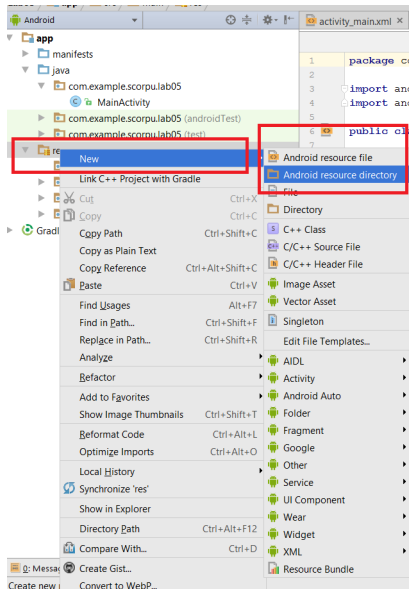


Programowanie aplikacji mobilnych

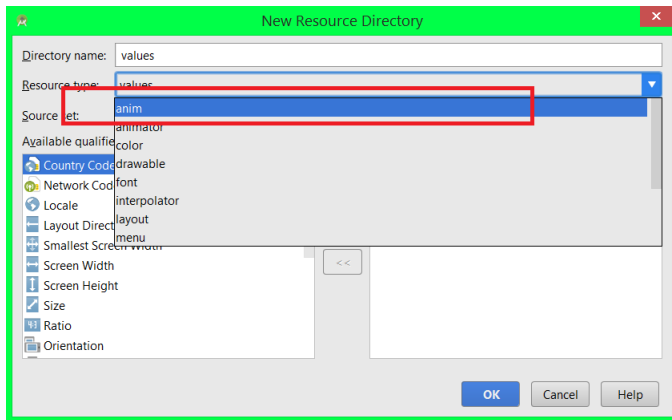
dr Przemysław Juszczuk

Katedra Inżynierii Wiedzy

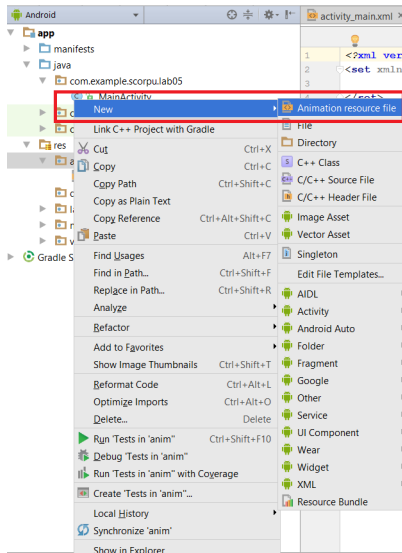
laborki 6



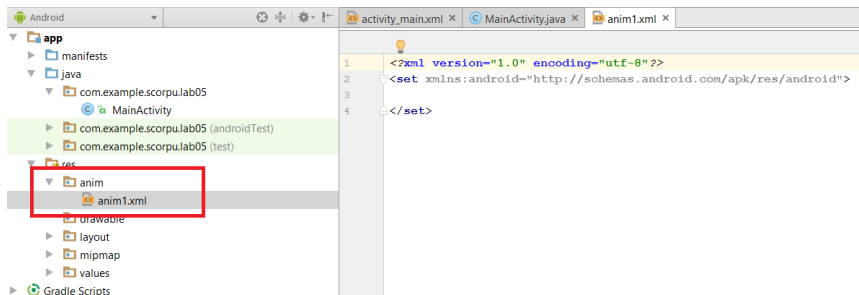
Rysunek: Korzystamy z Android resource directory – do przechowywania animacji



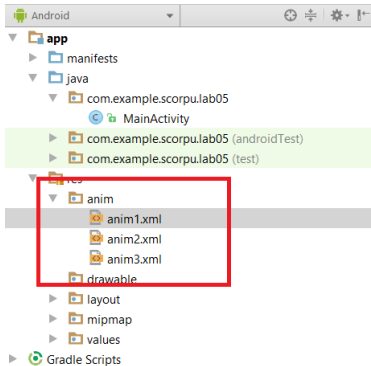
Rysunek: Tym razem wybieramy typ zasobu jako "anim"



Rysunek: Do utworzonego katalogu anim dodajemy plim XML Animation resource file



Rysunek: Do utworzonego katalogu anim dodajemy plim XML Animation resource file



Rysunek: Dodajemy dwa kolejne pliki animacji

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <set xmlns:android="http://schemas.android.com/apk/res/android">
3   android:fillAfter="true" >
4     <alpha
5       android:fromAlpha="0.0"
6       android:interpolator="@android:anim/accelerate_interpolator"
7       android:toAlpha="1.0" />
8 </set>
```

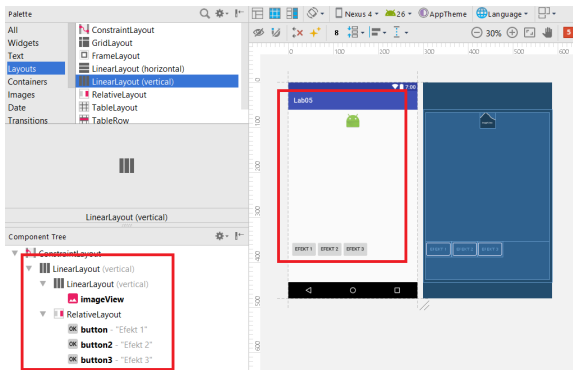
Rysunek: Animacja 1 – pojawienie się ikonki na ekranie

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <set xmlns:android="http://schemas.android.com/apk/res/android">
3   android:fillAfter="true" >
4     <alpha
5       android:fromAlpha="1.0"
6       android:interpolator="@android:anim/accelerate_interpolator"
7       android:toAlpha="0.0" />
8 </set>
```

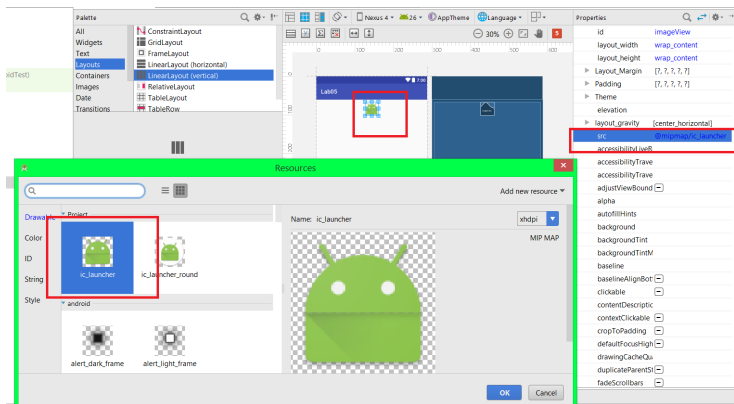
Rysunek: Animacja 2 – efekt znikania ikonki


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <set xmlns:android="http://schemas.android.com/apk/res/android">
3   <translate
4     android:fillAfter="false"
5     android:fromXDelta="500%"
6     android:toXDelta="0%"/>
7 </set>
```

Rysunek: Animacja 3 – poruszanie się po ekranie



Rysunek: Ustawiamy obrazek do animowania – właściwość src kontrolki ImageView



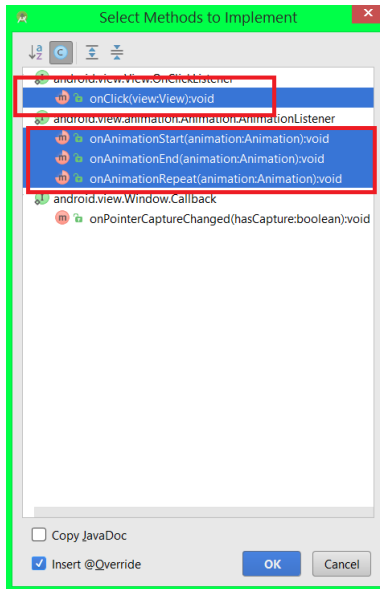
Rysunek: Zaczynamy od importu klas oraz stworzenia obiektów Button oraz EditText

```
1 package com.example.scorpu.lab05;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.animation.Animation;
6 import android.widget.Button;
7 import android.widget.ImageView;
8
9 public class MainActivity extends AppCompatActivity {
10
11     Animation anim1;
12     Animation anim2;
13     Animation anim3;
14
15
16     ImageView imageView;
17     Button animacja1;
18     Button animacja2;
19     Button animacja3;
20
21
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_main);
26     }
27 }
28
```

Rysunek: Wymagane importy oraz dodanie obiektów Button i Animation

```
1 package com.example.scorpu.lab05;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.view.animation.Animation;
7 import android.widget.Button;
8 import android.widget.ImageView;
9
10 public class MainActivity extends AppCompatActivity implements View.OnClickListener,
11     Animation.AnimationListener {
12
13     // Implement methods
14     // Make 'MainActivity' abstract
15     // Add import for 'android.view.animation.Animation.AnimationListener'
16     // Create Test
17     // Create subclass
18     // Insert App Indexing API Code
19     // Unimplement Interface
20     // Make package-private
21     // Button animacja4;
22     Button animacja4;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_main);
28     }
29 }
30
```

Rysunek: Implementujemy dwa interfejsy: AnimationListener i onClickListener.



Rysunek: Opcja automatycznego generowania potrzebnych metod

```
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_main);
28     }
29
30     @Override
31     public void onClick(View view) {
32
33     }
34
35     @Override
36     public void onAnimationStart(Animation animation) {
37
38     }
39
40     @Override
41     public void onAnimationEnd(Animation animation) {
42
43     }
44
45     @Override
46     public void onAnimationRepeat(Animation animation) {
47
48     }
```

Rysunek: Wymagane metody – na razie bez kodu

```

24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_main);
28     }
29
30     private void loadUI()
31     {
32         imageView = (ImageView) findViewById(R.id.imageView);
33         animacja1 = (Button) findViewById(R.id.button);
34         animacja2 = (Button) findViewById(R.id.button2);
35         animacja3 = (Button) findViewById(R.id.button3);
36
37         animacja1.setOnClickListener(this);
38         animacja2.setOnClickListener(this);
39         animacja3.setOnClickListener(this);
40     }
41
42     @Override
43     public void onClick(View view) {
44
45     }

```

Rysunek: Wykorzystamy dodatkową metodę – którą wywołamy w metodzie onCreate – tylko łącznie obiektów z kontrolkami z pliku XML


```
8 import android.widget.Button;
9 import android.widget.ImageView;
10
11 import android.view.animation.AnimationUtils;
12
13 public class MainActivity extends AppCompatActivity implements View.OnClickListener,
14     Animation.AnimationListener {
15
16     Animation anim1;
17     Animation anim2;
18     Animation anim3;
19     ImageView imageView;
20     Button animacja1;
21     Button animacja2;
22     Button animacja3;
23
24
25
26 @Override
27 protected void onCreate(Bundle savedInstanceState) {...}
28
29
30
31 private void loadUI ()
32 {...}
33
34
35
36
37
38
39
40
41
42
43 private void loadAnimations()
44 {
45     anim1 = AnimationUtils.loadAnimation(this, R.anim.anim1);
46     anim2 = AnimationUtils.loadAnimation(this, R.anim.anim2);
47     anim3 = AnimationUtils.loadAnimation(this, R.anim.anim3);
48 }
49
```

Rysunek: Dodatkowy import a także połączenie plików animacji (pliki XML) z obiektami – to też w osobnej metodzie

```
11 import android.view.animation.AnimationUtils;
12
13 public class MainActivity extends AppCompatActivity implements View.OnClickListener,
14     Animation.AnimationListener {
15
16     Animation anim1;
17     Animation anim2;
18     Animation anim3;
19     ImageView imageView;
20     Button animacja1;
21     Button animacja2;
22     Button animacja3;
23
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29
30         loadAnimations();
31         loadUI();
32     }
33 }
```

Rysunek: Dwie wcześniejsze metody loadAnimations i loadUI wywołujemy wewnątrz metody onCreate

```
34 private void loadUI()
35 { ... }
45
46 private void loadAnimations()
47 { ... }
53
54 @Override
55 public void onClick(View view) {
56     switch (view.getId()) {
57         case R.id.button:
58             anim1.setDuration(3000);
59             anim1.setAnimationListener(this);
60             imageView.startAnimation(anim1);
61             break;
62         case R.id.button2:
63             anim2.setDuration(3000);
64             anim2.setAnimationListener(this);
65             imageView.startAnimation(anim2);
66             break;
67         case R.id.button3:
68             anim3.setDuration(3000);
69             anim3.setAnimationListener(this);
70             imageView.startAnimation(anim3);
71             break;
72     }
73 }
```

Rysunek: Metody do animacji zostają puste. Potrzebujemy tylko obsługę przycisków – w metodzie onClick