
Aplikacje Mobilne – programowanie w środowisku Android Studio
rezultat projektu

Nowe Technologie wsparciem dla edukacji

nr umowy - POWERSE-2018-1-PL01-KA101-049291

realizowanego ze środków POWER na zasadach programu Erasmus+

sektor Edukacja szkolna

„Ponadnarodowa mobilność kadry edukacji szkolnej”

Co nam będzie potrzebne do dalszej pracy?

<https://www.geeksforgeeks.org/differences-jdk-jre-jvm/>

- Java Development Kit (JDK).
- Rozbudowany edytor tekstowy, np...
- ...Sublime Text
- Android Studio, czyli IntelliJ IDEA, tyle że ubrane na zielono.

Instalacja Java Development Kit (JDK)

1. Sprawdzamy, czy na naszym komputerze jest zainstalowany JDK.
W tym celu w terminali cmd wpisujemy: `javac -version`.
A może wersja jest zainstalowana, ale nie dodana do ścieżki:
Sprawdzamy: `where javac.exe`

```
C:\Users\Jola>javac -version
Nazwa 'javac' nie jest rozpoznawana jako polecenie wewnętrzne lub zewnętrzne,
program wykonywalny lub plik wsadowy.

C:\Users\Jola>where javac.exe
INFORMACJE: Nie można odnaleźć plików danych wzorców.

C:\Users\Jola>
```

2. Jeżeli nie ma, to szukamy plików instalacyjnych:

download jdk



Wszystko

Filmy

Wiadomości

Grafika

Książki

Więcej

Ustawienia

Narzędzia

Okolo 21 900 000 wyników (0,37 s)

Java SE - Downloads | Oracle Technology Network | Oracle

<https://www.oracle.com/.../java/javase/downloads/index.html> ▼ Tłumaczenie strony

Java SE downloads including: **Java** Development Kit (JDK), Server **Java** Runtime Environment (Server JRE), and **Java** Runtime Environment (JRE).

Java SE 11.0.2(LTS)

Java SE 11.0.2 is the latest release for the Java SE 11 Platforms

[Learn more](#) ➔

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle JDK License](#)
- [Java SE Licensing Information User Manual](#)
 - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme](#)

Oracle JDK









DOWNLOAD ⬇



Java SE Development Kit 11.0.2

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.

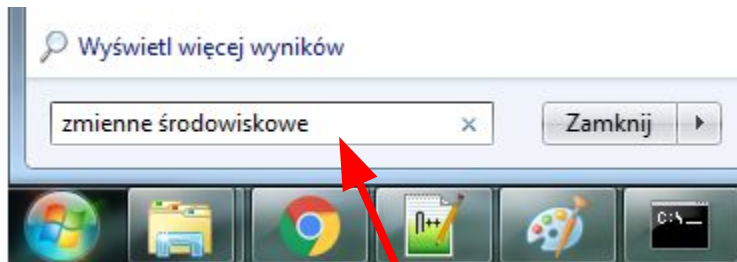
Product / File Description	File Size	Download
Linux	147.28 MB	 jdk-11.0.2_linux-x64_bin.deb
Linux	154.01 MB	 jdk-11.0.2_linux-x64_bin.rpm
Linux	171.32 MB	 jdk-11.0.2_linux-x64_bin.tar.gz
macOS	166.13 MB	 jdk-11.0.2_osx-x64_bin.dmg
macOS	166.49 MB	 jdk-11.0.2_osx-x64_bin.tar.gz
Solaris SPARC	186.78 MB	 jdk-11.0.2_solaris-sparcv9_bin.tar.gz
Windows	150.94 MB	 jdk-11.0.2_windows-x64_bin.exe
Windows	170.96 MB	 jdk-11.0.2_windows-x64_bin.zip

▶ Komputer ▶ Dysk lokalny (C:) ▶ Program Files ▶ Java ▶ jdk-11.0.2 ▶ bin ▶				
Otwórz Nagraj Nowy folder				
miejsca	Nazwa	Data modyfikacji	Typ	Rozmiar
	javac	2019-02-12 11:45	Aplikacja	21 KB
	javadoc	2019-02-12 11:45	Aplikacja	21 KB

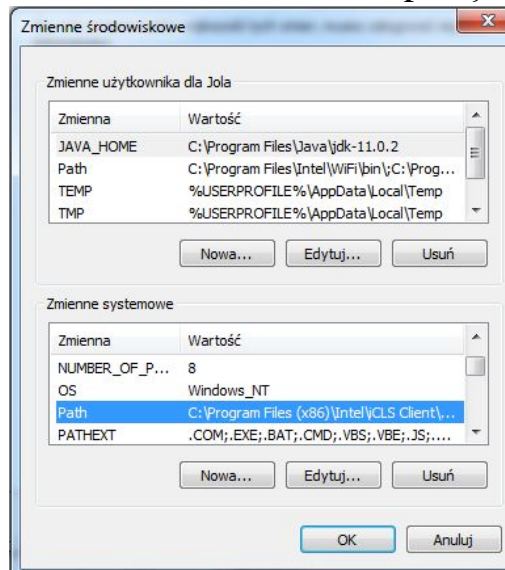
Teraz musimy dodać do ścieżki systemowej path i do zmiennej użytkownika, ścieżkę do kompilatora javac

Kopiuujemy ścieżkę do katalogu kompilatora

C:\Program Files\Java\jdk-11.0.2\bin



Wpisujemy:

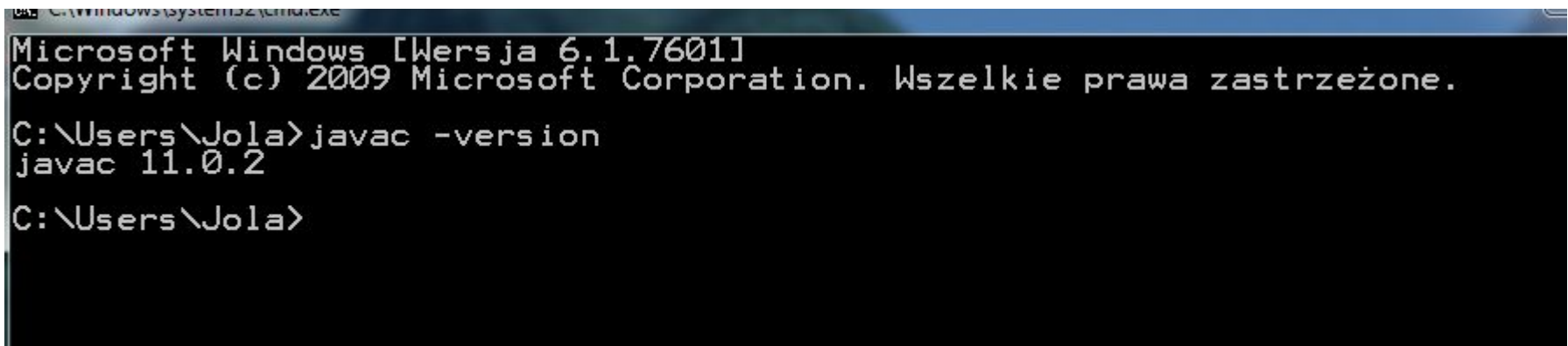


Dla zmiennej systemu szukamy path(jeżeli nie ma, to należy utworzyć) edytuj i na koniec aktualnego wpisu, po średniku, wklej ścieżkę

C:\Program Files\Java\jdk-11.0.2\bin

Dla zmiennych środowiskowych konta szukamy JAVA_HOME, jeżeli nie ma, to tworzymy i wpisujemy C:\Program Files\Java\jdk-11.0.2\bin

Otwieramy terminal i sprawdzamy, czy wszystko jest dobrze zainstalowane.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Jola>javac -version
javac 11.0.2

C:\Users\Jola>
```


Instalacja Android Studio

android studio download



Wszystko

Filmy

Grafika

Wiadomości

Książki

Więcej

Ustawienia

Narzę

Okolo 805 000 000 wyników (0,36 s)

Download Android Studio and SDK tools | Android Developers

<https://developer.android.com/studio/> ▼ Tłumaczenie strony

If you do not need **Android Studio**, you can **download** the basic Android command line tools below. You can use the included sdkmanager to **download** other ...

[Install Android Studio](#) · [Android SDK Manager](#) · [Android Studio release notes](#)



Android Studio

DOWNLOAD

WHAT'S NEW

USER GUIDE

PREVIEW

androidstudio

Android Studio provides the fastest tools for building apps on every type of Android device.

DOWNLOAD ANDROID STUDIO

3.3.1 for Windows 64-bit (948 MB)

DOWNLOAD OPTIONS

RELEASE NOTES



Install Type

Choose the type of setup you want for Android Studio:

☐ Standard

Android Studio will be installed with the most common settings and options.
Recommended for most users.

☒ Custom

You can customize installation settings and components installed.

Previ...

Next

Cancel

Finish



SDK Components Setup

Check the components you want to update/install. Click Next to continue.

- ☒ Android SDK – (416 MB)
- ☒ Android SDK Platform
- ☒ API 28: Android 9.0 (Pie) – (168 MB)
- ☒ Performance (Intel® HAXM) – (2,62 MB)
- ☒ Android Virtual Device – (881 MB)

Enables a hardware-assisted virtualization engine (hypervisor) to speed up Android app emulation on your development computer. (Recommended)

**Pozwala na przyspieszenie
pracy emulatora**

Android SDK Location:

C:\Users\Jola\AppData\Local\Android\Sdk

Total download size: 1,43 GB

Disk space available on drive : 417 GB

Target folder is neither empty nor does it point to an existing SDK installation.

Previous

Next

Cancel

Finish



Downloading Components

Unzipping...

tools/lib/monitor-x86_64/plugins/org.eclipse.ui.intro.universal_3.2...rces48.gif

Preparing "Install Android SDK Tools (revision: 26.1.1)".

Downloading <https://dl.google.com/android/repository/sdk-tools-windows-4333796.zip>



Kliknij :Show Details - Informacje o zainstalowanych elementach. Gdyby był błąd, należy skopiować informacje, wkleić do Google i w internecie szukać rozwiązania

Previous

Next

Cancel

Finish



Android Studio

Version 3.3.1

- + Start a new Android Studio project
- Open an existing Android Studio project
- Check out project from Version Control ▾
- Profile or debug APK
- Import project (Gradle, Eclipse ADT, etc.)
- Import an Android code sample

⚙ Configure ▾ Get Help ▾

Choose your project

Phone and Tablet

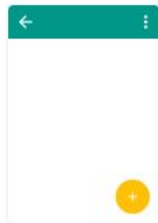
Wear OS

TV

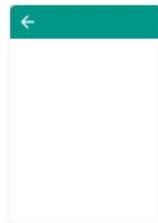
Android Auto

Android Things

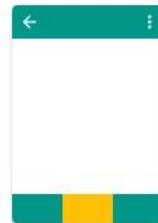
Add No Activity



Basic Activity



Empty Activity



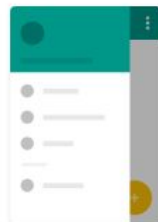
Bottom Navigation Activity



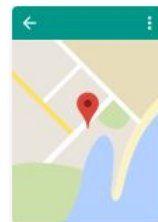
Fullscreen Activity



Master/Detail Flow



Navigation Drawer Activity



Google Maps Activity

Empty Activity

Creates a new empty activity

Previous


Next

Cancel

Finish

Create New Project

Configure your project



Empty Activity

Creates a new empty activity


Name
My Application

Package name
com.example.myapplication

Save location
C:\Users\Jola\AndroidStudioProjects\MyApplication

Language
Java

Minimum API level
API15: Android 4.0.3 (IceCreamSandwich)

 Your app will run on approximately 100% of devices.
[Help me choose](#)

☐ This project will support instant apps

☐ Use AndroidX artifacts

[Previous](#) [Next](#) [Cancel](#) [Finish](#)

Można wybrać wersję urządzenia,
na które programujemy

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%

Ice Cream Sandwich

Contacts Provider

Social APIs
User profile
Invite intent
Large photos

Calendar Provider

Calendar APIs
Event intents

Voicemail Provider

Add voicemails to the device

Multimedia

Media effects for images and videos
Remote control client
Improved media player

Camera

Face detection
Focus and metering areas
Continuous auto focus
Camera broadcast intents

Connectivity

Android Beam for NDEF push with NFC
Wi-Fi P2P connections
Bluetooth health profile
Network usage and controls

Accessibility

Explore-by-touch mode
Accessibility for views
Accessibility services
Improved text-to-speech engine support

User Interface

Spell checker services
Improved action bar
Grid layout
Texture view
Switch widget
Improved popup menus
System themes
Controls for system UI visibility
Hover event support
Hardware acceleration for all windows

Enterprise

VPN services
Device policies
Certificate management

Device Sensors

Improved sensors
Temperature sensor
Humidity sensor

<https://developer.android.com/about/versions/android-4.0.html>

OK

Cancel

MyApplication [C:\Users\Jola\AndroidStudioProjects\MyApplication] - ...app\src\main\java\com\example\myapplication\MainActivity.java [app] - Android Studio (Administrator)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

MyApplication > app > src > main > java > com > example > myapplication > MainActivity

Project

- Android
 - app
 - manifests
 - java
 - com.example.myapplication
 - MainActivity
 - com.example.myapplication (androidTest)
 - com.example.myapplication (test)
 - generatedJava
 - res
 - Gradle Scripts

activity_main.xml x MainActivity.java x

```
1 package com.example.myapplication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13
14 }
```

Assistant What's New

What's New in 3.3

This panel describes some of the new features and behavior changes included in this update.

To open this panel again later, select **Help > What's New in Android Studio** from the main menu.

[Read in a browser](#)

Navigation Editor

Build Sync

- Build: completed successfully at 2019-02-12 16:11
 - Run build C:\Users\Jola\AndroidStudioProjects\MyApplication
 - Load build 17 s 297 ms
 - Configure build 17 s 188 ms
 - Calculate task graph 5 ms
 - Run tasks 342 ms

16 s 715 ms

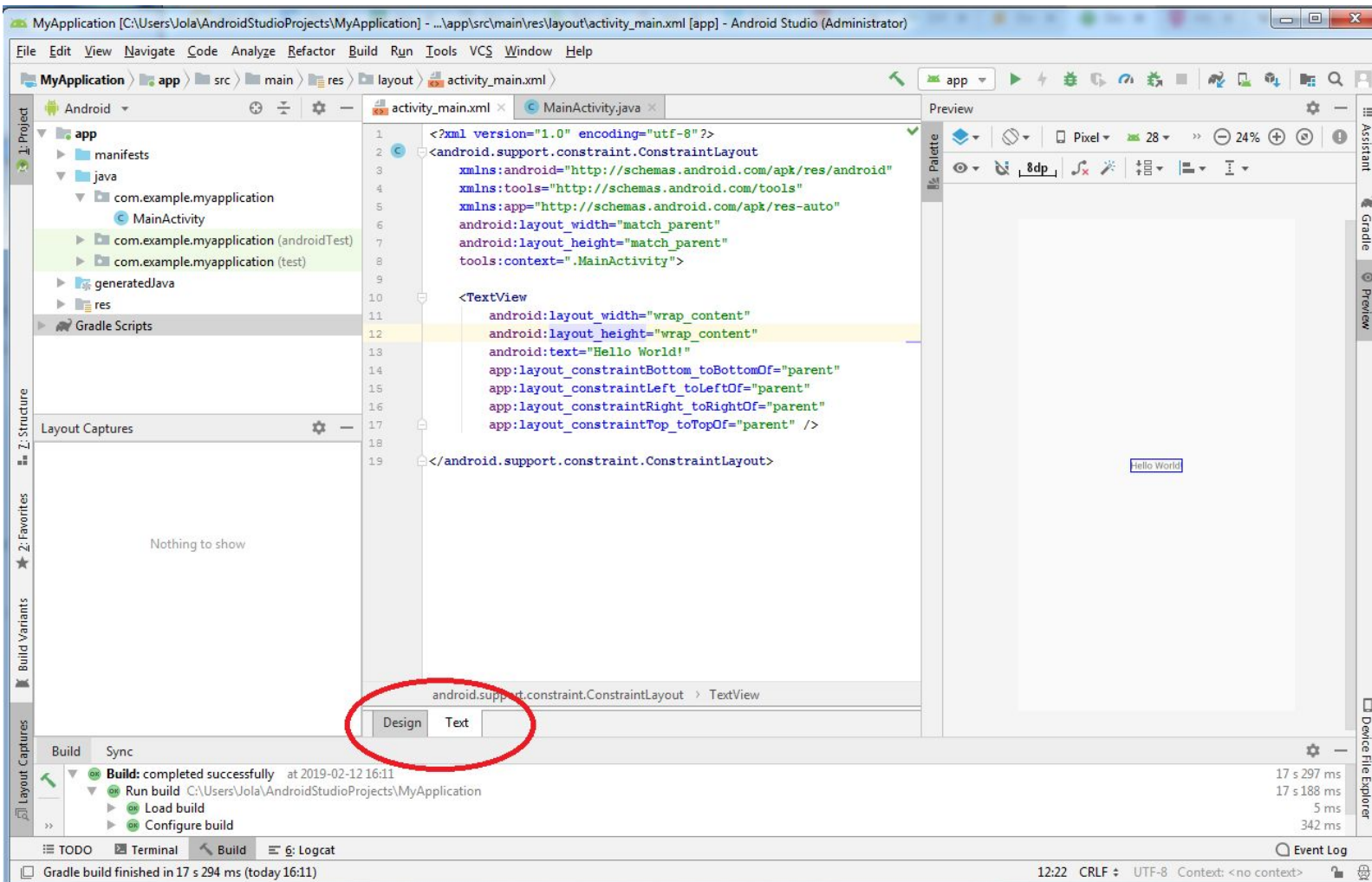
TODO Terminal Build Logcat Event Log

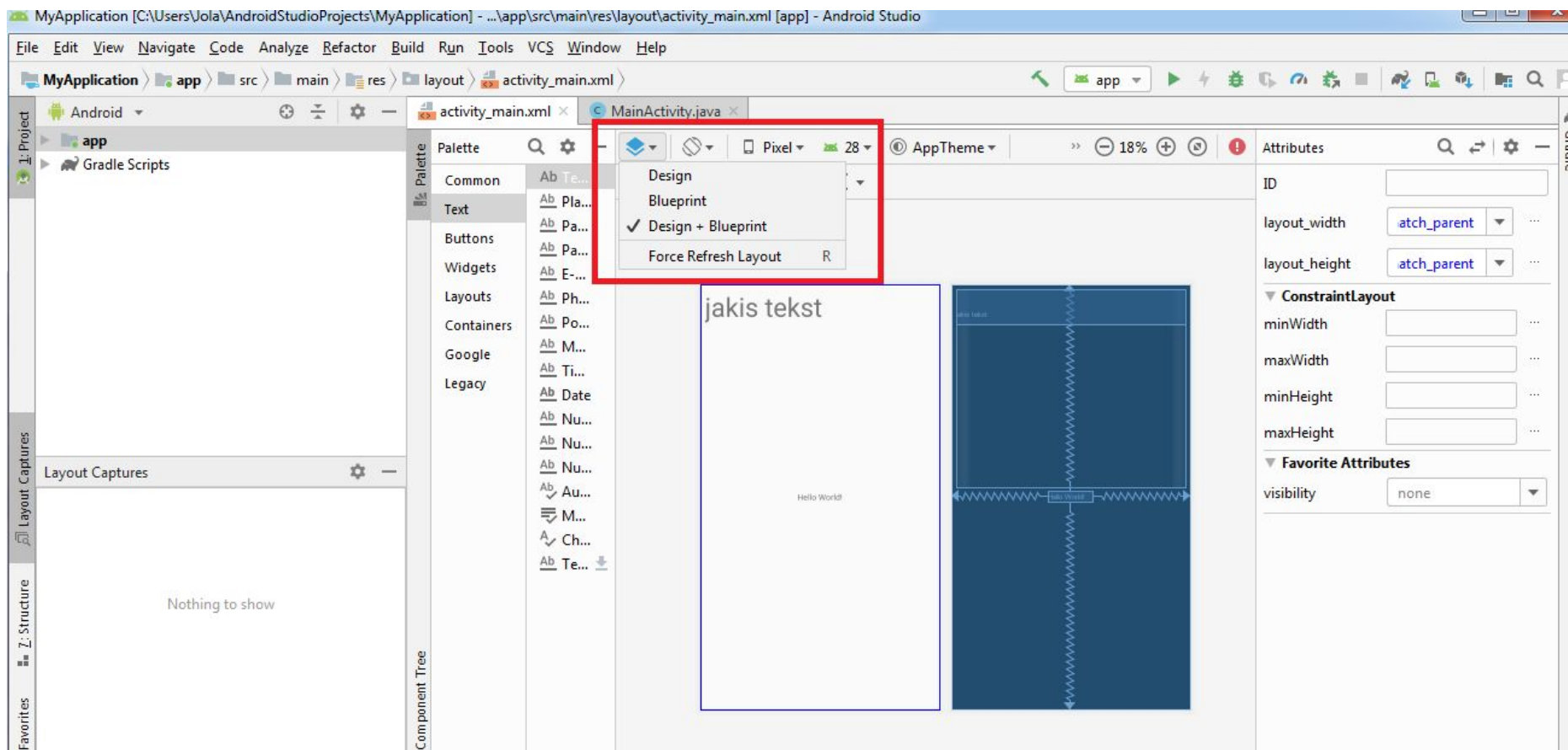
Gradle build finished in 17 s 294 ms (moments ago)

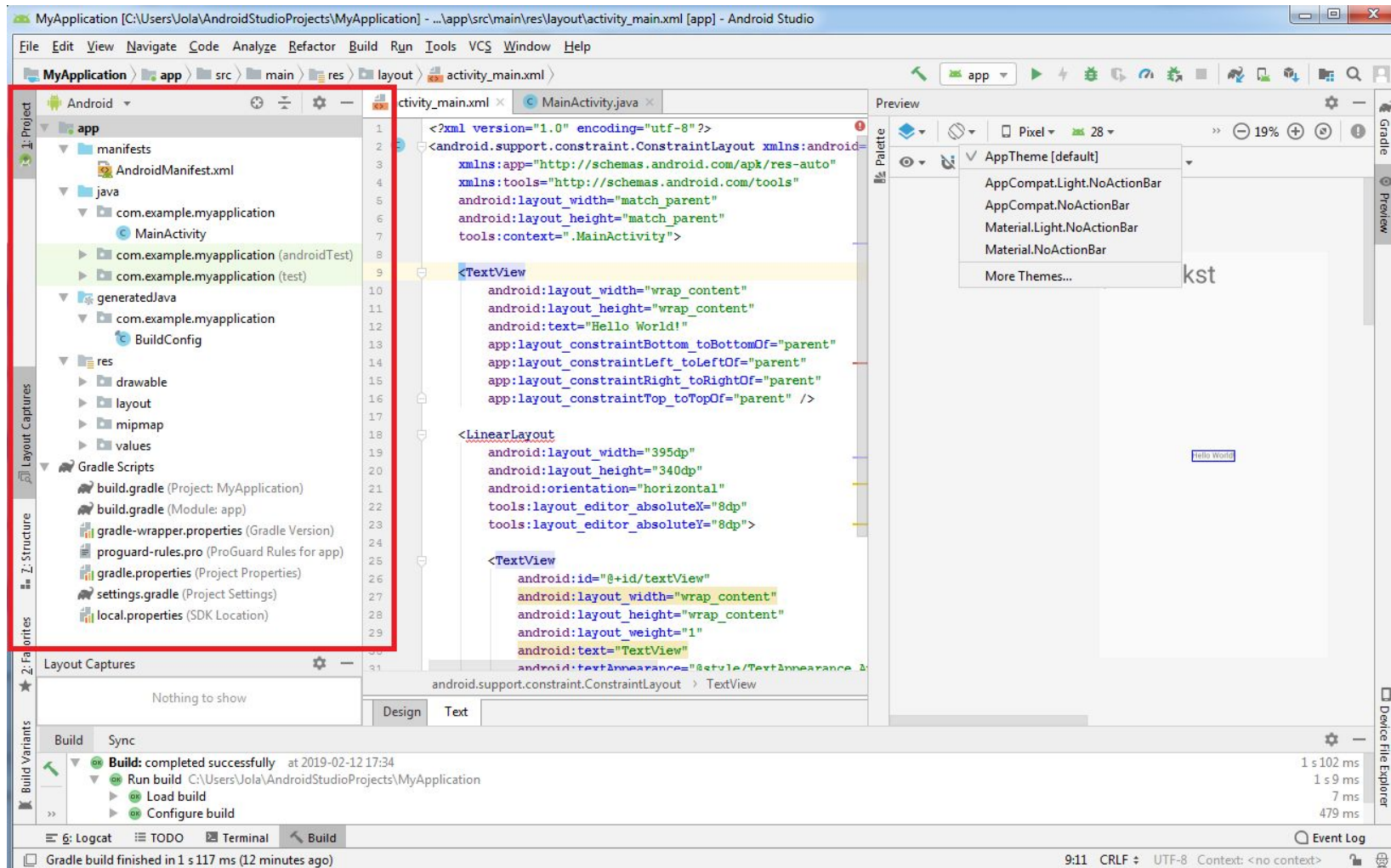
1:1 CRLF UTF-8 Context: <no context>

Gracie

Device File Explorer







Gradle

From Wikipedia, the free encyclopedia

Gradle is an open-source [build automation](#) system that builds upon the concepts of [Apache Ant](#) and [Apache Maven](#) and introduces a [Groovy-based domain-specific language](#) (DSL) instead of the [XML](#) form used by [Apache Maven](#) for declaring the project configuration.^[1] Gradle uses a [directed acyclic graph](#) ("DAG") to determine the order in which tasks can be run.

Gradle was designed for multi-project builds, which can grow to be quite large. It supports incremental builds by intelligently determining which parts of the build tree are up to date; any task dependent only on those parts does not need to be re-executed.

Android application package

From Wikipedia, the free encyclopedia

Android Package (APK)^[1] is the [package file format](#) used by the [Android](#) operating system for distribution and installation of [mobile apps](#) and [middleware](#).

APK files are analogous to other [software packages](#) such as [APPX](#) in [Microsoft Windows](#) or a [Debian package](#) in [Debian](#)-based operating system. To make an APK file, a program for Android is first compiled, and then all of its part are packaged into one container file. An APK file contains all of a program's code (such as [.dex](#) files), resources, assets, certificates, and [manifest file](#). As is the case with many file formats, APK files can have any name needed, provided that the file name ends in the [file extension](#) ".apk".^{[2][3][4][5]}

APK files are a type of [archive file](#), specifically in [zip format](#)-type packages, based on the [JAR file format](#), with `.apk` as the [filename extension](#). The [MIME type](#) associated with APK files is `application/vnd.android.package-archive`.^[6]

APK files can be installed on [Android](#)-powered devices just like installing software on a [PC](#). When a user downloads and installs an Android application, from either an official source (such as the [Google Play Store](#)), or from an unofficial site, they are installing an APK file on to their device. A user or developer can also install an APK file directly to a device (that is, not via download from the network) from a desktop computer, using a communication program such as [adb](#), or from within a file manager app in a process known as [sideloading](#). The installation of APK files downloaded outside the Google Play is disabled by default. Users can install unknown APK files by enabling "Unknown sources" from "Accounts and Security" in Settings.^[7]

APK



Filename extension

`.apk` , `.xapk`

Internet media type

`application/com.android.package-install` always

Type of format

Package format

Container for

Mobile apps

Extended from

JAR and ZIP

Podstawy konfiguracji

MyApplication [C:\Users\Jola\AndroidStudioProjects\MyApplication] - ...app\src\main\res\layout\activity_main.xml [app] - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

res layout activity_main.xml

Settings

Appearance & Behavior

Appearance

Menus and Toolbars

System Settings

File Colors

Scopes

Notifications

Quick Lists

Path Variables

Keymap

Editor

Plugins

Version Control

Build, Execution, Deployment

Languages & Frameworks

Tools

Kotlin Compiler

Experimental

Settings

Appearance & Behavior

Appearance

Menus and Toolbars

System Settings

File Colors

Scopes

Notifications

Quick Lists

Path Variables

Keymap

Editor

Plugins

Version Control

Build, Execution, Deployment

Languages & Frameworks

Tools

Kotlin Compiler

Experimental

Appearance > Appearance

UI Options

Theme: IntelliJ

☐ Adjust colors for red-green vision deficiency (protanopia, deuteranopia) [How it works](#)

☐ Override default fonts by (not recommended):

Name: Segoe UI Size: 12

☒ Cyclic scrolling in list

☒ Show icons in quick navigation ☐ Show Flags for Languages

☐ Automatically position mouse cursor on default button

☒ Hide navigation popups on focus loss

☐ Drag-n-Drop with ALT pressed only

Background Image...

Tooltip initial delay (ms): 0 1200

Antialiasing

IDE: Subpixel Editor: Subpixel

Window Options

☒ Animate windows

☒ Show memory indicator

☐ Disable mnemonics in menu

☐ Disable mnemonics in controls

☒ Show tool window bars

☒ Show tool window numbers

☒ Allow merging buttons on dialogs

☐ Small labels in editor tabs

OK Cancel Apply Help

Build Variants

Build Sync

Build: completed successfully at 2019-02-12 17:34

Run build C:\Users\Jola\AndroidStudioProjects\MyApplication

Load build

Configure build

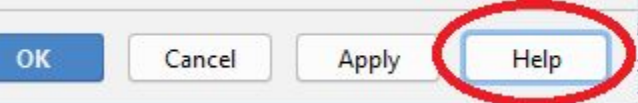
Logcat TODO Terminal Build

Edit application settings

Gradle build finished

9:11 CRLF UTF-8 Context: <no context>

Zwalnianie
pamięci



← → ↻ <https://www.jetbrains.com/help/idea/2018.2/settings-appearance.html>

IntelliJ IDEA 2018.2 Help

[Get IntelliJ IDEA](#)

Shortcuts: [Windows/Linux](#)

- ✓ Meet IntelliJ IDEA
- ✓ Configuring the IDE
- ✓ Configuring projects
- ✓ Working with source code
- ✓ Building, running, testing and deploying applications
- ✓ Analyzing applications
- ✓ Version control with IntelliJ IDEA
 - Managing tasks and contexts
- ✓ Migration Guides
- ✓ Language and framework-specific guidelines
- ✓ Tutorials
- ^ Reference
 - ✓ Basic Concepts
 - ✓ New Project Wizard
 - ✓ New Module Wizard
 - ✓ Tool windows reference
 - ✓ Import Project or Module Wizard
 - ✓ Dialogs
 - ^ Settings / Preferences Dialog
 - ^ Appearance and Behavior

Appearance

- ✓ Menus and Toolbars

Appearance

[File](#) | [Settings](#) | [Appearance and Behavior](#) | [Appearance for Windows and Linux](#)

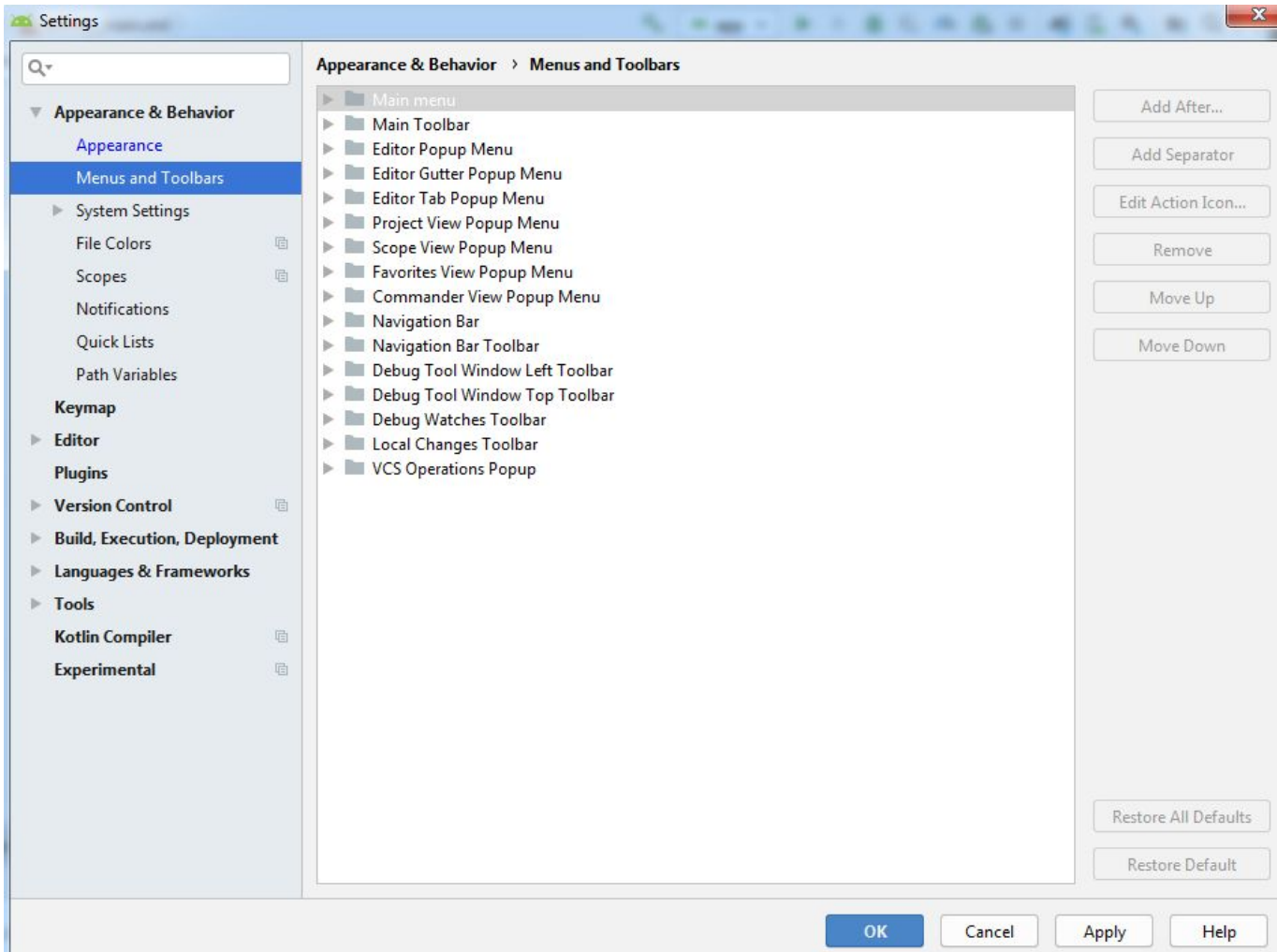
[IntelliJ IDEA](#) | [Preferences](#) | [Appearance and Behavior](#) | [Appearance for macOS](#)

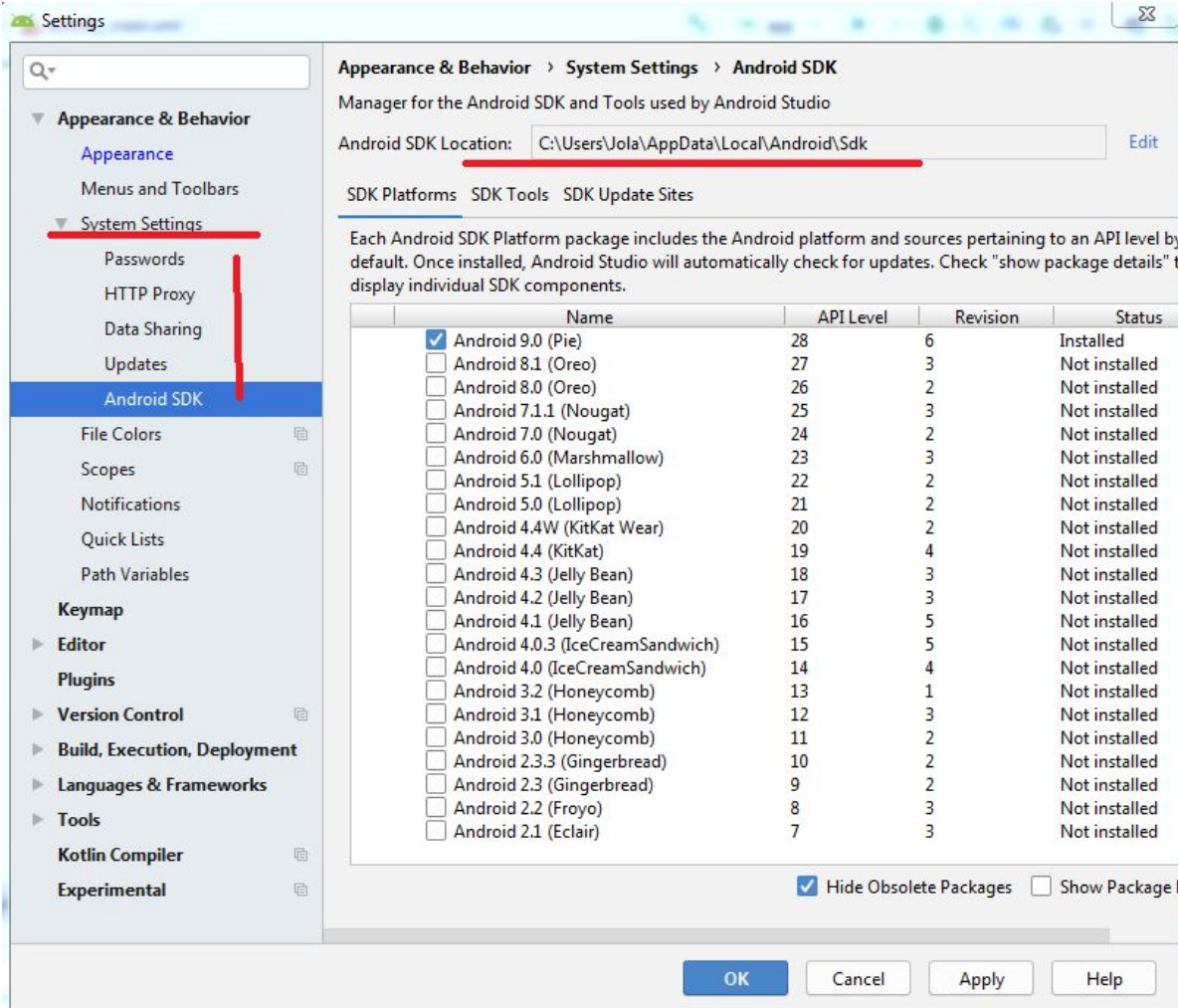
[Ctrl+Alt+S](#) 

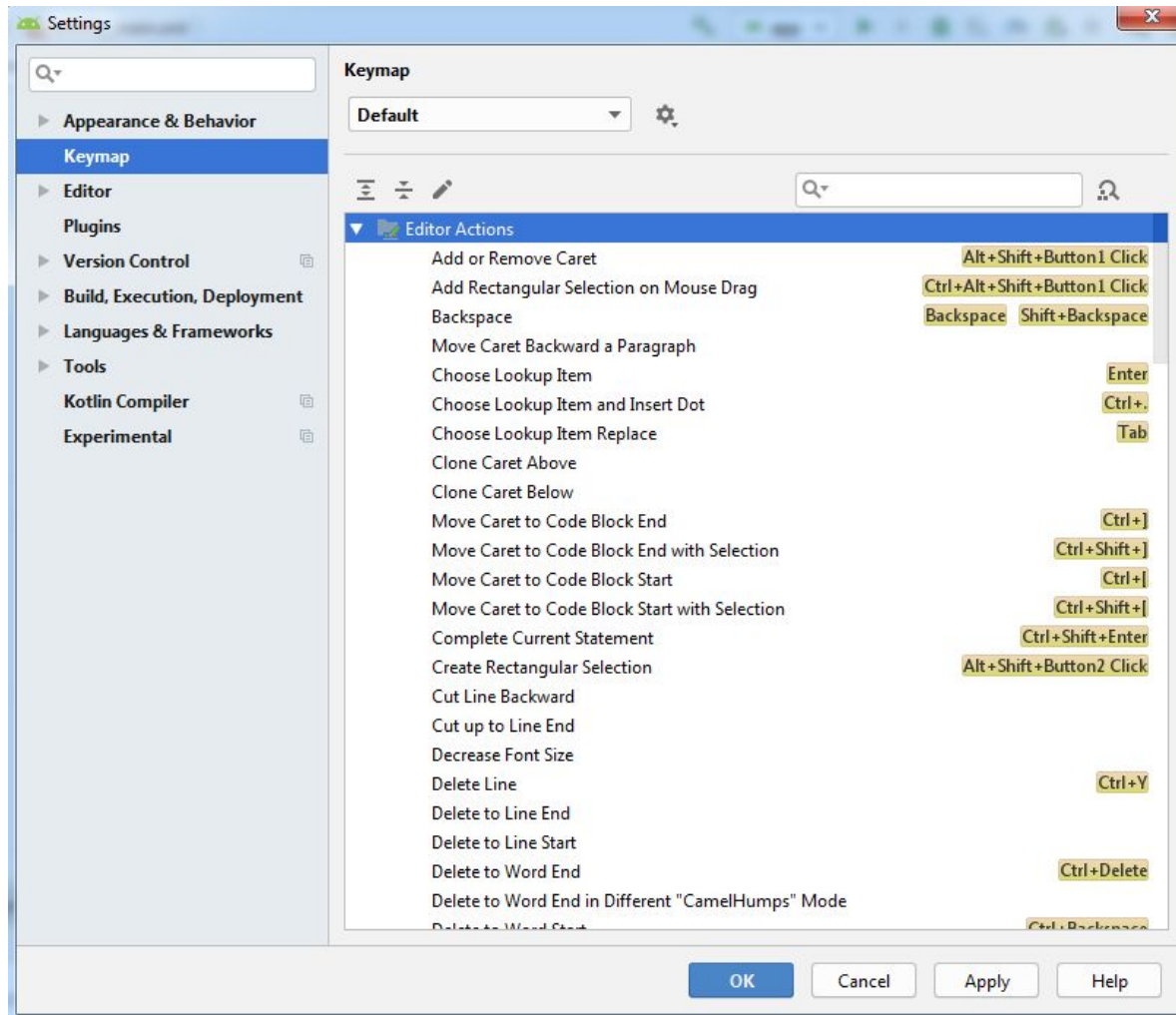
Use this page to change the overall look and feel of your IDE.

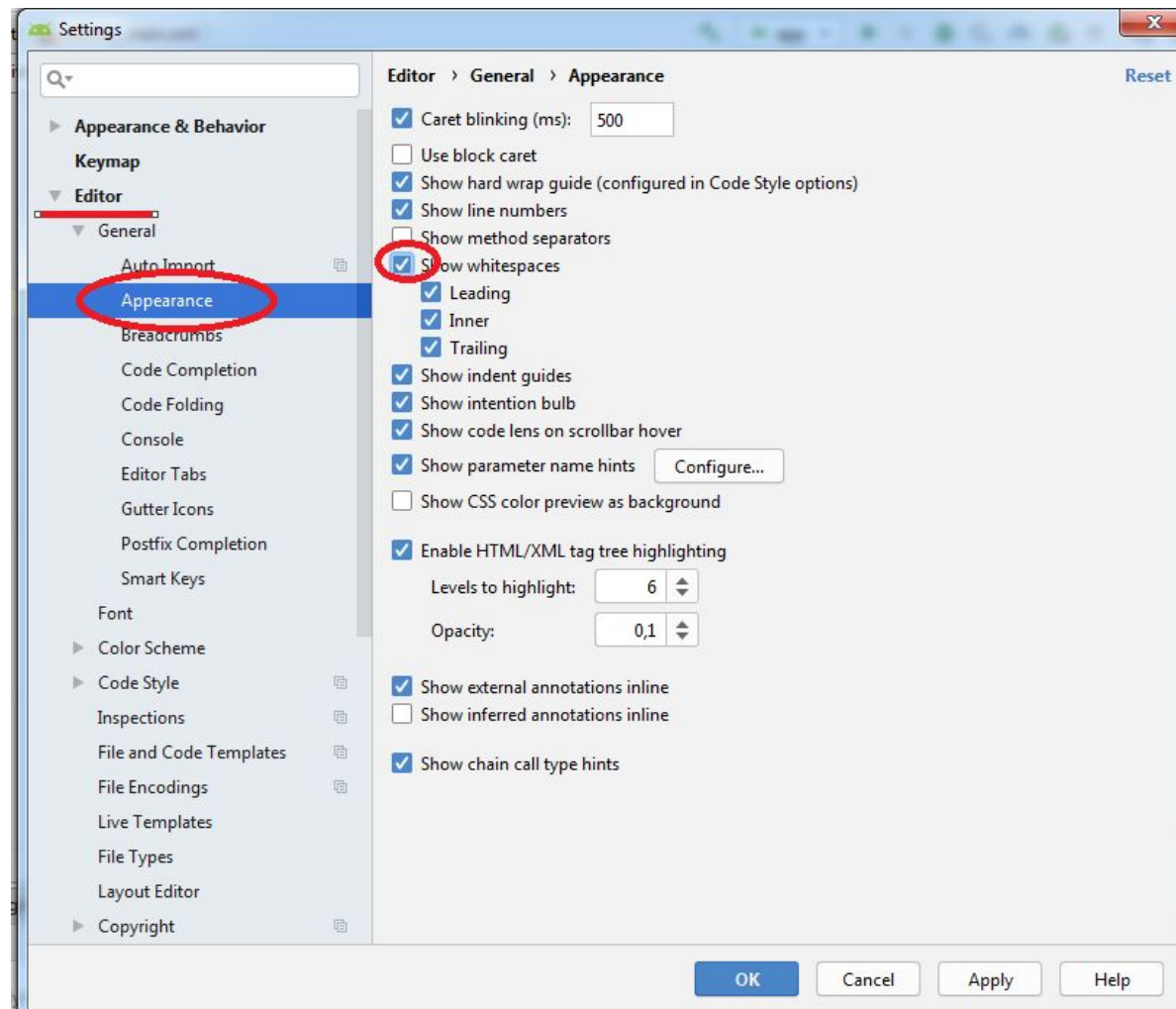
UI Options

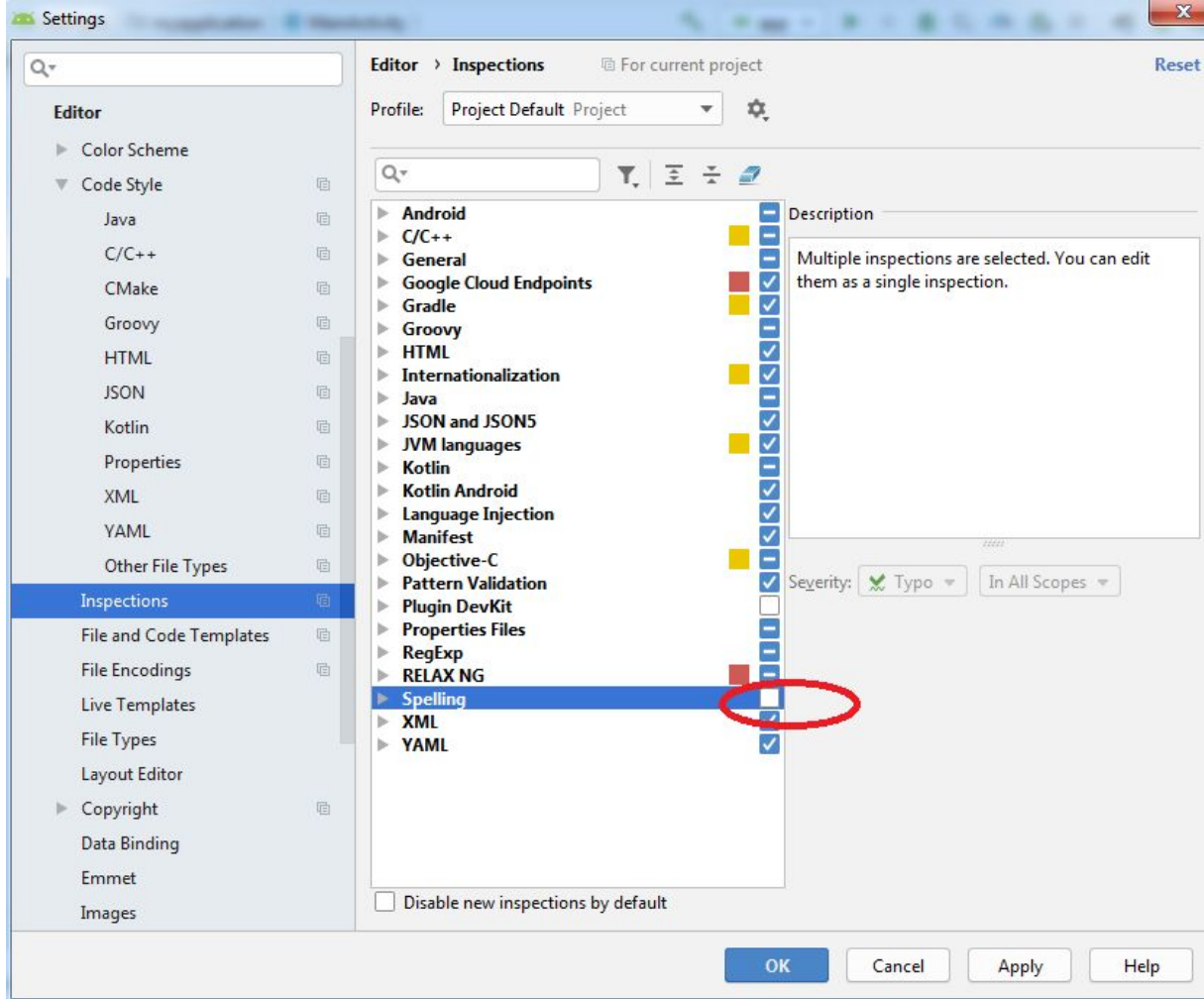
Option	Description
Theme	Use this drop-down list to select the desired theme from the list.
Adjust colors for red-green vision deficiency	Select this option to adjust the IDE colors (code highlighting in the editor, text notifications, etc.) for people with the red-green color deficiency. For more information, see Adjust colors for red-green color vision deficiency .
Override default fonts by (not recommended)	Select this checkbox to enable specifying font family and size to be used instead of the default one. When first installed, IntelliJ IDEA takes Windows default font size and style.
Cyclic scrolling in list	Select this checkbox to enable scrolling through a list by jumping from the last item to the first one and vice versa.

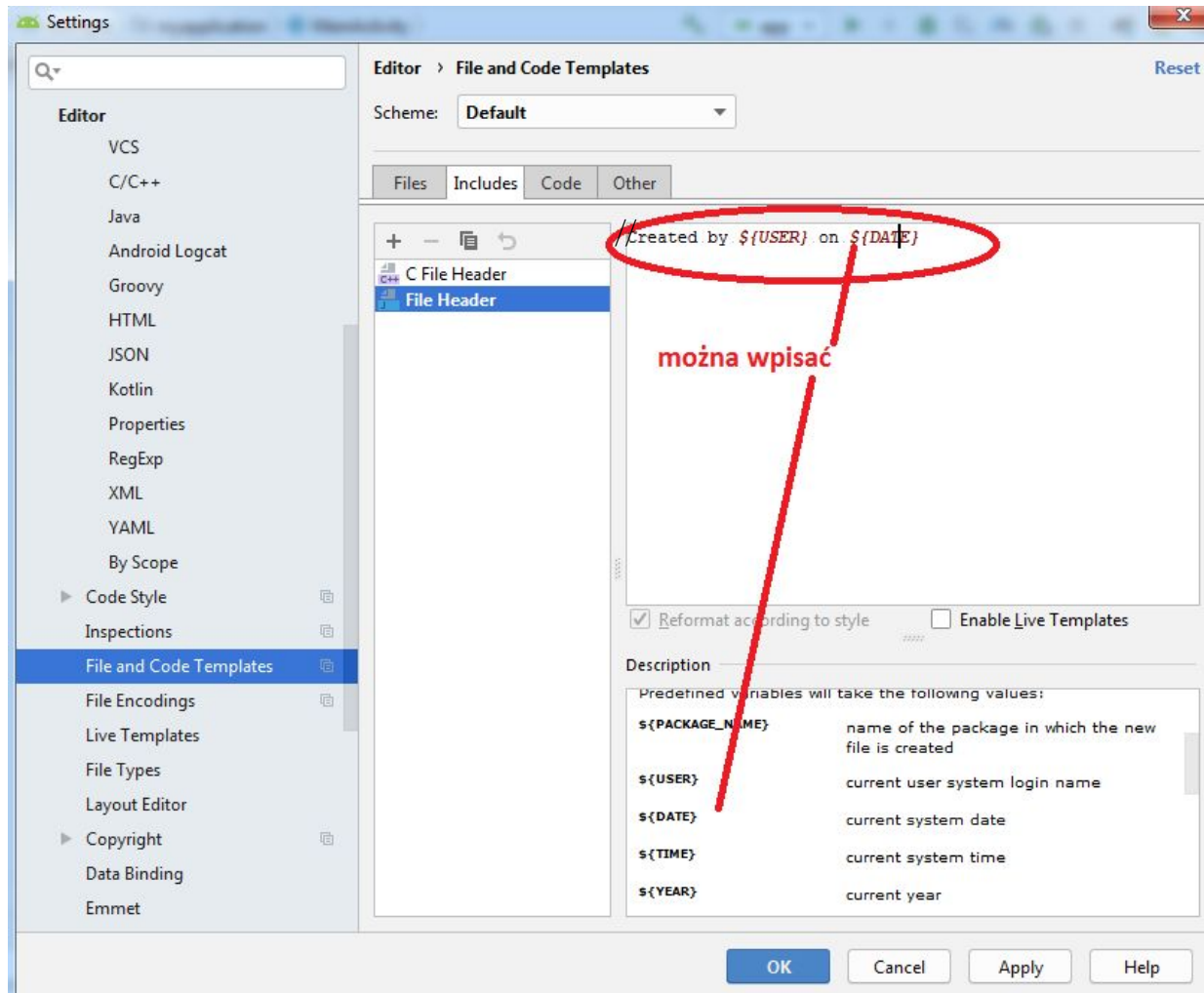




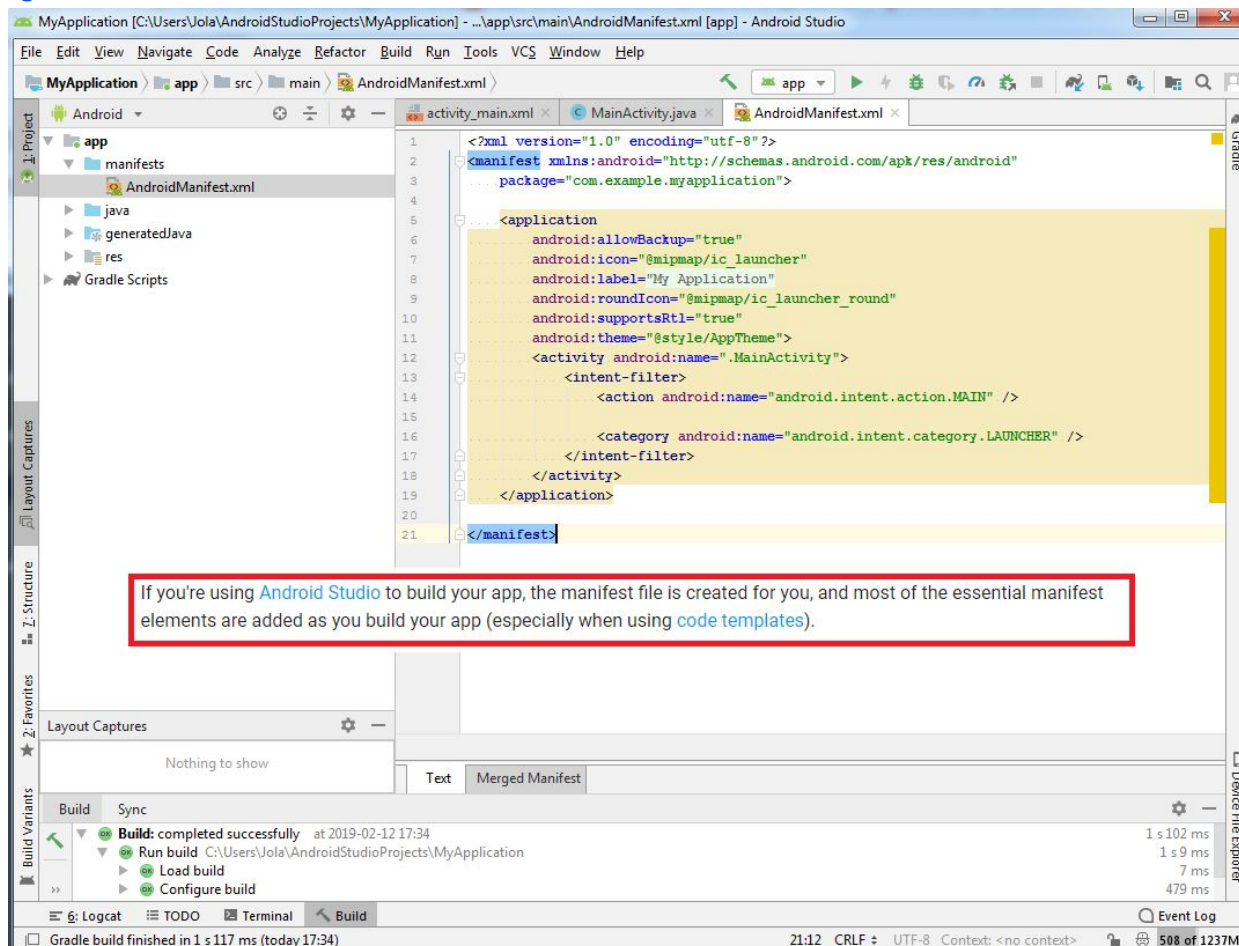








Struktura projektu



MyApplication [C:\Users\Vola\AndroidStudioProjects\MyApplication] - ...\app\src\main\AndroidManifest.xml [app] - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

MyApplication > app > src > main > AndroidManifest.xml

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.myapplication">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="My Application"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportRtl="true"
11        android:theme="@style/AppTheme">
12         <activity android:name=".MainActivity">
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15
16                 <category android:name="android.intent.category.LAUNCHER" />
17             </intent-filter>
18         </activity>
19     </application>
20
21 </manifest>
```

If you're using [Android Studio](#) to build your app, the manifest file is created for you, and most of the essential manifest elements are added as you build your app (especially when using [code templates](#)).

Layout Captures

Nothing to show

Text Merged Manifest

Build Sync

Build: completed successfully at 2019-02-12 17:34

- Run build C:\Users\Vola\AndroidStudioProjects\MyApplication
- Load build
- Configure build

1 s 102 ms

1 s 9 ms

7 ms

479 ms

Logcat TODO Terminal Build

Event Log

Gradle build finished in 1 s 117 ms (today 17:34)

21:12 CRLF UTF-8 Context: <no context>

508 of 1237M

App Manifest Overview

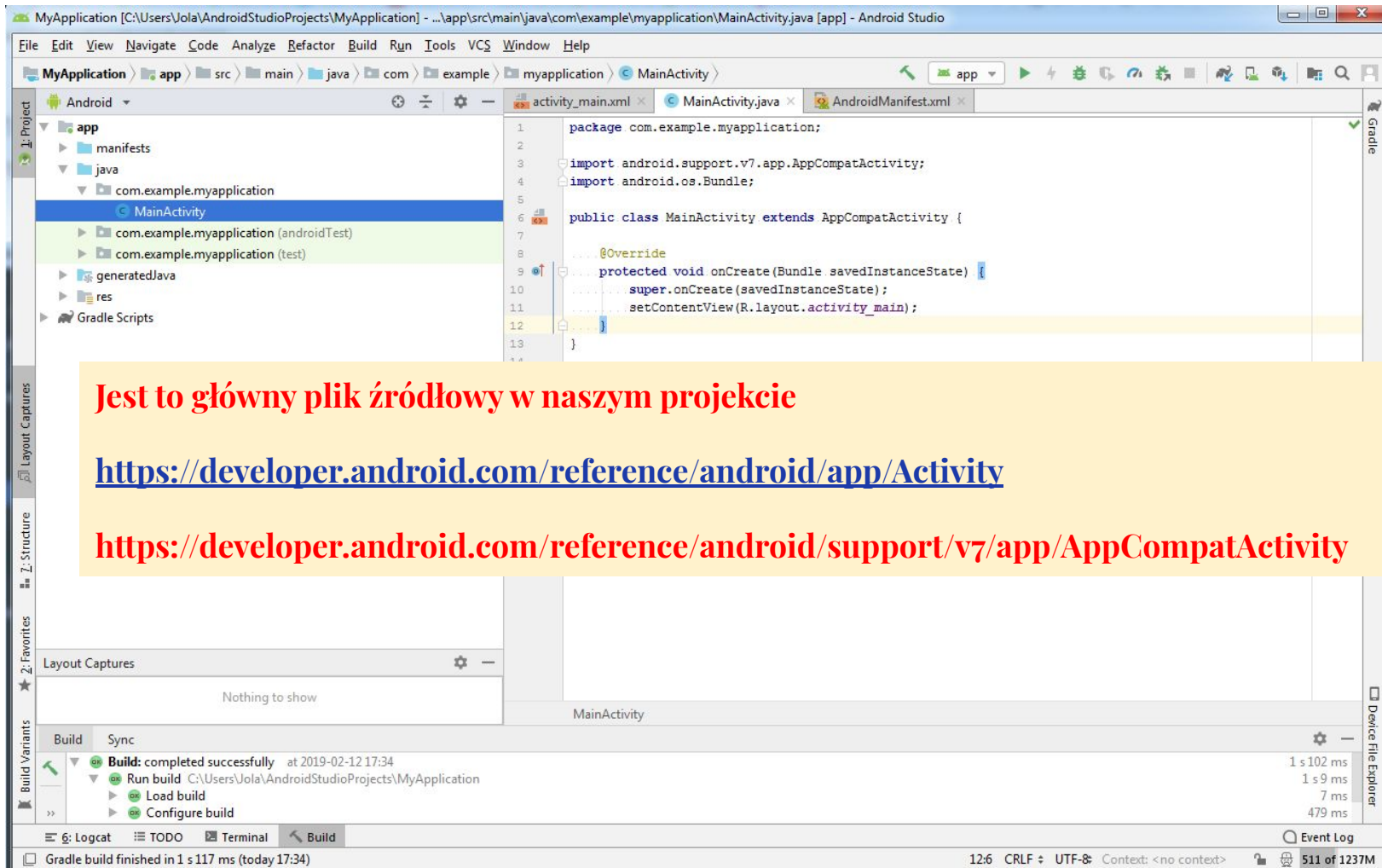


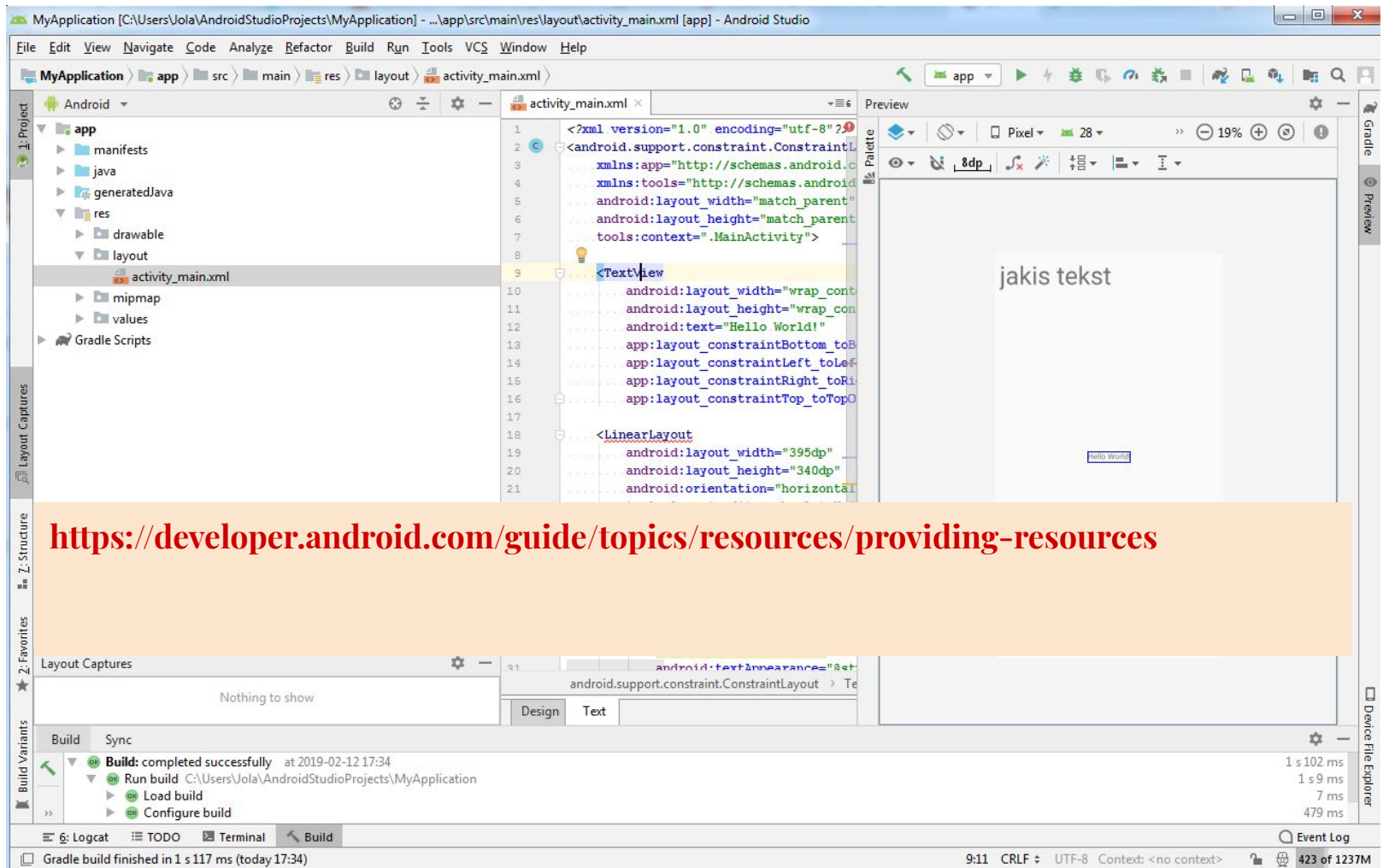
Every app project must have an `AndroidManifest.xml` file (with precisely that name) at the root of the [project source set](#). The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.

Among many other things, the manifest file is required to declare the following:

- The app's package name, which usually matches your code's namespace. The Android build tools use this to determine the location of code entities when building your project. When packaging the app, the build tools replace this value with the application ID from the Gradle build files, which is used as the unique app identifier on the system and on Google Play. [Read more about the package name and app ID.](#)
- The components of the app, which include all activities, services, broadcast receivers, and content providers. Each component must define basic properties such as the name of its Kotlin or Java class. It can also declare capabilities such as which device configurations it can handle, and intent filters that describe how the component can be started. [Read more about app components.](#)
- The permissions that the app needs in order to access protected parts of the system or other apps. It also declares any permissions that other apps must have if they want to access content from this app. [Read more about permissions.](#)
- The hardware and software features the app requires, which affects which devices can install the app from Google Play. [Read more about device compatibility.](#)

If you're using [Android Studio](#) to build your app, the manifest file is created for you, and most of the essential manifest elements are added as you build your app (especially when using [code templates](#)).



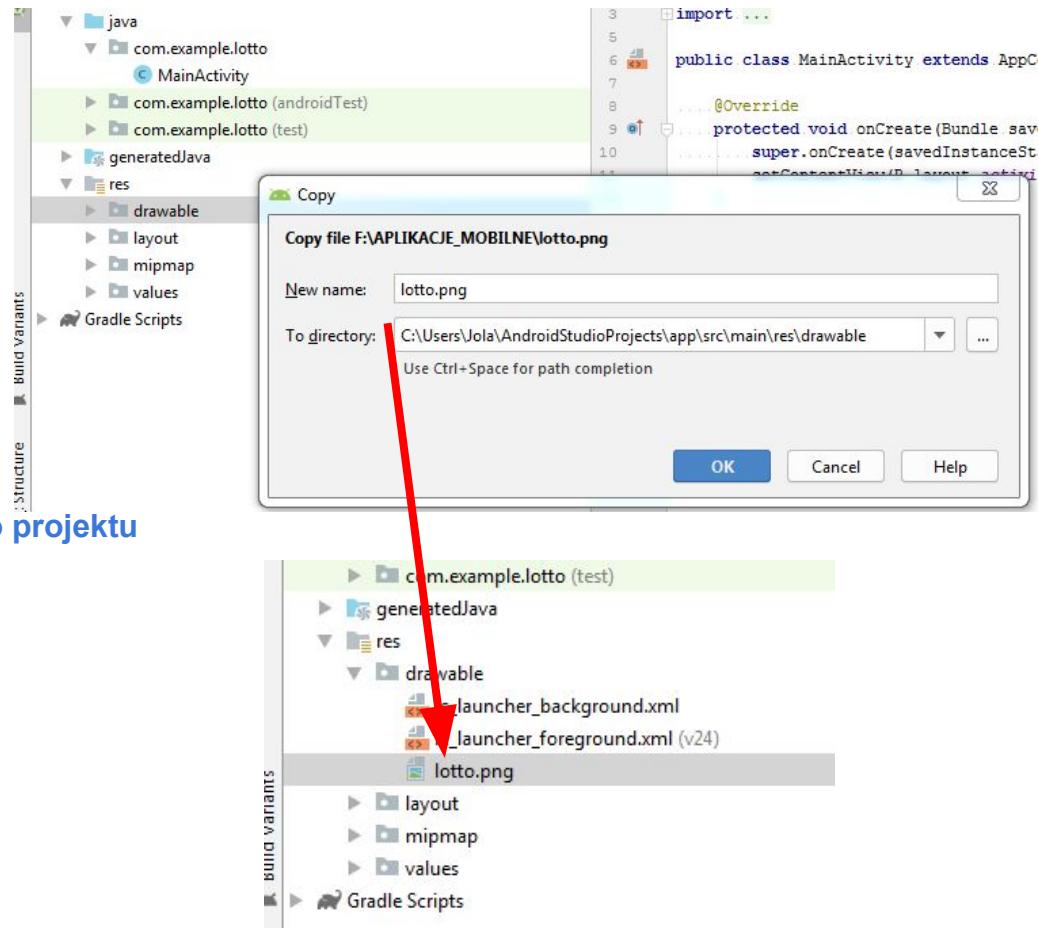


<https://developer.android.com/guide/topics/resources/providing-resources>

Utworzenie pierwszej aplikacji - rozkład

Aplikacja Lotto:

1. File->new project
2. Lotto-nazwa projektu
3. Pozostałe opcje bez zmian
4. Empty Activity
5. Szukamy obrazka Lotto.
6. Kopiujemy i wklejamy do folderu
7. Obrazek osadzimy w ImageView
8. Activity_main.xml - w trybie edycji
-przechodzimy do głównego rozkładu naszego projektu
Zmieniamy rozkład na RelativeLayout



Write the XML

Using Android's XML vocabulary, you can quickly design UI layouts and the screen elements they contain, in the same way you create web pages in HTML — with a series of nested elements.

Each layout file must contain exactly one root element, which must be a View or ViewGroup object. Once you've defined the root element, you can add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines your layout. For example, here's an XML layout that uses a vertical `LinearLayout` to hold a `TextView` and a `Button`:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```



ID

Any View object may have an integer ID associated with it, to uniquely identify the View within the tree. When the app is compiled, this ID is referenced as an integer, but the ID is typically assigned in the layout XML file as a string, in the `id` attribute. This is an XML attribute common to all View objects (defined by the `View` class) and you will use it very often. The syntax for an ID, inside an XML tag is:

```
android:id="@+id/my_button"
```



The at-symbol (@) at the beginning of the string indicates that the XML parser should parse and expand the rest of the ID string and identify it as an ID resource. The plus-symbol (+) means that this is a new resource name that must be created and added to our resources (in the `R.java` file). There are a number of other ID resources that are offered by the Android framework. When referencing an Android resource ID, you do not need the plus-symbol, but must add the `android` package namespace, like so:

```
android:id="@android:id/empty"
```



With the `android` package namespace in place, we're now referencing an ID from the `android.R` resources class, rather than the local resources class.

In order to create views and reference them from the app, a common pattern is to:

1. Define a view/widget in the layout file and assign it a unique ID:

```
<Button android:id="@+id/my_button"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/my_button_text" />
```

2. Then create an instance of the view object and capture it from the layout (typically in the `onCreate()` method):

KOTLIN

JAVA


```
Button myButton = (Button) findViewById(R.id.my_button);
```

Defining IDs for view objects is important when creating a `RelativeLayout`. In a relative layout, sibling views can define their layout relative to another sibling view, which is referenced by the unique ID.


An ID need not be unique throughout the entire tree, but it should be unique within the part of the tree you are searching (which may often be the entire tree, so it's best to be completely unique when possible).

Główny rozkład aplikacji

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#FFF"
tools:context=".MainActivity">
</RelativeLayout>
```



Dodajemy TextView - jest to główny, według którego będą pozycjonowane inne widoki



```
<ImageView
    android:id="@+id/imageview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/lotto"
    android:layout_above="@id/textview"
    android:layout_centerInParent="true" />
```

```
<ImageView
    android:id="@+id/imageview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/lotto"
    android:layout_above="@id/textview"
    android:layout_centerInParent="true" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/textview"
    android:layout_centerInParent="true"
    android:text="Generuj"
    android:textSize="16sp" />
```

Android Developers > Docs > Guides

Relative Layout

RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent **RelativeLayout** area (such as aligned to the bottom, left or center).



Note: For better performance and tooling support, you should instead [build your layout with ConstraintLayout](#).

A **RelativeLayout** is a very powerful utility for designing a user interface because it can eliminate nested view groups and keep your layout hierarchy flat, which improves performance. If you find yourself using several nested **LinearLayout** groups, you may be able to replace them with a single **RelativeLayout**.

Positioning Views

`RelativeLayout` lets child views specify their position relative to the parent view or to each other (specified by ID). So you can align two elements by right border, or make one below another, centered in the screen, centered left, and so on. By default, all child views are drawn at the top-left of the layout, so you must define the position of each view using the various layout properties available from `RelativeLayout.LayoutParams`.

Some of the many layout properties available to views in a `RelativeLayout` include:

`android:layout_alignParentTop`

If `"true"`, makes the top edge of this view match the top edge of the parent.

`android:layout_centerVertical`

If `"true"`, centers this child vertically within its parent.

`android:layout_below`

Positions the top edge of this view below the view specified with a resource ID.

`android:layout_toRightOf`

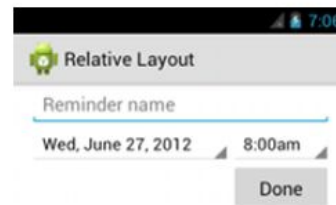
Positions the left edge of this view to the right of the view specified with a resource ID.

These are just a few examples. All layout attributes are documented at `RelativeLayout.LayoutParams`.

The value for each layout property is either a boolean to enable a layout position relative to the parent `RelativeLayout` or an ID that references another view in the layout against which the view should be positioned.

In your XML layout, dependencies against other views in the layout can be declared in any order. For example, you can declare that "view1" be positioned below "view2" even if "view2" is the last view declared in the hierarchy. The example below demonstrates such a scenario.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```



Utworzenie pierwszej aplikacji - aktywność



android.widget

added in API level 1

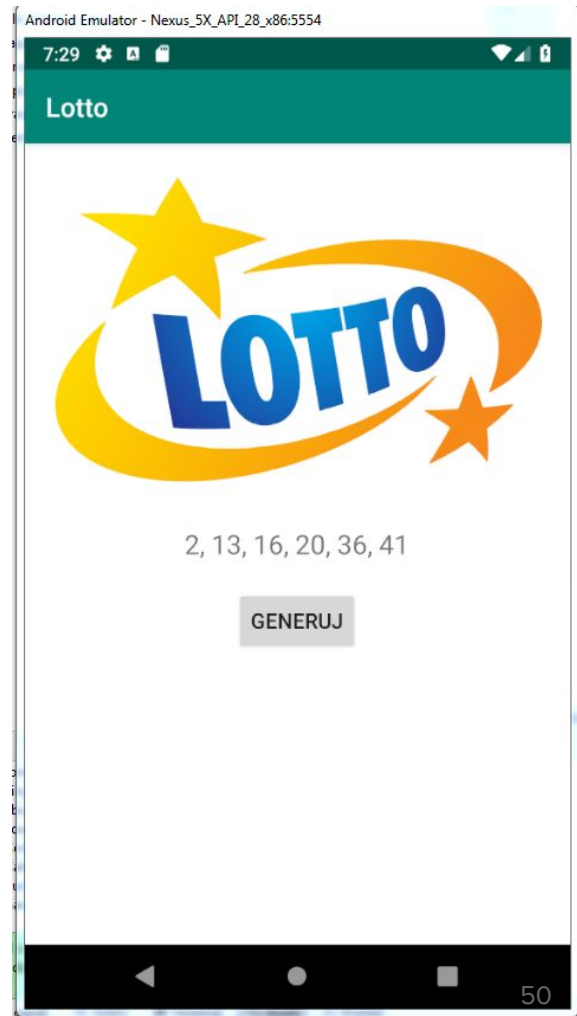
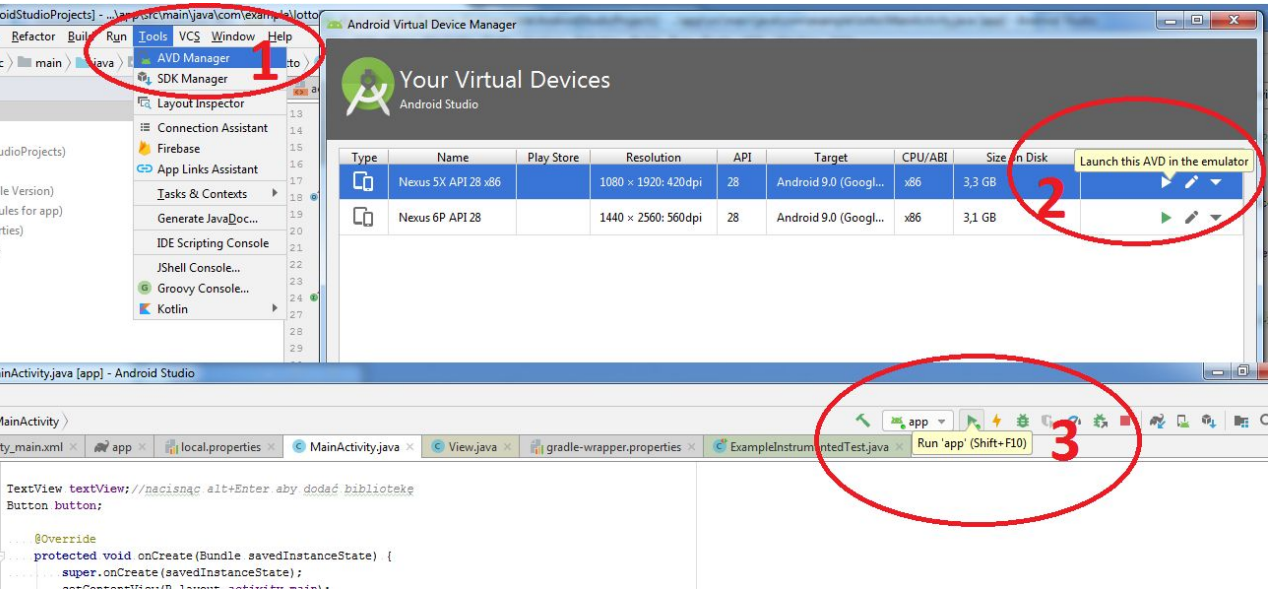


The widget package contains (mostly visual) UI elements to use on your Application screen. You can also design your own.

To create your own widget, extend [View](#) or a subclass. To use your widget in layout XML, there are two additional files for you to create. Here is a list of files you'll need to create to implement a custom widget:

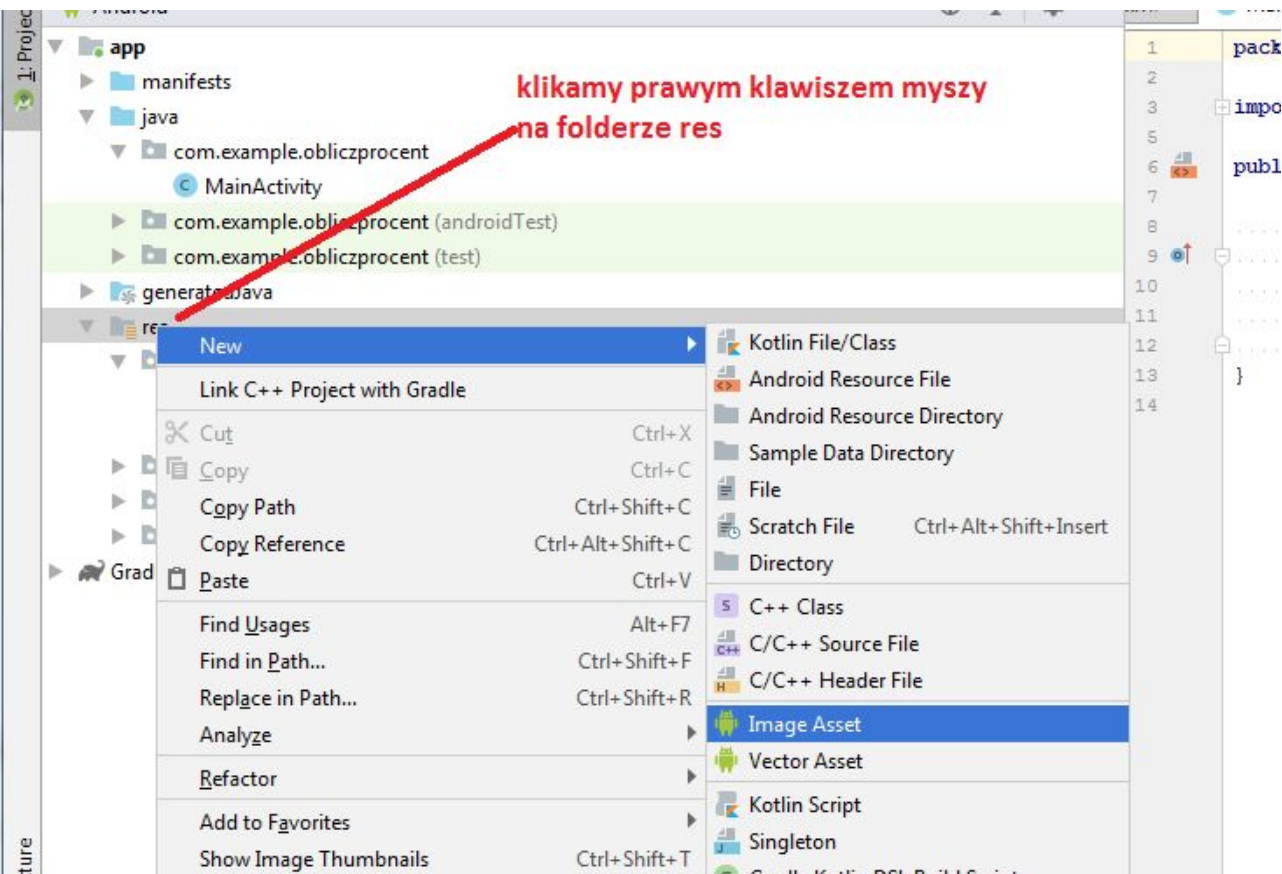
- **Java implementation file** - This is the file that implements the behavior of the widget. If you can instantiate the object from layout XML, you will also have to code a constructor that retrieves all the attribute values from the layout XML file.
- **XML definition file** - An XML file in `res/values/` that defines the XML element used to instantiate your widget, and the attributes that it supports. Other applications will use this element and attributes in their in another in their layout XML.
- **Layout XML [optional]** - An optional XML file inside `res/layout/` that describes the layout of your widget. You could also do this in code in your Java file.

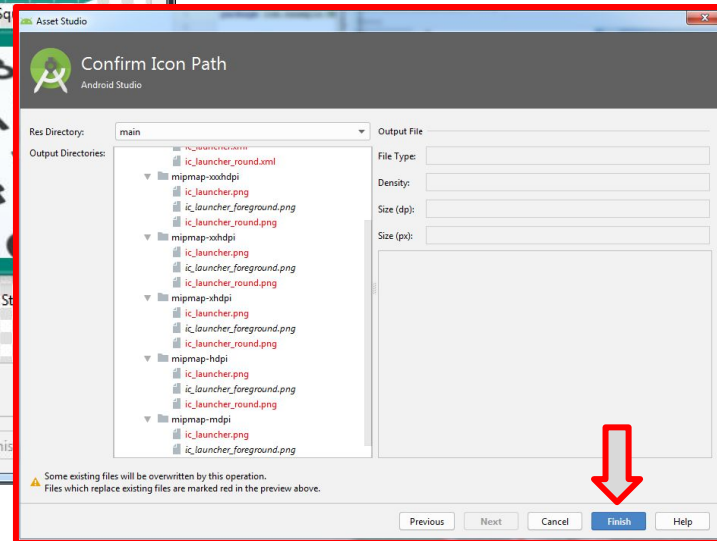
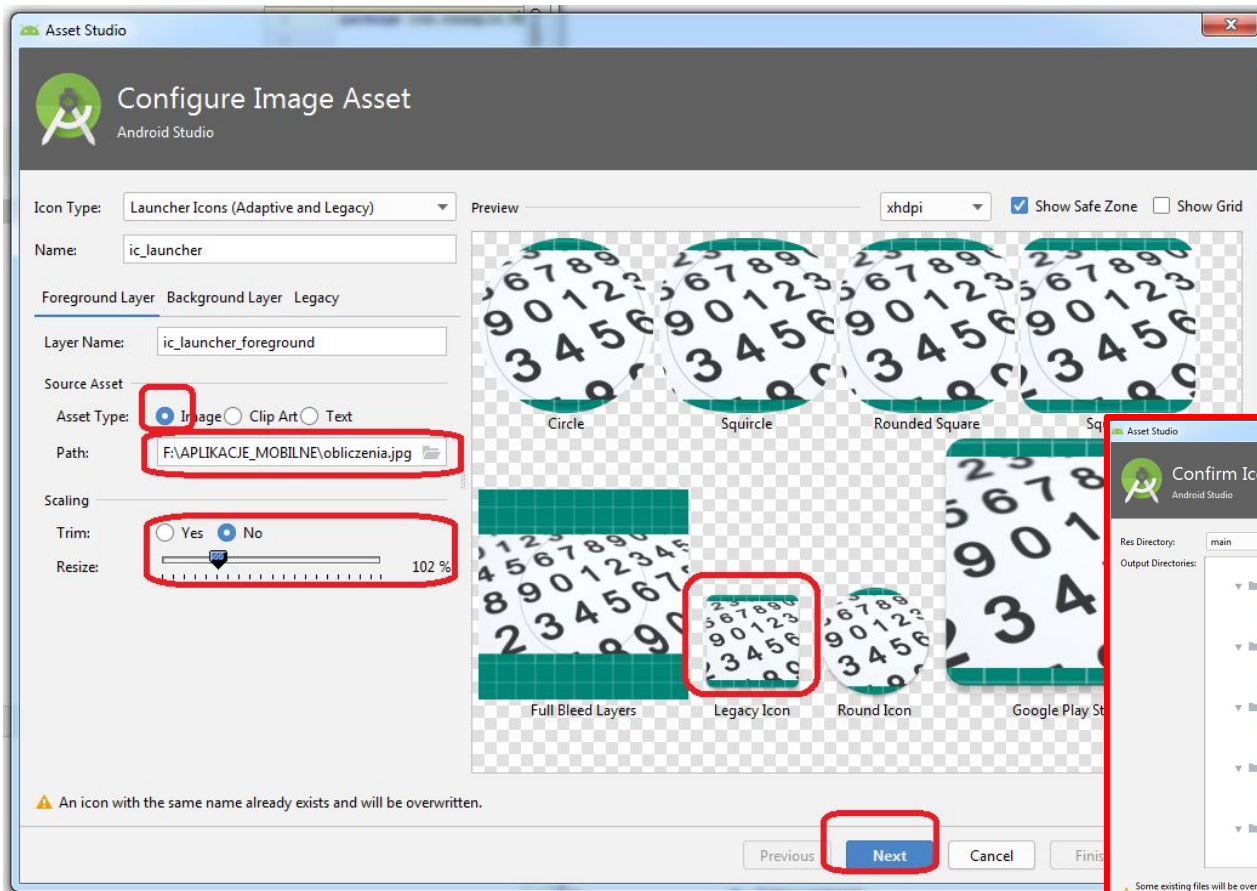
Uruchamianie aplikacji na emulatorze



Zadanie: Analiza aplikacji

1. Utworzenie nowego projektu i dodanie ikony aplikacji





Zaprojektowanie interfejsu aplikacji z użyciem rozkładu GridLayout

<https://developer.android.com/reference/android/widget/GridLayout>

← → ↺

https://developer.android.com/reference/android/widget/GridLayout

🔍 ⚙️ ⌵

Android Developers

Platform Android Studio Google Play Android Jetpack Docs News

🔍 Szukaj

Documentation

OVERVIEW GUIDES REFERENCE SAMPLES DESIGN & QUALITY

Overview

Android Platform

API level 28

Class Index

Package Index

android

android.accessibilityservice

android.accounts

android.animation

android.annotation

android.app

android.app.admin

android.app.assist

android.app.backup

android.app.job

android.app.slice

android.app.usage

android.appwidget

android.bluetooth

android.bluetooth.le

android.companion

android.content

android.content.pm

GridLayout

added in API level 14

☆☆☆☆☆

public class GridLayout

extends ViewGroup

java.lang.Object

↳ android.view.View

↳ android.view.ViewGroup

↳ android.widget.GridLayout

A layout that places its children in a rectangular *grid*.

The grid is composed of a set of infinitely thin lines that separate the viewing area into *cells*. Throughout the API, grid lines are referenced by grid *indices*. A grid with *N* columns has *N* + 1 grid indices that run from 0 through *N* inclusive. Regardless of how GridLayout is configured, grid index 0 is fixed to the leading edge of the container and grid index *N* is fixed to its trailing edge (after padding is taken into account).

Row and Column Specs

Children occupy one or more contiguous cells, as defined by their `rowSpec` and `columnSpec` layout parameters. Each spec defines the set of rows or columns that are to be occupied; and how children should be aligned within the resulting group of cells. Although cells do not normally overlap in a GridLayout, GridLayout does not prevent children being defined to occupy the same cell or group of cells. In this case however, there is no guarantee that children will not themselves overlap after the layout operation completes.

Spis treści

Summary

Nested classes

XML attributes

Inherited XML attributes

Constants

Inherited constants

Fields

Inherited fields

Public constructors

Public methods

Protected methods

Inherited methods

XML attributes

android:alignmentM...

android:columnCount

android:columnOrde...

android:orientation

android:rowCount

android:rowOrderPre...

android:useDefault...

Constants

ALIGN_BOUNDS

ALIGN_MARGINS

HORIZONTAL



Introduction

Material studies

Material Foundation

Foundation overview

Environment

Layout

Understanding layout

Density & resolution

Responsive layout grid

Columns, gutters, and margins

Grid customization

Breakpoints

Grid behavior

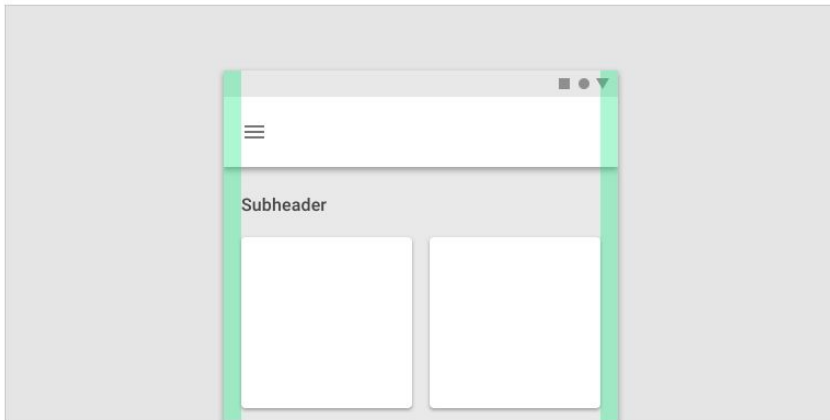
UI regions

Whiteframes

Margins

Margins are the space between content and the left and right edges of the screen.

Margin widths are defined as fixed values at each breakpoint range. To better adapt to the screen, the margin width can change at different breakpoints. Wider margins are more appropriate for larger screens, as they create more whitespace around the perimeter of content.



On mobile, at a breakpoint of 360dp, this layout grid uses 16dp margins.

Material System

Introduction

Material studies

Material Foundation

Foundation overview

Environment

Layout

Understanding layout

Density & resolution

Responsive layout grid

Spacing methods

Baseline

Spacing

Containers and aspect ratios

Touch and click targets

Component behavior

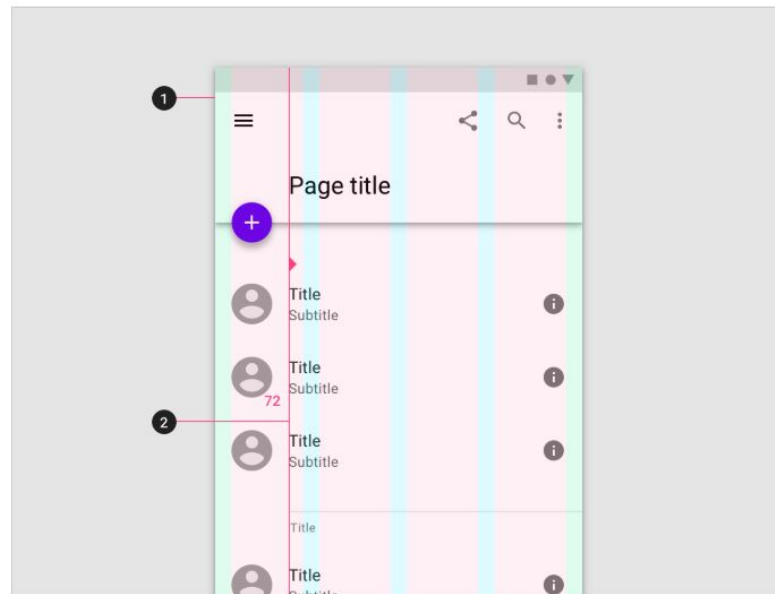
Density

Navigation

Keylines

Keylines enable the consistent placement of elements outside of the **layout grid**. They are vertical lines that show where elements are placed in a design that doesn't align to the grid. Keylines are determined by each element's distance from the edge of the screen, measured in increments of 8dp.

Keylines should be used in combination with the responsive layout grid to place elements consistently across a design.



- 1. Layout Grid
- 2. Keyline

Zmieniamy domyślny Layout na GridLayout

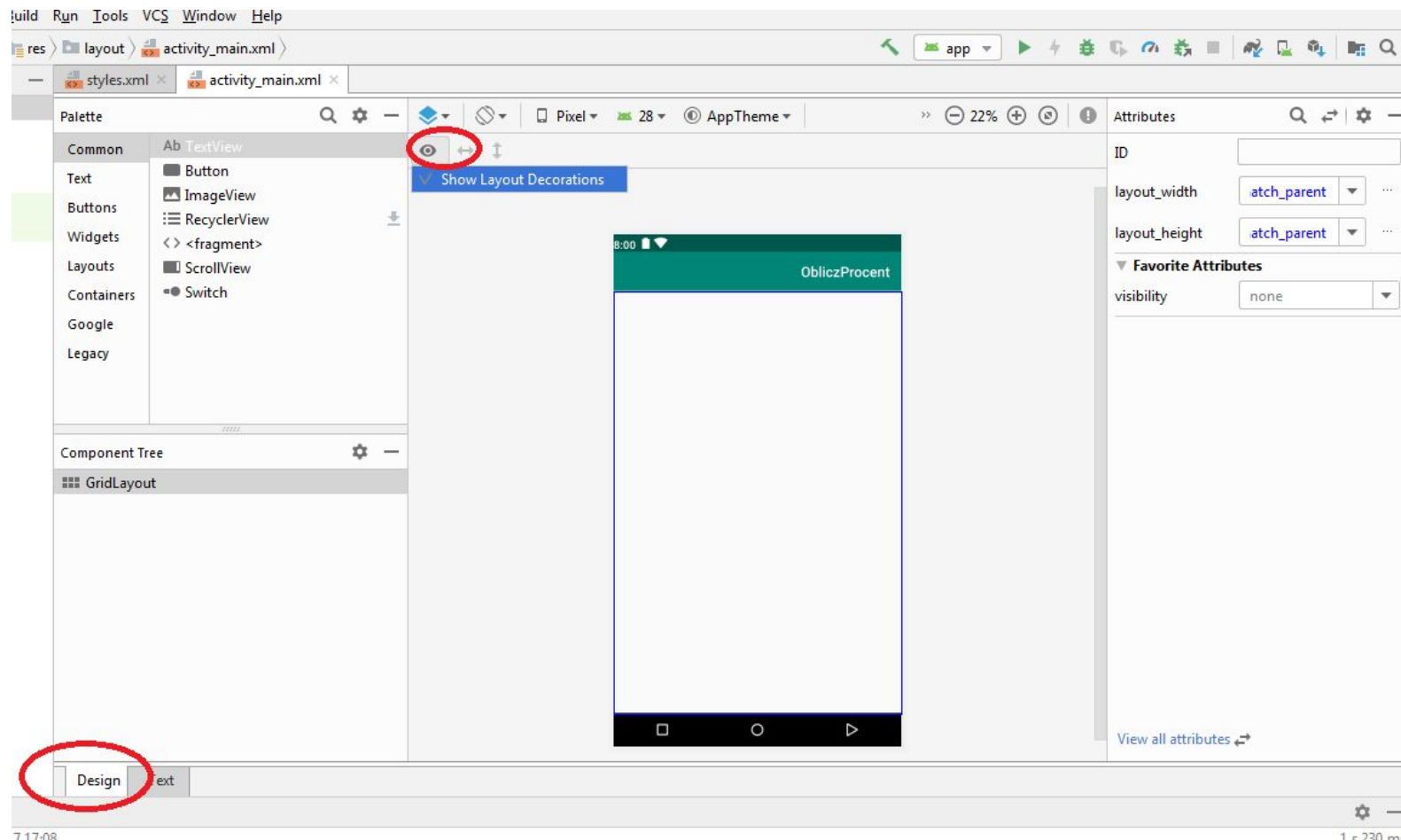
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
</android.support.constraint.ConstraintLayout>
```

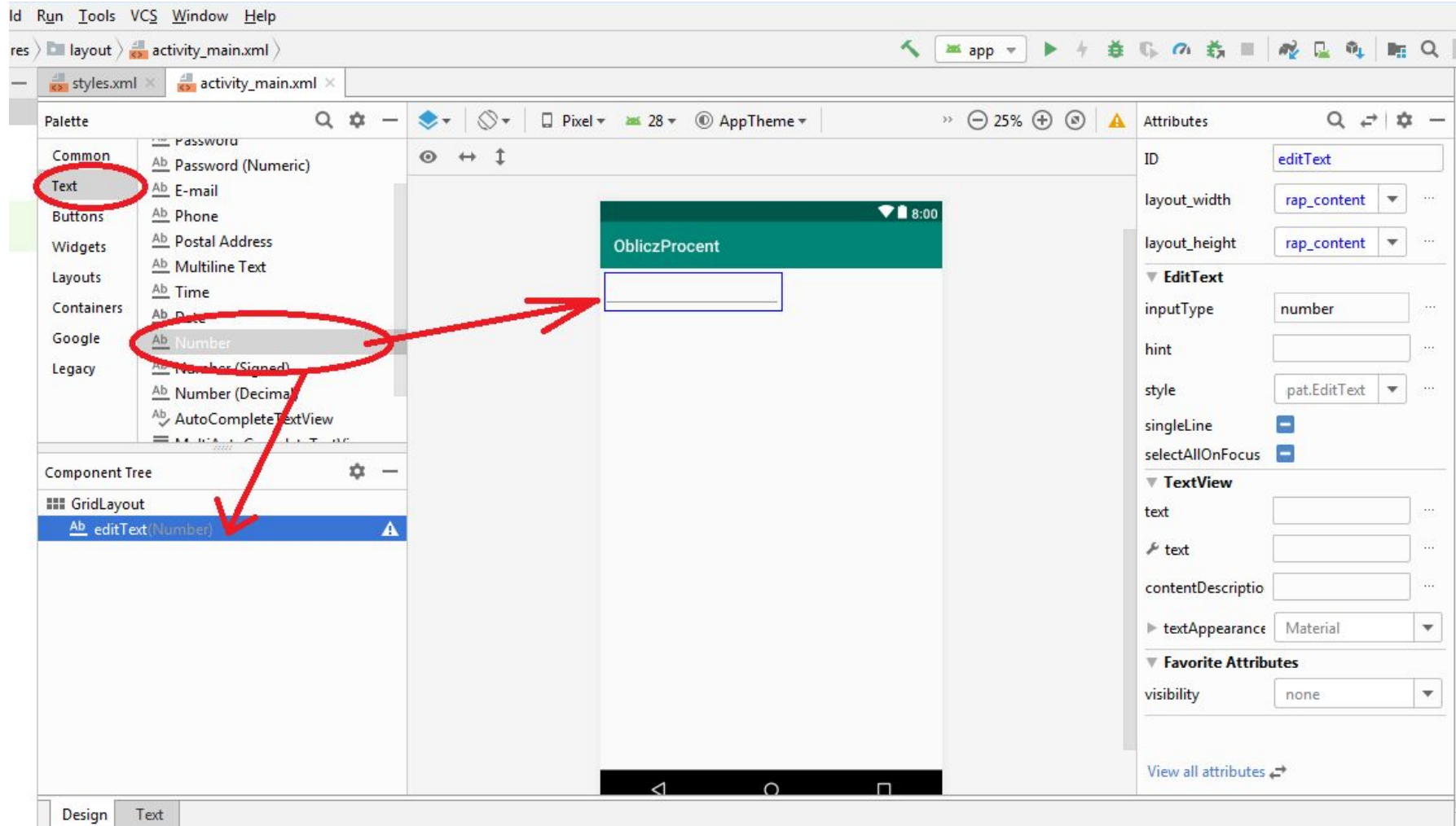
Na:

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:columnCount="2"
  android:useDefaultMargins="true" <!--pomiędzy wszystkimi widokami w rozkładzie będzie 8dp wolnej
                                przestrzeni, co jest standardem w zaleceniach MATERIAL DESIGN -->
  tools:context=".MainActivity">
</GridLayout>
```

Cały rozkład zbudujemy w edytorze rozkładu:



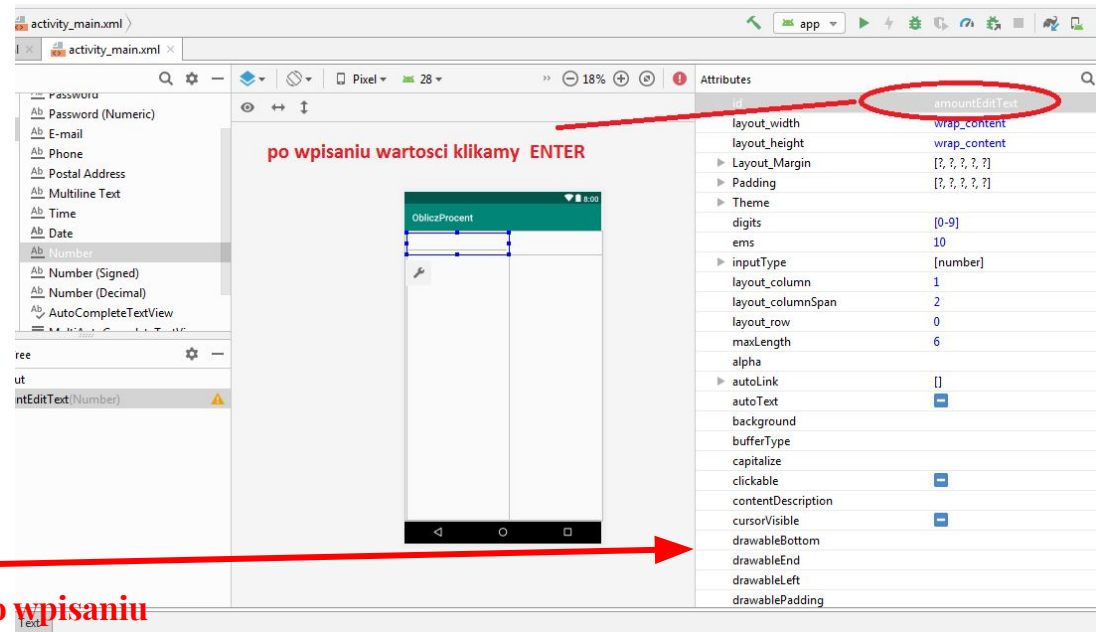


Widok EditText

<EditText

```
    android:id="@+id/amountEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_row="0"
    android:layout_column="0"
    android:layout_columnSpan="2"
    android:digits="[0-9]"
    android:ems="10"
    android:inputType="number"
    android:maxLength="6" />
```

Szukamy własności i wpisujemy wartość-po wpisaniu wciskamy ENTER



Widok textView, który będzie przykrywał widok EditText

Palette

Common

Text

Buttons

Widgets

Layouts

Containers

Google

Legacy

Ab TextView

Ab Plain Text

Ab Password

Ab Password (Numeric)

Ab E-mail

Ab Phone

Ab Postal Address

Ab Multiline Text

Ab Time

Ab Date

Ab Number

Ab Number (Signed)

Ab Number (Decimal)

Ab AutoCompleteTextView

MultiAutoCompleteTextView

Component Tree

GridLayout

Ab amountEditText(Number)

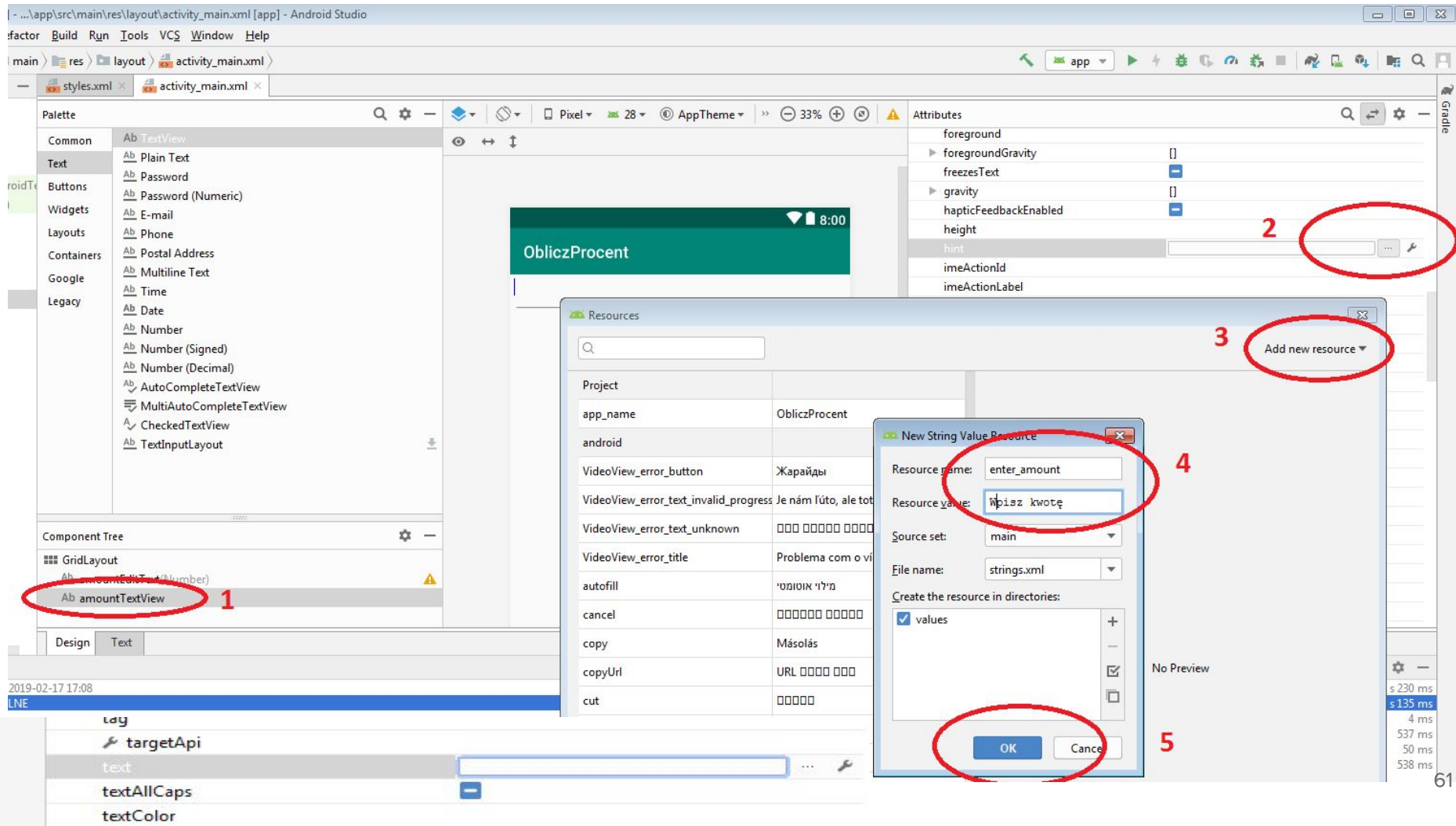
Ab textView- "TextView"

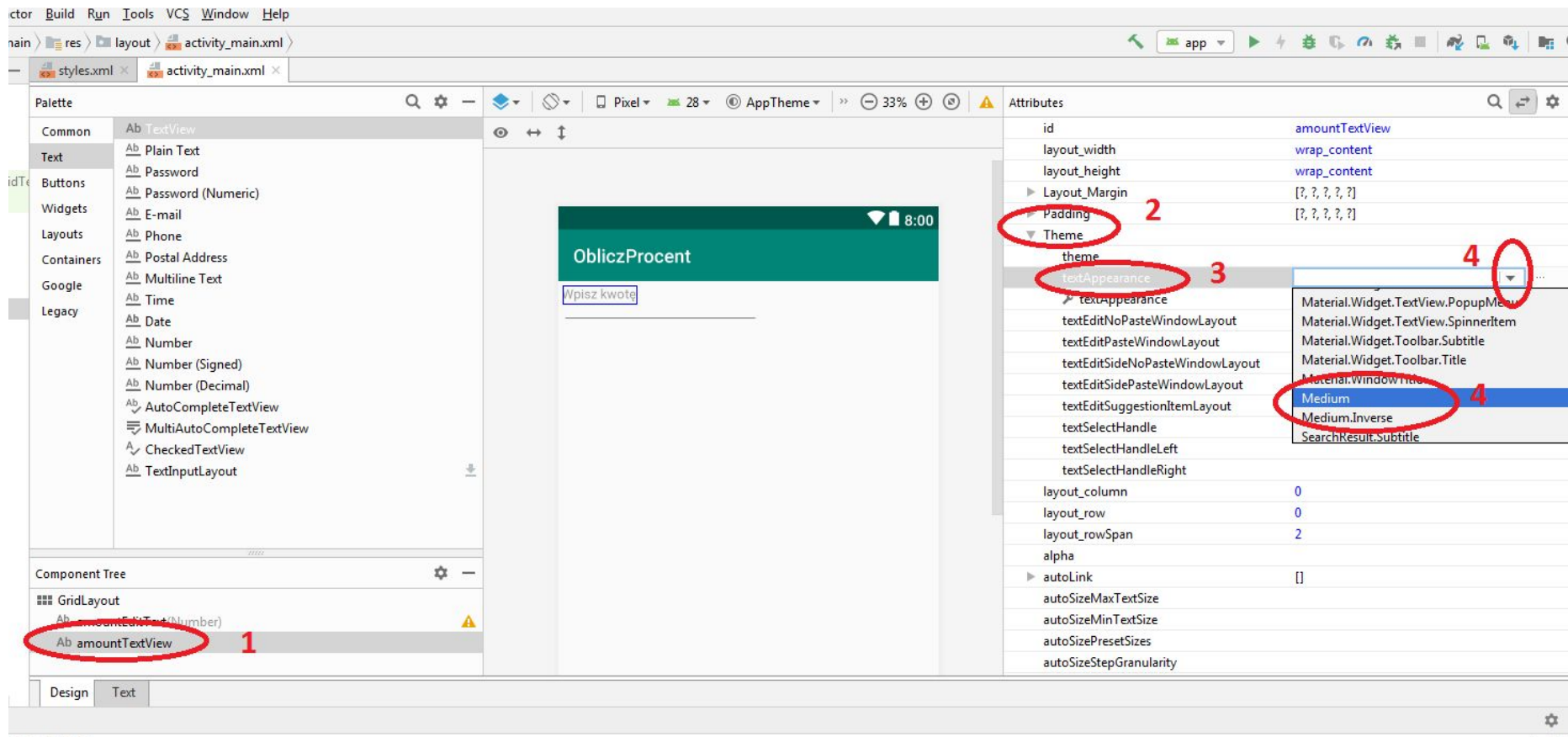
`<TextView
 android:id="@+id/amountTextView"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_row="0"
 android:layout_columnSpan="2"
 android:layout_column="0" />`

Attributes

id	textView
layout_width	wrap_content
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?]
Padding	[?, ?, ?, ?]
Theme	
text	TextView
alpha	
autoLink	[]
autoSizeMaxTextSize	
autoSizeMinTextSize	
autoSizePresetSizes	
autoSizeStepGranularity	
autoSizeTextType	
autoText	[-]
background	
bufferType	
capitalize	
clickable	[-]
contentDescription	
cursorVisible	[-]
digits	
drawableBottom	
drawableEnd	
drawableLeft	
drawablePadding	
drawableRight	
drawableStart	
drawableTop	

Ustawić wartości zgodnie z plikiem xml





Palette

















































































































































































































































































































































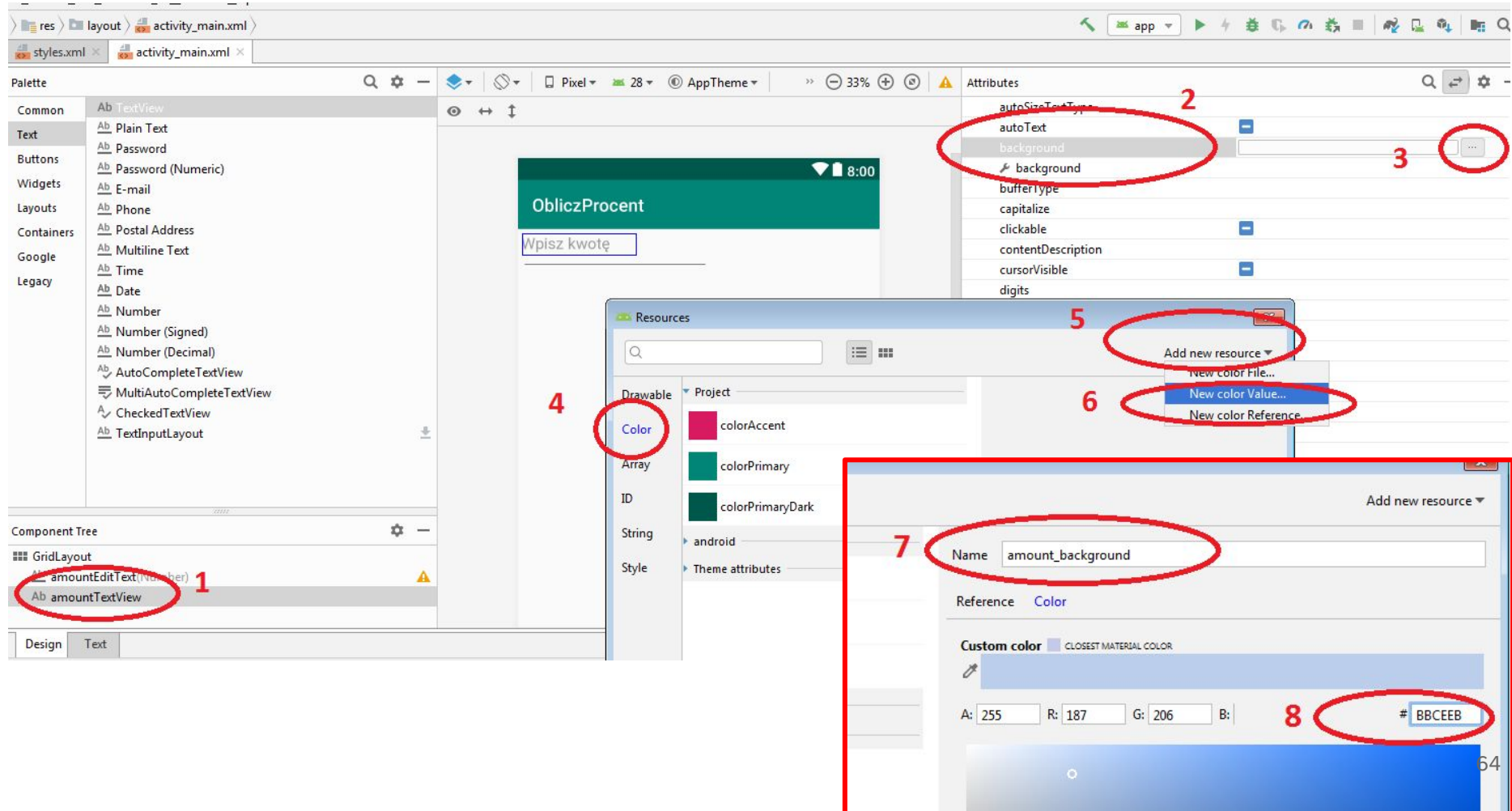


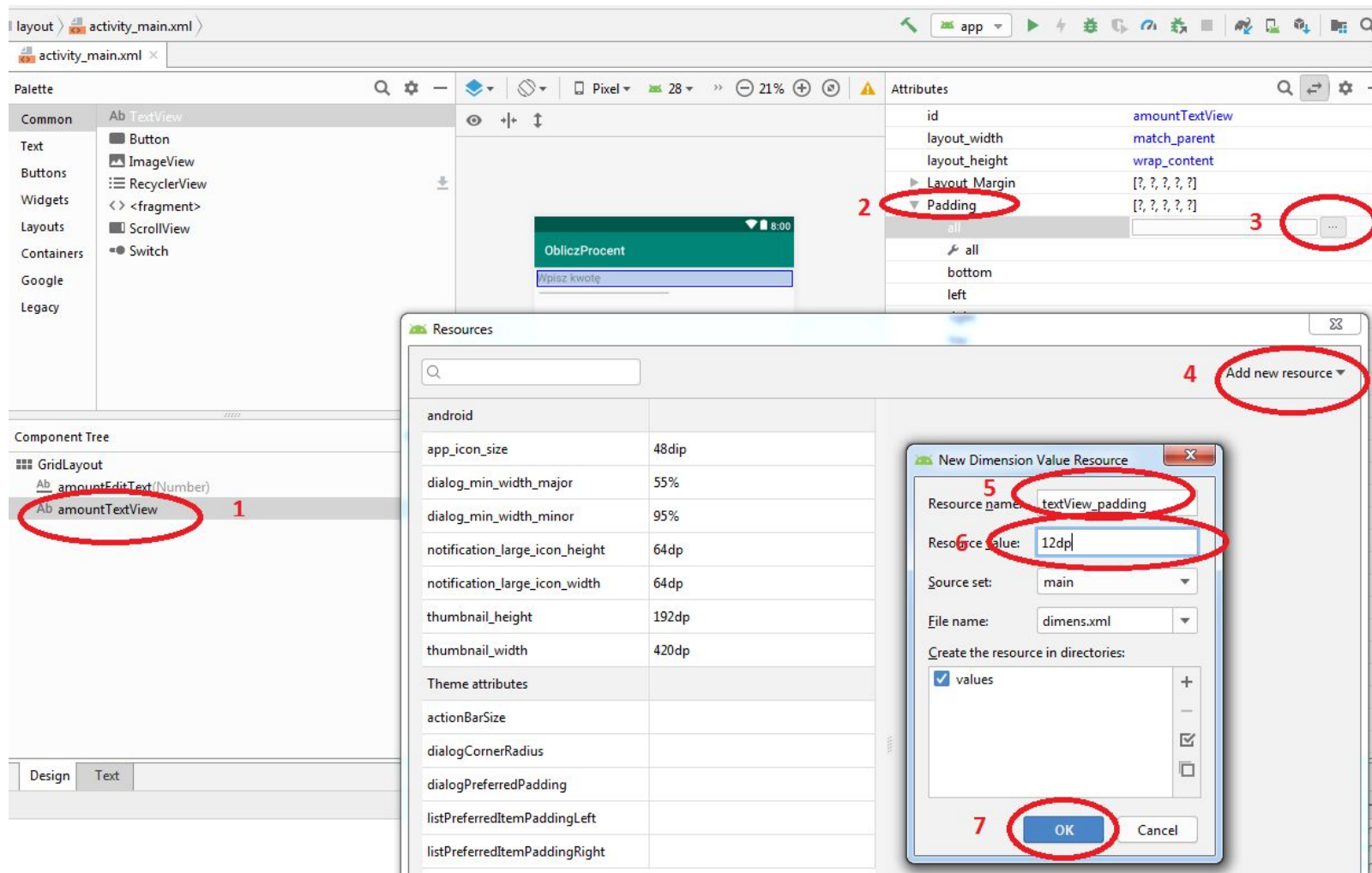




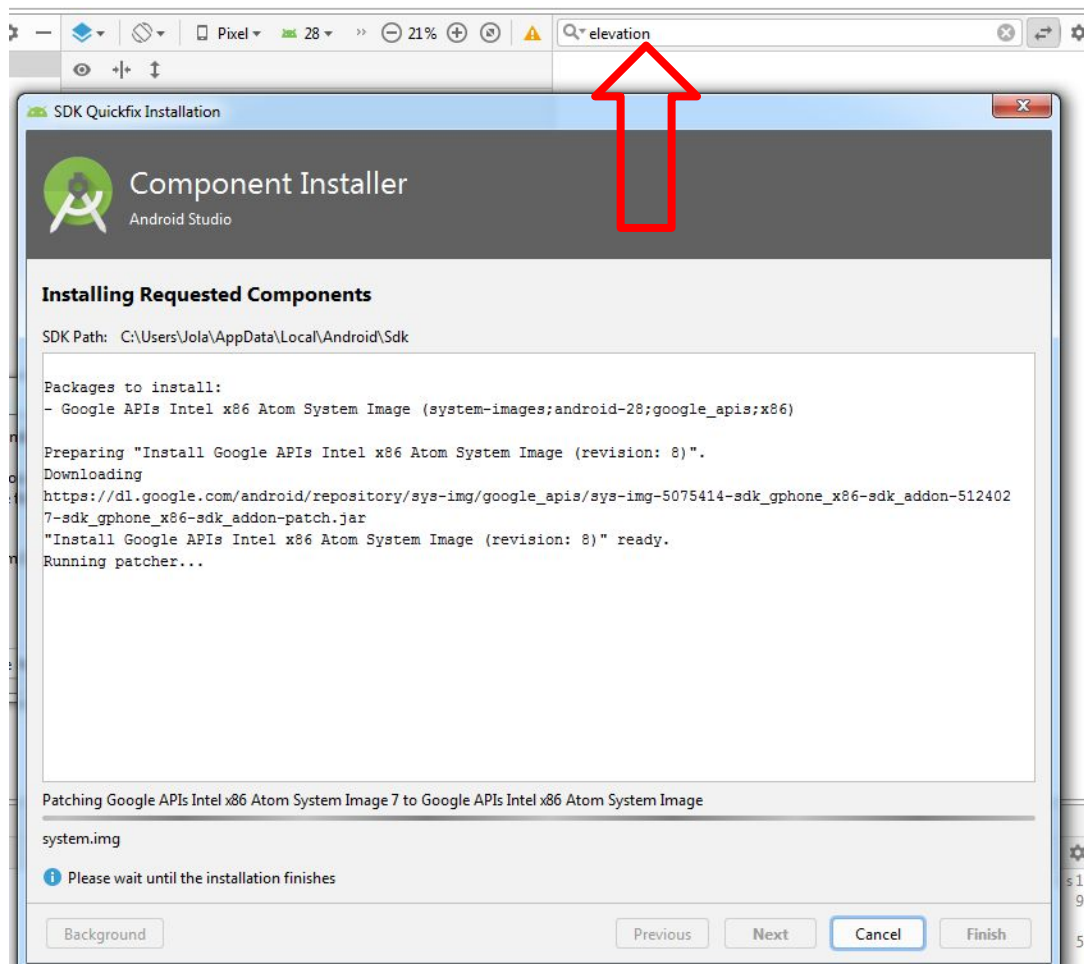






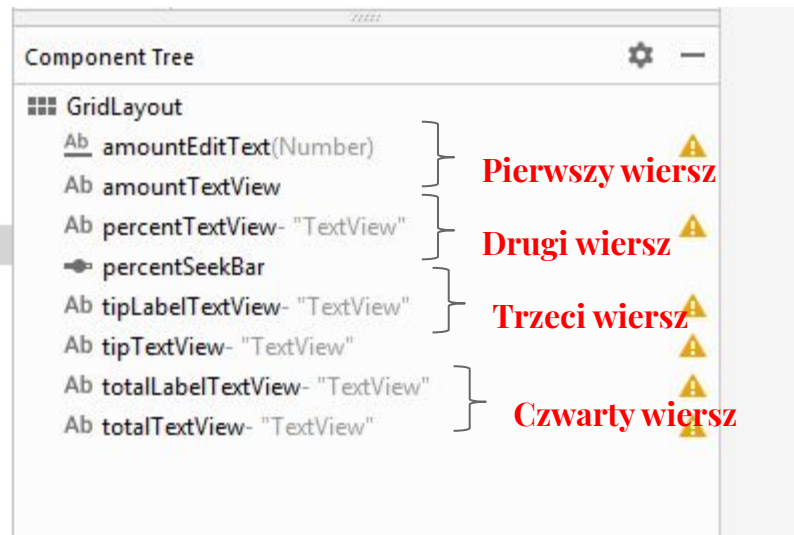
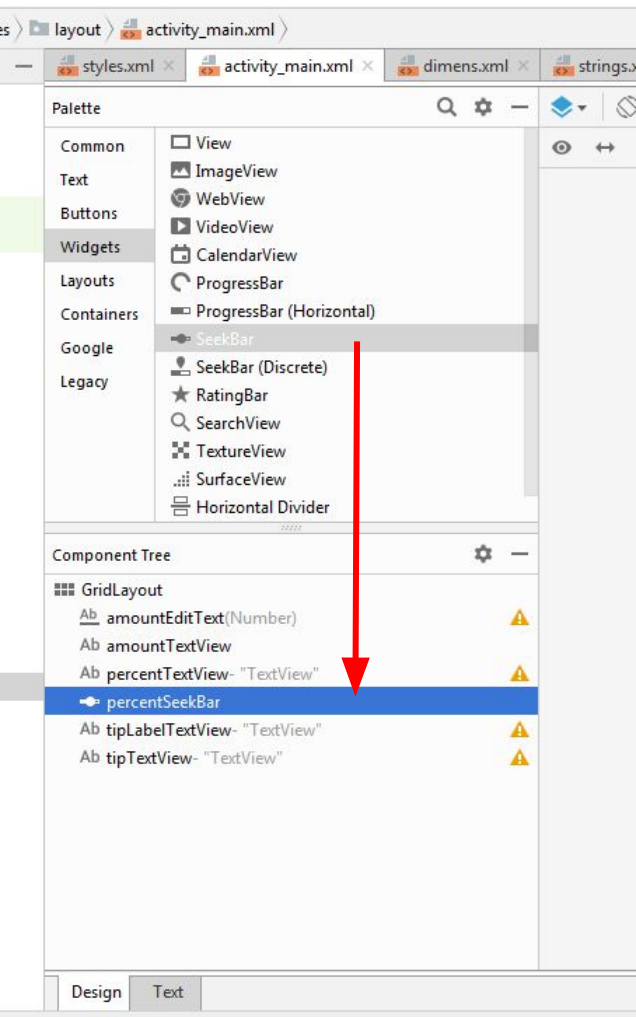
Dodawanie cienia do widoku – dodanie brakującej biblioteki:



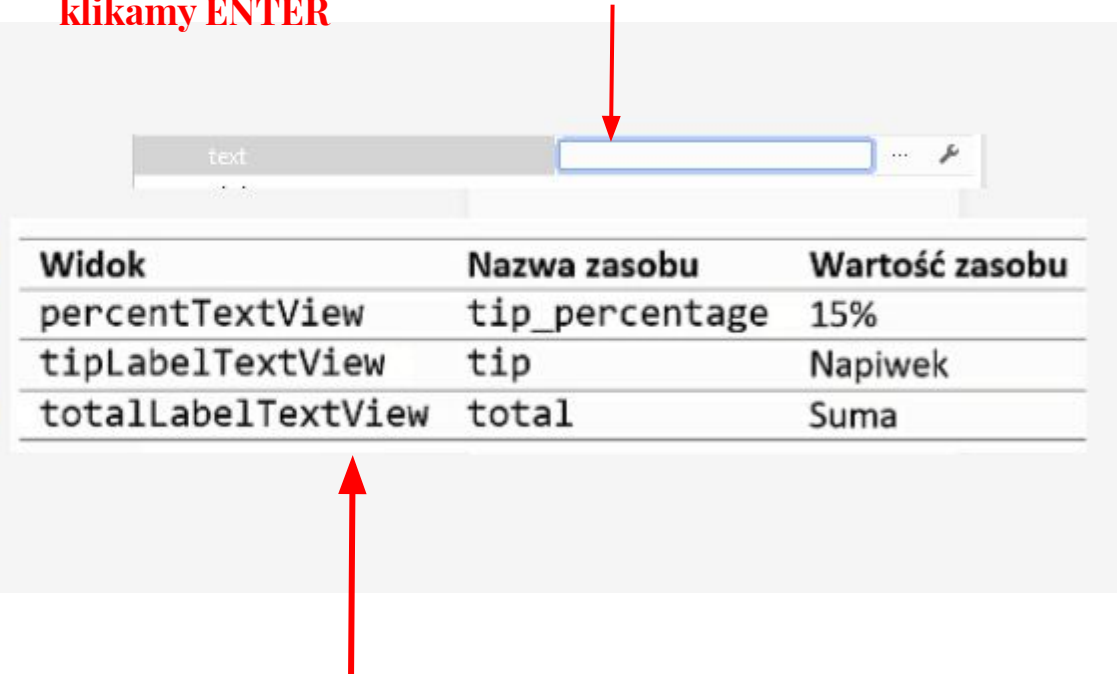
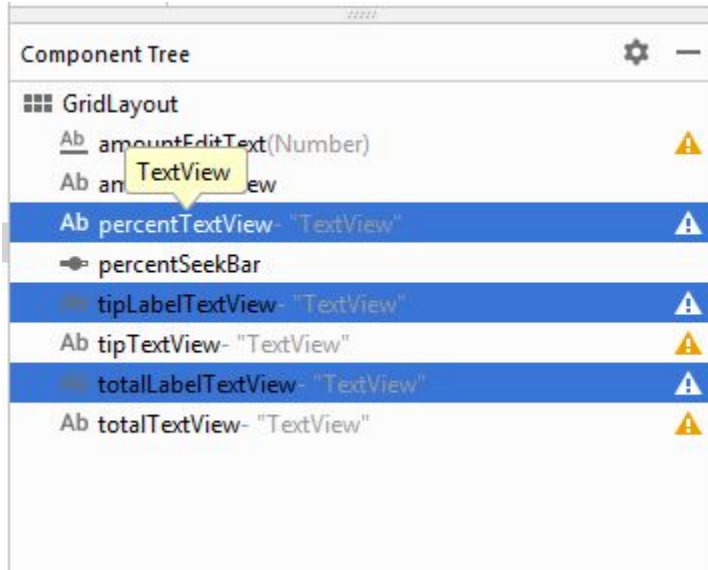
<TextView

```
    android:id="@+id/amountTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_row="0"
    android:layout_rowSpan="2"
    android:layout_column="0"
    android:layout_gravity="fill_horizontal"
    android:background="@color/amount_background"
    android:elevations="@dimens/elevation"
    android:hint="@string/enter_amount"
    android:padding="@dimen/textView_padding"
    android:textAppearance="@android:style/TextAppearance.Medium" />
```

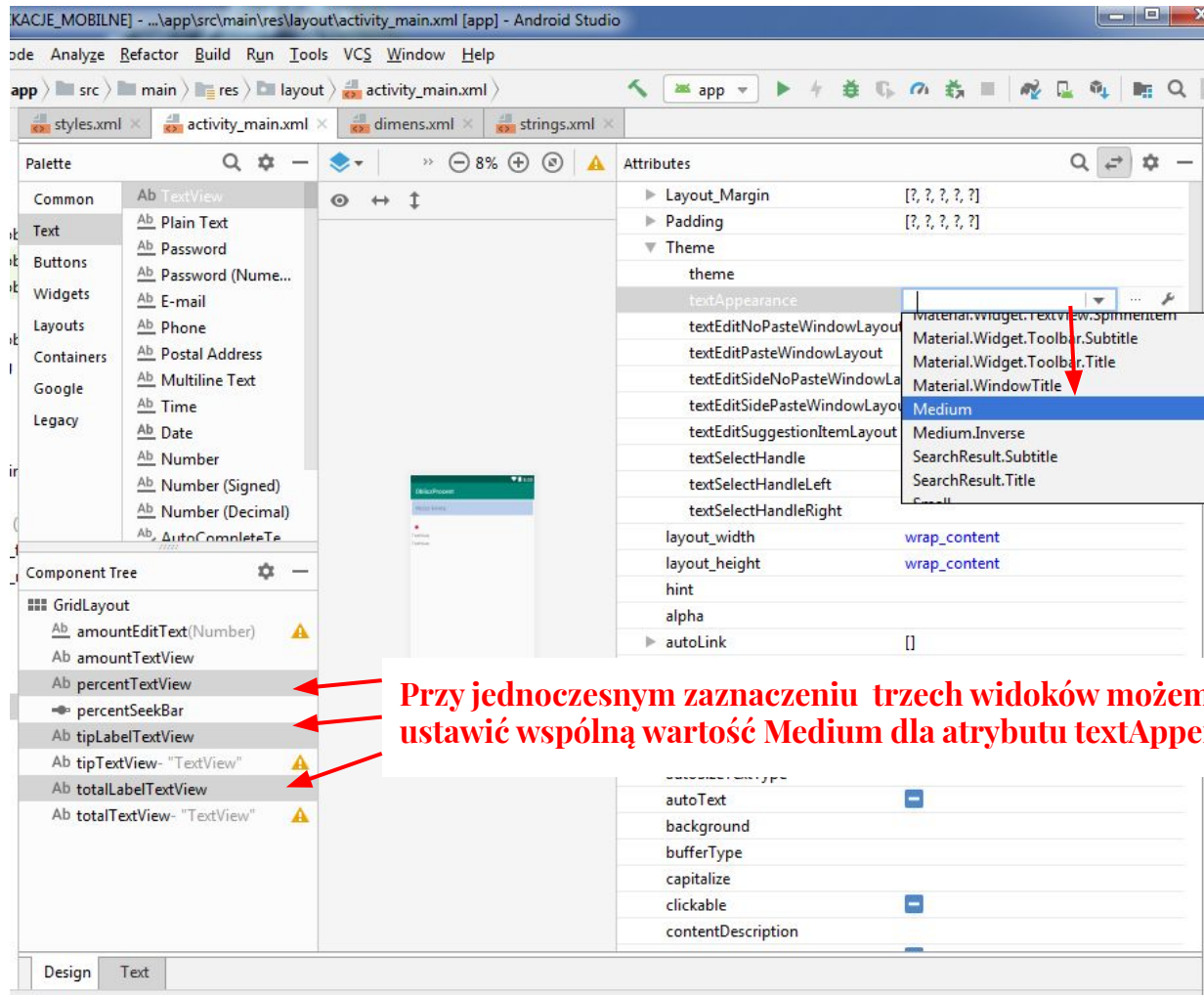




Przy zaznaczonych trzech widokach, znajdujemy atrybut text kasujemy zawartość. Następnie klikamy ENTER



Należy stworzyć zasoby dla atrybutu hint dla trzech zaznaczonych widoków



Przy jednoczesnym zaznaczeniu trzech widoków możemy jednocześnie ustawić wspólną wartość Medium dla atrybutu textAppearance

Wyrównanie do prawej:

The screenshot displays the Android Studio IDE with the following components:

- Palette:** A list of widget categories on the left, including Common, Text, Buttons, Widgets, Layouts, Containers, Google, and Legacy. The 'Text' category is selected, showing various text-related widgets like Plain Text, Password, E-mail, Phone, Postal Address, Multiline Text, Time, Date, Number, and Number (Signed/Decimal).
- Component Tree:** A list of components in the layout, including GridLayout, amountEditText(Number), amountTextView, percentTextView, percentSeekBar, tipLabelTextView, tipTextView- "TextView", totalLabelTextView, and totalTextView- "TextView".
- Design View:** A central area showing a mobile app interface with a green header bar labeled 'ObliczProcent', a blue input field labeled 'Wpisz kwotę', and a red dot below it. The status bar at the top shows '8:00' and the bottom navigation bar is visible.
- Attributes:** A panel on the right showing the attributes for the selected widget. The 'right' attribute under the 'layout_gravity' group is selected and checked.

Attribute	Value
freezesText	<input type="checkbox"/>
gravity	<input type="checkbox"/>
hapticFeedbackEnabled	<input type="checkbox"/>
height	<input type="checkbox"/>
imeActionId	<input type="checkbox"/>
imeActionLabel	<input type="checkbox"/>
imeOptions	<input type="checkbox"/>
includeFontPadding	<input type="checkbox"/>
inputMethod	<input type="checkbox"/>
inputType	<input type="checkbox"/>
isScrollContainer	<input type="checkbox"/>
keepScreenOn	<input type="checkbox"/>
lastBaselineToBottomHeight	<input type="checkbox"/>
layerType	<input type="checkbox"/>
layout_column	<input type="checkbox"/>
layout_columnSpan	<input type="checkbox"/>
layout_gravity	[right]
bottom	<input type="checkbox"/>
clip_horizontal	<input type="checkbox"/>
center	<input type="checkbox"/>
clip_vertical	<input type="checkbox"/>
start	<input type="checkbox"/>
right	<input checked="" type="checkbox"/>
center_horizontal	<input checked="" type="checkbox"/>
fill	<input type="checkbox"/>
fill_horizontal	<input type="checkbox"/>
top	<input type="checkbox"/>
left	<input type="checkbox"/>
center_vertical	<input type="checkbox"/>

styles.xml × activity_main.xml × dimens.xml × strings.xml ×

Palette

Common

Text

Buttons

Widgets

Layouts

Containers

Google

Legacy

Ab TextView

Ab Plain Text

Ab Password

Ab Password (Nume...

Ab E-mail

Ab Phone

Ab Postal Address

Ab Multiline Text

Ab Time

Ab Date

Ab Number

Ab Number (Signed)

Ab Number (Decimal)

Ab AutoCompleteTe

Component Tree

GridLayout

Ab amountEditText(Number) ⚠

Ab amountTextView ⚠

Ab percentTextView ⚠

Ab percentSeekBar ⚠

Ab tipLabelTextView ⚠

Ab tipTextView- "TextView" ⚠

Ab totalLabelTextView ⚠

Ab totalTextView- "TextView" ⚠

Oblicz Procent

Wpisz kwotę

TextView

TextView

Attributes

id percentTextView

layout_width wrap_content

layout_height wrap_content

Layout_Margin [?, ?, ?, ?]

Padding [?, ?, ?, ?]

Theme

hint @string/tip_percentage

layout_gravity [right, center_horizontal]

bottom

clip_horizontal

center

clip_vertical

start

right

center_horizontal

fill

fill_horizontal

top

left

center_vertical

fill_vertical

end

alpha

autoLink []

autoSizeMaxTextSize

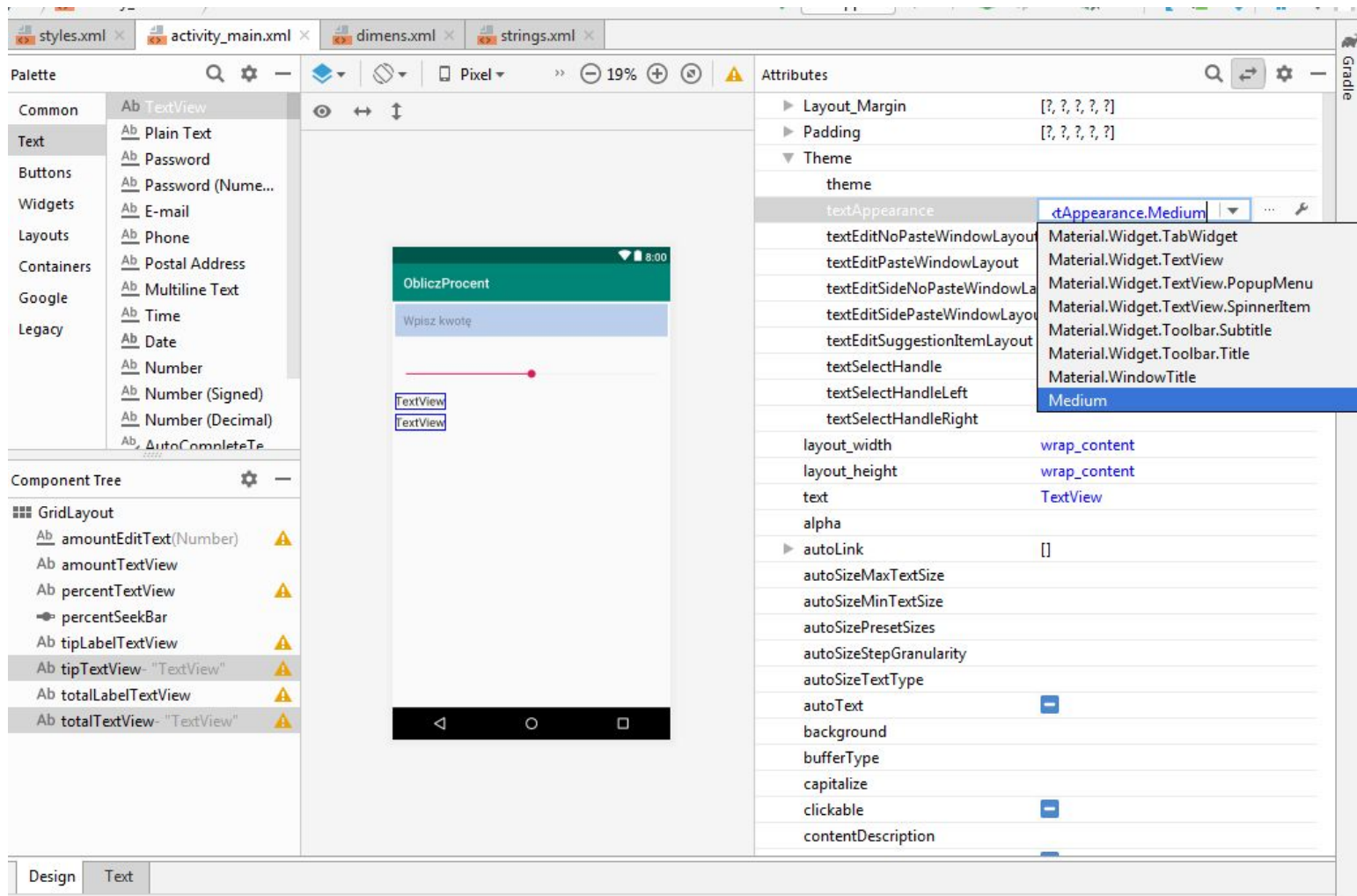
autoSizeMinTextSize

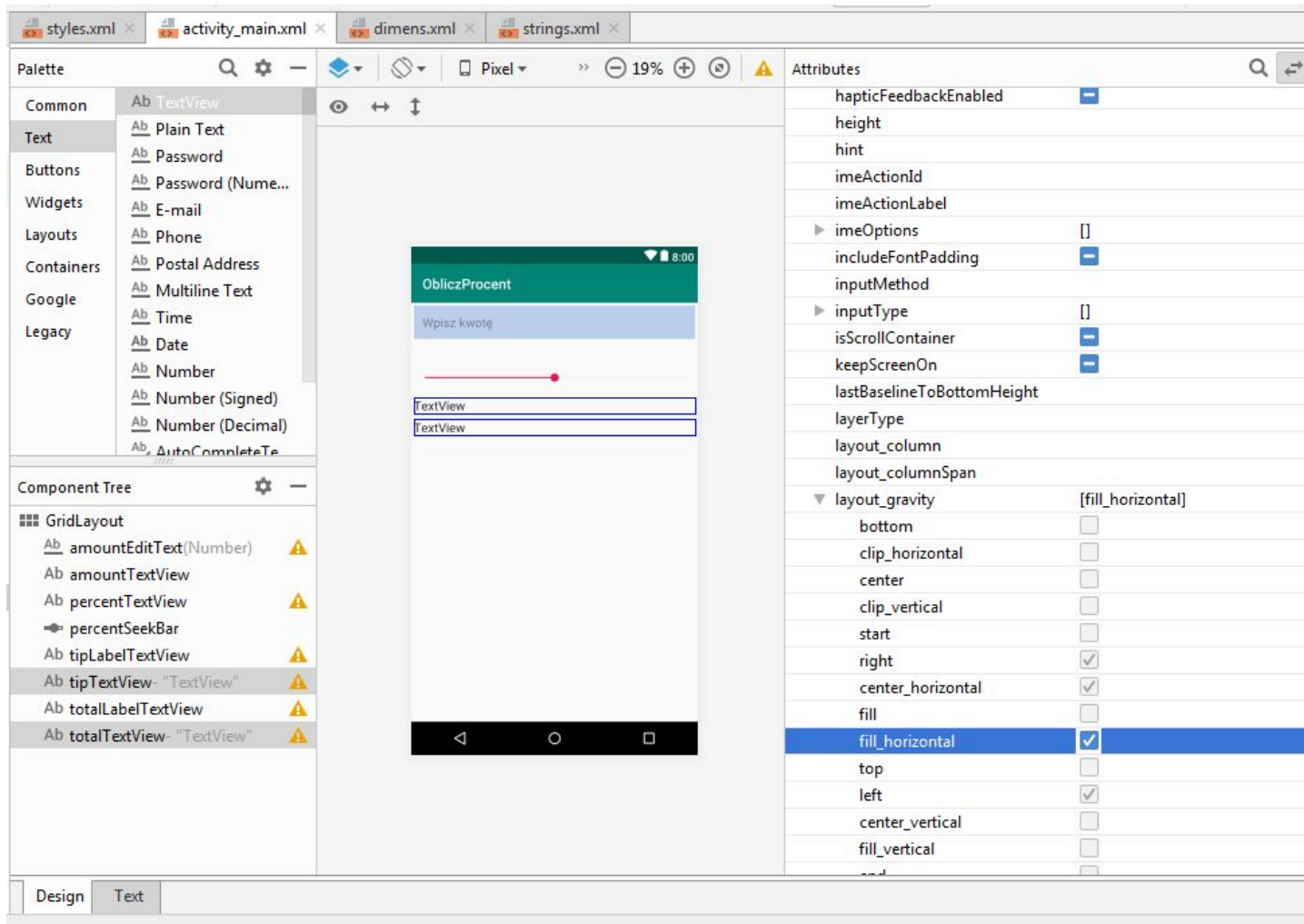
autoSizePresetSizes

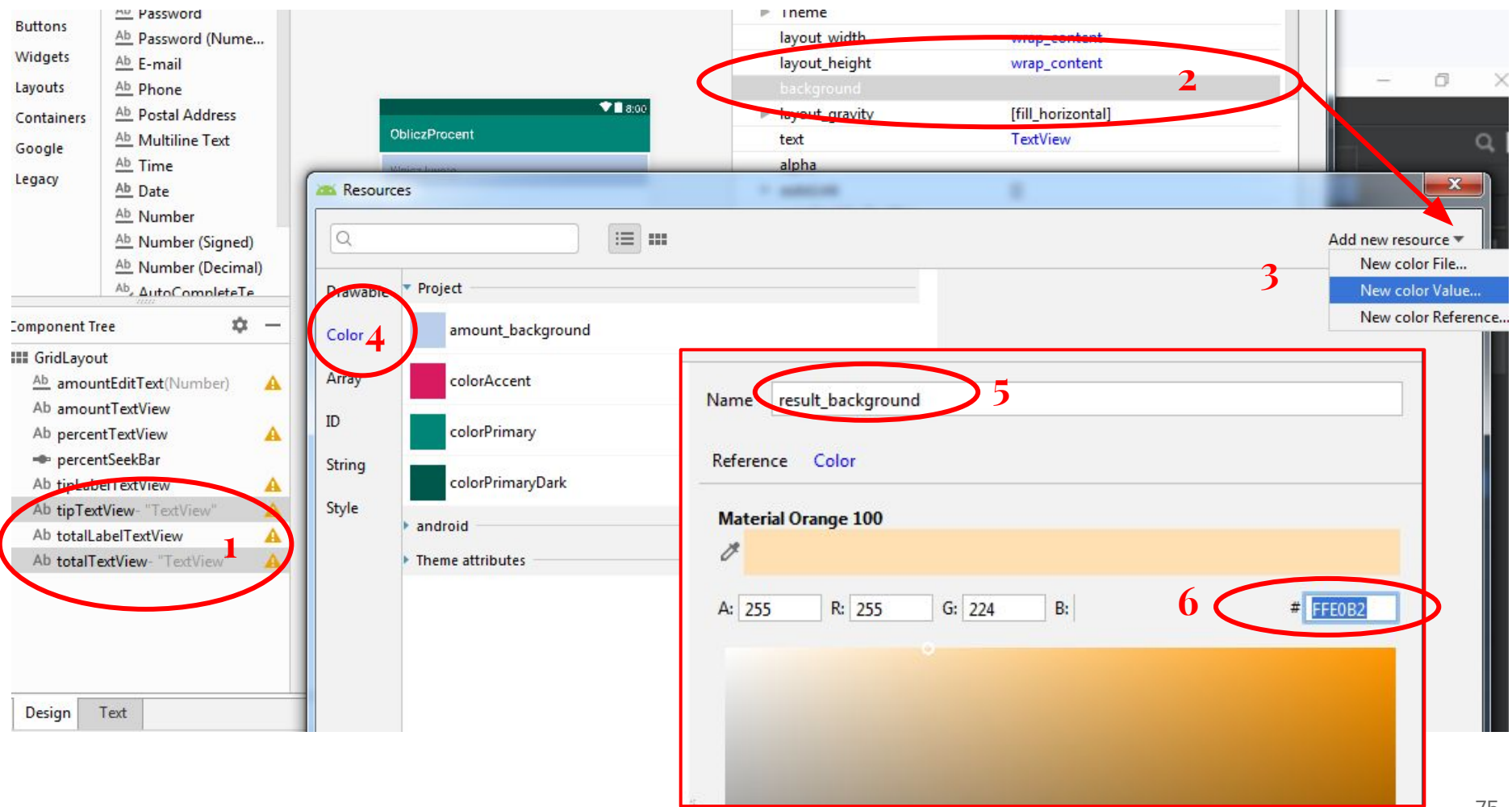
autoSizeStepGranularity

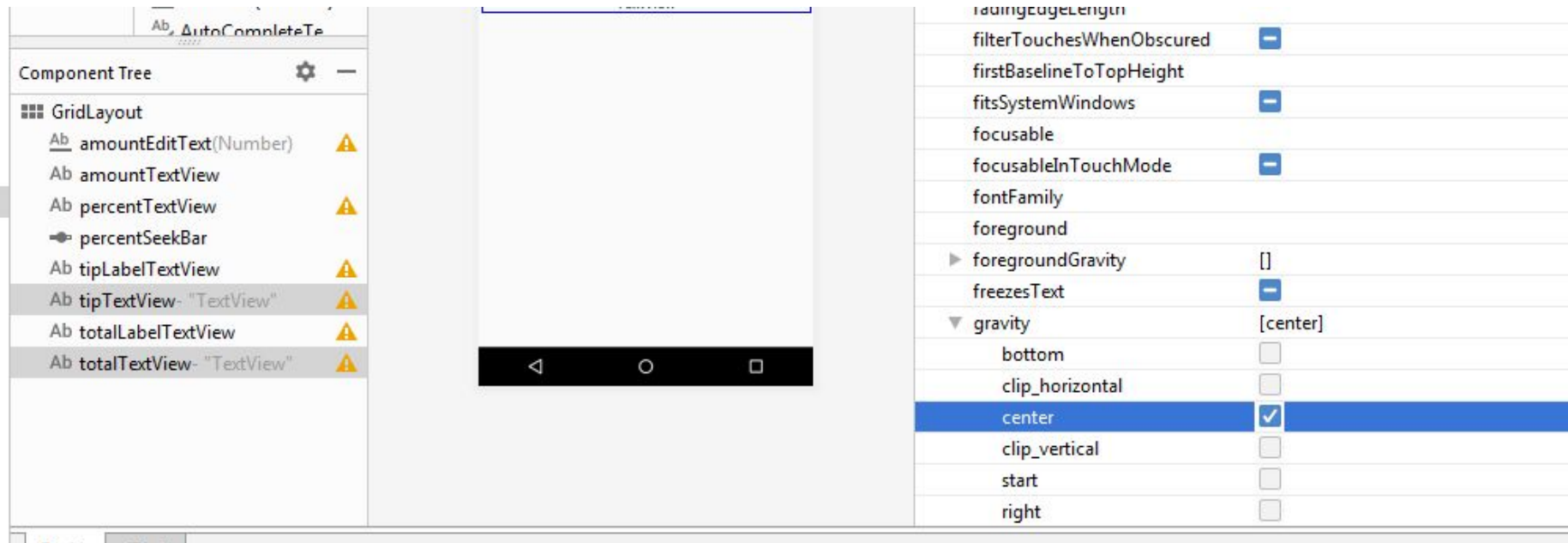
autoSizeTextType

Design Text









src > main > res > layout > activity_main.xml

styles.xml x activity_main.xml x colors.xml x dimens.xml x refs.xml x strings.xml x

Palette

Common

Text

Buttons

Widgets

Layouts

Containers

Google

Legacy

Ab TextView

Ab Plain Text

Ab Password

Ab Password (Numer...

Ab E-mail

Ab Phone

Ab Postal Address

Ab Multiline Text

Ab Time

Ab Date

Ab Number

Ab Number (Signed)

Ab Number (Decimal)

Ab AutoCompleteTex...

Ab MultiAutoComple...

Ab CheckedTextView

Ab TextInputLayout

Component Tree

GridLayout

Ab amountEditText(Number)

Ab amountTextView

Ab percentTextView

Ab percentSeekBar

Ab tipLabelTextView

Ab tipTextView

Ab totalLabelTextView

Ab totalTextView

Pixel 28 AppTheme Default (en-us)

34%

Attributes

Layout_Margin [?, ?, ?, ?]

Padding [12dp, ?, ?, ?]

all @dimen/te

bottom

Resources

Project

elevation 4dp

seekBar_height 40dp

textView_padding 12dp

android

app_icon_size 48dp

dialog_min_width_major 55%

dialog_min_width_minor 95%

notification_large_icon_height 64dp

notification_large_icon_width 64dp

thumbnail_height 192dp

thumbnail_width 192dp

Theme attributes

actionBarSize

dialogCornerRadius

dialogPreferredPadding

listPreferredItemPaddingLeft


Name: textView_padding

Default 12dp

← → ↺

https://developer.android.com/training/material/shadows-clipping

🔍 ⚙️ ⭐️

 Developers

Platform

Android Studio

Google Play

Android Jetpack

Docs

News

🔍 Szukaj

Documentation

OVERVIEW

GUIDES

REFERENCE

SAMPLES

DESIGN & QUALITY

Create shadows and clip views

▶ Text

Buttons

Checkboxes

Radio buttons

Toggle buttons

Spinners

Pickers

Tooltips

▶ Notifications

▶ Add the app bar

▶ Control the system UI visibility

▶ Designing effective navigation

Implementing effective navigation

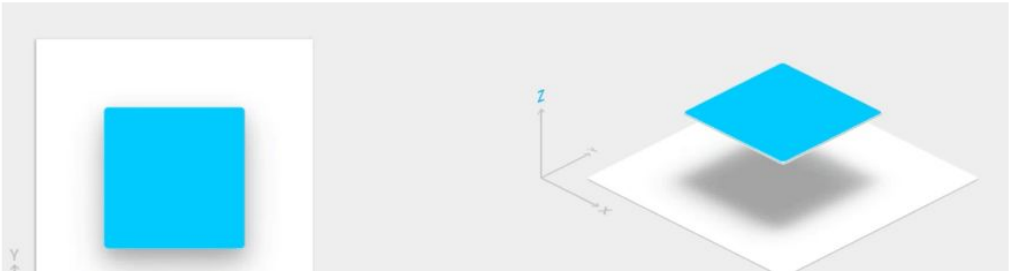
▶ Slide between fragments using ViewPager

Create Shadows and Clip Views

☆☆☆☆☆

Material design introduces elevation for UI elements. Elevation helps users understand the relative importance of each element and focus their attention to the task at hand.

The elevation of a view, represented by the Z property, determines the visual appearance of its shadow: views with higher Z values cast larger, softer shadows. Views with higher Z values occlude views with lower Z values; however, the Z value of a view does not affect the view's size.



Spis treści

Assign Elevation to Your Views

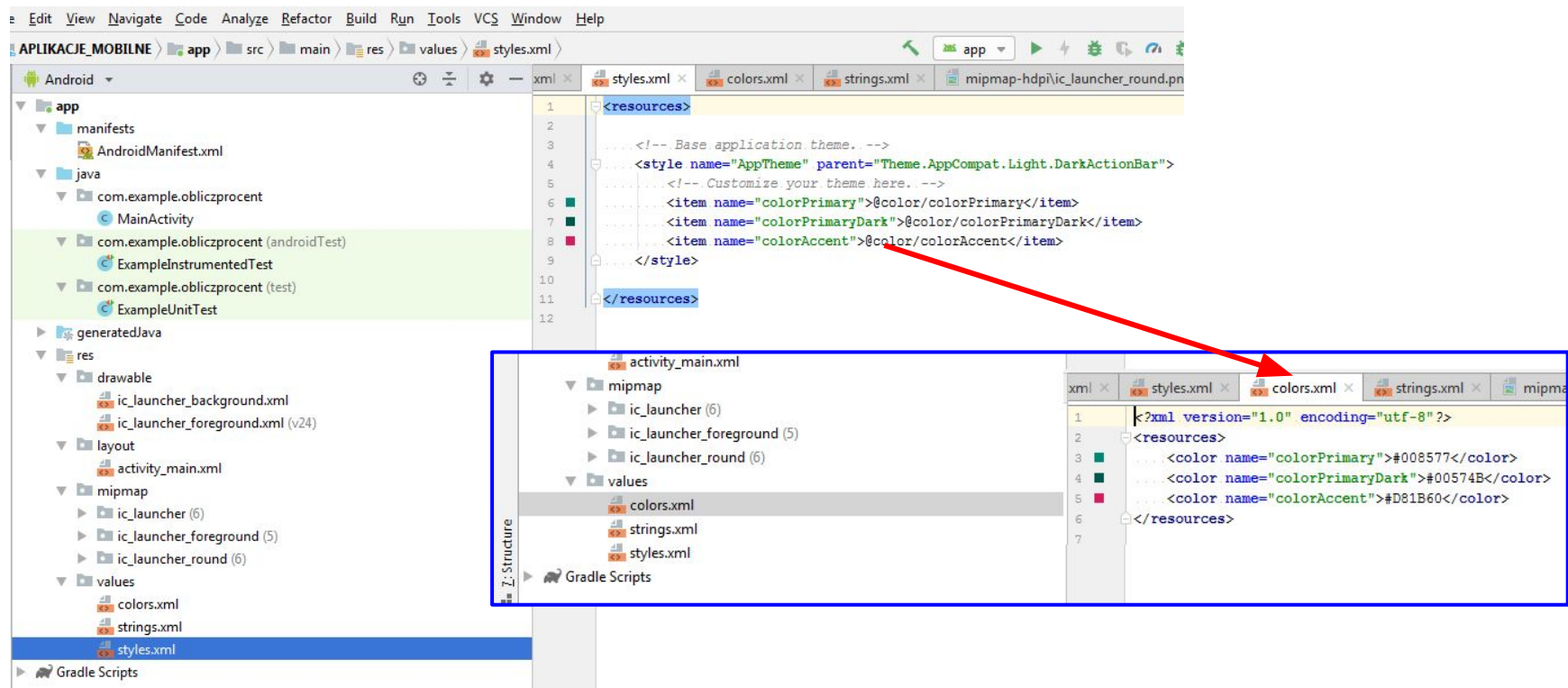
Customize View Shadows and Outlines

Clip Views

Personalizacja kolorów motywu graficznego interfejsu użytkownika

Każda aplikacja posiada główny motyw definiujący jej domyślny wygląd. Motyw kolorystyczny theme jest określony w pliku manifest.xml. Styl zdefiniowany jest w pliku style.xml

```
android:roundIcon="@mipmap/ic_launcher_round"  
android:supportsRtl="true"  
android:theme="@style/AppTheme">  
<activity android:name=".MainActivity">
```



Nazwa stylu.

Motyw AppTheme jest jednym z wielu zdefiniowanych w bibliotece AppCompatActivity. Aby sprawdzić jakie są inne domyślne style, należy, przy przyciśniętym przycisku ctrl kliknąć na wybranym motywie.

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

<!-- Customize your theme here. -->

<item name="colorPrimary">@color/colorPrimary</ item>

<item name="colorPrimaryDark">@color/colorPrimaryDark</ item>

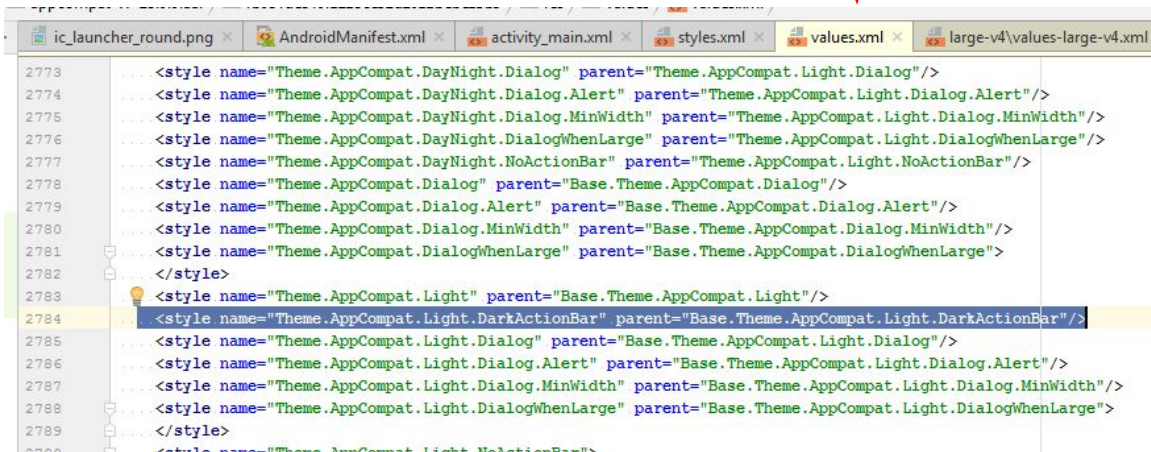
<item name="colorAccent">@color/colorAccent</ item>

</style>

</resources>

Nadpisane wartości

Wokół znajdują się wszystkie inne dostępne motywy



<https://developer.android.com/guide/topics/ui/look-and-feel/themes> -
do zapoznania się

Styles and Themes | Android Dev

← → ↻ <https://developer.android.com/guide/topics/ui/look-and-feel/themes>

Android Developers

Platform Android Studio Google Play Android Jetpack Docs News

🔍 Szukaj

Documentation

OVERVIEW GUIDES REFERENCE SAMPLES DESIGN & QUALITY

Architecture Components

Intents and intent filters

User interface & navigation

Overview

Layouts

Look and feel

Material design

Styles and themes

Adaptive icons

Add a floating action button

Create shadows and clip views

Text

Buttons

Checkboxes

Radio buttons

Toggle buttons

Spinners

Pickers

Tooltips

Notifications

Add the app bar

Control the system UI visibility

Designing effective navigation

Implementing effective navigation

Slide between fragments using

Styles and Themes

☆☆☆☆☆

Styles and themes on Android allow you to separate the details of your app design from the UI structure and behavior, similar to stylesheets in web design.

A *style* is a collection of attributes that specify the appearance for a single [View](#). A style can specify attributes such as font color, font size, background color, and much more.

A *theme* is a type of style that's applied to an entire app, activity, or view hierarchy—not just an individual view. When you apply your style as a theme, every view in the app or activity applies each style attribute that it supports. Themes can also apply styles to non-view elements, such as the status bar and window background.

Styles and themes are declared in a [style resource file](#) in `res/values/`, usually named `styles.xml`.

Material Dark

First Name:

Last Name:

Visit Type: ☒ Business ☐ Social

CONTINUE CANCEL

Material Light

First Name:

Last Name:

Visit Type: ☒ Business ☐ Social

CONTINUE CANCEL

Spis treści

Create and apply a st

Extend and customiz

style

Apply a style as a the

Style hierarchy

TextAppearance

Customize the default

theme

Add version-specifi

styles

Customize widget sty

81

Zaprogramowanie logiki aplikacji

Zdefiniujemy pola dla wszystkich kluczowych elementów naszego interface użytkownika


```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.obliczprocent">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity" android:screenOrientation="portrait"
android:windowSoftInputMode="stateAlwaysVisible">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Funkcje ułatwień dostępu do aplikacji

<https://developer.android.com/guide/topics/ui/accessibility/apps>