# WSTĘP DO ANDROIDA

Laboratorium 3 Systemy i aplikacje bez granic

Tworzymy nowy projekt typu Empty Activity

- Tworzymy nowy projekt typu Empty Activity
- Poniżej Text View wstawiamy przycisk Button

- Tworzymy nowy projekt typu Empty Activity
- Poniżej Text View wstawiamy przycisk Button
- Tekst zmieniamy mu na Uruchom a id na myButton i wciskamy symbol



- Tworzymy nowy projekt typu Empty Activity
- Poniżej Text View wstawiamy przycisk Button
- Tekst zmieniamy mu na Uruchom a id na myButton i wciskamy symbol



Text View zmieniamy id na status Text

 W atrybucie onClick przycisku wpisujemy myButtonClick  W atrybucie onClick przycisku wpisujemy myButtonClick onClick myButtonClick

- W atrybucie onClick przycisku wpisujemy myButtonClick onClick myButtonClick
- W kodzie klasy aktywności dodajemy metodę myButtonClick

- W atrybucie onClick przycisku wpisujemy myButtonClick onClick myButtonClick
- W kodzie klasy aktywności dodajemy metodę myButtonClick
   fun myButtonClick (v: View) {

fun myButtonClick (v: View) {
 statusText.<u>text</u> = "Uruchamiam"
 showActivity()
}

val REQUEST\_CODE=10000

val REQUEST\_CODE=10000

 Do projektu dodajemy nową aktywność Second Activity typu Empty Activity

val REQUEST\_CODE=10000

- Do projektu dodajemy nową aktywność Second Activity typu Empty Activity
- Wstawiamy do niej Text View o id status Text

fun showActivity() {
 val i=Intent( packageContext: this,SecondActivity::class.java)
 i.putExtra( name: "Parametr", value: "Twoje dane")
 startActivityForResult(i,REQUEST\_CODE)
}

fun showActivity() {
 val i=Intent( packageContext: this,SecondActivity::class.java)
 i.putExtra( name: "Parametr", value: "Twoje dane")
 startActivityForResult(i,REQUEST\_CODE)
}

A w SecondActivity dodajemy do onCreate

fun showActivity() {
 val i=Intent( packageContext: this,SecondActivity::class.java)
 i.putExtra( name: "Parametr", value: "Twoje dane")
 startActivityForResult(i,REQUEST\_CODE)
}

## A w SecondActivity dodajemy do onCreate

```
val extras = <u>intent</u>.extras ?: return
val message = extras.getString( key: "Parametr")
statusText.<u>text</u> = message
```

• W SecondActivity dodajemy metodę:

• W SecondActivity dodajemy metodę:

```
override fun finish() {
    val data = Intent()
    data.putExtra( name: "returnString1", value: "Wiadomość dla ciebie")
    setResult(Activity.RESULT_OK,data)
    super.finish()
}
```

W SecondActivity dodajemy metodę:

```
override fun finish() {
    val data = Intent()
    data.putExtra( name: "returnString1", value: "Wiadomość dla ciebie")
    setResult(Activity.RESULT_OK,data)
    super.finish()
}
```

• A w MainActivity:

W SecondActivity dodajemy metodę:

```
override fun finish() {
    val data = Intent()
    data.putExtra( name: "returnString1", value: "Wiadomość dla ciebie")
    setResult(Activity.RESULT_OK,data)
    super.finish()
}
```

• A w MainActivity:

• Uruchamiamy

- Uruchamiamy
- Do MainActivity dodajemy metodę:

- Uruchamiamy
- Do MainActivity dodajemy metodę:

```
fun showImplicitActivity() {
    val i = Intent(Intent.ACTION_VIEW, Uri.parse( uriString: "http://www.predki.com"))
    startActivity(i)
}
```

- Uruchamiamy
- Do MainActivity dodajemy metodę:

```
fun showImplicitActivity() {
    val i = Intent(Intent.ACTION_VIEW, Uri.parse( uriString: "http://www.predki.com"))
    startActivity(i)
}
```

I podmieniamy jej wywołanie w obsłudze przycisku

 Utworzyć nowy projekt typu Empty Activity o nazwie BroadcastSample i aktywności SendBroadcastActivity

- Utworzyć nowy projekt typu Empty Activity o nazwie BroadcastSample i aktywności SendBroadcastActivity
- Usunąć Text View i wstawić w to miejsce Button z tekstem "Wyślij"

- Utworzyć nowy projekt typu Empty Activity o nazwie BroadcastSample i aktywności SendBroadcastActivity
- Usunąć Text View i wstawić w to miejsce Button z tekstem "Wyślij"
- W atrybucie on Click wpisać "broadcastIntent"
Ш

import android.content.BroadcastReceiver import android.content.Intent import android.content.IntentFilter import android.support.v7.app.AppCompatActivity import android.os.Bundle import android.view.View

import android.content.BroadcastReceiver import android.content.Intent import android.content.IntentFilter import android.support.v7.app.AppCompatActivity import android.os.Bundle import android.view.View

Dodajemy metodę

import android.content.BroadcastReceiver import android.content.Intent import android.content.IntentFilter import android.support.v7.app.AppCompatActivity import android.os.Bundle import android.view.View

Dodajemy metodę

```
fun broadcastIntent(view: View)
{
    val intent = Intent()
    intent.action = "com.predki.sendbroadcast"
    intent.flags = Intent.FLAG_INCLUDE_STOPPED_PACKAGES
    sendBroadcast(intent)
}
```

Dodajemy nową klasę do projektu

Ш

## Dodajemy nową klasę do projektu

▶ manifests ▼ miava		1	<pre>package com.predki.broadcastsample</pre>	2
	nom prodki brogdogstegmala		<pre>import android.content.Intent</pre>	
<ul> <li>Dic</li> <li>Dic</li> <li>Dic</li> <li>Dic</li> </ul>	New Link C++ Project with Gradle	×	G Java Class Kotlin File/Class Android resource file	mpatActivity
Gradle	<ul> <li>✓ Cut</li> <li>☑ Copy</li> <li>Copy Path</li> <li>Copy as Plain Text</li> <li>Copy Reference</li> </ul>	X೫ ጋ೫ ጋ೫ጉ ጋ೫ጉፓ	<ul> <li>Android resource directory</li> <li>Sample Data directory</li> <li>File</li> <li>Scratch File</li> <li>Package</li> </ul>	<pre>mpatActivity() { anceState: Bundle?) { eState) ivity_send_broadcast)</pre>
	<ul> <li>Paste</li> <li>Find Usages</li> <li>Find in Path</li> <li>Replace in Path</li> </ul>	策V ℃F7 企業F 企業R	<ul> <li>C++ Class</li> <li>C/C++ Source File</li> <li>C/C++ Header File</li> <li>Image Asset</li> </ul>	.sendbroadcast" INCLUDE_STOPPED_PACKAGES
	Analyze		뼦 Vector Asset	
	Refactor Add to Favorites Show Image Thumbnails	◄ ♦ ℃೫T	<ul> <li>Singleton</li> <li>Gradle Kotlin DSL Build Script</li> <li>Gradle Kotlin DSL Settings</li> </ul>	l Script ings
	Reformat Code Optimize Imports	て第L ヘンの	Edit File Templates	
	Delete	$\boxtimes$	Activity	•
	<ul> <li>Run 'Tests in 'com.predki.broadcast'</li> <li>Debug 'Tests in 'com.predki.broadcast'</li> <li>Run 'Tests in 'com.predki.broadcast' with Com.predki.broadcast'</li> </ul>	个仓R 个仓D overage	Android Auto     Android Auto     Folder     Fragment     Google	> > >
	Create 'Tests in 'com.predki.broadcastsample		🔮 Other	📕 🛄 Android Manifest File
	Local History Synchronize 'broadcastsample'	Þ	Service     Vi Component     Wear	<ul> <li>Broadcast Receiver</li> <li>Content Provider</li> <li>Daydream</li> </ul>
	Compare With	ЖD	₩ Widget XML Resource Bundle	

Modyfikujemy kod tej klasy

### Modyfikujemy kod tej klasy

### package com.predki.broadcastsample

```
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast
```

```
class MyReceiver : BroadcastReceiver() {
```

override fun onReceive(context: Context, intent: Intent) {
 Toast.makeText(context, text: "Broadcast Intent Detected.",
 Toast.LENGTH\_LONG).show()

Ш

pole

• pole

var receiver: BroadcastReceiver? = null

Ш

- Dodajemy do klasy aktywności:
  - pole

var receiver: BroadcastReceiver? = null

Н

• metody

• pole

### var receiver: BroadcastReceiver? = null

metody

```
private fun configureReceiver() {
    val filter = IntentFilter()
    filter.addAction( action: "com.predki.sendbroadcast")
    receiver = MyReceiver()
    registerReceiver(receiver, filter)
}

override fun onDestroy() {
    super.onDestroy()
    unregisterReceiver(receiver)
}
```

 i wywołanie metody configureReceiver() na końcu onCreate

Н

 i wywołanie metody configureReceiver() na końcu onCreate

override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.activity\_send\_broadcast)
 configureReceiver()
}

 i wywołanie metody configureReceiver() na końcu onCreate

override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.activity\_send\_broadcast)
 configureReceiver()

• Uruchamiamy



# 

 Tworzymy nowy projekt typu Basic Activity o nazwie MenuExample

- Tworzymy nowy projekt typu Basic Activity o nazwie MenuExample
- Rozwijamy projekt app\res\menu i otwieramy menu\_main.xml

- Tworzymy nowy projekt typu Basic Activity o nazwie MenuExample
- Rozwijamy projekt app\res\menu i otwieramy menu\_main.xml
- Usuwamy z Component Tree dotychczasowe menu Settings

- Tworzymy nowy projekt typu Basic Activity o nazwie MenuExample
- Rozwijamy projekt app\res\menu i otwieramy menu\_main.xml
- Usuwamy z Component Tree dotychczasowe menu Settings
- Z palety dodajemy nowe Menu



### Do menu dodajemy element Group

- Do menu dodajemy element Group
- Dla grupy ustawiamy atrybut checkableBehavior na single

Ш

- Do menu dodajemy element Group
- Dla grupy ustawiamy atrybut checkableBehavior na group id single checkableBeh single



- Do menu dodajemy element Group
- Dla grupy ustawiamy atrybut checkableBehavior na group id single checkableBeh single



Do grupy dodajemy po kolei 4 elementy Menu Item •

- Do menu dodajemy element Group
- Dla grupy ustawiamy atrybut checkableBehavior na id single checkableBeh single



- Do grupy dodajemy po kolei 4 elementy Menu Item
- Ustawiamy ich id i text, żeby odpowiadały 4 kolorom: • czerwony, zielony, żółty, niebieski

- Do menu dodajemy element Group
- Dla grupy ustawiamy atrybut checkableBehavior na single
- Do grupy dodajemy po kolei 4 elementy Menu Item
- Ustawiamy ich id i text, żeby odpowiadały 4 kolorom: czerwony, zielony, żółty, niebieski

item		
id	menu_red	
title	Czerwony	
icon		
showAsActior		

enabled



```
<menu xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.predki.menuapplication.MainActivity">
    <group android:checkableBehavior="single" >
        <item
            android:id="@+id/menu_red"
            android:title="Czerwony" />
        <item
            android:id="@+id/menu_green"
            android:title="Zielony" />
        <item
            android:id="@+id/menu_yellow"
            android:title="Żółty" />
        <item
            android:id="@+id/menu_blue"
            android:title="Niebieski" />
    </group>
</menu>
```



# Zmieniamy id CoordinatorLayout w activity\_main na layoutView

Ш
## Zmieniamy id CoordinatorLayout w activity\_main na layoutView

Ш

Zmieniamy kod metody



