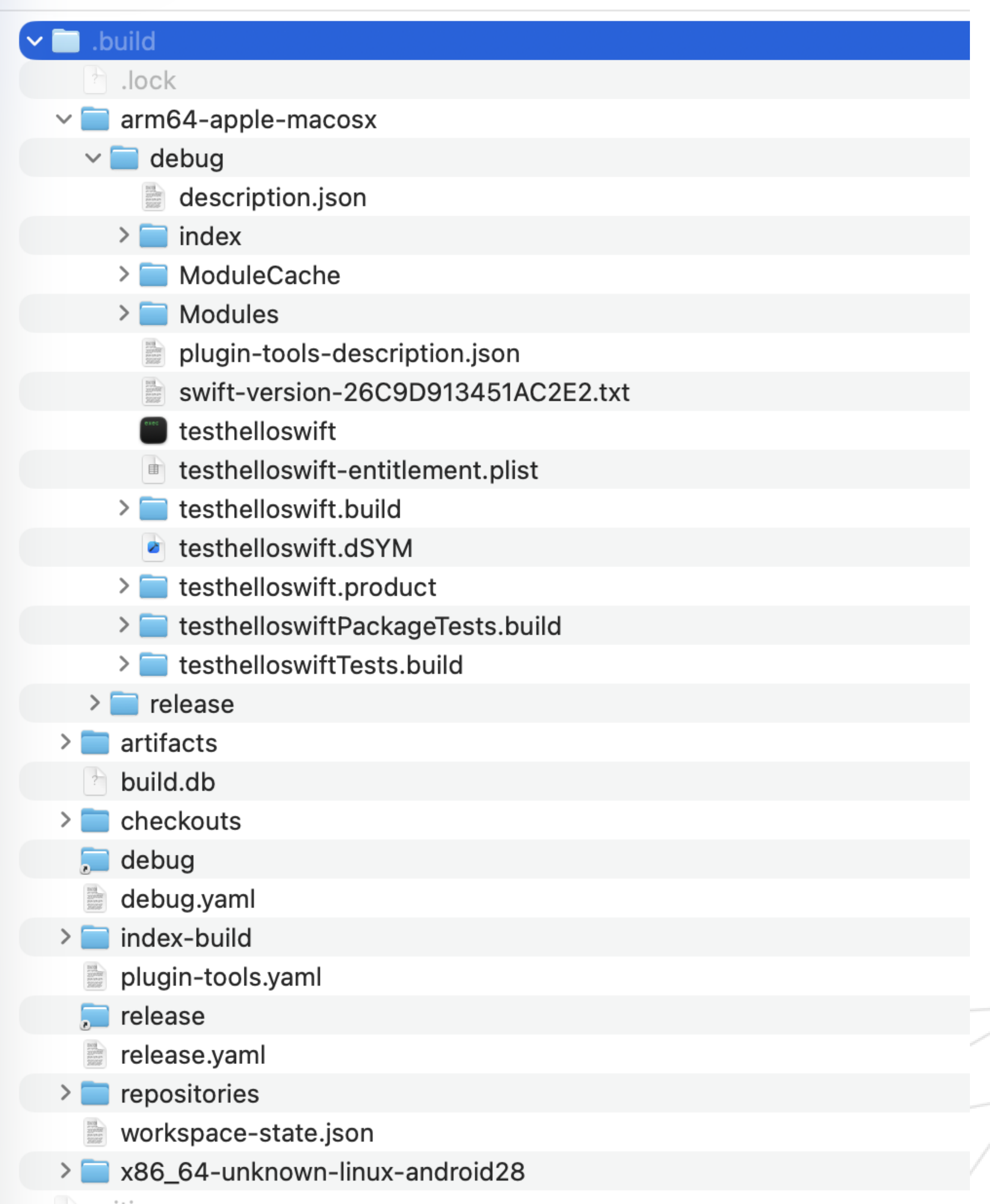
Вызываем сборку под нужную архитектуру aarch64



```
annazharkova@MacBook-Pro-Anna-4 testhelloswift % swiftly run swift build --swift-sdk aarch64-
unknown-linux-android28 --static-swift-stdlib
Building for debugging...
[8/8] Linking testhelloswift
Build complete! (5.99s)

annazharkova@MacBook-Pro-Anna-4 testhelloswift % file
/Users/annazharkova/MobiusJavaSwift/testhelloswift/.build/aarch64-unknown-linux-
android28/debug/testhelloswift
/Users/annazharkova/MobiusJavaSwift/testhelloswift/.build/aarch64-unknown-linux-
android28/debug/testhelloswift: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV),
dynamically linked, interpreter /system/bin/linker64,
BuildID[sha1]=cf4f3818fc0ece553f42481905f63976ab01b8be, with debug_info, not stripped
```

Тестовый пример на Swift Android

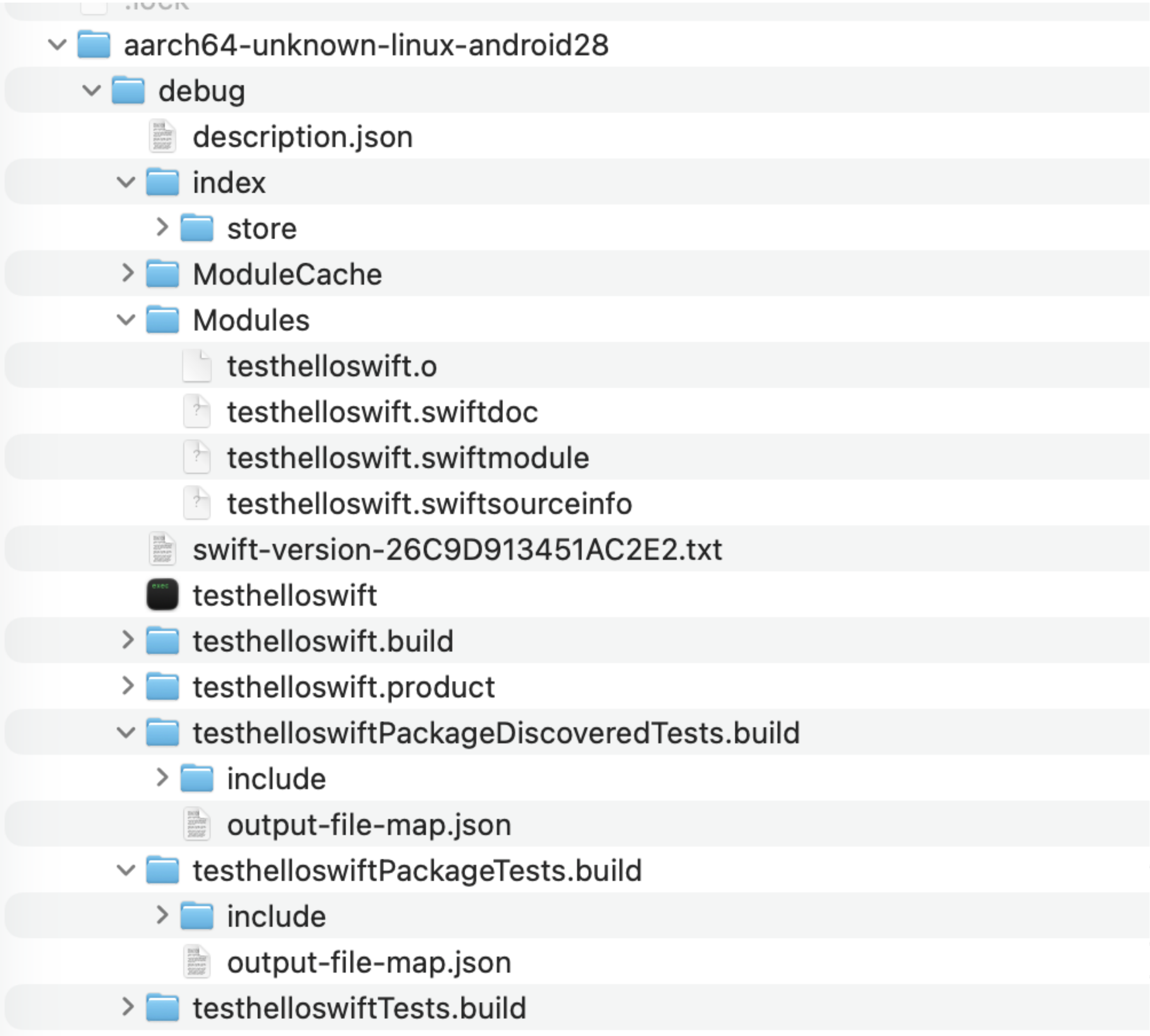
После сборки



```
.build
  .lock
  arm64-apple-macosx
    debug
      description.json
      index
      ModuleCache
      Modules
      plugin-tools-description.json
      swift-version-26C9D913451AC2E2.txt
      testhelloswift
      testhelloswift-entitlement.plist
      testhelloswift.build
      testhelloswift.dSYM
      testhelloswift.product
      testhelloswiftPackageTests.build
      testhelloswiftTests.build
    release
  artifacts
  build.db
  checkouts
  debug
  debug.yaml
  index-build
  plugin-tools.yaml
  release
  release.yaml
  repositories
  workspace-state.json
  x86_64-unknown-linux-android28
```

Тестовый пример на Swift Android

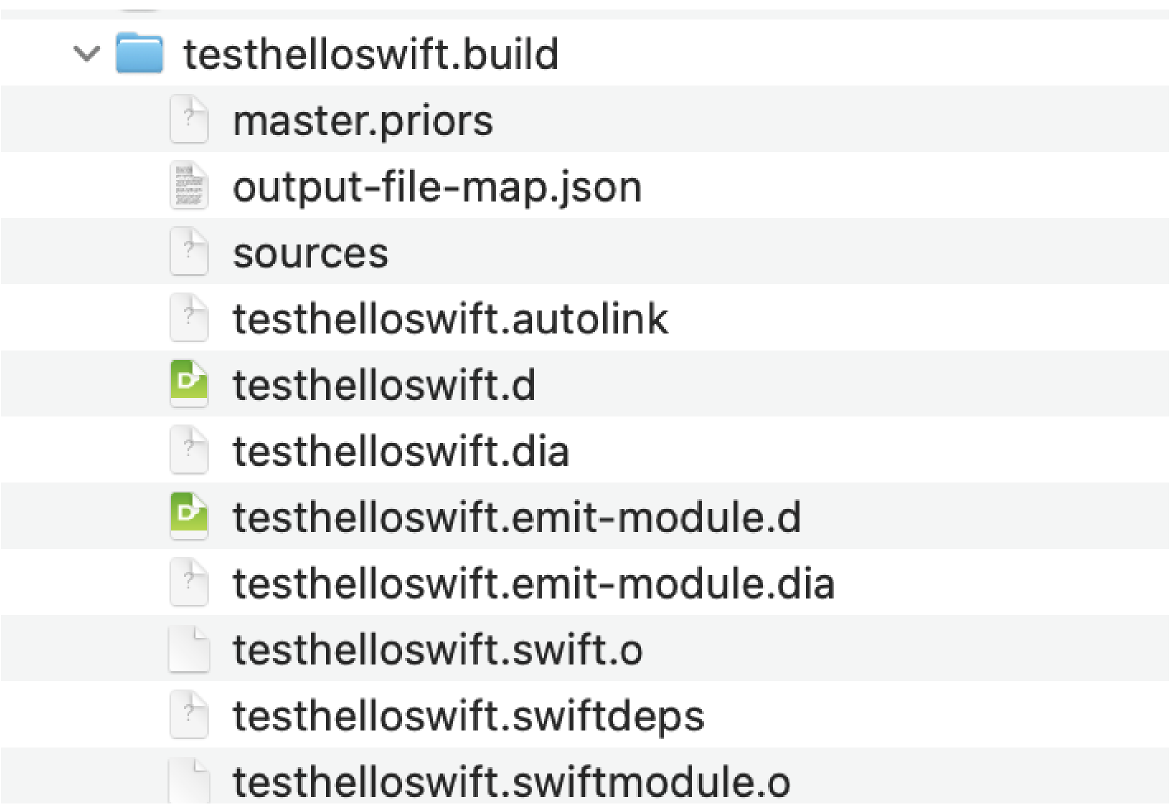
После сборки



```
.lock
aarch64-unknown-linux-android28
  debug
    description.json
    index
      store
    ModuleCache
    Modules
      testhelloswift.o
      testhelloswift.swiftdoc
      testhelloswift.swiftmodule
      testhelloswift.swiftsourceinfo
    swift-version-26C9D913451AC2E2.txt
    testhelloswift
    testhelloswift.build
    testhelloswift.product
    testhelloswiftPackageDiscoveredTests.build
      include
      output-file-map.json
    testhelloswiftPackageTests.build
      include
      output-file-map.json
    testhelloswiftTests.build
```

Тестовый пример на Swift Android

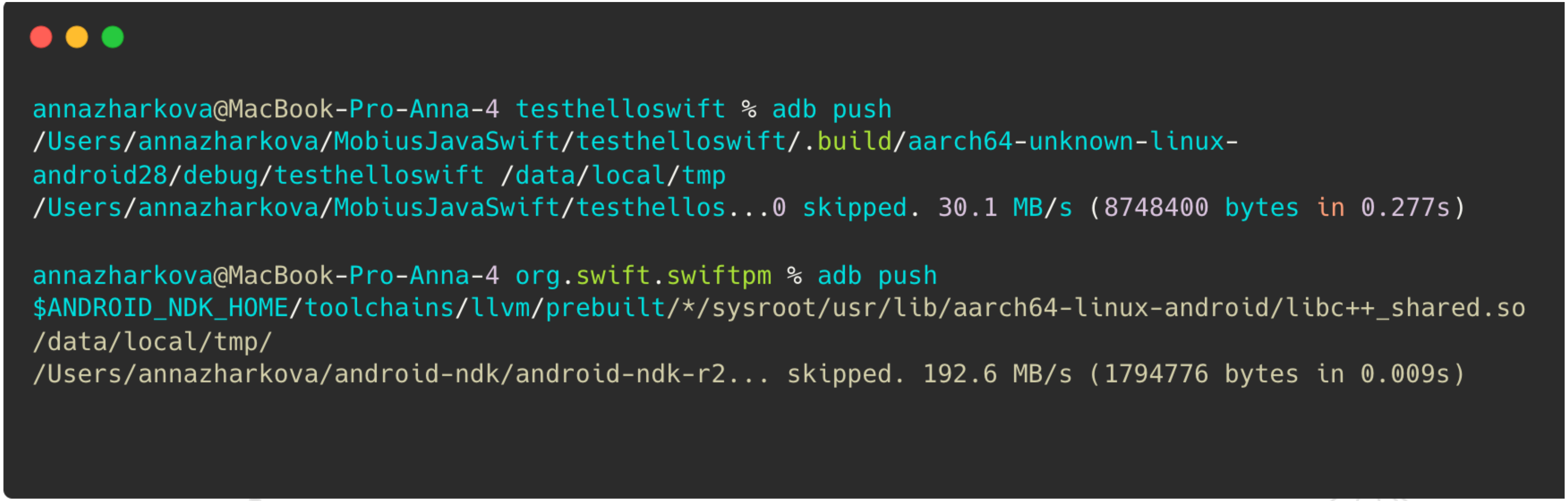
После сборки



```
testhelloswift.build
    master.priors
    output-file-map.json
    sources
    testhelloswift.autolink
    testhelloswift.d
    testhelloswift.dia
    testhelloswift.emit-module.d
    testhelloswift.emit-module.dia
    testhelloswift.swift.o
    testhelloswift.swiftdeps
    testhelloswift.swiftmodule.o
```

Тестовый пример на Swift Android

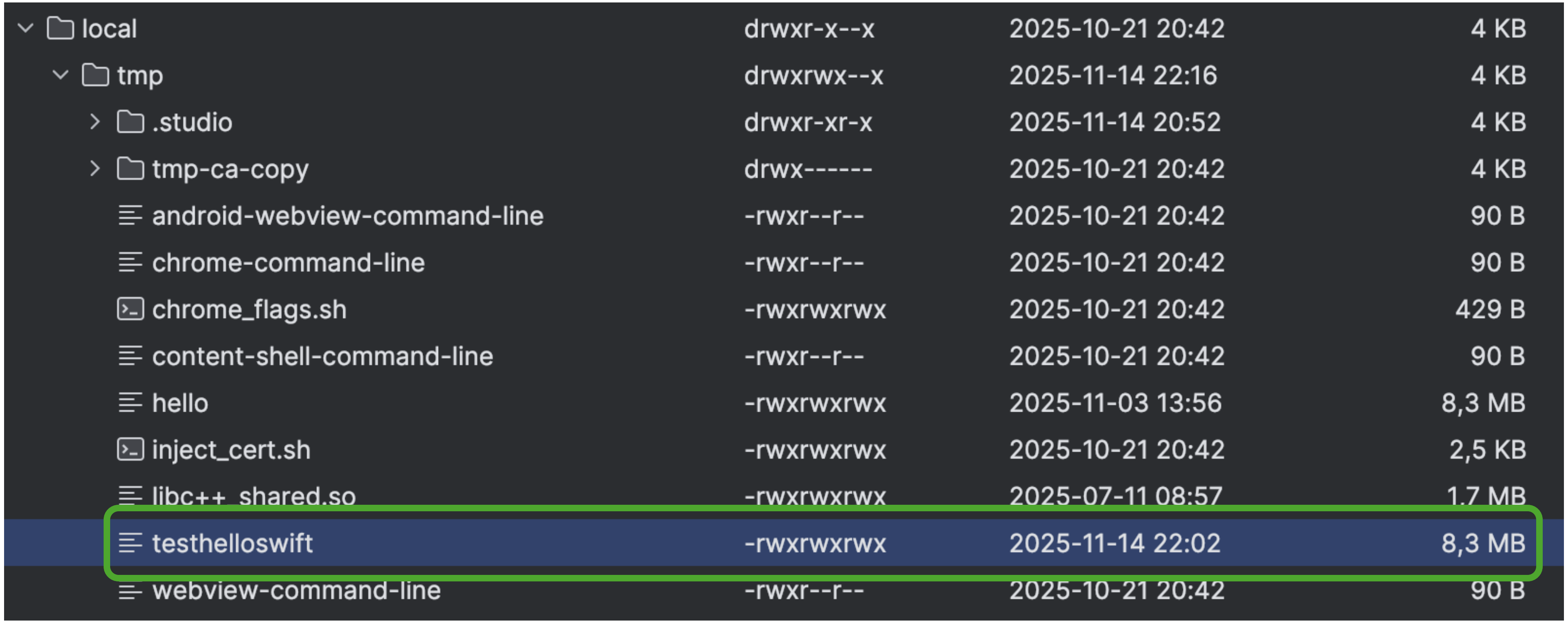
Деплоим



```
annazharkova@MacBook-Pro-Anna-4 testhelloswift % adb push
/Users/annazharkova/MobiusJavaSwift/testhelloswift/.build/aarch64-unknown-linux-
android28/debug/testhelloswift /data/local/tmp
/Users/annazharkova/MobiusJavaSwift/testhellos...0 skipped. 30.1 MB/s (8748400 bytes in 0.277s)

annazharkova@MacBook-Pro-Anna-4 org.swift.swiftpm % adb push
$ANDROID_NDK_HOME/toolchains/llvm/prebuilt/*/sysroot/usr/lib/aarch64-linux-android/libc++_shared.so
/data/local/tmp/
/Users/annazharkova/android-ndk/android-ndk-r2... skipped. 192.6 MB/s (1794776 bytes in 0.009s)
```

Тестовый пример на Swift Android

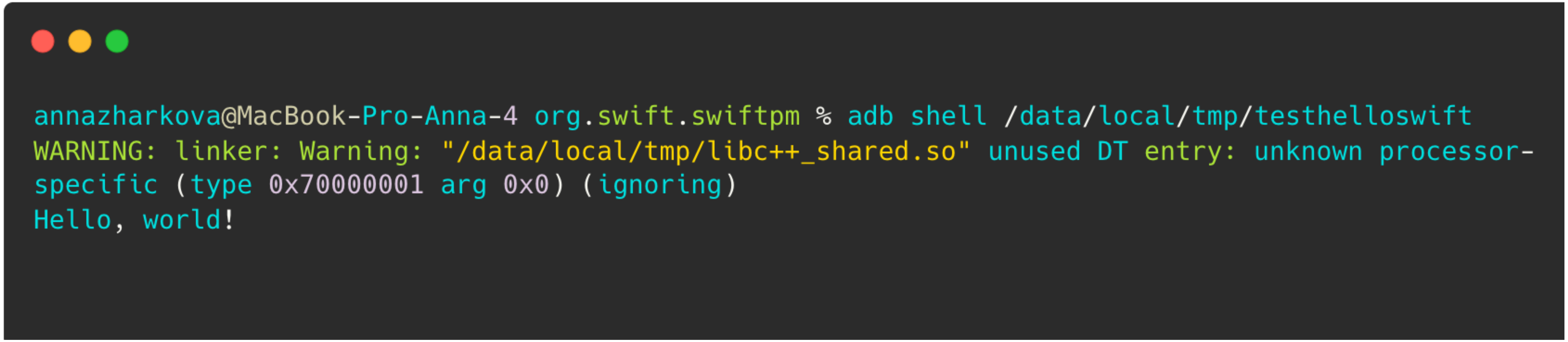
Деплоим



```
v 🗀 local                              drwxr-x--x    2025-10-21 20:42    4 KB
  v 🗀 tmp                              drwxrwx--x    2025-11-14 22:16    4 KB
    > 🗀 .studio                        drwxr-xr-x    2025-11-14 20:52    4 KB
    > 🗀 tmp-ca-copy                    drwx------    2025-10-21 20:42    4 KB
      android-webview-command-line      -rwxr--r--    2025-10-21 20:42    90 B
      chrome-command-line               -rwxr--r--    2025-10-21 20:42    90 B
      chrome_flags.sh                   -rwxrwxrwx    2025-10-21 20:42    429 B
      content-shell-command-line        -rwxr--r--    2025-10-21 20:42    90 B
      hello                             -rwxrwxrwx    2025-11-03 13:56    8,3 MB
      inject_cert.sh                    -rwxrwxrwx    2025-10-21 20:42    2,5 KB
      libc++_shared.so                  -rwxrwxrwx    2025-07-11 08:57    1.7 MB
      testhelloswift                    -rwxrwxrwx    2025-11-14 22:02    8,3 MB
      webview-command-line              -rwxr--r--    2025-10-21 20:42    90 B
```

Тестовый пример на Swift Android

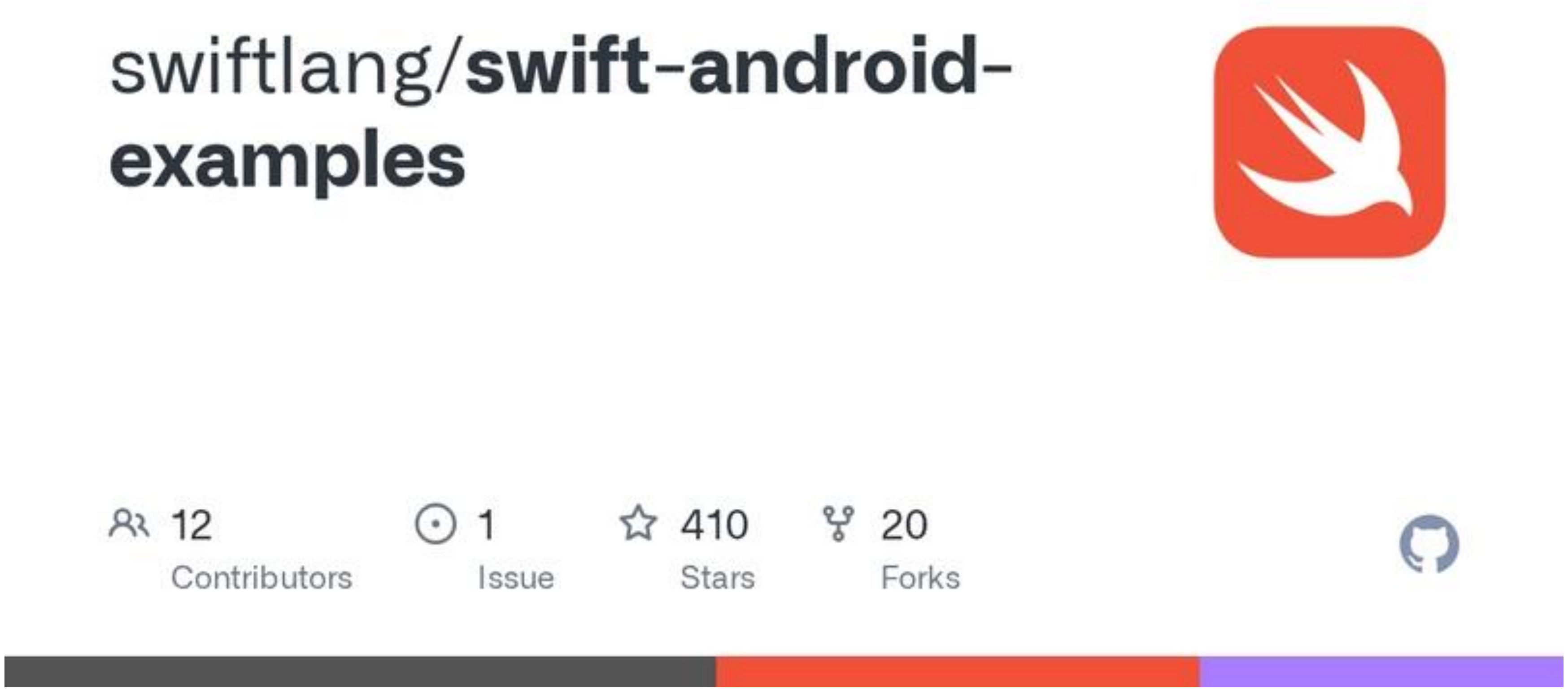
Запускаем



```
annazharkova@MacBook-Pro-Anna-4 org.swift.swiftpm % adb shell /data/local/tmp/testhelloswift
WARNING: linker: Warning: "/data/local/tmp/libc++_shared.so" unused DT entry: unknown processor-
specific (type 0x70000001 arg 0x0) (ignoring)
Hello, world!
```

Расширенные примеры с Activity

Это и документация



swiftlang/**swift-android-examples**

12 Contributors

1 Issue

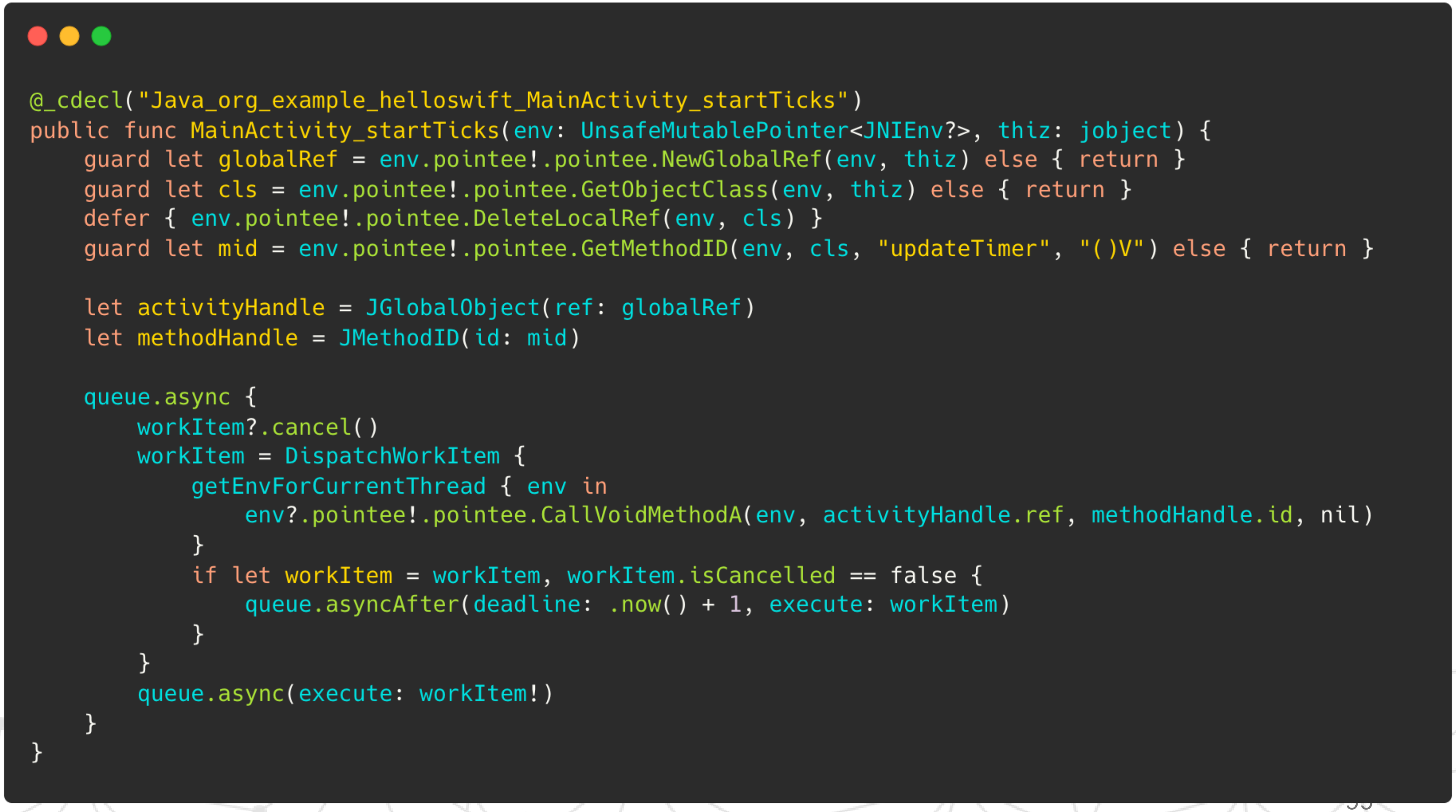
410 Stars

20 Forks

Добавления в существующий Activity

@_cdecl – макрос для новой функции в Activity

Import Android???

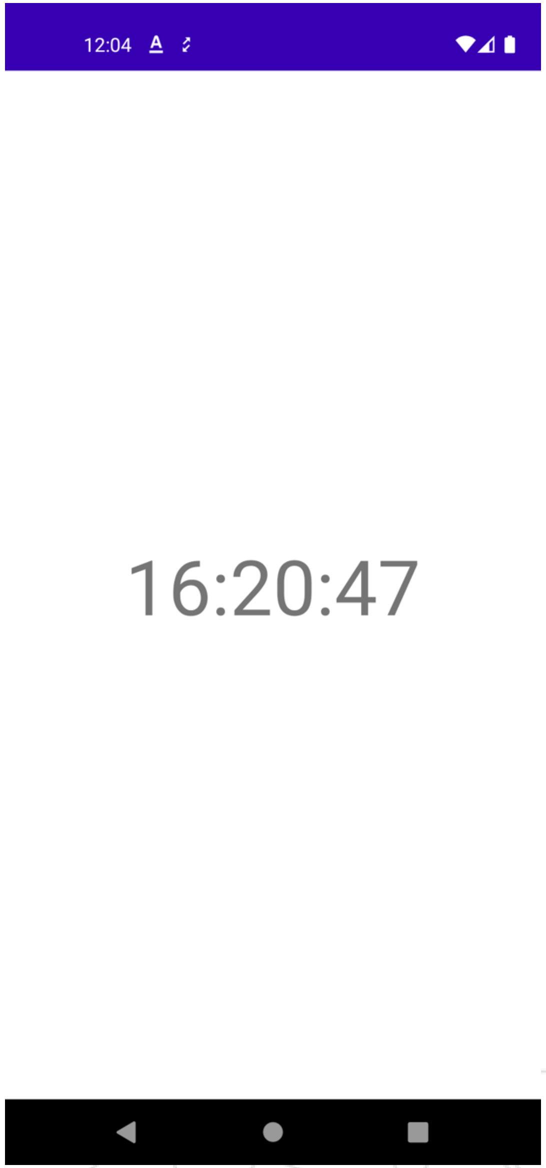


```
@_cdecl("Java_org_example_helloswift_MainActivity_startTicks")
public func MainActivity_startTicks(env: UnsafeMutablePointer<JNIEnv?>, thiz: jobject) {
    guard let globalRef = env.pointee!.pointee.NewGlobalRef(env, thiz) else { return }
    guard let cls = env.pointee!.pointee.GetObjectClass(env, thiz) else { return }
    defer { env.pointee!.pointee.DeleteLocalRef(env, cls) }
    guard let mid = env.pointee!.pointee.GetMethodID(env, cls, "updateTimer", "()V") else { return }

    let activityHandle = JGlobalObject(ref: globalRef)
    let methodHandle = JMethodID(id: mid)

    queue.async {
        workItem?.cancel()
        workItem = DispatchWorkItem {
            getEnvForCurrentThread { env in
                env?.pointee!.pointee.CallVoidMethodA(env, activityHandle.ref, methodHandle.id, nil)
            }
            if let workItem = workItem, workItem.isCancelled == false {
                queue.asyncAfter(deadline: .now() + 1, execute: workItem)
            }
        }
        queue.async(execute: workItem!)
    }
}
```

Добавления в существующий Activity

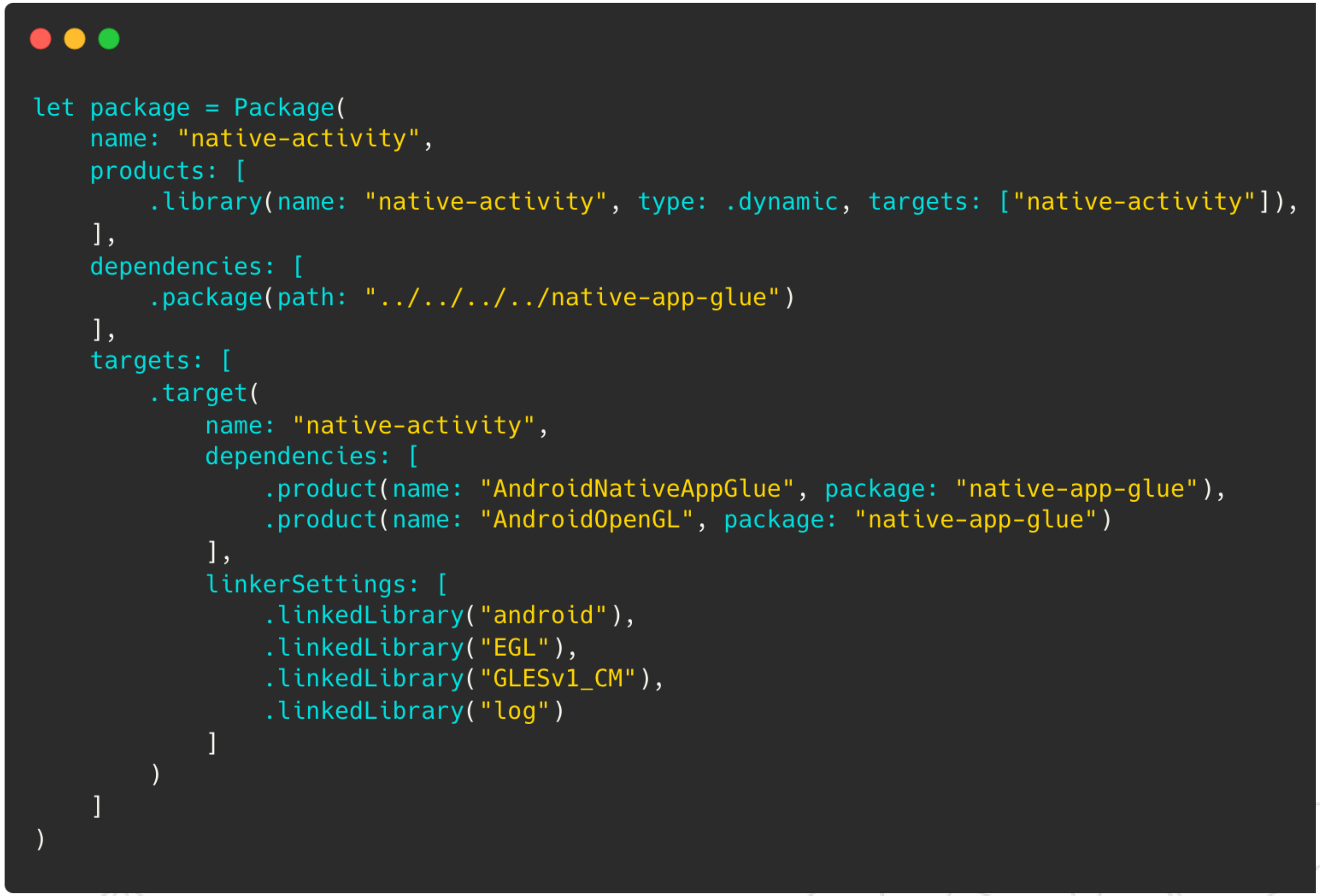
@_cdecl – макрос для новой функции в Activity

Import Android???



NDKActivity

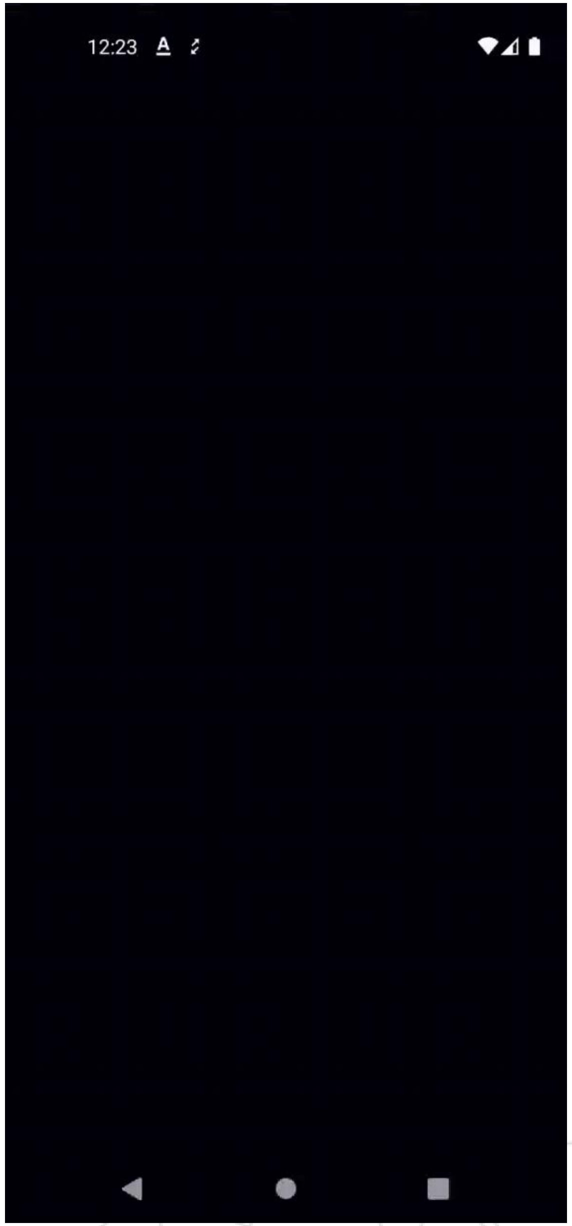
SPM с нативной Activity



```
let package = Package(
    name: "native-activity",
    products: [
        .library(name: "native-activity", type: .dynamic, targets: ["native-activity"]),
    ],
    dependencies: [
        .package(path: "../../../../native-app-glue")
    ],
    targets: [
        .target(
            name: "native-activity",
            dependencies: [
                .product(name: "AndroidNativeAppGlue", package: "native-app-glue"),
                .product(name: "AndroidOpenGL", package: "native-app-glue")
            ],
            linkerSettings: [
                .linkedLibrary("android"),
                .linkedLibrary("EGL"),
                .linkedLibrary("GLESv1_CM"),
                .linkedLibrary("log")
            ]
        )
    ]
)
```

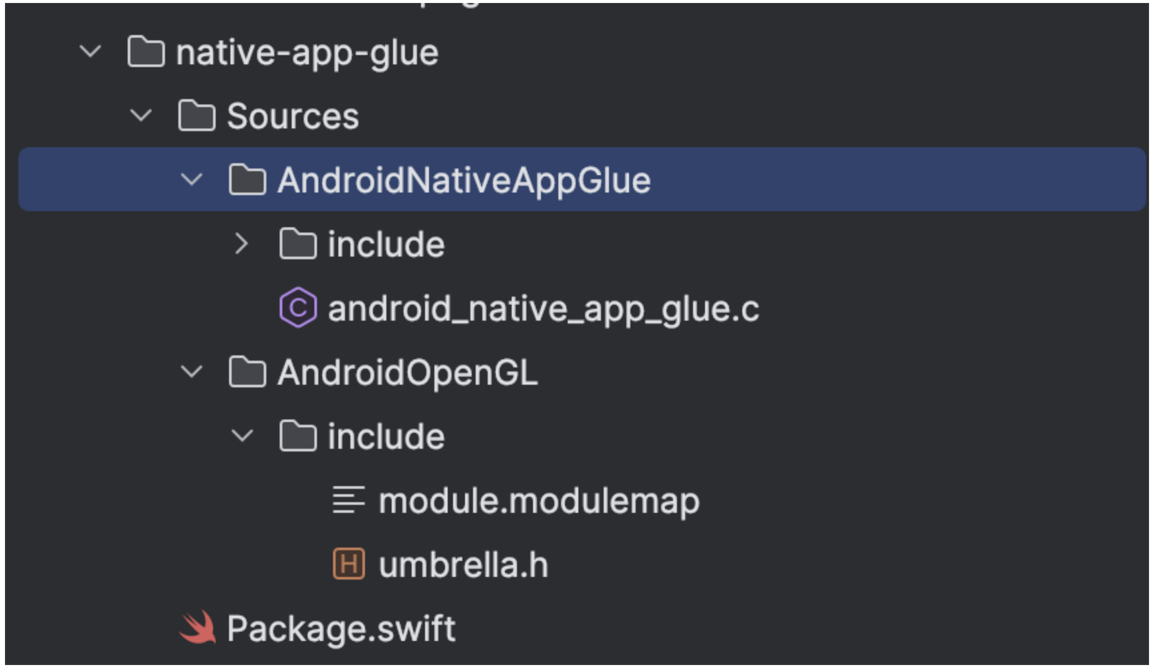
NDKActivity

SPM с нативной Activity



NDKActivity

C++ -> JNI -> Swift -> Android

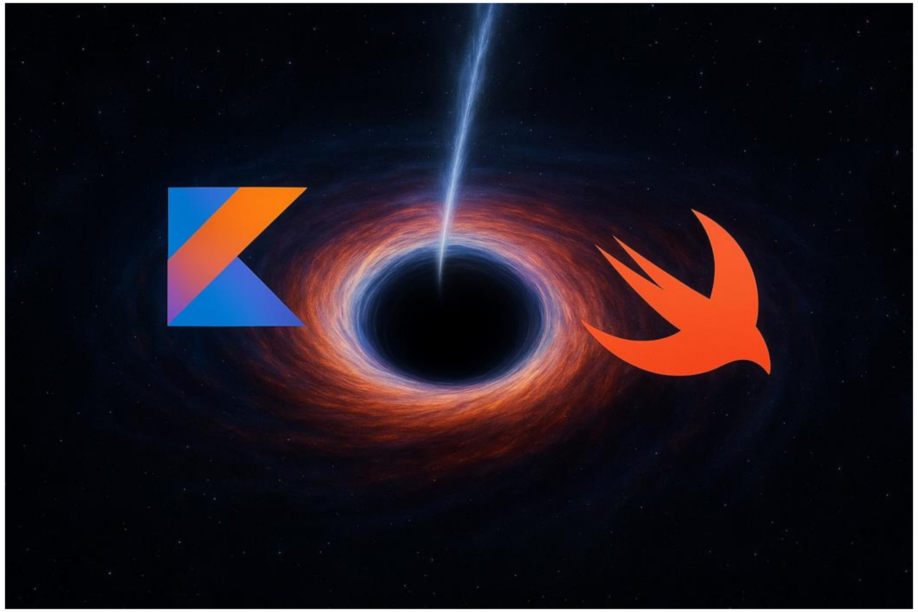


```
native-app-glue
  Sources
    AndroidNativeAppGlue
      include
      android_native_app_glue.c
    AndroidOpenGL
      include
        module.modulemap
        umbrella.h
  Package.swift
```

Что в итоге с задумками от Apple

- Swift -> Java работает лучше, чем Java -> Swift
- Документация -> 0
- Слишком много ручного труда

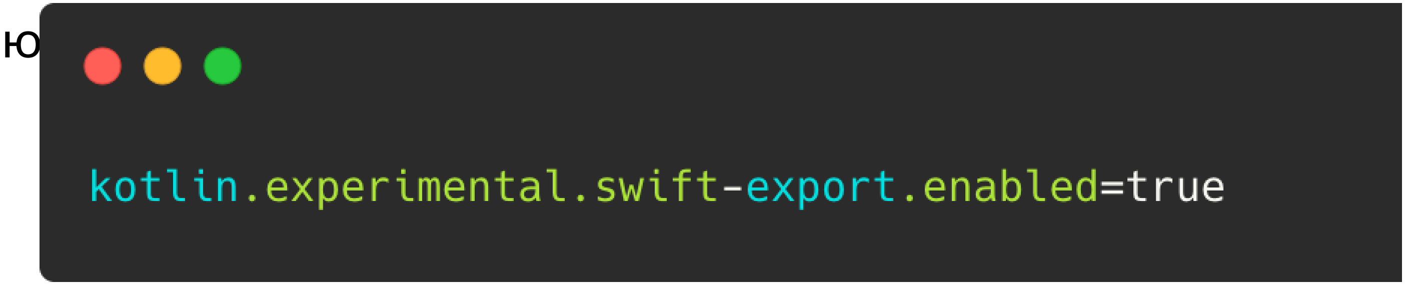
А что с прямым экспортом Kotlin – Swift в KMP?



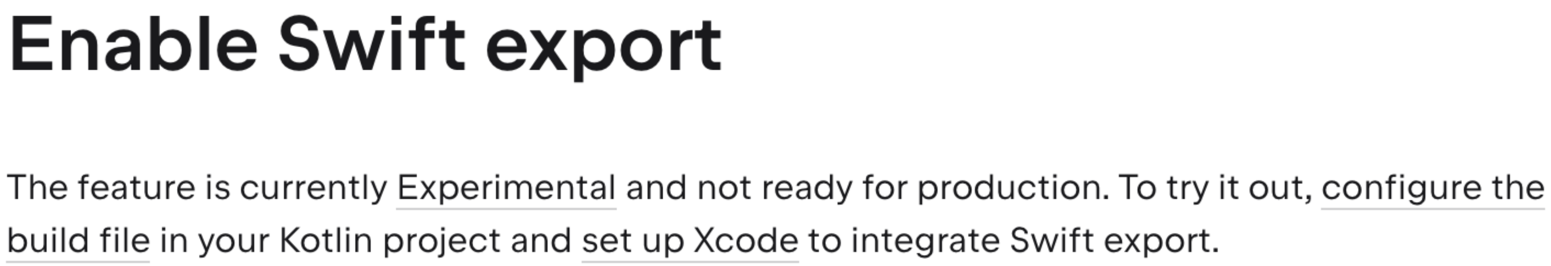
Kotlin-Swift.Прямой export

Experimental

Kotlin 2.2.21 поддерживает по умолчанию



```
kotlin.experimental.swift-export.enabled=true
```



Enable Swift export

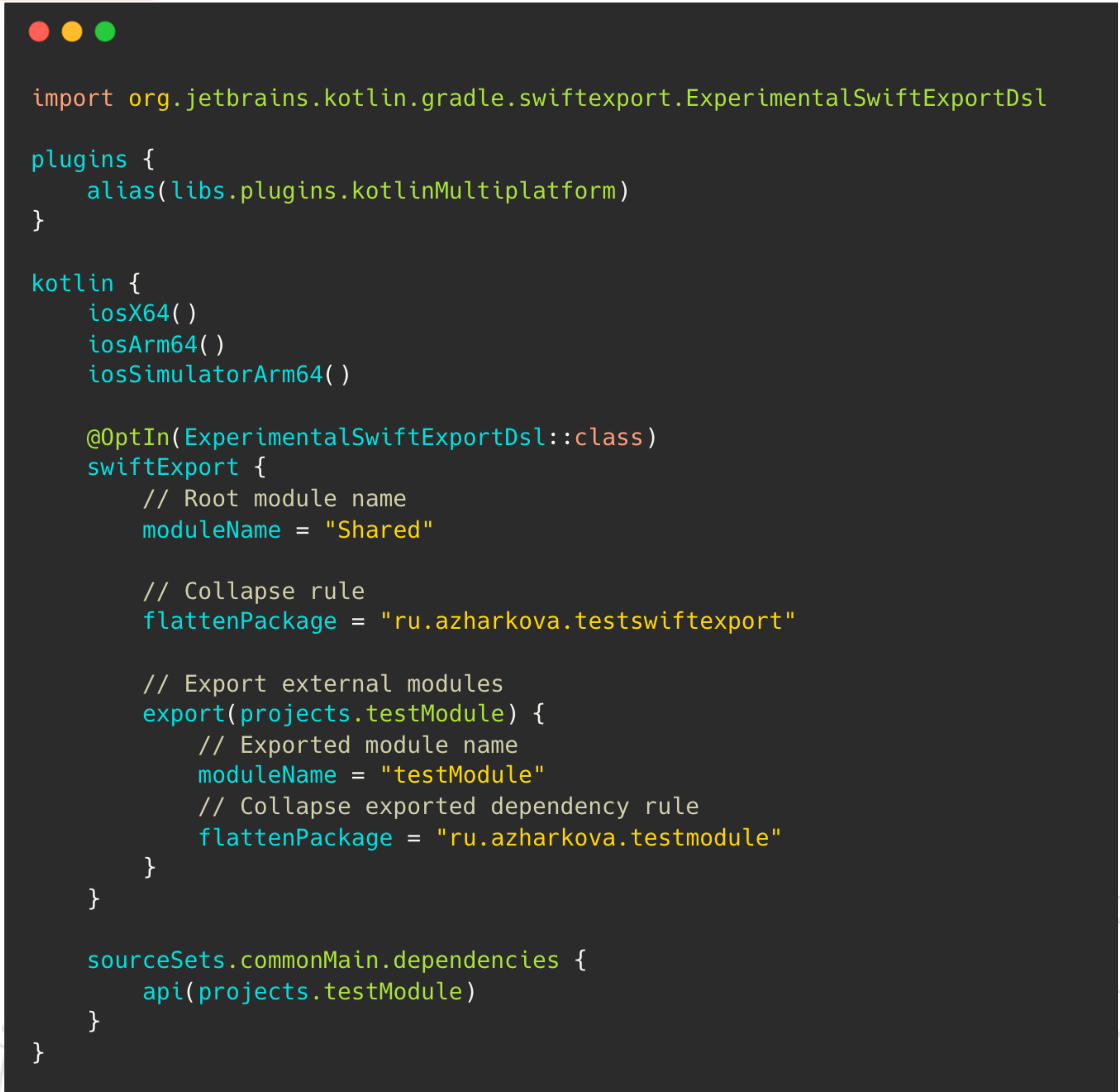
The feature is currently Experimental and not ready for production. To try it out, configure the build file in your Kotlin project and set up Xcode to integrate Swift export.

Не готов к продакшену

Kotlin-Swift.Прямой export

Experimental

swiftExport секция в build.gradle.kts

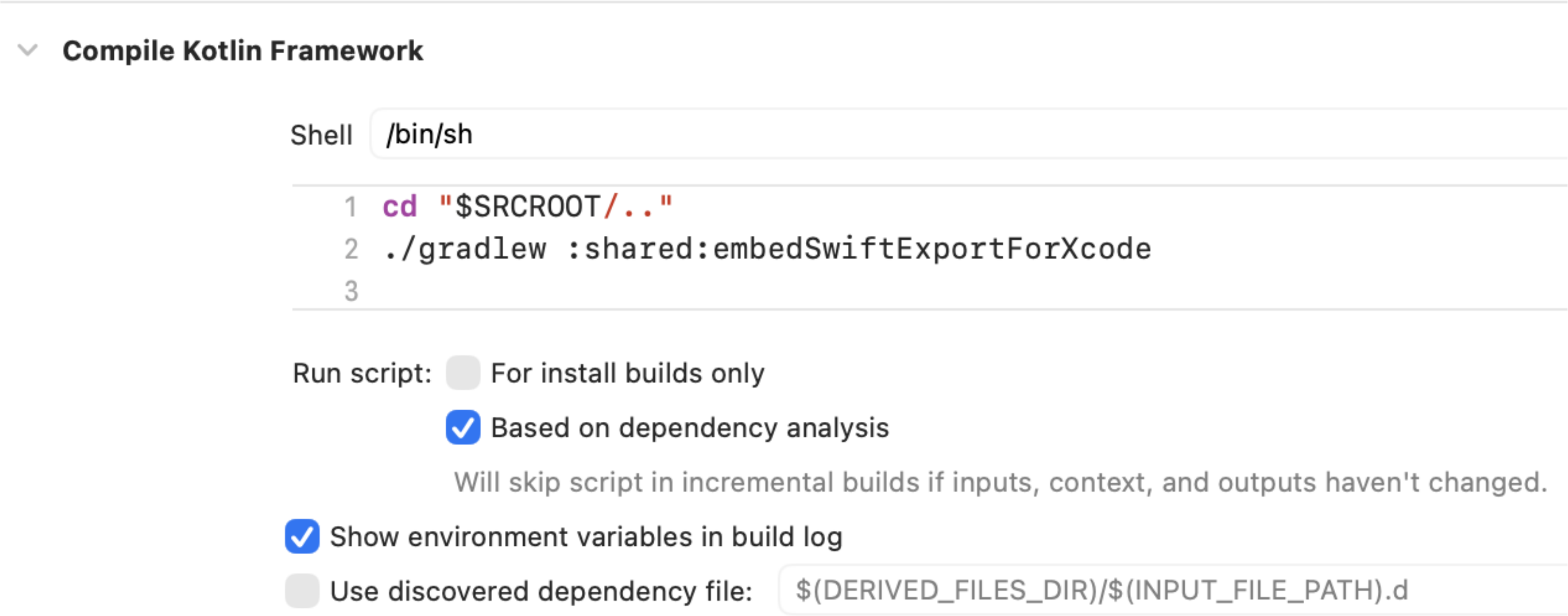


```kotlin
import org.jetbrains.kotlin.gradle.swiftexport.ExperimentalSwiftExportDsl

plugins {
    alias(libs.plugins.kotlinMultiplatform)
}

kotlin {
    iosX64()
    iosArm64()
    iosSimulatorArm64()

    @OptIn(ExperimentalSwiftExportDsl::class)
    swiftExport {
        // Root module name
        moduleName = "Shared"

        // Collapse rule
        flattenPackage = "ru.azharkova.testswiftexport"

        // Export external modules
        export(projects.testModule) {
            // Exported module name
            moduleName = "testModule"
            // Collapse exported dependency rule
            flattenPackage = "ru.azharkova.testmodule"
        }
    }

    sourceSets.commonMain.dependencies {
        api(projects.testModule)
    }
}
```

Kotlin-Swift.Прямой export

Experimental

Новый скрипт



Compile Kotlin Framework

Shell /bin/sh

```
cd "$SRCROOT/.."
./gradlew :shared:embedSwiftExportForXcode

```

Run script: □ For install builds only

☑ Based on dependency analysis

Will skip script in incremental builds if inputs, context, and outputs haven't changed.

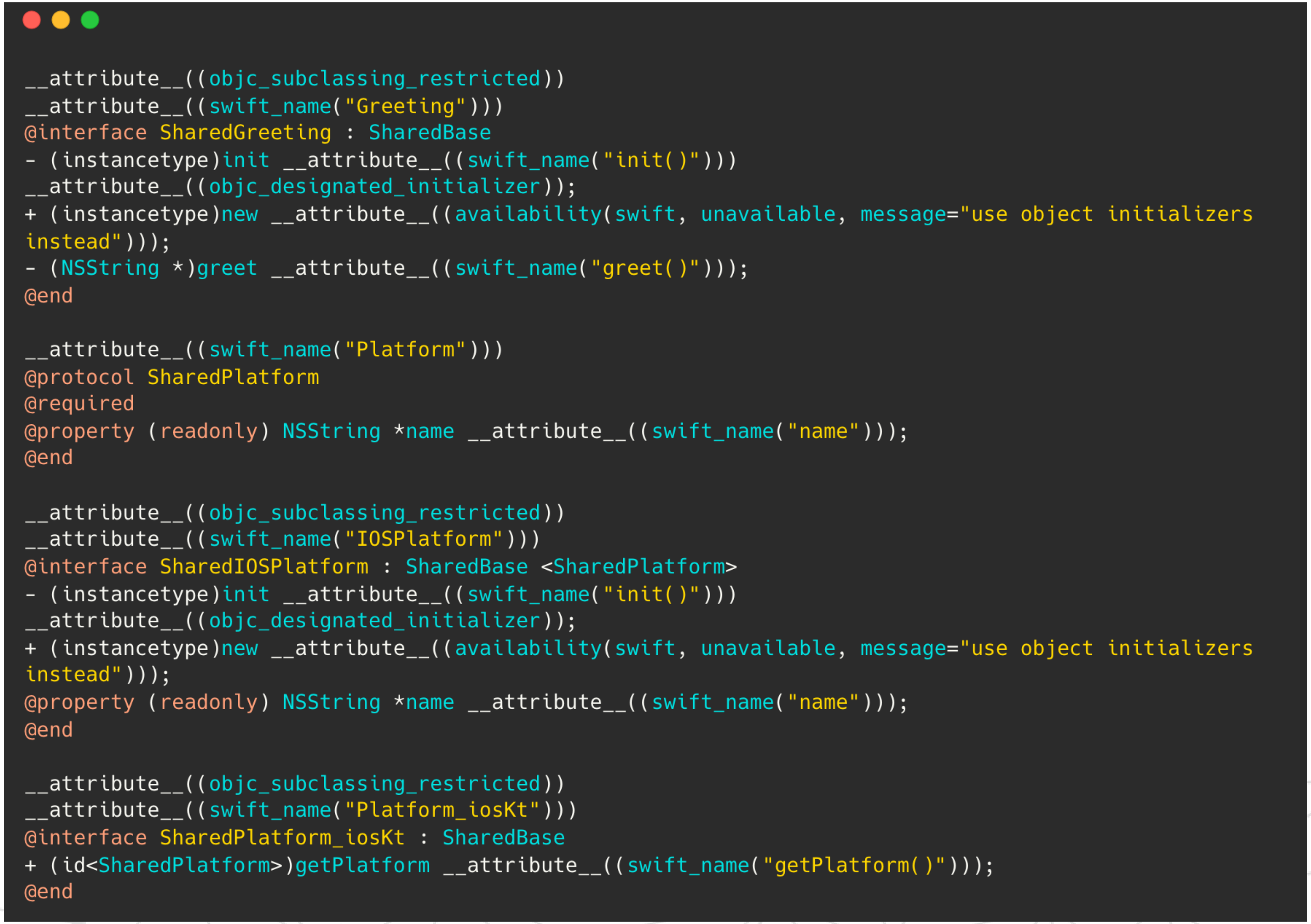
☑ Show environment variables in build log

□ Use discovered dependency file: $(DERIVED_FILES_DIR)/$(INPUT_FILE_PATH).d

Kotlin-Swift.Код до

Experimental

Полная версия > 170 строк

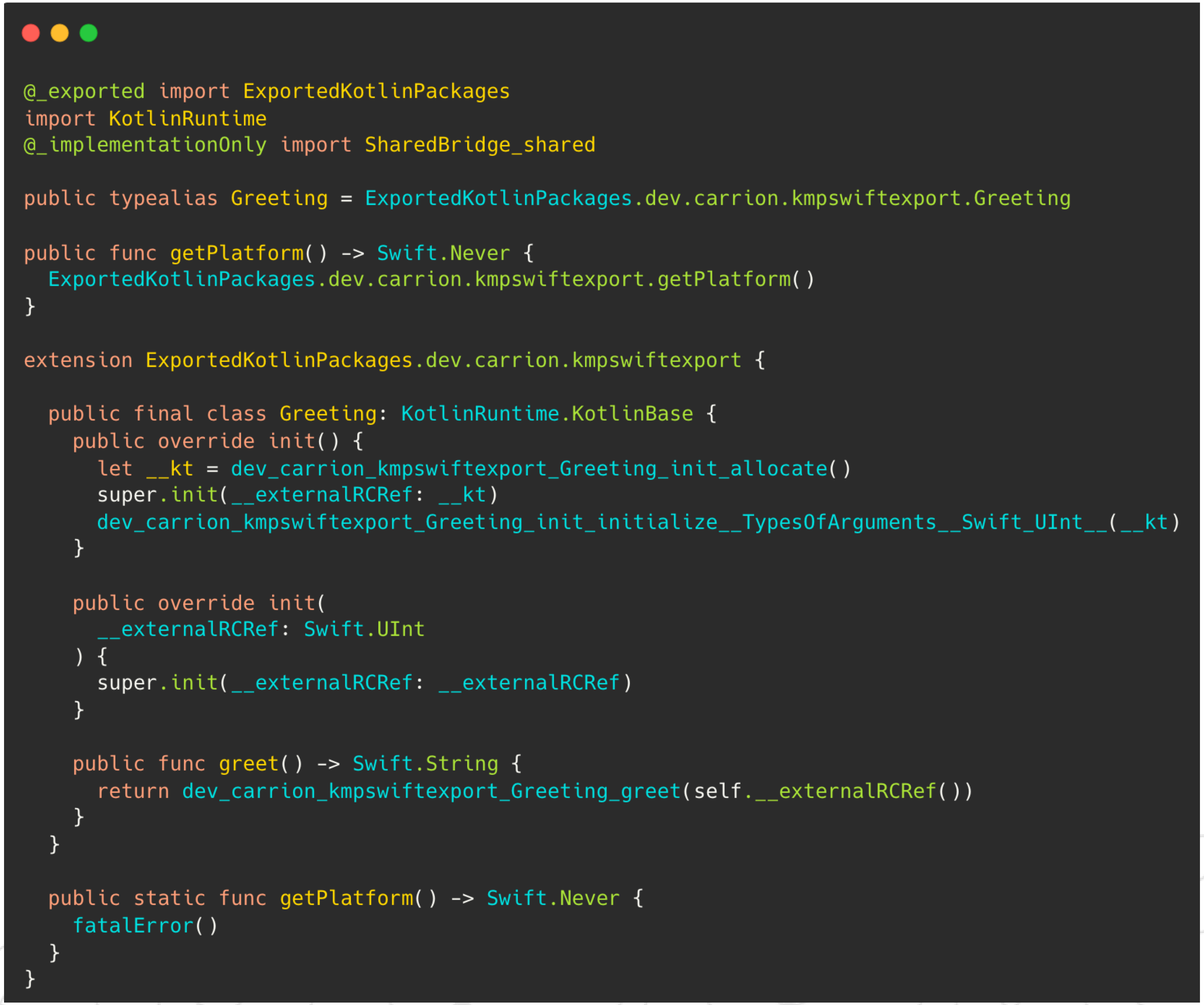


```
__attribute__((objc_subclassing_restricted))
__attribute__((swift_name("Greeting")))
@interface SharedGreeting : SharedBase
- (instancetype)init __attribute__((swift_name("init()")))
__attribute__((objc_designated_initializer));
+ (instancetype)new __attribute__((availability(swift, unavailable, message="use object initializers
instead")));
- (NSString *)greet __attribute__((swift_name("greet()")));
@end

__attribute__((swift_name("Platform")))
@protocol SharedPlatform
@required
@property (readonly) NSString *name __attribute__((swift_name("name")));
@end

__attribute__((objc_subclassing_restricted))
__attribute__((swift_name("IOSPlatform")))
@interface SharedIOSPlatform : SharedBase <SharedPlatform>
- (instancetype)init __attribute__((swift_name("init()")))
__attribute__((objc_designated_initializer));
+ (instancetype)new __attribute__((availability(swift, unavailable, message="use object initializers
instead")));
@property (readonly) NSString *name __attribute__((swift_name("name")));
@end

__attribute__((objc_subclassing_restricted))
__attribute__((swift_name("Platform_iosKt")))
@interface SharedPlatform_iosKt : SharedBase
+ (id<SharedPlatform>)getPlatform __attribute__((swift_name("getPlatform()")));
@end
```

Kotlin-Swift.Код после

Experimental

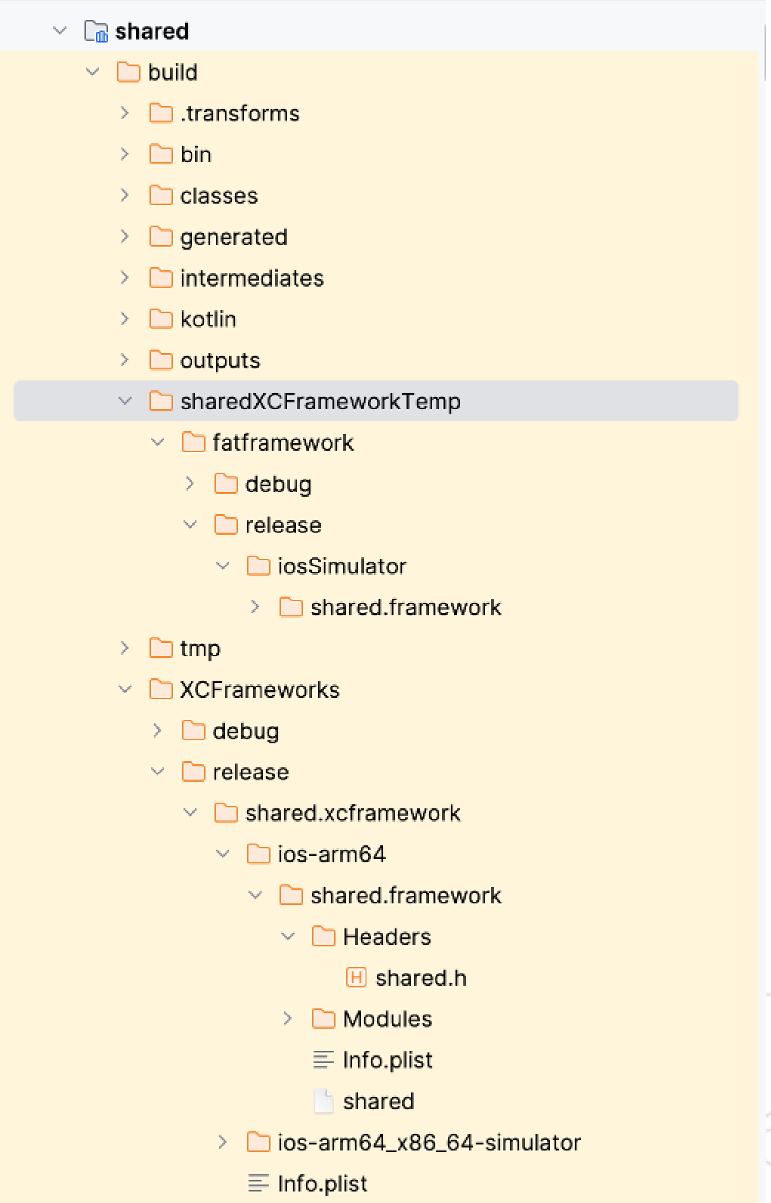
28 строк



```
@_exported import ExportedKotlinPackages
import KotlinRuntime
@_implementationOnly import SharedBridge_shared

public typealias Greeting = ExportedKotlinPackages.dev.carrion.kmpswiftexport.Greeting

public func getPlatform() -> Swift.Never {
  ExportedKotlinPackages.dev.carrion.kmpswiftexport.getPlatform()
}

extension ExportedKotlinPackages.dev.carrion.kmpswiftexport {

  public final class Greeting: KotlinRuntime.KotlinBase {
    public override init() {
      let __kt = dev_carrion_kmpswiftexport_Greeting_init_allocate()
      super.init(__externalRCRef: __kt)
      dev_carrion_kmpswiftexport_Greeting_init_initialize__TypesOfArguments__Swift_UInt__(__kt)
    }

    public override init(
      __externalRCRef: Swift.UInt
    ) {
      super.init(__externalRCRef: __externalRCRef)
    }

    public func greet() -> Swift.String {
      return dev_carrion_kmpswiftexport_Greeting_greet(self.__externalRCRef())
    }
  }

  public static func getPlatform() -> Swift.Never {
    fatalError()
  }
}
```

Готовим SPM библиотеку из нативной

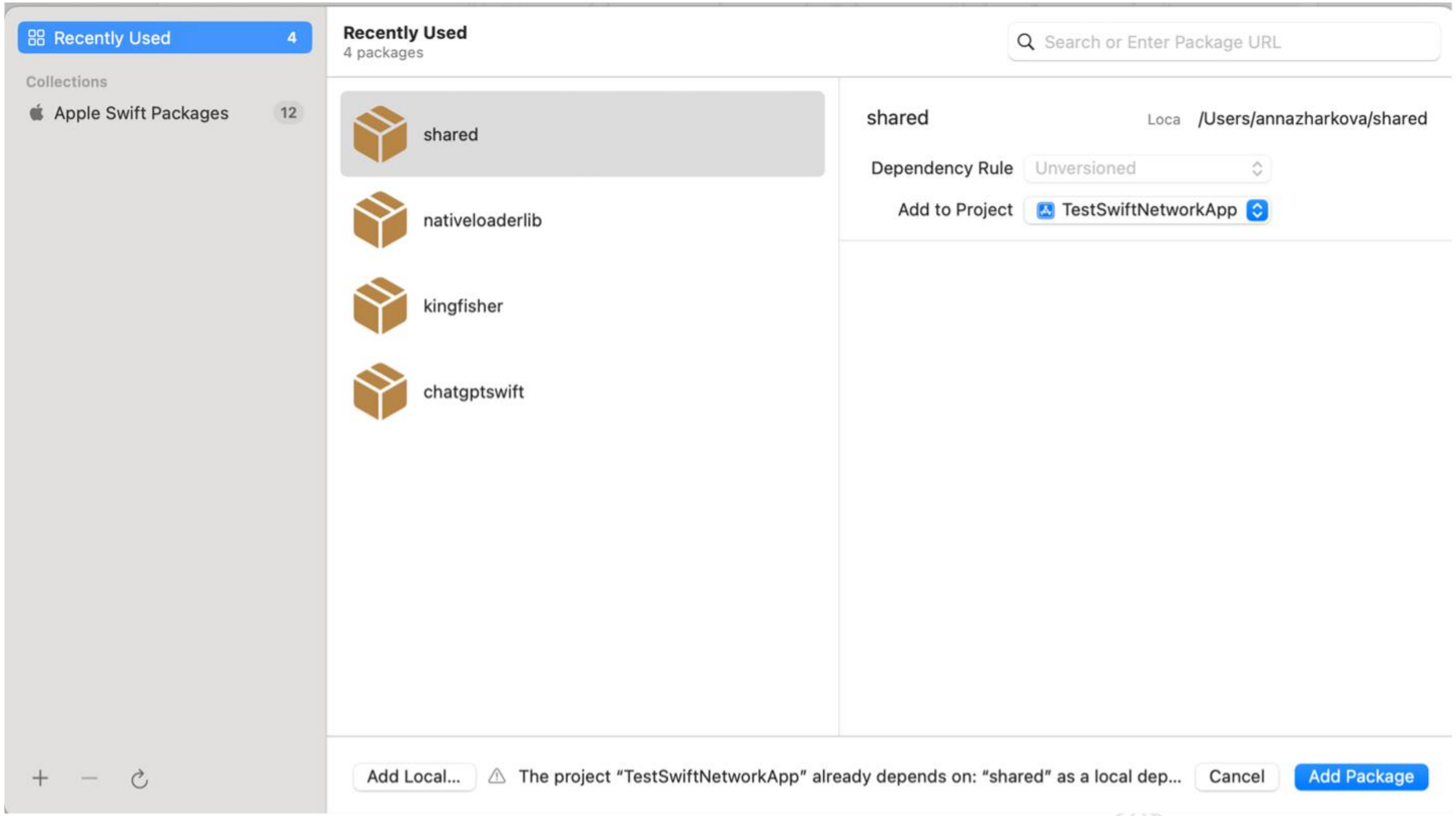
Собираем shared-модуль



```
shared
  build
    .transforms
    bin
    classes
    generated
    intermediates
    kotlin
    outputs
    sharedXCFrameworkTemp
      fatframework
        debug
        release
          iosSimulator
            shared.framework
    tmp
    XCFrameworks
      debug
      release
        shared.xcframework
          ios-arm64
            shared.framework
              Headers
                shared.h
              Modules
              Info.plist
              shared
          ios-arm64_x86_64-simulator
          Info.plist
```

Готовим SPM библиотеку из нативной

Создаем новый пакет



shared Loca /Users/annazharkova/shared

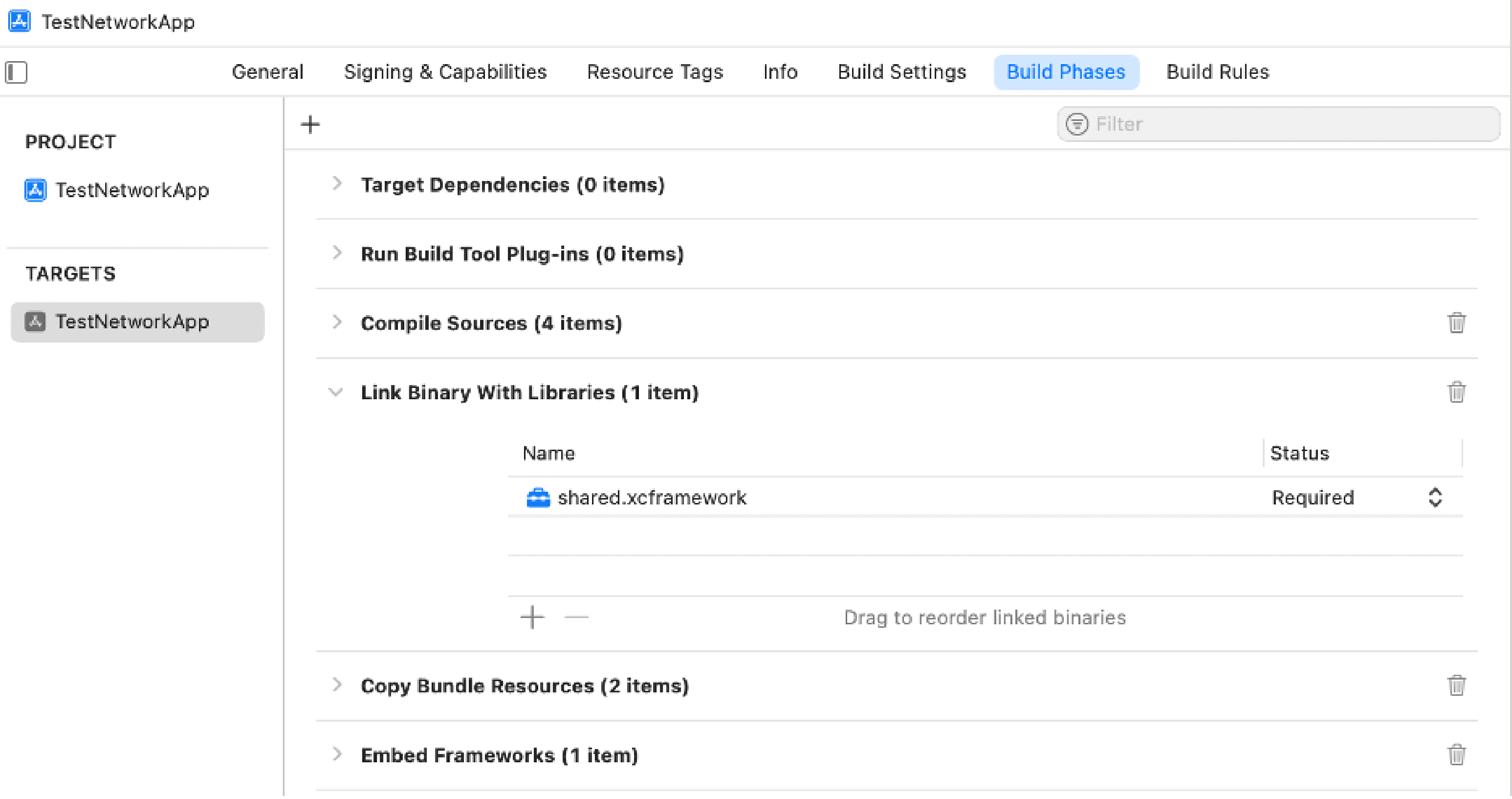
Dependency Rule: Unversioned

Add to Project: TestSwiftNetworkApp

⚠ The project "TestSwiftNetworkApp" already depends on: "shared" as a local dep...

Готовим SPM библиотеку из нативной

Подключаем



TestNetworkApp

General | Signing & Capabilities | Resource Tags | Info | Build Settings | Build Phases | Build Rules

PROJECT
- TestNetworkApp

TARGETS
- TestNetworkApp

- Target Dependencies (0 items)
- Run Build Tool Plug-ins (0 items)
- Compile Sources (4 items)
- Link Binary With Libraries (1 item)

Name	Status
shared.xcframework	Required

Drag to reorder linked binaries

- Copy Bundle Resources (2 items)
- Embed Frameworks (1 item)

Kotlin-Swift.Прямой export. Ограничения

- Подключение и настройка вручную
- Типы-наследники List, Set или Map, игнорируются при экспорте (KT-80416)
- В generics типы затираются до верхней границы
- Swift классы не могут напрямую наследовать от экспортированных Kotlin классов или интерфейсов
- Kotlin не может экспортировать функциональные типы в Swift
- Export ломается при одинаковых именах (SQLDelight Runtime и Compose Runtime (KT-80185))

Summary

- Swift-Java перспективный инструмент, но нужно много доработок
- Много настроек вручную
- Много лишнего кода и действий (Swift -> Swift -> Java, огромные билды)
- Swift-Java, Swift-Android не готовы к продакшену
- Kotlin-Swift – есть ограничения, нужны доработки, но выглядит более целостным

Список полезных источников

https://www.swift.org/blog/gsoc-2025-showcase-swift-java/

https://www.swift.org/blog/nightly-swift-sdk-for-android/

https://www.youtube.com/watch?v=tOH6V1IvTAc

https://docs.oracle.com/en/java/javase/21/core/foreign-function-and-memory-api.html

https://github.com/swiftlang/swift-android-examples

https://github.com/anioutkazharkova/swift-java-network

Спасибо за внимание!

@anioutkajarkova

azharkova
prettygeeknotes