

Android Studio

(ako začat')

Jazyk Kotlin

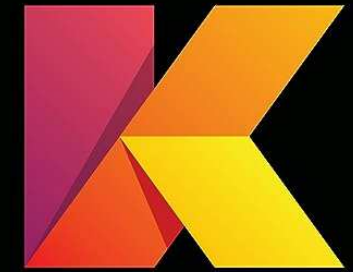
(ako neskončit')



Peter Borovanský
KAI, I-18

MS-Teams: [2sf3ph4](#), [List](#), [github](#)

borovan 'at' ii.fmph.uniba.sk



Základné info o kurze

- Stránka predmetu
 - <https://dai.fmph.uniba.sk/courses/VMA/>
- Prémiovo-orientované vyučovanie
 - prihláste sa do [L.I.S.T.](#)
 - ak ste v ňom nikdy neboli, ozvite sa mi mailom
 - sledujte LIST, všetky zadania budú v ňom
 - sledujte Teams [2sf3ph4](#), komunikácia/prednášky/oznamy budú tam
 - prémie sa budú nepravidelne objavovať a sú **plnohodnotná** alternatíva k domácim úlohám

veľká časť kurzu bude dobre sledovateľná z knihy

Android Studio Giraffe Essentials - Kotlin Edition: Developing Android Apps Using Android Studio 2022.3.1 and Kotlin , Neil Smyth

.pdf je k dispozícii...

<https://www.amazon.com/Android-Studio-Giraffe-Essentials-Developing/dp/1951442776>

Vývojové jazyky/nástroje

- Symbian
 - C++, Java ME, Python, ...
- Windows Mobile 6
 - C# (MS Visual Studio)
- iOS
 - Objective-C -> Swift 3/4/5 (Xcode)
- Android
 - scratch (MIT Inventor)
 - java (Android SDK + plugin pre Eclipse) -> (Android Studio)
 - java (A.I.D.E.)
 - Kotlin (Android Studio 4+)
 - C++ (Android NDK)
- Multi-platform
 - C# (Xamarin iOS, Android, Windows) – fy. MS, Visual Studio 2015
 - Pascal (Delphi XE5 iOS, Android, Windows 10) – fy. Embarcadero
 - JavaScript/TypeScript (React Native)
 - Flutter od Google
- game engine
 - C# (Unity 2D/3D)

iOS - Apple Center kurz

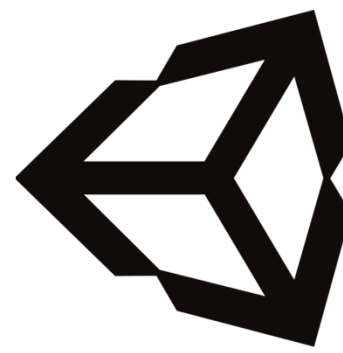
1-AIN-303/15

...



Unity

(game engine)



unity

#1 Unreal Engine

#2 Unity

Oblíbený nástroj pre tvorbu multi-platform aplikácií pre bakalárske práce

- 2-INF-263/15 magisterský predmet: Tvorba a dizajn počítačových hier
 - <http://sccg.sk/~mferko/tdh/>
 - <https://candle.fmph.uniba.sk/ucitelia/Michal-Ferko>
- 1-AIN-303/15 bakalársky predmet: Game Engines
 - Šimko, Gajdošech

Vývoj a nástroje

(detailnejšie)

■ natívne aplikácie

- Android



- Java
- Kotlin



- iOS



- Objective-C
- Swift



Priamy prístup k všetkým fičúrkam a komponentom OS, aj tým najnovším ...

■ hybridné aplikácie

- Cordova

- ionic



APACHE
CORDOVA™



Web-app na báze .html, .css, .js, ktoré púšťame v prostredí WebView=browser/wrapper bez browserových ovládačov

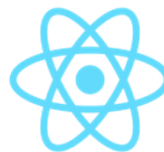
Vývoj a nástroje

(detailnejšie)

■ kompilované aplikácie

■ React Native

- JavaScript
- Facebook
- nekompiluje do natívneho kódu
- obmedzená množina widgets



■ Flutter (Dart framework od Googlu)

- Dart
- Google
- kompilované do ARM C++
- bohatšia množina widgets
- Material design (Quantum Paper)– Google 2014



■ NativeScript

- JavaScript



QUANTUM
and the building blocks of a unified interface



Aspekty programátora

Code sharing (write once, use everywhere)

- Cordova, ionic
- Flutter (Material Design)
- ReactNative
- Java, Swift



Knowledge sharing (learn once, use everywhere)

- Cordova, ionic, ReactNative (.js), Flutter (Dart)
- Java, Kotlin, Swift



Widget library

- Java, Swift, Cordova, ionic
- Flutter (Dart)
- ReactNative (.js)



<https://www.youtube.com/watch?v=bnYJRYFsrSw&feature=youtu.be>

Aspekty programátora

Eco-system (schopnosť nájsť riešenie/radu/blog na stackoverflow,..)

- Java, Swift, Kotlin 😊
- Cordova, ionic (.js) 😐
- ReactNative (.js, React) 😐
- Flutter (nové ale zlepšuje sa) 😞

Popularita

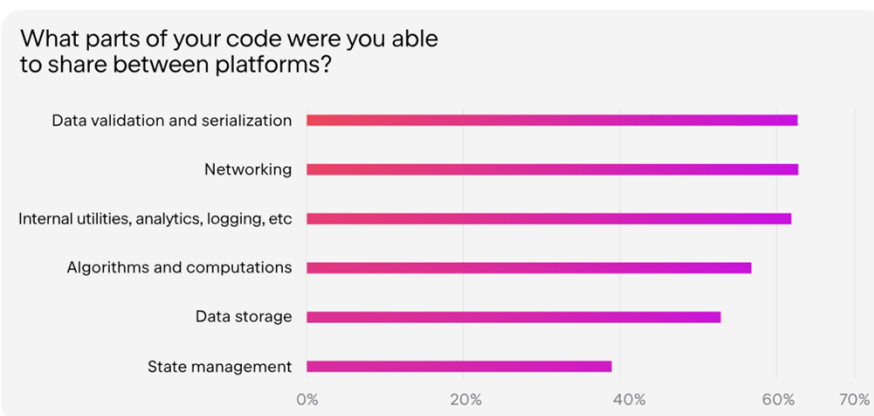
- Java, Swift, Kotlin 😊
- Cordova, ionic, ReactNative, Flutter (pushujú FB a Google) 😐

Kotlin Multiplatform

is in Beta

- *KMM* is an SDK designed to simplify creating cross-platform mobile applications (Android, iOS iPhone, watchOS, Windows, Linux)
- share common code between iOS and Android apps
- write platform-specific code
- platforms jvm, js, wasm
- xcode (mac) is necessary to build an iOS app
- KMM news 2021

https://www.youtube.com/watch?v=QJqLpTw3vwI&list=PLIFc5cFwUnmy_oVc9YQzjasSNoAk4hk_C&index=1&t=47s



Based on the results of the Kotlin Multiplatform Survey 2022

<https://kotlinlang.org/lp/multiplatform/>

Android Studio



How to use Android studio on low-end machines 4GB 8 GB of RAM

Java vs. Kotlin



tradičný VMA kurz postavený na Java už štvrtý rok beží v jazyku Kotlin 1.9

Dôvody:

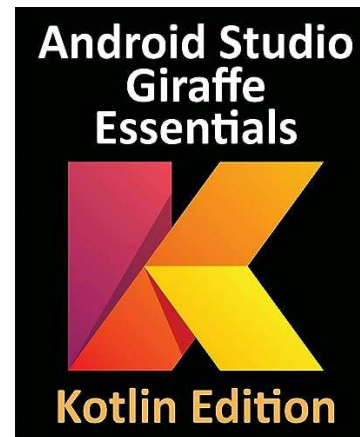
- ako iOS má svoj moderný jazyk Swift (3/4/5), aj Android má svoj Kotlin
 - Java je trochu *skamenelina* medzi modernými jazykmi (Swift, Kotlin, Scala)
 - Kotlin je Googlom oficiálne podporovaným vývojový nástroj pre Android
 - projekt Kotlin má už >12 rokov
 - kompiluje do JVM
 - funguje s Android Studiom
 - na JetBrains produkty ste si asi zvykli, a sú top
 - oboznámite sa s niektorými princípmi moderných jazykov
-
- Reference: <https://kotlinlang.org/docs/reference/>
 - Online: <https://play.kotlinlang.org/byExample/>

Android Studio Giraffe Kotlin Edition

- <https://www.amazon.com/Android-Studio-Giraffe-Essentials-Developing/dp/1951442776>
- <https://www.ebookfrenzy.com/errata/giraffekotlin.html>
- sources: <https://www.ebookfrenzy.com/retail/giraffekotlin/index.php>

Inštalácia Android Studio:

<https://developer.android.com/studio>



Predmet má cvičenie, ale aj tak:

- ozvite sa v prípade problémov inštalácie na platformy, napr. Linux, Mac.
- Dominika, Jožo, ja sa vám posnažíme problém vyriešiť

Inštalácia Android Studia:

- SDK Packages: Tools/SDK Manager tab SDK Platforms
- AVD: Android Virtual Device

Android Studio Giraffe Kotlin Edition

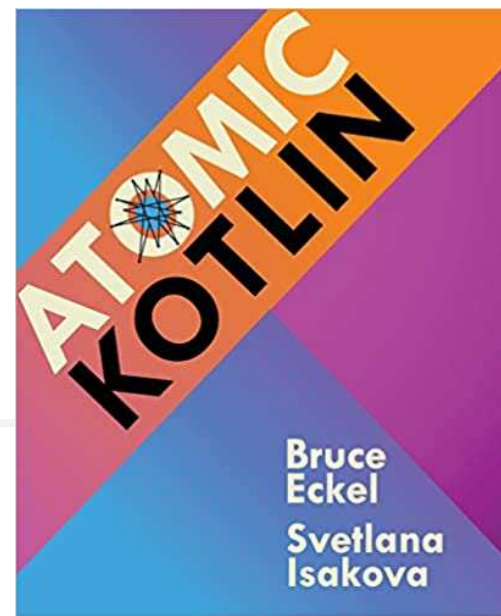
<https://www.amazon.com/Android-Studio-Giraffe-Essentials-Developing/dp/1951442776>

- <https://www.ebookfrenzy.com/errata/giraffekotlin.html>
- sources: <https://www.ebookfrenzy.com/retail/giraffekotlin/index.php>

2. Setting up an Android Studio Development Environment (mac/Windows/Linux)
3. Creating an Example Android App in AS
4. Creating an Android Virtual Device (AVD) in AS
5. Using and Configuring the Android Studio AVD Emulator
6. A tour of the Android Studio User Interface
7. Testing Android Studio App on a Physical Android Device
8. The Basics of the Android Studio Code Editor.
9. An Overview of the Android Architecture
10. The Anatomy of an Android App
11. An Introduction to Kotlin
12. Kotlin Data Types, Variables, and Nullability
13. Kotlin Operators and Expressions
14. Kotlin Control Flow
15. An Overview of Kotlin Functions and Lambdas
16. The Basics of Object Oriented Programming in Kotlin
17. An Introduction to Kotlin Inheritance and Subclassing
- ...
91. An Overview of Gradle in Android Studio



Atomic Kotlin



<https://www.amazon.com/Atomic-Kotlin-Bruce-Eckel/dp/0981872557>

Section I: Programming Basics

- Introduction
- Why Kotlin?
- Hello, World!
- var & val
- Data Types
- Functions
- if Expressions
- String Templates
- Number Types
- Booleans
- Repetition with while
- Looping & Ranges
- The in Keyword
- Expressions & Statements
- Summary 1

Section II: Introduction to Objects

- Objects Everywhere
- Creating Classes
- Properties
- Constructors
- Constraining Visibility
- Packages
- Testing
- Exceptions
- Lists
- Variable Argument Lists
- Sets
- Maps
- Property Accessors
- Summary 2

Section III: Usability

- Extension Functions
- Named & Default Arguments
- Overloading
- when Expressions
- Enumerations
- Data Classes
- Destructuring Declarations
- Nullable Types
- Safe Calls & the Elvis Operator
- Non-Null Assertions
- Extensions for Nullable Types
- Introduction to Generics
- Extension Properties
- break & continue

Section IV: Functional Programming

- Lambdas
- The Importance of Lambdas
- Operations on Collections
- Member References
- Higher-Order Functions
- Manipulating Lists
- Building Maps
- Sequences
- Local Functions
- Folding Lists
- Recursion

Section V: Object-Oriented Programming

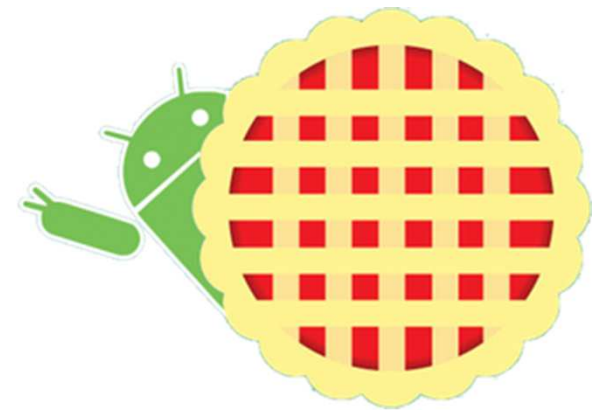
- Interfaces
- Complex Constructors
- Secondary Constructors
- Inheritance
- Base Class Initialization
- Abstract Classes
- Upcasting
- Polymorphism
- Composition
- Inheritance & Extensions
- Class Delegation
- Downcasting
- Sealed Classes



Why Teach Kotlin

- Kotlin is basically becoming the language of Android.
- Decrease in boilerplate helps us to quickly identify which fundamental Android concepts students are missing.
- On numerous courses, where we proceed through Java to Kotlin, we are considering a Kotlin-first approach.
- Students are happy to have the chance to program in something they may have heard about.
- I think our students benefit in general from being exposed to a wide range of programming languages, and I think it is valuable for them to gain experience in using more modern languages alongside the more traditional ones like Java and C++.
- My Kotlin students in fact understand OO concepts better than my Java students do.
- One of Kotlin's advantages is a good combination of strong typing and nullability.

Android a Google



2005 [Google](#) acquired Android Inc. with Rubin, Miner et al.

- 2007 [Open Handset Alliance](#), a consortium
 - device manufacturers: [HTC](#), [Sony](#) and [Samsung](#),
 - wireless carriers: [T-Mobile](#), ...
 - chipset makers: [Qualcomm](#), [Texas Instruments](#),includes Google with a goal to develop open standards for mobile devices
- major release named in alphabetical order after a dessert or sugary treat
 - 2.3 [Gingerbread](#)
 - 4.3 *Jelly Bean*, July, 2012,
 - 4.4 [KitKat](#), announced, October, 2013,
 - 5.1 [Lollipop](#), November, 2014,
 - 6.0 [Marshmallow](#), October, 2015,
 - 7.0 [Nougat](#), August, 2016.
 - 8.0 [Oreo](#), August, 2017,
 - 9.0 [Pie](#), August, 2018,
 - 10.0 [Android 10](#), September 2019
 - 11.0 [Android 11](#), tba

Version history by API level

- 2.1 Android 1.0 (API 1)
- 2.2 Android 1.1 (API 2)
- 2.3 Android 1.5 Cupcake (API 3)
- 2.4 Android 1.6 Donut (API 4)
- 2.5 Android 2.0 Eclair (API 5)
- 2.6 Android 2.2 Froyo (API 8)
- 2.7 Android 2.3 Gingerbread (API 9)
- 2.8 Android 3.0 Honeycomb (API 11)
- 2.9 Android 4.0 Ice Cream Sandwich (API 14)
- 2.10 Android 4.1 Jelly Bean (API 16)
- 2.11 Android 4.4 KitKat (API 19)
- 2.12 Android 5.0 Lollipop (API 21)
- 2.13 Android 6.0 Marshmallow (API 23)
- 2.14 Android 7.0 Nougat (API 24)
- 2.15 Android 8.0 Oreo (API 26)
- 2.16 Android 9 Pie (API 28)
- 2.17 Android 10 (API 29)
- 2.18 Android 11 (API 30)

API Levels

Version	SDK / API level	Version code	Codename	Cumulative usage ¹	Year
Android 14 DEV	Level 34	UPSIDE_DOWN_CAKE	Upside Down Cake	—	TBD
Android 13	Level 33	TIRAMISU	Tiramisu ²	30.33%	2022
	▪ targetSdk will need to be 33+ for new apps and app updates by August 2023.				
Android 12	Level 32 Android 12L	S_V2	Snow Cone ²	50.91%	2021
	Level 31 Android 12	S			
	▪ targetSdk must be 31+ for new apps and app updates.				
Android 11	Level 30	R	Red Velvet Cake ²	70.89%	2020
Android 10	Level 29	Q	Quince Tart ²	80.16%	2019
Android 9	Level 28	P	Pie	87.71%	2018
Android 8	Level 27 Android 8.1	O_MR1	Oreo	90.49%	2017
	Level 26 Android 8.0	O		93.7%	
Android 7	Level 25 Android 7.1	N_MR1	Nougat	94.31%	2016
	Level 24 Android 7.0	N		96.19%	
Android 6	Level 23	M	Marshmallow	97.83%	2015

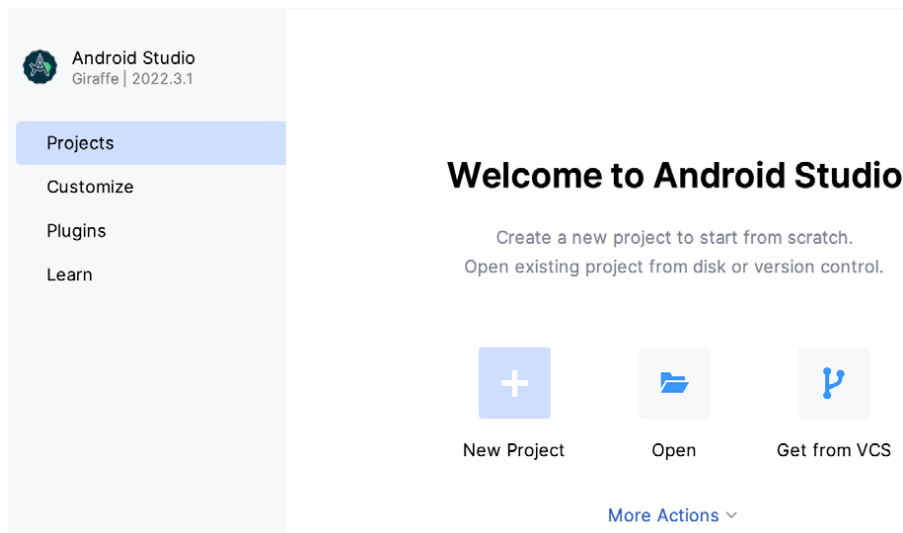
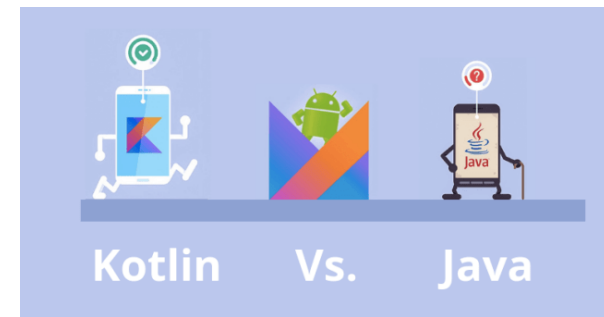
2. Setting up an Android Studio Development Environment

Inštalácia

System requirements

- Windows 8/10/11 64-bit
- macOS 10.14 or later running on Intel or Apple silicon
- Chrome OS device with Intel i5 or higher
- Linux systems with version 2.31 or later of the GNU C Library (glibc)
- **Minimum of 8GB of RAM**
- **Approximately 8GB of available disk space**
- 1280 x 800 minimum screen resolution

<https://developer.android.com/studio/index.html>



2. Setting up an Android Studio Development Environment

Android SDK Packages

Tools/SDK Manager tab SDK Platforms - API 33



Settings

Appearance & Behavior

Keymap

Editor

Plugins

Version Control

Build, Execution, Deployment

Languages & Frameworks

C/C++

Schemas and DTDs

Android SDK

GitHub Copilot

Kotlin

Markdown

Template Data Languages

Tools

Advanced Settings

Kotlin Compiler

Experimental

Languages & Frameworks > Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\borovan\AppData\Local\Android\Sdk

SDK Platforms

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Size	Status
<input type="checkbox"/> Android UpsideDownCakePrivacySandbox Preview			
<input type="checkbox"/> Android SDK Platform UpsideDownCakePrivacySandbox	UpsideDownCakePrivacySandbox		
<input type="checkbox"/> Google Play Intel x86_64 Atom System Image	UpsideDownCakePrivacySandbox		
<input checked="" type="checkbox"/> Android API 34			
<input type="checkbox"/> Android TiramisuPrivacySandbox Preview			
<input type="checkbox"/> Android SDK Platform TiramisuPrivacySandbox	TiramisuPrivacySandbox	9	Not installed
<input type="checkbox"/> Google Play Intel x86_64 Atom System Image	TiramisuPrivacySandbox	9	Not installed
<input checked="" type="checkbox"/> Android 13.0 ("Tiramisu")			
<input checked="" type="checkbox"/> Android SDK Platform 33	33	3	Installed
<input checked="" type="checkbox"/> Sources for Android 33	33	1	Not installed
<input type="checkbox"/> Android TV ARM 64 v8a System Image	33	5	Not installed
<input type="checkbox"/> Android TV Intel x86 Atom System Image	33	5	Not installed
<input type="checkbox"/> Google TV ARM 64 v8a System Image	33	5	Not installed
<input type="checkbox"/> Google TV Intel x86 Atom System Image	33	5	Not installed
<input type="checkbox"/> Google APIs ARM 64 v8a System Image	33	13	Not installed
<input checked="" type="checkbox"/> Google APIs Intel x86 Atom_64 System Image	33	7	Update Available: ...
<input checked="" type="checkbox"/> Google Play Intel x86_64 Atom System Image	33	7	Not installed

☒ Hide Obsolete Packages ☒ Show Package Details

OK Cancel Apply

Confirm Change

The following components will be installed:

- Google Play Intel x86_64 Atom System Image API 33 (revision 7)
- Sources for Android 33 (revision 1)

Disk usage:

- Estimated download size: 1,4 GB
- Estimated disk space to be additionally occupied on SDK partition after installation: 5,7 GB
- Currently available disk space in SDK root (C:\Users\borovan\AppData\Local\Android\Sdk): 24,0 GB

Cancel OK

2. Setting up an Android Studio Development Environment

Android SDK Packages

Tools/SDK Manager tab SDK Tools



Settings

Search:

Appearance & Behavior

Keymap

Editor

Plugins

Version Control

Build, Execution, Deployment

Languages & Frameworks

C/C++

Schemas and DTDs

Android SDK

GitHub Copilot

Kotlin

Markdown

Template Data Languages

Tools

Advanced Settings

Kotlin Compiler

Experimental

Languages & Frameworks > Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: [Edit](#) [Optimize disk space](#)

SDK Platforms SDK Tools SDK Update Sites

Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
<input checked="" type="checkbox"/> Android SDK Build-Tools 34		Update Available: 34.0.0
<input checked="" type="checkbox"/> GPU Debugging tools		Installed
<input type="checkbox"/> NDK (Side by side)		Not Installed
<input checked="" type="checkbox"/> Android SDK Command-line Tools (latest)		Installed
<input type="checkbox"/> CMake		Not Installed
<input type="checkbox"/> Android Auto API Simulators	1	Not installed
<input type="checkbox"/> Android Auto Desktop Head Unit Emulator	2.0	Not installed
<input checked="" type="checkbox"/> Android Emulator	32.1.14	Installed
<input type="checkbox"/> Android Emulator hypervisor driver (installer)	2.0.0	Not installed
<input checked="" type="checkbox"/> Android SDK Platform-Tools	34.0.4	Installed
<input checked="" type="checkbox"/> Android SDK Tools	26.1.1	Installed
<input type="checkbox"/> Google Play APK Expansion library	1	Not installed
<input checked="" type="checkbox"/> Google Play Instant Development SDK	1.9.0	Installed
<input type="checkbox"/> Google Play Licensing Library	1	Not installed
<input checked="" type="checkbox"/> Google Play services	49	Installed
<input checked="" type="checkbox"/> Google USB Driver	13	Installed
<input type="checkbox"/> Google Web Driver	2	Not installed
<input checked="" type="checkbox"/> Intel x86 Emulator Accelerator (HAXM installer)	7.6.5	Installed
<input type="checkbox"/> Layout Inspector image server for API 29-30	6	Not installed
<input checked="" type="checkbox"/> Layout Inspector image server for API 31-34	3	Installed
<input type="checkbox"/> Layout Inspector image server for API S	3	Not installed

☒ Hide Obsolete Packages ☐ Show Package Details

OK Cancel Apply

Android Virtual Device

Tools/AVD manager

Nakonfigurujte si AVD zodpovedajúci vášmu zariadeniu

alebo si vyberte zo zoznamu

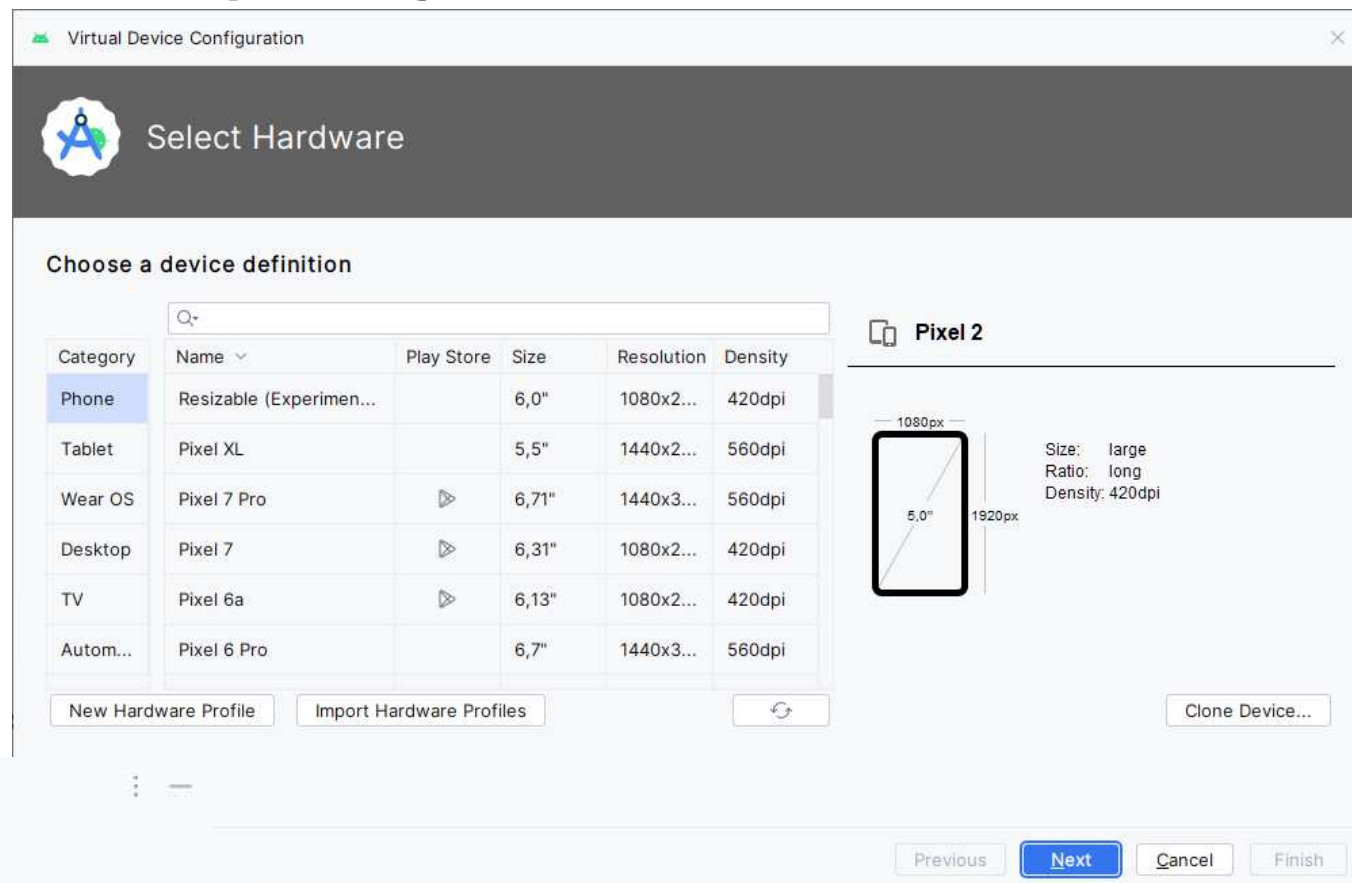
predvolených,

- Create Device
- modifikujte

nastavenia

podľa vášho

zariadenia

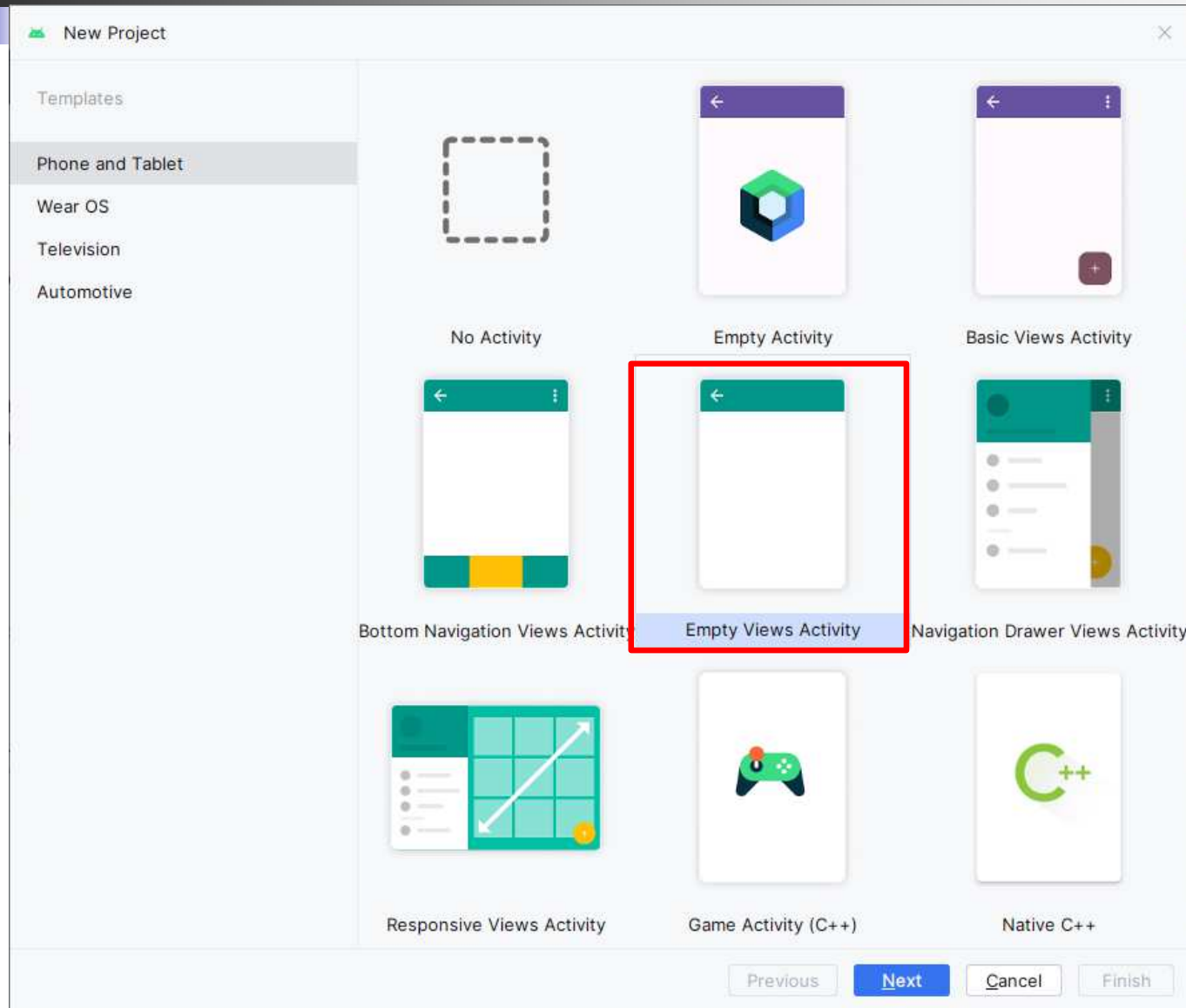


Chapter 3

3. Creating an Example Android App in Android Studio

Nový projekt

(File/New/New Android Project)



Nový projekt

(File/New/New Android Project)

New Project

Empty Views Activity

Creates a new empty activity

Name: EmptyApplication2023

Package name: com.example.emptyapplication2023

Save location: D:\borovan\workspace_AndroidStudio\EmptyApplication2023

Language: Kotlin

Minimum SDK: API 24 ("Nougat"; Android 7.0)

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

Pre

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.6%
4.2 Jelly Bean	17	98.1%
4.3 Jelly Bean	18	95.9%
4.4 KitKat	19	95.3%
5.0 Lollipop	21	85.0%
5.1 Lollipop	22	80.2%
6.0 Marshmallow	23	62.6%
7.0 Nougat	24	37.1%
7.1 Nougat	25	14.2%
8.0 Oreo	26	6.0%
8.1 Oreo	27	1.1%

Submitovanie riešení: Android SDK 13 (API 33), (compileSdkVersion 33, buildToolsVersion "33.*"), a min.požadované SDK (minSdkVersion 23)

API 34

Your app will run on < 1% of devices.
[Help me choose](#)

Nový projekt

(File/New/New Android Project)

```
plugins {  
    id("com.android.application")  
    id("org.jetbrains.kotlin.android")  
}
```

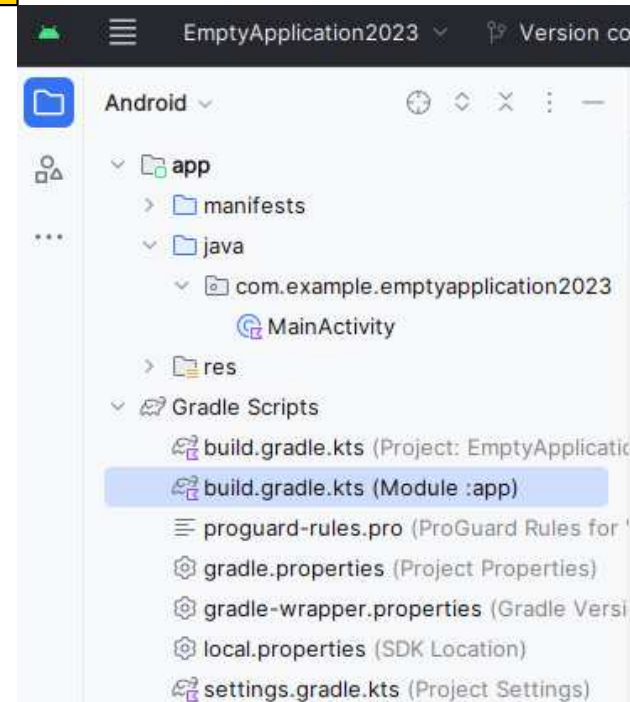
Submitovanie riešení: Android SDK 13 (API 33),
(compileSdkVersion 33, buildToolsVersion "33.*"),
a min.požadované SDK (minSdkVersion 23)

```
android {  
    namespace = "com.example.emptyapplication2023"  
    compileSdk = 33  
  
    defaultConfig {  
        applicationId = "com.example.emptyapplication2023"  
        minSdk = 24  
        targetSdk = 33  
        versionCode = 1  
        versionName = "1.0"  
    }  
}
```

acts be like...



ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.6%
4.2 Jelly Bean	17	98.1%
4.3 Jelly Bean	18	95.9%
4.4 KitKat	19	95.3%
5.0 Lollipop	21	85.0%
5.1 Lollipop	22	80.2%
6.0 Marshmallow	23	62.6%
7.0 Nougat	24	37.1%
7.1 Nougat	25	14.2%
8.0 Oreo	26	6.0%
8.1 Oreo	27	1.1%



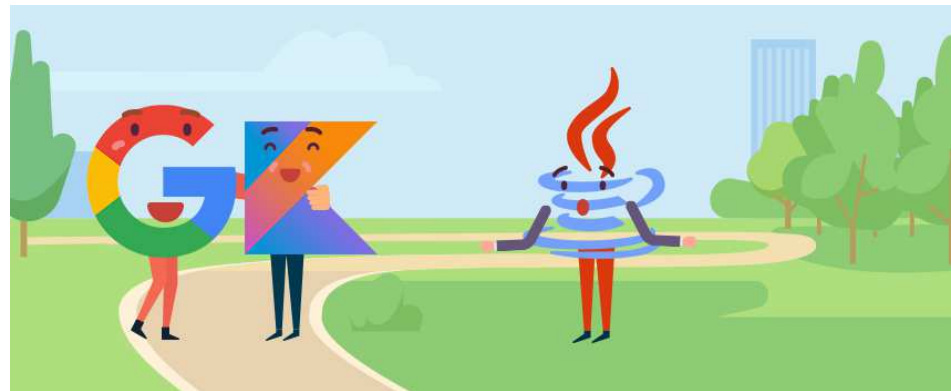
Nový projekt (Empty views activity)



The screenshot displays the Android Studio IDE with a new project named "EmptyApplication2023". The interface is configured for a "Pixel 6 API 33" device. The Project Explorer on the left shows the project structure, including the "activity_main.xml" file. The Palette on the top left lists various widgets, with "Ab TextView" selected. The central design view shows two preview windows: a light pink background on the left and a dark blue background on the right. A text view with the text "Hello World!" is positioned in the center of the light pink preview. The Properties panel on the right shows the attributes for the selected "Ab TextView" widget, including "id", "Declared Attributes", "Layout", "Constraint Widget", "Constraints (4)", and "Transforms". The "Constraints (4)" section shows the widget is constrained to the parent layout with "wrap_content" for both width and height. The "Transforms" section shows the widget's rotation and scale properties.

androidx.constraintlayout.widget.ConstraintLayout > TextView

Pýtajte sa kým nedostanete



EmptyApplication2023 Version control Pixel 6 API 33 app

Android

- app
 - manifests
 - java
 - com.example.emptyapplication2023
 - MainActivity
 - res
 - drawable
 - layout
 - activity_main.xml
 - mipmap
 - values
 - colors.xml
 - strings.xml
 - themes (2)
 - xml
 - res (generated)
 - Gradle Scripts
 - build.gradle.kts (Project: EmptyApplication2023)
 - build.gradle.kts (Module :app)
 - proguard-rules.pro (ProGuard Rules for ".app")
 - gradle.properties (Project Properties)
 - gradle-wrapper.properties (Gradle Version)
 - local.properties (SDK Location)
 - settings.gradle.kts (Project Settings)

activity_main.xml MainActivity.kt build.gradle.kts (:app)

Palette

- Common
 - Ab TextView...
 - Button
 - Image...
 - Recyc...
 - Fragm...
 - Scroll...
 - Switch
- Text
- Buttons
- Widgets
- Layouts
- Containers
- Helpers
- Google
- Legacy

Component Tree

- ConstraintLayout
 - Ab TextView "Hello World"

Attributes

Ab TextView <unnamed>

Declared Attributes

Layout

Constraint Widget

Constraints (4)

- layout_width wrap_content
- layout_height wrap_content
- visibility
- visibility

Transforms

Running Devices Pixel 6 API 33

7:57 Hello World!

Run

2023-08-28 09:57:15: Launching app on 'Pixel 6 API 33.'

```
$ adb shell am start -n "com.example.emptyapplication2023/com.example.emptyapplication2023.MainActivity" -a android.intent.action.MAIN -c android.int
```

Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.emptyapplication2023/.MainActivity }

[Open logcat panel for emulator Pixel 6 API 33](#)

Connected to process 4535 on device 'Pixel_6_API_33 [emulator-5554]'.

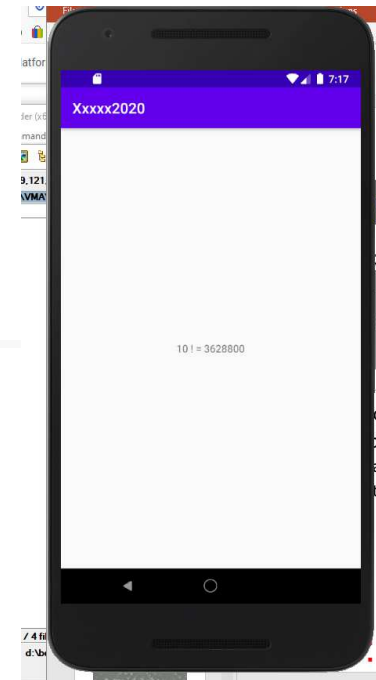
EmptyApplication2023 > app > src > main > res > layout > activity_main.xml

8:1 LF UTF-8 4 spaces

Ako si skúšať Kotlin v AS

(kým sa nedozvieme viac)

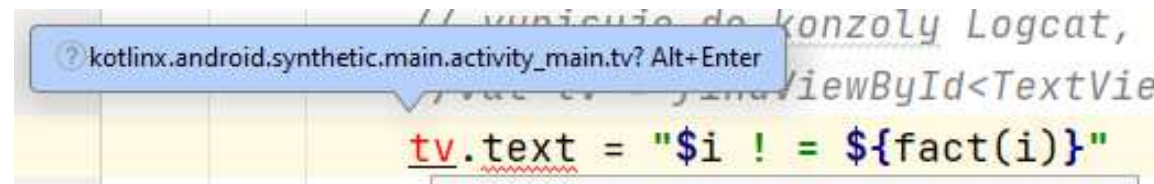
```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        //println(fact(10))  
        for (i in 0..10) {  
            Log.d("TAG", "$i ! = ${fact(i)}")  
            // vypisuje do konzoly Logcat, použite filter s "TAG"  
            val tv = findViewById<TextView>(R.id.tv)  
            tv.text = "$i ! = ${fact(i)}"  
            // vypise do View komponentu, ktorý je v Activite  
            Toast.makeText(this, "$i ! = ${fact(i)}",  
                Toast.LENGTH_SHORT).show()  
            // Toast alias Notifier (MITI)  
        }  
    }  
}  
fun fact(n : Int) : Int = if (n == 0) 1 else n * fact(n-1)  
}
```



Integrovanie Android Extensions

```
plugins {  
    id 'com.android.application'  
    id 'kotlin-android'  
    id 'kotlin-android-extensions'  
}  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.widget.TextView  
import android.widget.Toast  
import kotlinx.android.synthetic.main.activity_main.*
```

```
for (i in 0..10) {  
    Log.d("TAG", "$i ! = ${fact(i)}")  
    // vypisuje do konzoly Logcat, pouzite filter s "TAG"  
    val tv = findViewById<TextView>(R.id.tv)  
    tv.text = "$i ! = ${fact(i)}"  
    // vypise do View komponentu, ktory je v Aktivite  
    Toast.makeText(this, "$i ! = ${fact(i)}",  
        Toast.LENGTH_SHORT).show()  
    // Toast alias Notifier (MITI)  
}
```





Break point

(štruktúrou projektu pokračujeme na budúce)

- Switch to kotlin intro