



Курс Kotlin Jetpack Compose

Лекция 2

Компиляторы Kotlin
(Kotlin/JVM, Kotlin/JS,
Kotlin/Native)

Д. Макаренков

<https://dmpsy.club>

JetpackCompose <= (Kotlin && Android)



Целевая аудитория

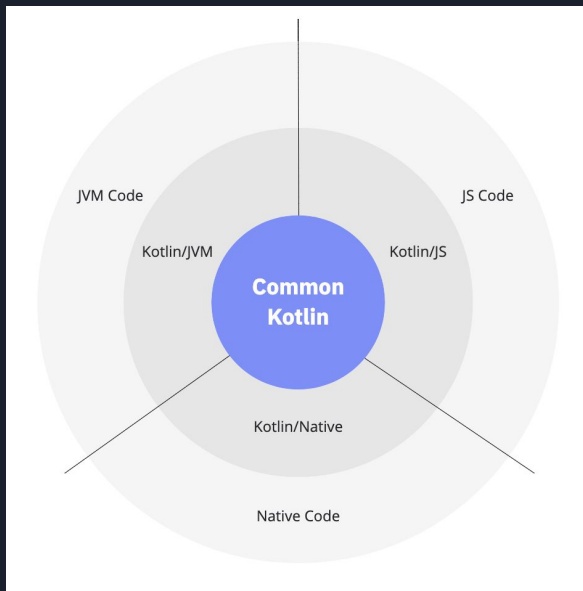
- Энтузиасты программирования на Kotlin для Android, желающие следовать современным принципам разработки пользовательских интерфейсов (UI) с использованием библиотеки Jetpack Compose

Девиз:

В начале было Слово...

Слово = Исходный код Kotlin (*.kt)

Триединство компиляторов Kotlin



Три компилятора реализуют одно естественное (приложение Java, Kotlin/JVM) и два, на первый взгляд, “противоестественных” (код JavaScript, компилятор Kotlin/JS) и (исполняемый файл, компилятор Kotlin/Native) “воплощения” исходного кода Kotlin

(см. <https://kotlinlang.org/docs/multiplatform.html#how-kotlin-multiplatform-works>)

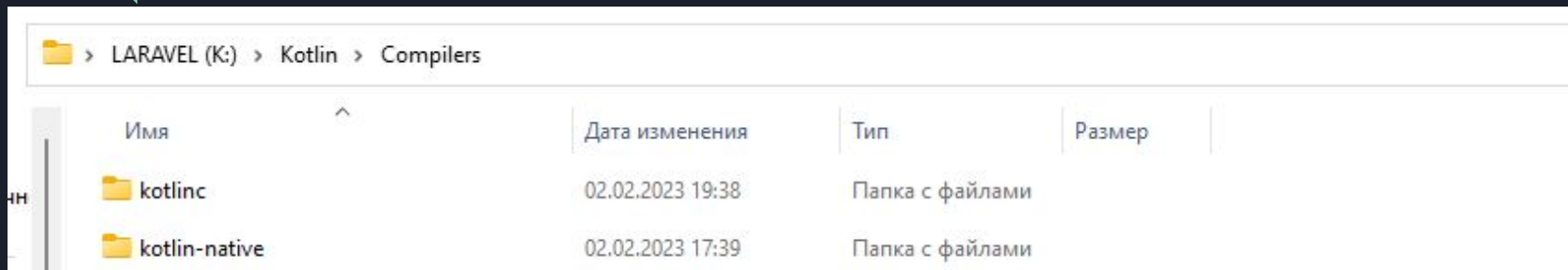
Установка компиляторов. Где скачать архивы

▼ Assets 8

 kotlin-compiler-1.8.0.zip	71.5 MB	Dec 28, 2022
 kotlin-native-linux-x86_64-1.8.0.tar.gz	181 MB	Dec 28, 2022
 kotlin-native-macos-aarch64-1.8.0.tar.gz	303 MB	Dec 28, 2022
 kotlin-native-macos-x86_64-1.8.0.tar.gz	304 MB	Jan 4
 kotlin-native-windows-x86_64-1.8.0.zip	175 MB	Dec 28, 2022
 maven-1.8.0-sha256.txt	88.1 KB	last month
 Source code (zip)		Dec 20, 2022
 Source code (tar.gz)		Dec 20, 2022

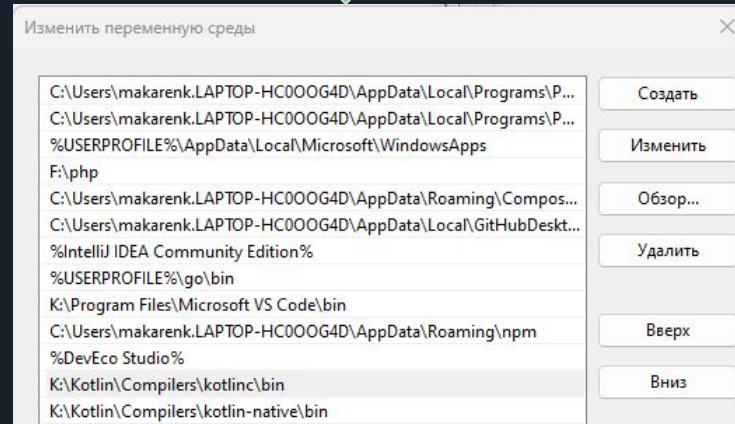
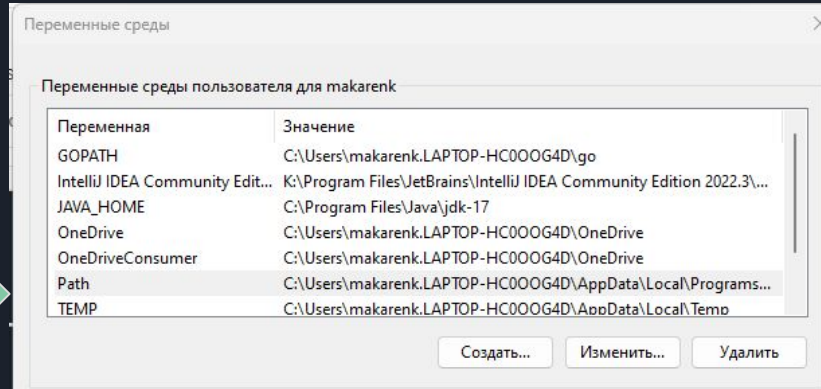
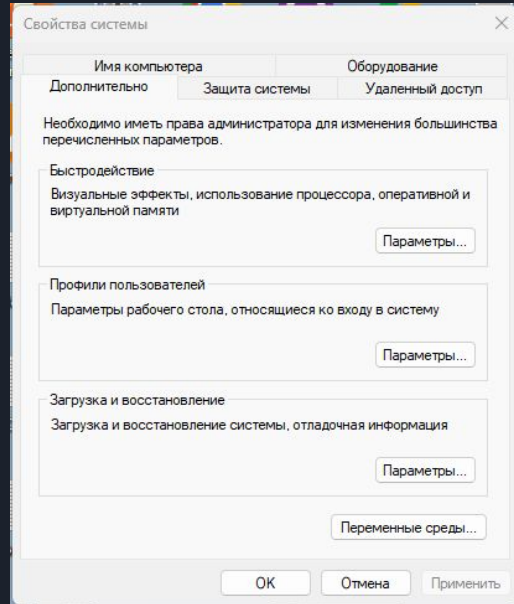
Скачивайте архивы 1.8.0 отсюда: <https://github.com/JetBrains/kotlin/releases/tag/v1.8.0>

Установка компиляторов. Windows 11 (1)



1. Скачать с github.com и раскрыть в удобную директорию (см. выше) `kotlin-compiler-1.8.0.zip` и `kotlin-native-windows-x86_64.1.8.0.zip`
2. Прописать и сохранить пути к компиляторам в переменной среды `path` (`K:\Kotlin\Compilers\kotlinc\bin`, `K:\Kotlin\Compilers\kotlinc-native\bin`):
Поиск | Изменение системных переменных среды... (см. далее))

Установка компиляторов. Windows 11 (2)



Установка компиляторов. Windows 11 (3)

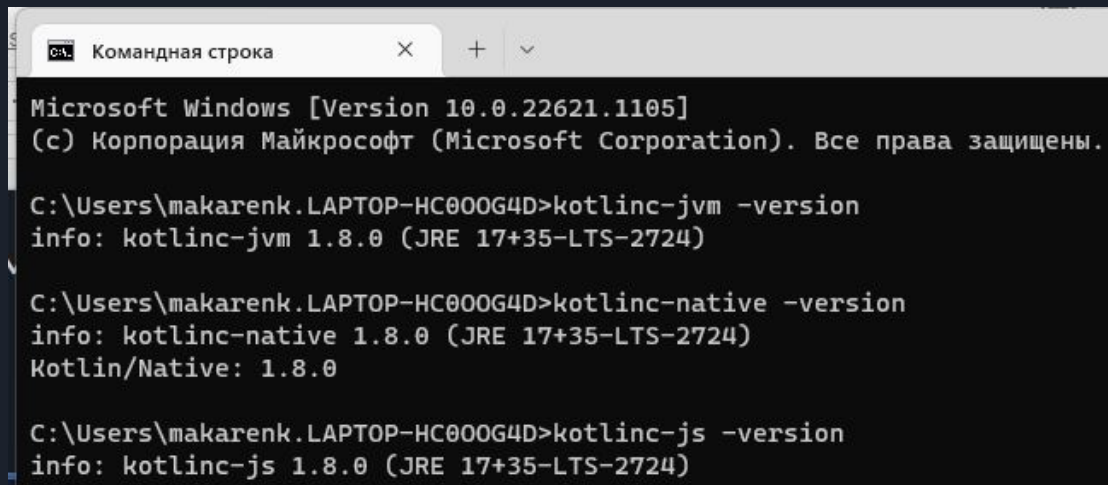
3. Убедиться, что пути работают:

Пуск | cmd

kotlinc-jvm -version

kotlinc-js -version

kotlinc-native -version



```
Microsoft Windows [Version 10.0.22621.1105]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\makarenk.LAPTOP-HC000G4D>kotlinc-jvm -version
info: kotlinc-jvm 1.8.0 (JRE 17+35-LTS-2724)

C:\Users\makarenk.LAPTOP-HC000G4D>kotlinc-native -version
info: kotlinc-native 1.8.0 (JRE 17+35-LTS-2724)
Kotlin/Native: 1.8.0

C:\Users\makarenk.LAPTOP-HC000G4D>kotlinc-js -version
info: kotlinc-js 1.8.0 (JRE 17+35-LTS-2724)
```

Установка компиляторов. Linux Ubuntu (1)

=====Linux-based (Ubuntu) installation / processing=====

1. Установить midnight commander - полезно для редактирования и не только
`sudo apt install mc`

2. Скачать и установить Java Open JDK:

`https://jdk.java.net/19/`

или

`sudo apt install default-jre`

`java -version`

`openjdk version "11.0.17" 2022-10-18`

3. Установить nvm для управления nodejs

`https://tecadmin.net/how-to-install-nvm-on-ubuntu-20-04/`

`sudo apt install curl // Если не установлен`

`curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash`

`source ~/.bashrc`

4. Установить node.js

`nvm install node`

`node -v`

`v19.5.0`

`npm -v`

`9.3.1`

`npmx -v`

`9.3.1`

Установка компиляторов. Linux Ubuntu (2)

5. Скачать и установить компиляторы kotlin (в /opt)

```
https://github.com/JetBrains/kotlin/releases/tag/v1.8.0
```

```
kotlin-compiler-1.8.0.zip
```

```
kotlin-native-linux-x86_64-1.8.0.tar.gz
```

```
sudo mkdir /opt
```

```
sudo chmod 777 /opt
```

```
cd ~/Downloads
```

```
mv kotlin-compiler* /opt/
```

```
mv kotlin-native-linux* /opt/
```

```
cd /opt
```

```
unzip kotlin-compiler-1.8.0.zip
```

```
tar -xzf kotlin-native*
```

```
mv kotlin-native-linux-x86_64-1.8.0 kotlin-native
```

6. Добавить пути в .bashrc (с помощью mcedit):

```
cd
```

```
mc
```

```
export PATH="$PATH:/opt/kotlinc/bin:/opt/kotlin-native/bin"
```

Для применения изменений перезапустить терминал или выполнить:

```
source ~/.bashrc
```

Установка компиляторов. Linux Ubuntu (3)

6. Добавить пути в `.bashrc` (с помощью `mcedit`):

```
cd
mc
  export PATH="$PATH:/opt/kotlinc/bin:/opt/kotlin-native/bin"
Для применения изменений перезапустить терминал или выполнить:
source ~/.bashrc
```

7. Убедиться в доступности компиляторов:

```
kotlinc-jvm -version
info: kotlinc-jvm 1.8.0...
kotlinc-js -version
info: kotlinc-js 1.8.0
error:=====
regarding IR-based compiler...
=====
kotlinc-native -version
info: kotlinc-native 1.8.0
Kotlin/Native: 1.8.0
konanc -version
info: kotlinc-native 1.8.0
Kotlin/Native: 1.8.0
=====end of installation=====
```

Установка компиляторов Kotlin JVM/JS. MacOS (1)

====Kotlin MacOS Installation====

1. Установить homebrew с brew.sh
`/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
2. Установить midnight commander
`brew install mc`
`LANG=ru_RU.UTF-8 mc`
3. Установить nvm для управления nodejs
`brew install nvm`
`mkdir ~/.nvm`
touch ~/.zshrc и добавить в файл следующие строки:
`export NVM_DIR="$HOME/.nvm"`
`[-s "/usr/local/opt/nvm/nvm.sh"] && \. "/usr/local/opt/nvm/nvm.sh" # This loads nvm`
`[-s "/usr/local/opt/nvm/etc/bash_completion.d/nvm"] && \. "/usr/local/opt/nvm/etc/bash_completion.d/nvm"`
This loads nvm bash_completion
Перезапустить клиента ssh для применения изменений
`nvm -v`
0.39.3
4. Установить nodejs
`nvm install node`
`nvm ls`
-> v19.5.0
`node -v`
v19.5.0
`npm -v`
9.3.1
`pnpm -v`
9.3.1

Установка компиляторов Kotlin JVM/JS. MacOS (2)

5. Установить компиляторы Kotlin JVM/Kotlin JS из brew
(это также установит и openjdk):

```
brew install kotlin
```

6. Убедиться в работоспособности

```
which kotlin
```

```
/usr/local/bin/kotlin
```

```
kotlin -version
```

```
Kotlin version 1.8.0-release-345 (JRE 19.0.2)
```

```
kotlinc -version
```

```
kotlinc-jvm 1.8.0 (JRE 19.0.2)
```

```
kotlinc-js 1.8.0 (JRE 19.0.2+7-44)
```

```
error: =====.kt-o
```

```
This project currently uses the Kotlin/JS Legacy compiler backend
```

```
Please migrate to the new IR-based compiler...
```

```
=====окончание установки Kotlin/JVM и Kotlin/JS=====
```

Установка компилятора Kotlin/Native MacOS BigSur (1)

```
=====Установка Kotlin/Native MacOS=====
```

```
Минимальные условия: BigSur macOS 11.X.X + Xcode_12.5.1.xip
```

```
=====
```

1. Скачать архив со страницы Kotlin github:

```
https://github.com/JetBrains/kotlin/releases/tag/v1.8.0
```

```
kotlin-native-macos-x86_64-1.8.0.tar.gz or
```

```
kotlin-native-macos-aarch64-1.8.0.tar.gz (if this arch is supported by your SW/HW)
```

2. cd Downloads

```
tar -xzf kotlin*
```

```
ls
```

```
kotlin-native-macos-x86_64-1.8.0
```

```
kotlin-native-macos-x86_64-1.8.0.tar.gz
```

3. sudo mkdir /opt

```
sudo chmod 777 /opt
```

```
mv kotlin-native-macos-x86_64-1.8.0 /opt/kotlinc-native
```

4. edit .zshrc:

```
export PATH=$PATH:/opt/kotlinc-native/bin
```

Установка компилятора Kotlin/Native MacOS BigSur (2)

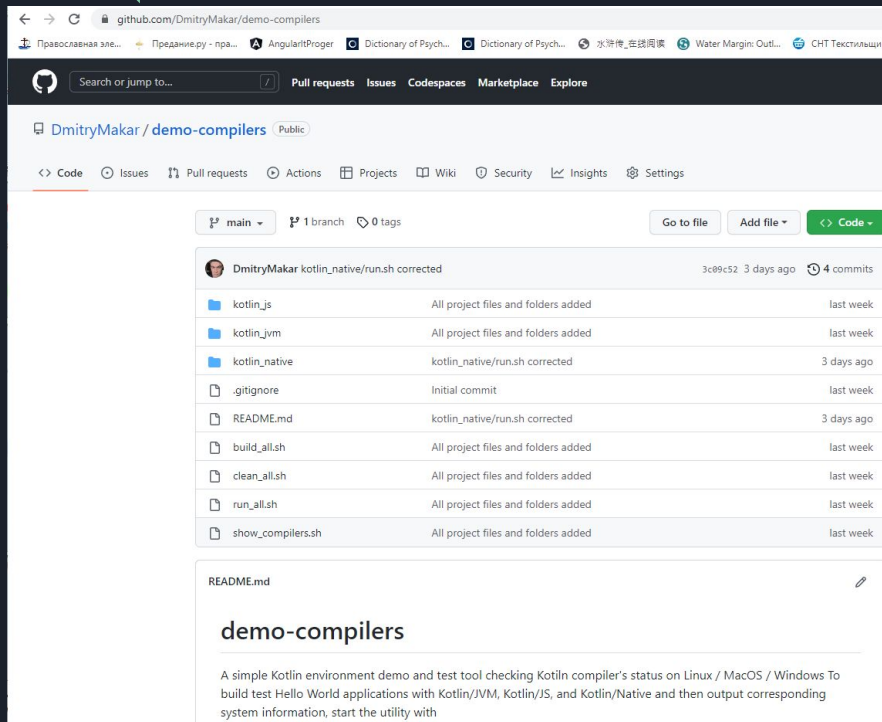
5. Перезапустить клиента ssh для применения изменений
6. Установить xcode с `apple.developer.com` (необходим для работы с Kotlin/Native)

7. Убедиться в доступности компилятора

```
kotlinc-native -version
kotlinc-native 1.8.0 (JRE 1.8.0_361-b09)
Kotlin/Native: 1.8.0
konanc -version
kotlinc-native 1.8.0 (JRE 1.8.0_361-b09xvode)
Kotlin/Native 1.8.0
```

```
=====
====Annex=====
---- Установка и удаление Command Line Tools-----
xcode-select --install # Устанавливает Command Line Tools
sudo rm -rf /Library/Developer/CommandLineTools # удаляет Command Line Tools
---- Удаление Xcode:-----
Найти Xcode.app и перетащить его в корзину (Trash)
~/Library/Developer/
~/Library/Caches/
----NB--Homebrew устанавливает свои CommandLineTools;
соответственно, путь должен быть изменен--
xcode-select -p # показывает текущий путь к Active Developer Directory
/Library/Developer/CommandLineTools
xcode-select -s /Applications/Xcode.app # меняет путь на правильный, к Xcode
xcode-select -p
/Applications/Xcode.app/Contents/Developer
=====end of MacOS Installation=====
```

Тестирование: demo_compilers/show_compilers.sh (1)



The screenshot shows the GitHub interface for the repository 'demo-compilers' by user 'DmitryMakar'. The repository is public and has 1 branch (main) and 0 tags. The commit history table is as follows:

File	Commit Message	Time	Commits
kotlin_js	All project files and folders added	last week	
kotlin_jvm	All project files and folders added	last week	
kotlin_native	kotlin_native/run.sh corrected	3 days ago	
.gitignore	Initial commit	last week	
README.md	kotlin_native/run.sh corrected	3 days ago	
build_all.sh	All project files and folders added	last week	
clean_all.sh	All project files and folders added	last week	
run_all.sh	All project files and folders added	last week	
show_compilers.sh	All project files and folders added	last week	

Below the table, the README.md file is partially visible, starting with the title 'demo-compilers' and a description: 'A simple Kotlin environment demo and test tool checking Kotlin compiler's status on Linux / MacOS / Windows To build test Hello World applications with Kotlin/JVM, Kotlin/JS, and Kotlin/Native and then output corresponding system information, start the utility with'.

1. Установите и запустите GitBash (если на Windows) или иную подобную оболочку:

<https://git-scm.com/downloads>

2. git clone <https://github.com/DmitryMakar/demo-compilers.git>
3. cd demo_compilers
4. ./show_compilers

Тестирование: вывод show_compilers.sh (1)

```
MINGW64/k/kotlin/demo-compilers
build_all.sh kotlin_js/ kotlin_native/ show_compilers.sh

makarenk@LAPTOP-HC000G4D MINGW64 /k/kotlin/demo-compilers (main)
$ ./show_compilers.sh
=====
We build scripts first
=====
hello.jar build passed
warning: =====
This project currently uses the Kotlin/JS Legacy compiler backend, which has been deprecated and will be removed in a future release.

Please migrate your project to the new IR-based compiler (https://kotlin.jsir). Because your build tool will not support the new Kotlin/JS compiler, you will also need to migrate to Gradle.

You can continue to use the deprecated legacy compiler in the current version of the toolchain by providing the compiler option -Xuse-deprecated-legacy-compiler.
=====

added 1 package in 1s
hello.js build passed
hello executable build passed
Everything is done!
=====
Now launch them
=====
=====
WELCOME TO KOTLIN COMPILERS!
=====
```

```
MINGW64/k/kotlin/demo-compilers
=====
=====
WELCOME TO KOTLIN COMPILERS!
=====
Hello Kotlin/JVM!
=====
Hello Kotlin/JS!
=====
Hello Kotlin/Native!
=====
CURRENT SOFTWARE VERSIONS:
=====
java version "17" 2021-09-14 LTS
Java(TM) SE Runtime Environment (build 17+35-LTS-2724)
Java HotSpot(TM) 64-Bit Server VM (build 17+35-LTS-2724, mixed mode, sharing)
info: kotlinc-jvm 1.8.0 (JRE 17+35-LTS-2724)
info: kotlinc-js 1.8.0 (JRE 17+35-LTS-2724)
error: =====
This project currently uses the Kotlin/JS Legacy compiler backend, which has been deprecated and will be removed in a future release.

Please migrate your project to the new IR-based compiler (https://kotlin.jsir). Because your build tool will not support the new Kotlin/JS compiler, you will also need to migrate to Gradle.

You can continue to use the deprecated legacy compiler in the current version of the toolchain by providing the compiler option -Xuse-deprecated-legacy-compiler.
=====
info: kotlinc-native 1.8.0 (JRE 17+35-LTS-2724)
Kotlin/Native: 1.8.0
```


Тестирование: вывод show_compilers.sh (2)

```
MINGW64/k/kotlin/demo-compilers
Please migrate your project to the new IR-based compiler (https://kotlin.in/jsir).
Because your build tool will not support the new Kotlin/JS compiler, you will al
so need to migrate to Gradle.

You can continue to use the deprecated legacy compiler in the current version of
the toolchain by providing the compiler option -Xuse-deprecated-legacy-compiler
.
=====
info: kotlinc-native 1.8.0 (JRE 17+35-LTS-2724)
Kotlin/Native: 1.8.0
===node version:===
v16.17.1
===npm version:===
9.1.3
=====
Now sleeping for 3 second
=====
Now cleaning the project
stoppint the script
=====
rm: cannot remove '*.class': No such file or directory
kotlin_jvm cleaned
kotlin_js cleaned
kotlin_native cleaned
Everything is done!
=====
Exiting now, Bye-bye!
=====
makarenk@LAPTOP-HC00G4D MINGW64 /k/kotlin/demo-compilers (main)
$
```

Тестирование: примеры исходников для практикума

```
it.kt x hello.kt x misc.xml x workspace.xml x Hi  
fun main() {  
    println("Hello from Windows Kotlin/Native!")  
    println("Please introduce yourself:")  
    println("Have a nice day, ${readln()}!")  
    println("\nPress any key to return")  
    readln()  
}
```

```
hellojs.kt x misc.xml x workspace.xml x t  
fun main() {  
    println("Hello from Windows Kotlin/JS!")  
}
```

Необходимые опции компиляторов

Компилятор	Пример строки сборки	Строка запуска	Комментарий
Kotlin/JVM	<code>kotlinc-jvm -include-runtime hello.kt -d hello.jar</code>	<code>java -jar hello.jar</code>	Кроссплатформенность, “ДУХ”
Kotlin/JS	<code>kotlinc-js hellojs.kt -Xuse-deprecated-legacy-compiler -module-kind commonjs -output hello.js npm install kotlin</code>	<code>node hello.js</code>	Кроссплатформенность, “ДУША”
Kotlin/Native	<code>kotlinc-native hello.kt -o hello</code>	<code>hello.exe, ./hello.exe, ./hello.keexe</code>	Мультиплатформенность, “ТЕЛО”

NB. Дух-Душа-Тело -условно, по мере “уплотнения” результирующей “материю”: содержимое jar универсально, требуется только JVM для исполнения; JavaScript подразумевает наличие (стороннего для Kotlin/Java) движка nodejs v80 и / или Webserver’a; исполняемые же файлы “прибиты гвоздями” к своей операционке локальными компиляторами (например, Xcode)

Тестирование: сгенерированный архив hello.jar

```
ArcLite:zip:hello.jar
n      Name
..
kotlin
META-INF
HelloKt.class
Up 07.02.23 13:50
Bytes: 1,04 K, files: 1, folders: 2

ArcLite:zip:hello.jar:\kotlin
n      Name
..
annotation
collections
comparisons
concurrent
contracts
coroutines
enums
experimental
internal
io
jdk7
js
jvm
math
properties
random
ranges
reflect
sequences
streams
system
text
time
_Assertions.class
_ArrayIntrinsicsKt.class
_BuilderInference.class
_CharCodeJVMKt.class
_CharCodeKt.class
_CompareToKt.class
_ContextFunctionTypeParams.cla}
_DeepRecursiveFunction.class
_DeepRecursiveKt.class
_DeepRecursiveScope.class
_DeepRecursiveScopeImpl$crossF}
_DeepRecursiveScopeImpl.class
_Deprecated.class
_DeprecatedSinceKotlin.class
_DeprecationLevel.class
_DslMarker.class
_ExceptionsKt.class
_ExceptionsKt__ExceptionsKt.cl}
_ExperimentalMultiplatform.cla}
_ExperimentalStdlibApi.class
_ExperimentalSubclassOptIn.cla}
_ExperimentalUnsignedTypes.cla}
Up 07.02.23 13:50
Bytes: 280 K, files: 113, folders: 23

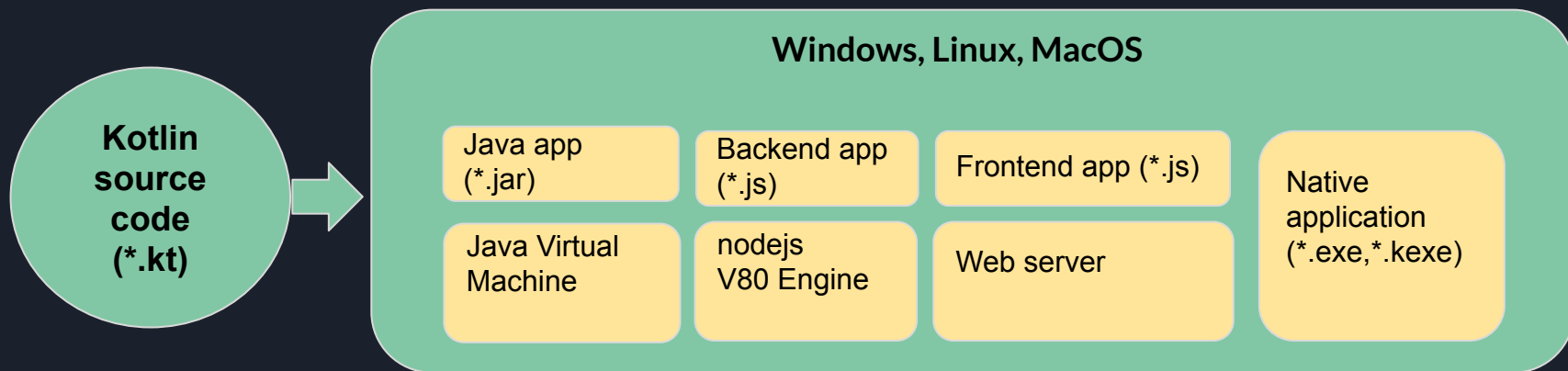
\kotlin>
1Help 2UserMn 3View 4Edit 5Copy 6RenMov 7MkFold 8Delete 9ConfMn 10Quit 11Plugin 12Screen
```

Тестирование: сгенерированный JavaScript

К: > Kotlin > Compilers > JS hello.js > ...

```
1 (function (_, Kotlin) {
2   'use strict';
3   var println = Kotlin.kotlin.io.println_s8jyv4$;
4   function main() {
5     println('Hello from Windows Kotlin/JS!');
6   }
7   _.main = main;
8   main();
9   Kotlin.defineModule('hello', _);
10  return _;
11 }(module.exports, require('kotlin')));
12
```

Кросс- и мультиплатформенность Kotlin



- **Кроссплатформенность** - возможность использования приложения без пересборки на разных платформах (Java/JVM, JS/node, JS/Webserver)
- **Мультиплатформенность** - возможность сборки приложения под разные платформы (Native application под Windows / Linux/MacOS)

Варианты сборки





Полезные ресурсы

Ссылка	Комментарий
https://kotlinlang.org/docs/multiplatform.html	Kotlin как мультиплатформенное решение (ENG)
https://blog.jetbrains.com/kotlin/2023/01/kotlin-1-8-0-released/?_gl=1*13ysm3k*_ga*MTMyNTQ3ODkzOC4xNjcyMTIzNjAz*_ga_9J976DJZ68*MTY3NTc2NDc1NS41NS4wLjE2NzU3NjQ3NTUuMC4wLjA.&_ga=2.80286303.935657555.1675764755-1325478938.1672123603	О компиляторах 1.8.0 (ENG)
https://github.com/DmitryMakar/demo-compilers	Утилита для быстрого тестирования компиляторов



Следующее видео

- Лекция 2 Практикум

P.S. Презентация доступна для скачивания здесь:

https://dmpsy.club/references/Kotlin/lesson_002_compilers_rus.pdf

<https://dmpsy.club>

postmaster@dmpsy.club

