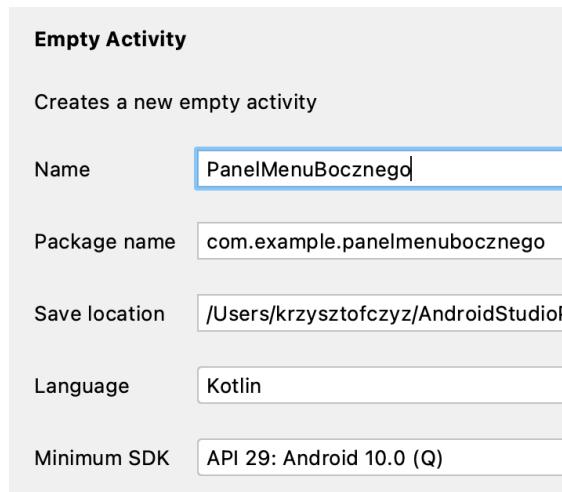
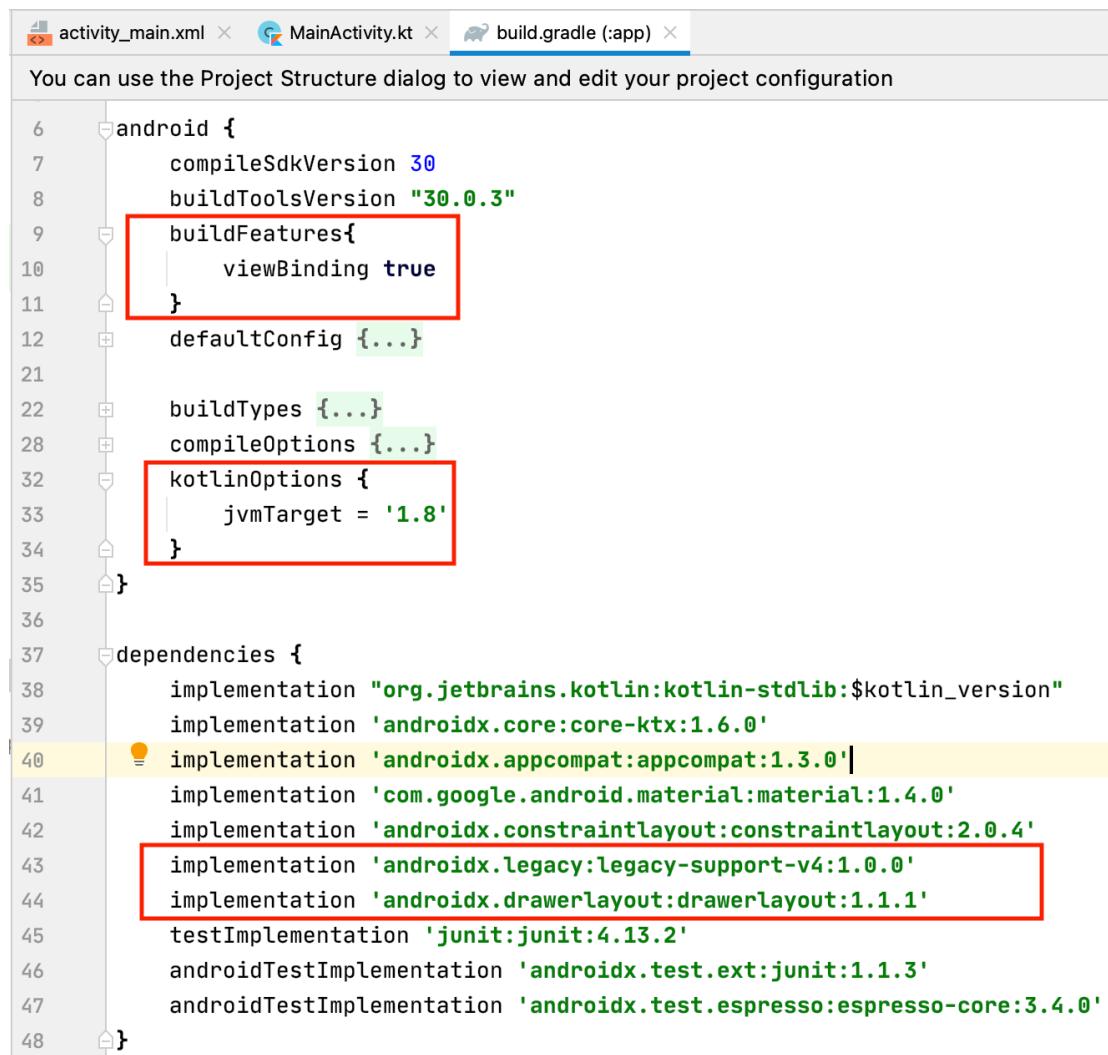


## Boczny panel zawierający Menu

Uruchamiamy nowy projekt z pustą aktywnością:



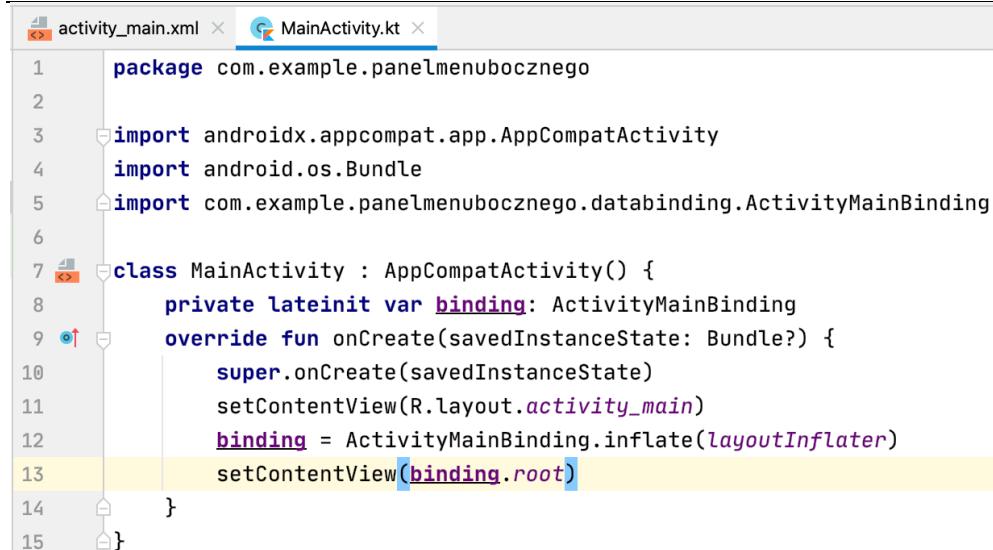
W pliku build.gradle dodajemy obsługę viewBinding oraz bibliotekę obsługującą design



```

6  android {
7      compileSdkVersion 30
8      buildToolsVersion "30.0.3"
9      buildFeatures{
10         viewBinding true
11     }
12     defaultConfig {...}
13
14     buildTypes {...}
15     compileOptions {...}
16     kotlinOptions {
17         jvmTarget = '1.8'
18     }
19
20 }
21
22 dependencies {
23     implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
24     implementation 'androidx.core:core-ktx:1.6.0'
25     implementation 'androidx.appcompat:appcompat:1.3.0'
26     implementation 'com.google.android.material:material:1.4.0'
27     implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
28     implementation 'androidx.legacy:legacy-support-v4:1.0.0'
29     implementation 'androidx.drawerlayout:drawerlayout:1.1.1'
30     testImplementation 'junit:junit:4.13.2'
31     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
32     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
33 }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

```

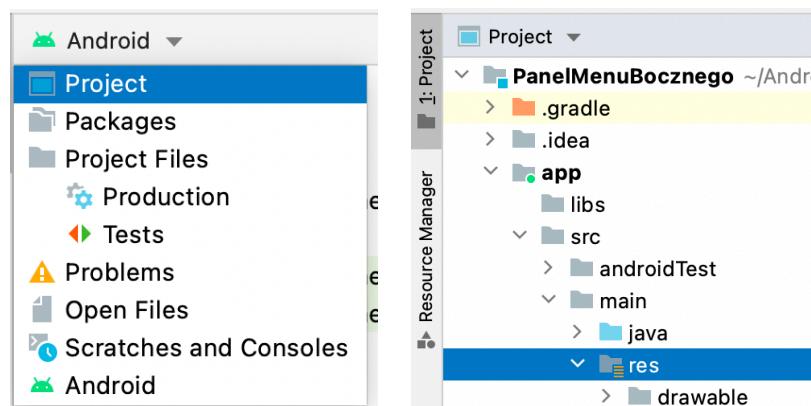


```

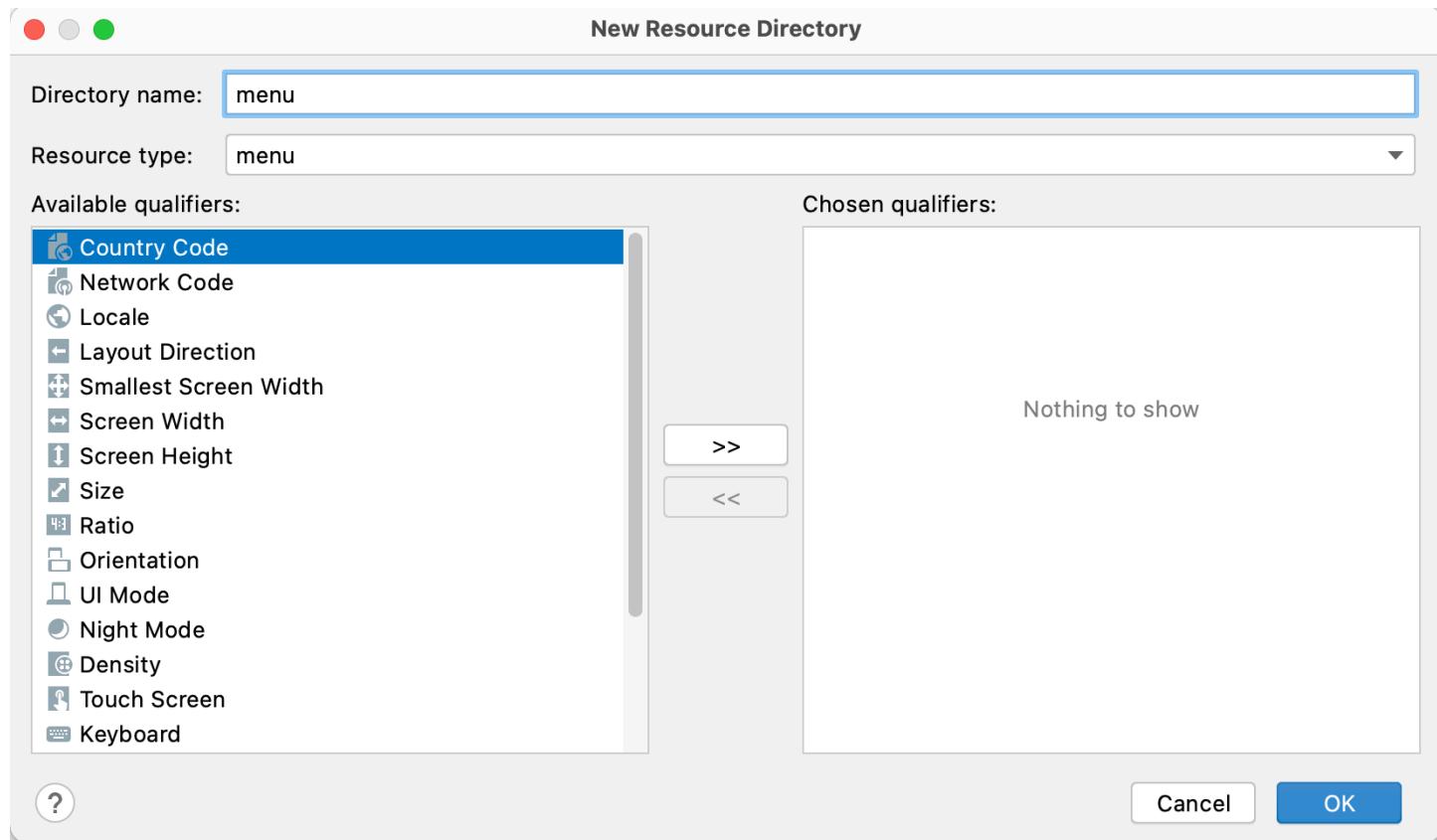
1 package com.example.panelmenubocznego
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import com.example.panelmenubocznego.databinding.ActivityMainBinding
6
7 class MainActivity : AppCompatActivity() {
8     private lateinit var binding: ActivityMainBinding
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12         binding = ActivityMainBinding.inflate(layoutInflater)
13         setContentView(binding.root)
14     }
15 }

```

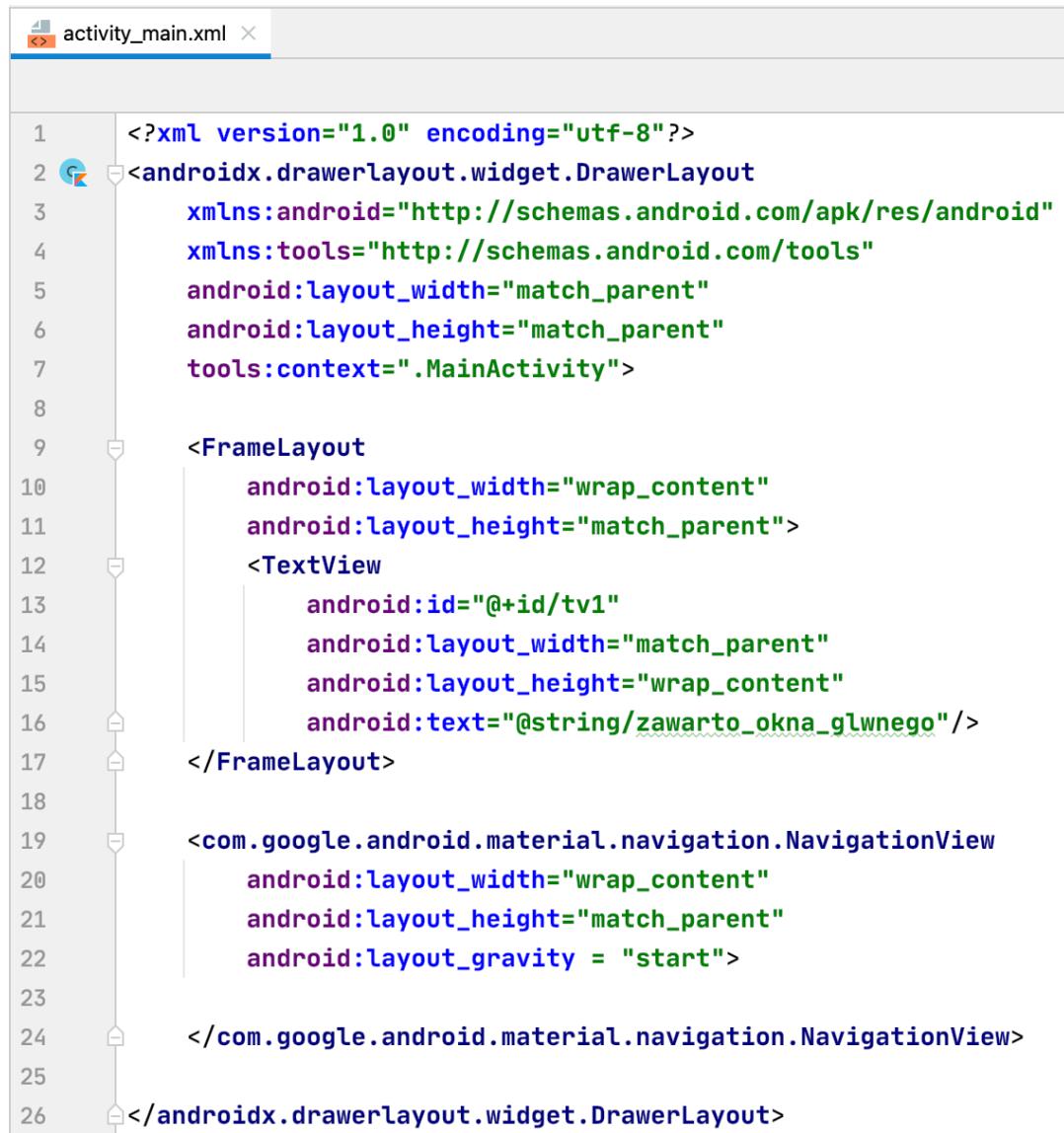
Utworzmy teraz folder menu w którym będzie się znajdować menu paska bocznego. Przełączamy się na widok projektu i tworzymy folder:



klikamy prawym przyciskiem w 'res':

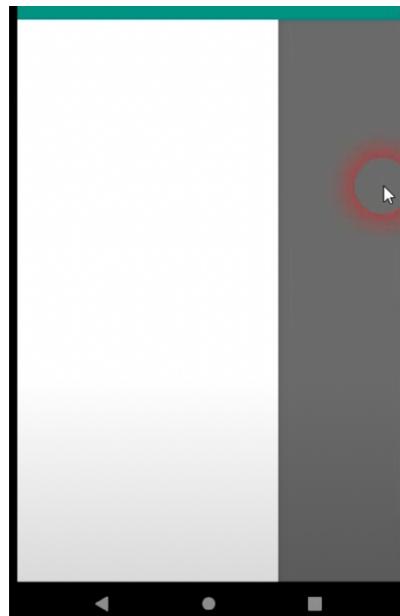


Modyfikujemy nasz główny layout

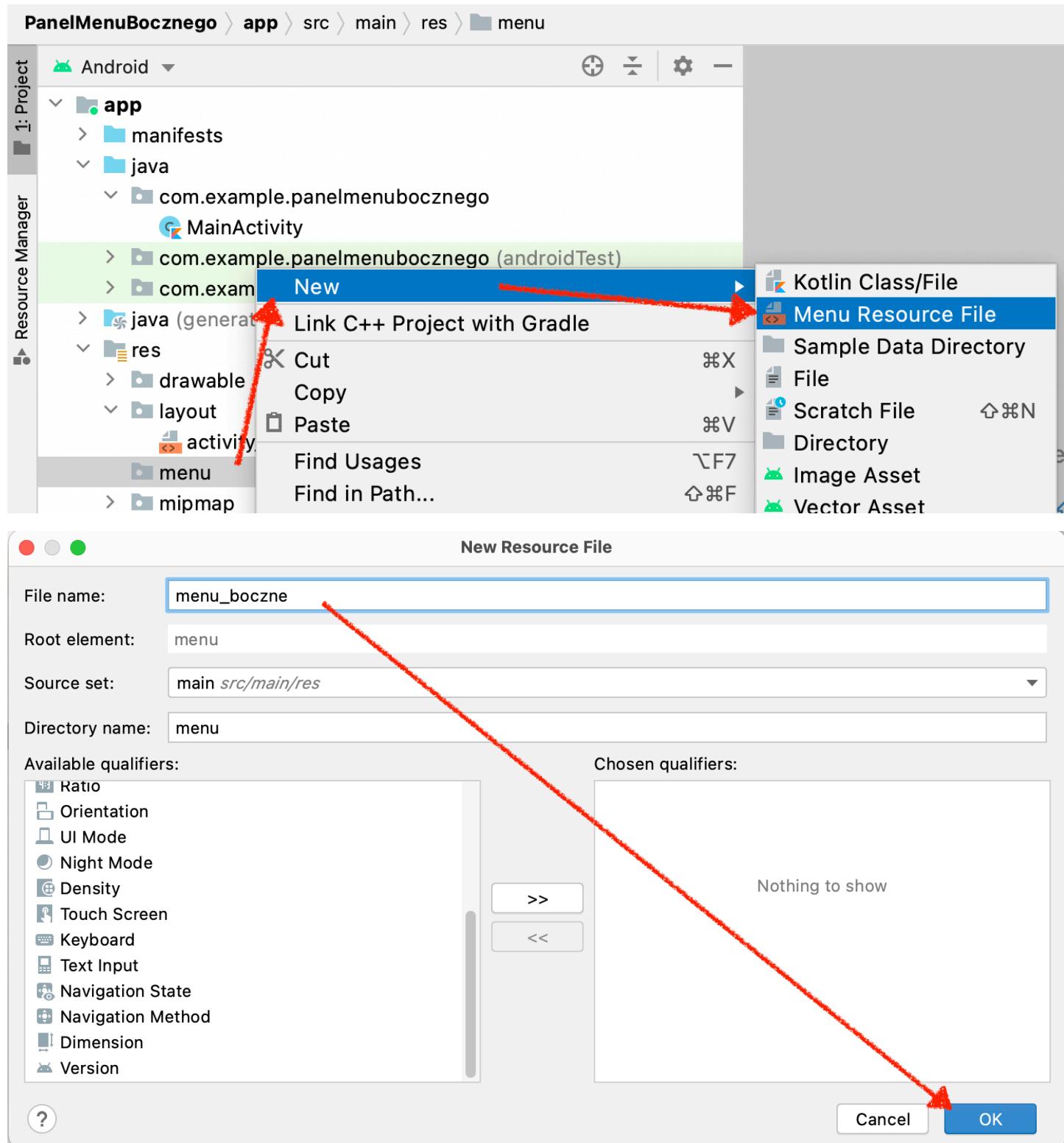


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.drawerlayout.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <FrameLayout
10        android:layout_width="wrap_content"
11        android:layout_height="match_parent">
12         <TextView
13             android:id="@+id/tv1"
14             android:layout_width="match_parent"
15             android:layout_height="wrap_content"
16             android:text="@string/zawarto_okna_glwnego"/>
17     </FrameLayout>
18
19     <com.google.android.material.navigation.NavigationView
20         android:layout_width="wrap_content"
21         android:layout_height="match_parent"
22         android:layout_gravity = "start">
23
24     </com.google.android.material.navigation.NavigationView>
25
26 </androidx.drawerlayout.widget.DrawerLayout>
```

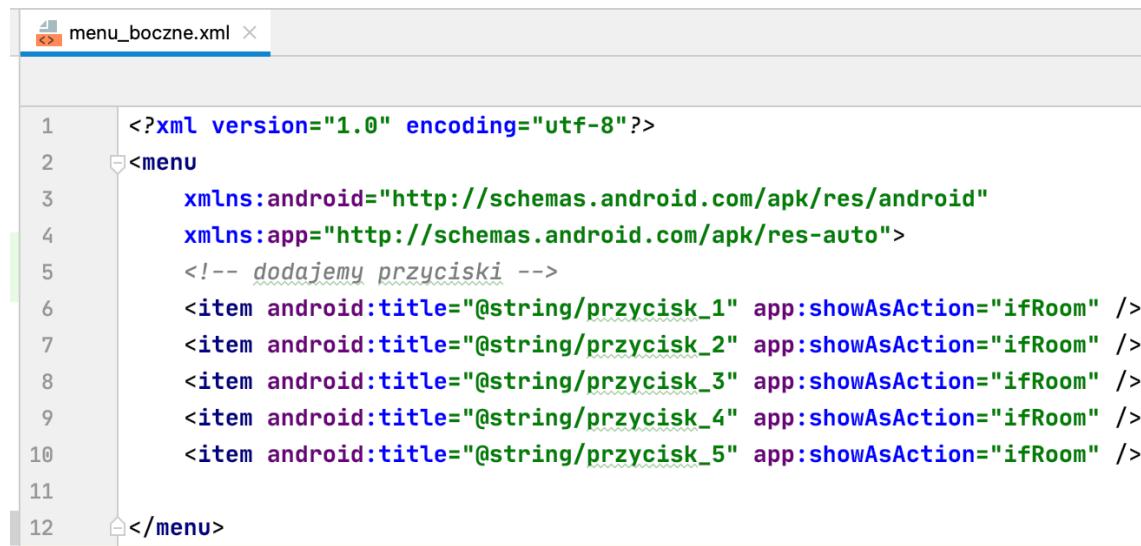
Powinniśmy uzyskać możliwość wysuwania z lewej strony layoutu.



Utworzymy menu w wysuwany panelu bocznym:



Modyfikujemy Layout przycisków według wzoru:



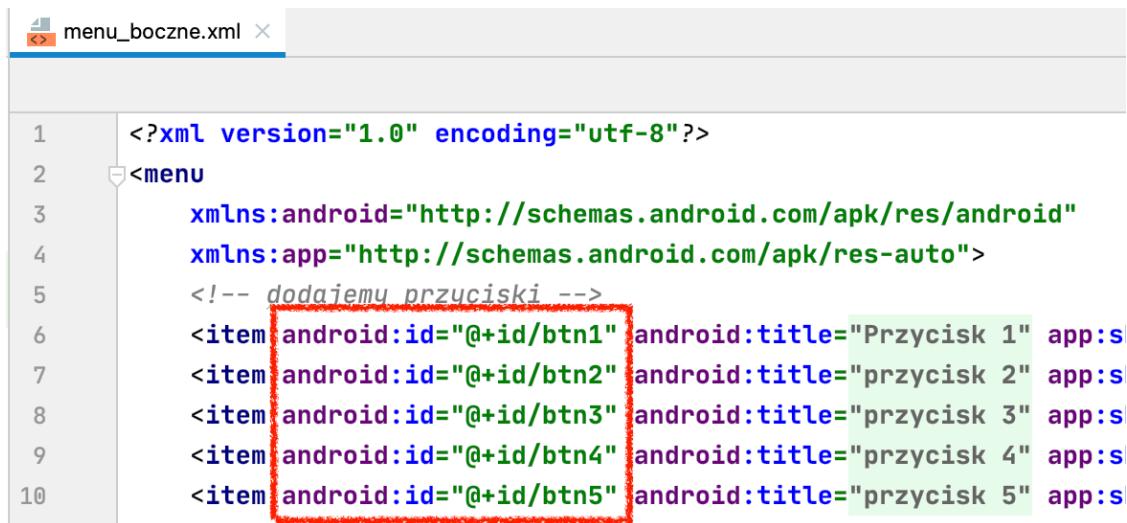
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <menu
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto">
5     <!-- dodajemy przyciski --&gt;
6     &lt;item android:title="@string/przycisk_1" app:showAsAction="ifRoom" /&gt;
7     &lt;item android:title="@string/przycisk_2" app:showAsAction="ifRoom" /&gt;
8     &lt;item android:title="@string/przycisk_3" app:showAsAction="ifRoom" /&gt;
9     &lt;item android:title="@string/przycisk_4" app:showAsAction="ifRoom" /&gt;
10    &lt;item android:title="@string/przycisk_5" app:showAsAction="ifRoom" /&gt;
11
12 &lt;/menu&gt;</pre>
```

oraz panel boczny w main\_activity.xml :



```
2 <androidx.drawerlayout.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     xmlns:app="http://schemas.android.com/apk/res-auto"
8     tools:context=".MainActivity">
9
10    <FrameLayout
11        android:layout_width="wrap_content"
12        android:layout_height="match_parent">
13        <TextView
14            android:id="@+id/tv1"
15            android:layout_width="match_parent"
16            android:layout_height="wrap_content"
17            android:text="Zawartość okna głównego"/>
18    </FrameLayout>
19
20    <com.google.android.material.navigation.NavigationView
21        android:layout_width="wrap_content"
22        android:layout_height="match_parent"
23        android:layout_gravity = "start"
24        app:menu="@menu/menu_boczne">
25
26    </com.google.android.material.navigation.NavigationView>
27
28 </androidx.drawerlayout.widget.DrawerLayout>
```

Dodamy identyfikatory przycisków:



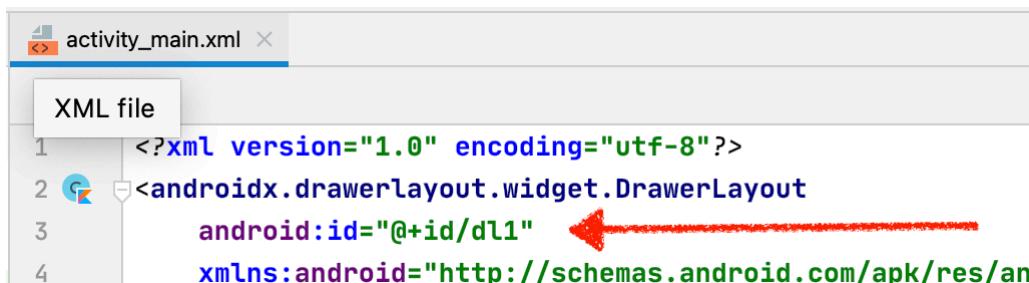
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <menu
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto">
5     <!-- dodajemy przyciski -->
6     <item android:id="@+id(btn5" android:title="Przycisk 1" app:s
7     <item android:id="@+id(btn2" android:title="przycisk 2" app:s
8     <item android:id="@+id(btn3" android:title="przycisk 3" app:s
9     <item android:id="@+id(btn4" android:title="przycisk 4" app:s
10    <item android:id="@+id(btn1" android:title="przycisk 5" app:s
```

oraz dla nawigacji



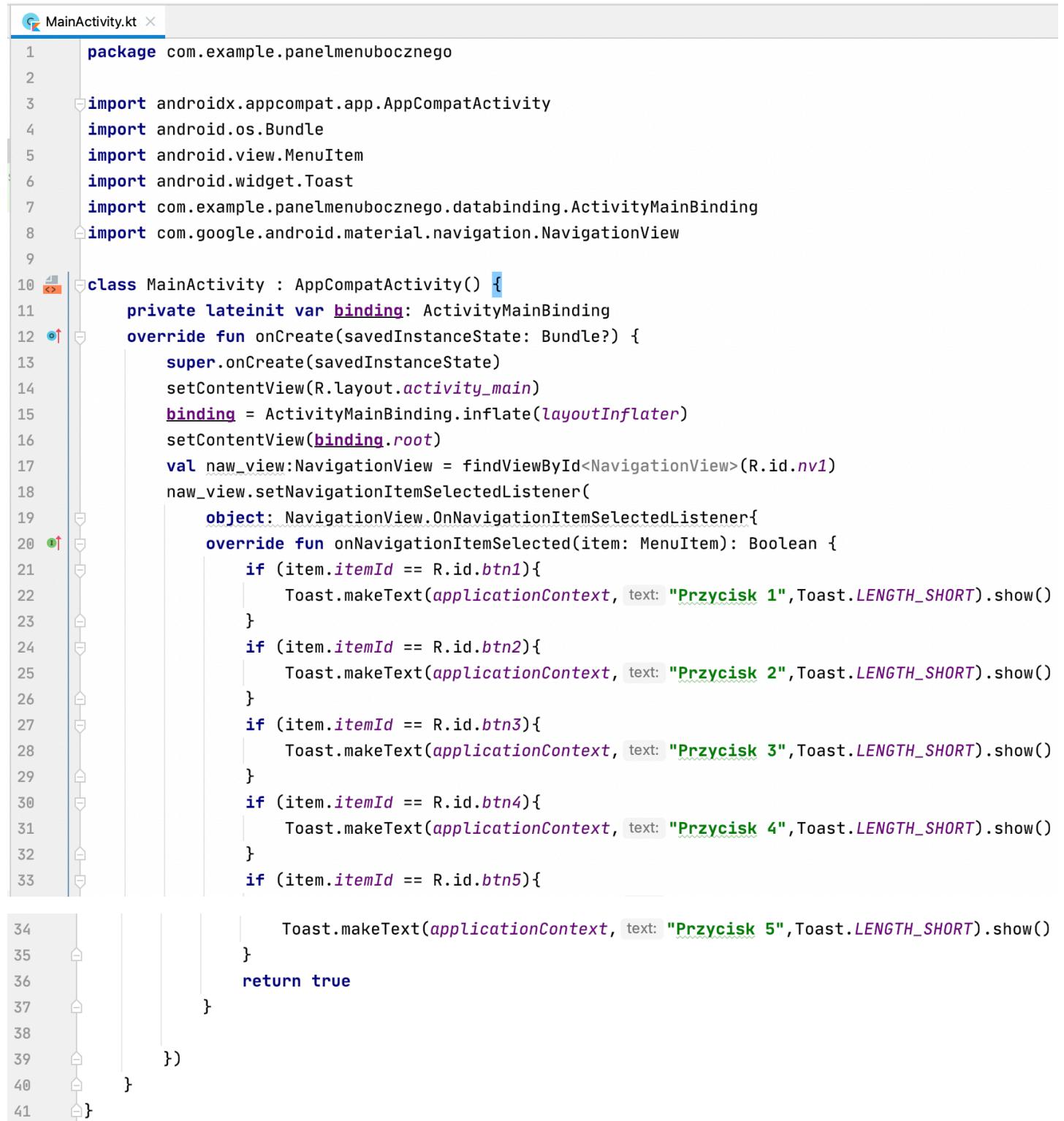
```
20 <com.google.android.material.navigation.NavigationView
21     android:id="@+id/nv1" ←
22     android:layout_width="wrap_content"
23     android:layout_height="match_parent"
24     android:layout_gravity = "start"
25     app:menu="@menu/menu_boczne">
```

i dla layoutu



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.drawerlayout.widget.DrawerLayout
3     android:id="@+id/dl1" ←
4     xmlns:android="http://schemas.android.com/apk/res/an
```

W MainActivity.kt zmieniamy zachowanie przycisków. Oprogramowujemy zdarzenia wywołane przez kliknięcie w danego Itema:



```
1 package com.example.panelmenubocznego
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.MenuItem
6 import android.widget.Toast
7 import com.example.panelmenubocznego.databinding.ActivityMainBinding
8 import com.google.android.material.navigation.NavigationView
9
10 class MainActivity : AppCompatActivity() {
11     private lateinit var binding: ActivityMainBinding
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15         binding = ActivityMainBinding.inflate(layoutInflater)
16         setContentView(binding.root)
17         val nav_view: NavigationView = findViewById<NavigationView>(R.id.nv1)
18         nav_view.setNavigationItemSelectedListener(
19             object: NavigationView.OnNavigationItemSelectedListener{
20                 override fun onNavigationItemSelected(item: MenuItem): Boolean {
21                     if (item.itemId == R.id.btn1){
22                         Toast.makeText(applicationContext, text: "Przycisk 1",Toast.LENGTH_SHORT).show()
23                     }
24                     if (item.itemId == R.id.btn2){
25                         Toast.makeText(applicationContext, text: "Przycisk 2",Toast.LENGTH_SHORT).show()
26                     }
27                     if (item.itemId == R.id.btn3){
28                         Toast.makeText(applicationContext, text: "Przycisk 3",Toast.LENGTH_SHORT).show()
29                     }
30                     if (item.itemId == R.id.btn4){
31                         Toast.makeText(applicationContext, text: "Przycisk 4",Toast.LENGTH_SHORT).show()
32                     }
33                     if (item.itemId == R.id.btn5){
34                         Toast.makeText(applicationContext, text: "Przycisk 5",Toast.LENGTH_SHORT).show()
35                     }
36                 }
37             }
38         }
39     }
40 }
41 }
```

Zmienimy działanie przycisku nr.5 tak aby losował on liczbę i wypisywał w głównej aktywności. Zmodyfikujemy zawartość if-a:

```
34
35
36
37
38
39
40
41
    if (item.itemId == R.id.btn5){
        val generator = ThreadLocalRandom.current().nextInt( bound: 100)
        binding.tv1.text = generator.toString()
        Toast.makeText(
            applicationContext, text: "Liczba została wylosowana",Toast.LENGTH_SHORT
        ).show()
    }
    return true
```