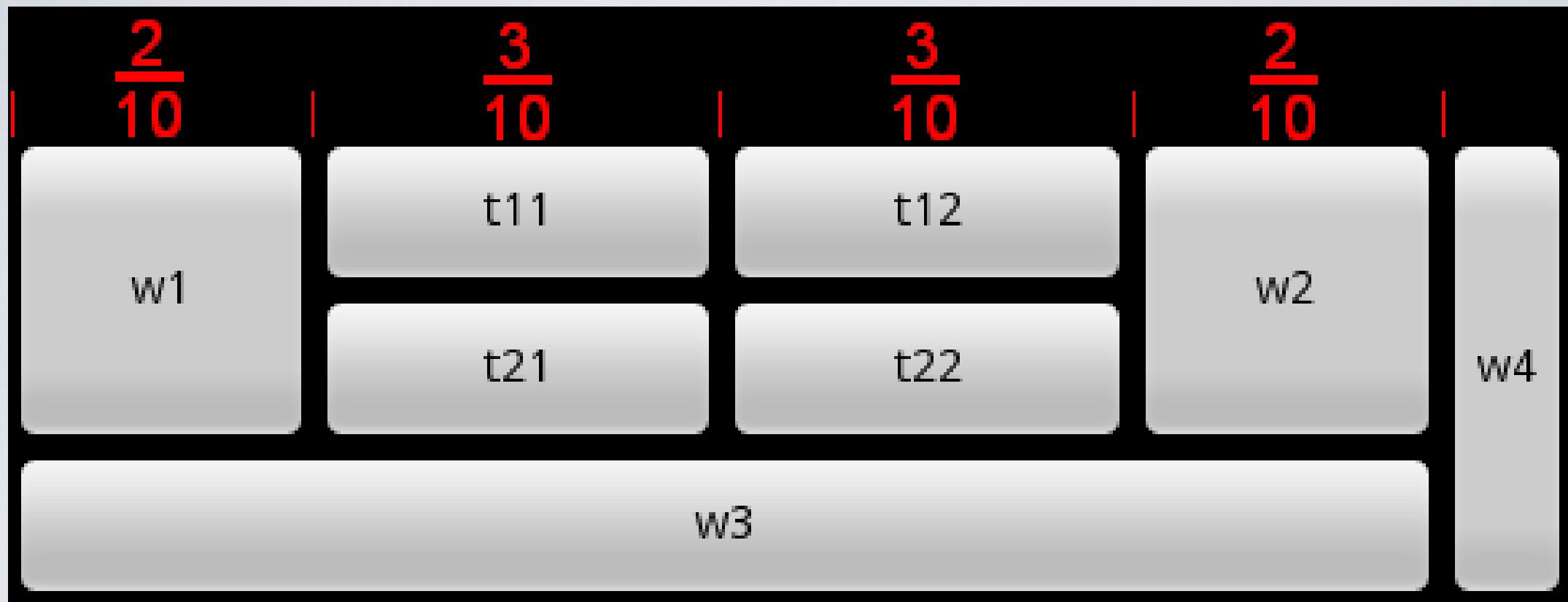


# Programowanie urządzeń mobilnych

dr inż. Andrzej Grosser  
na podstawie wykładu  
dr inż. Juliusza Mikody

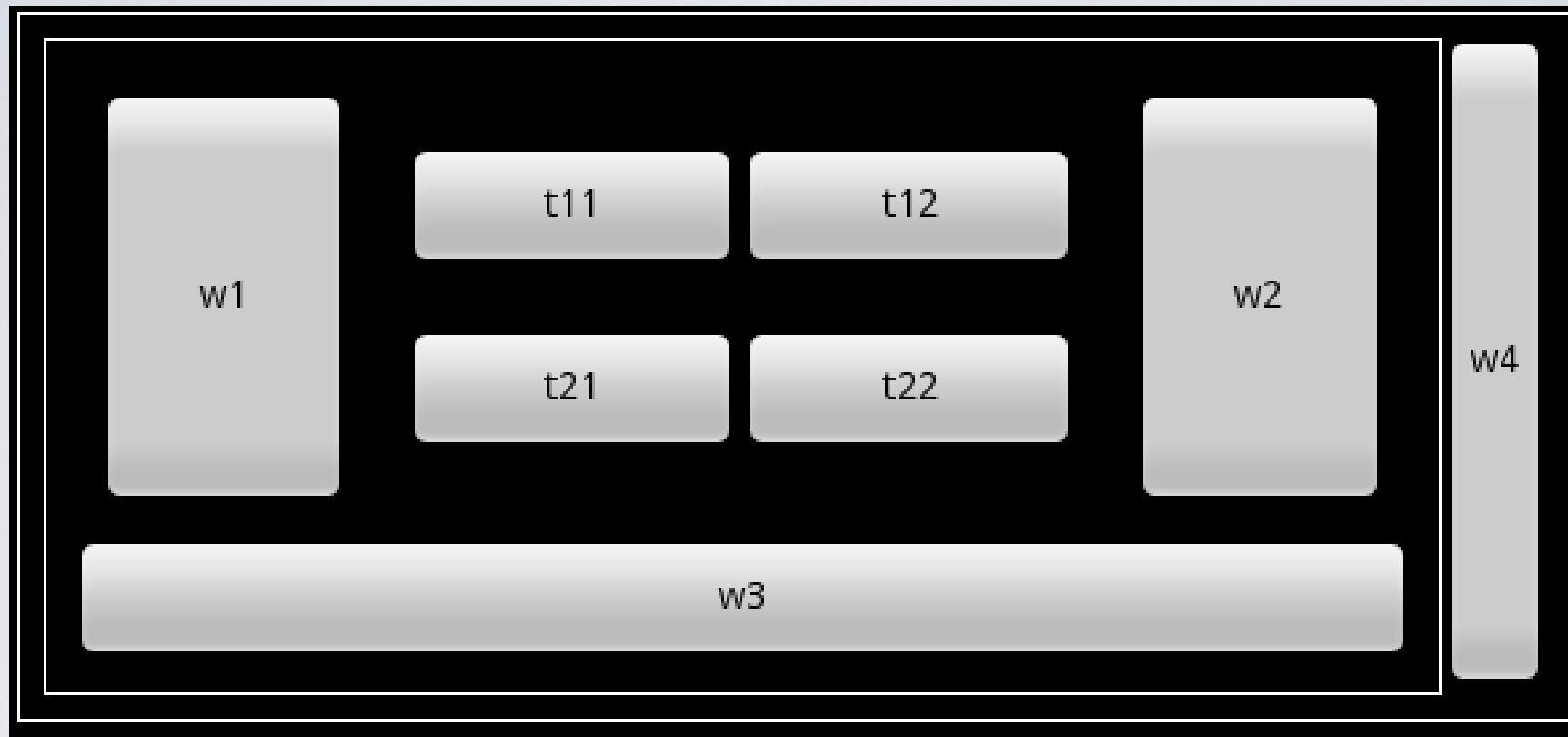
# Rozmieszczenie - przykład



- Jaki układu rozmieszczający powinien zostać wykorzystany?
  - AbsoluteLayout, TableLayout, RelativeLayout
- Jakie są ich wady i zalety

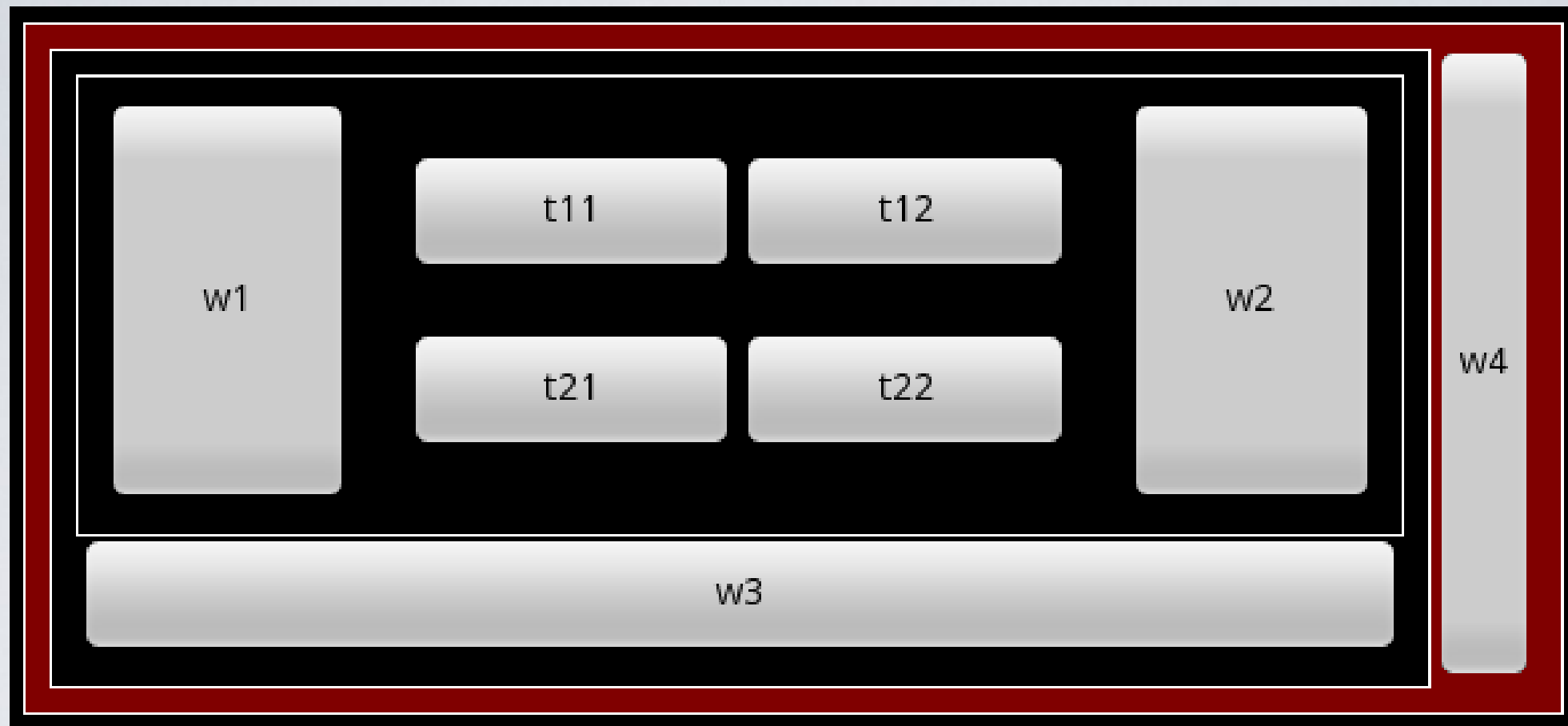


# Rozmieszczenie - przykład



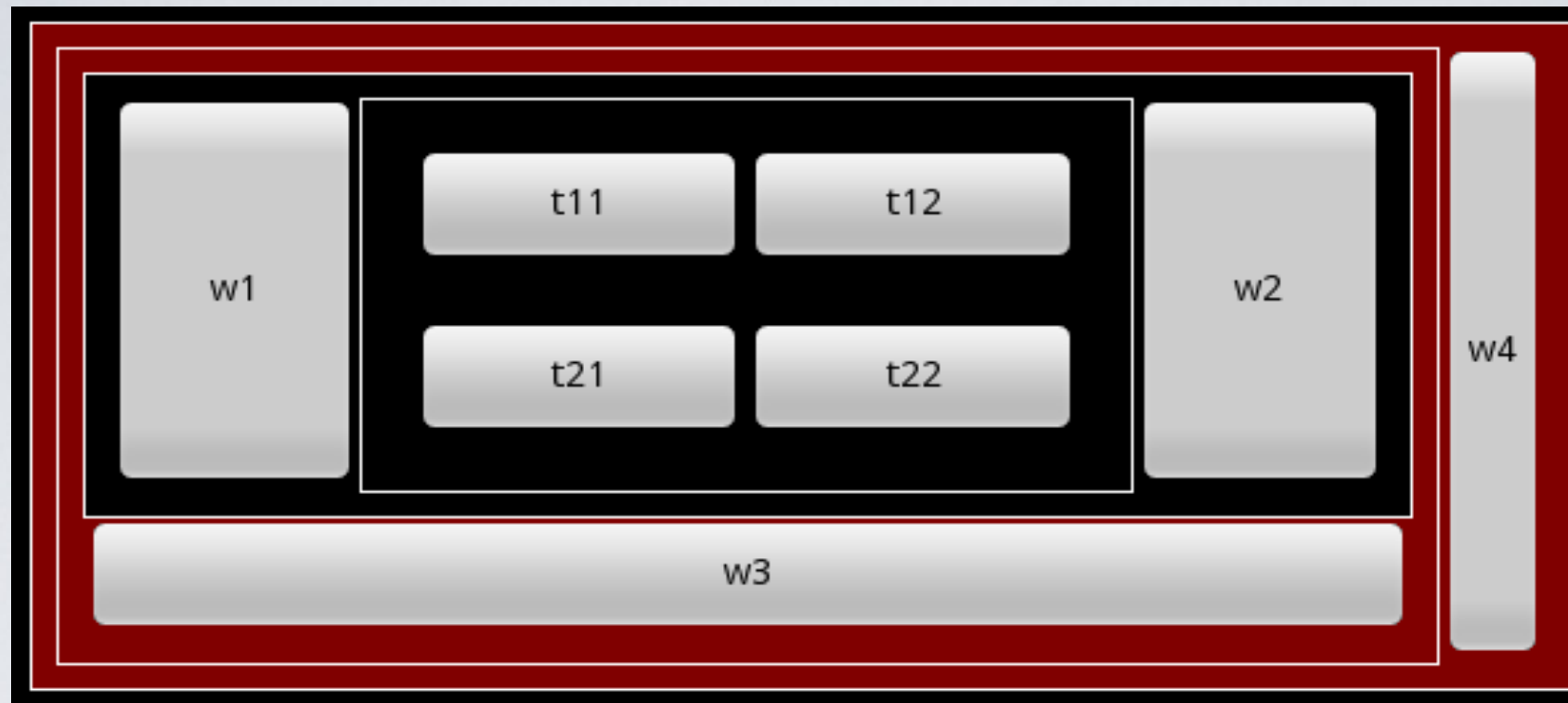
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <LinearLayout >
        <!-- ..... -->
    </LinearLayout>
    <Button android:text="w4"
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"/>
</LinearLayout>
```

# Rozmieszczenie - przykład



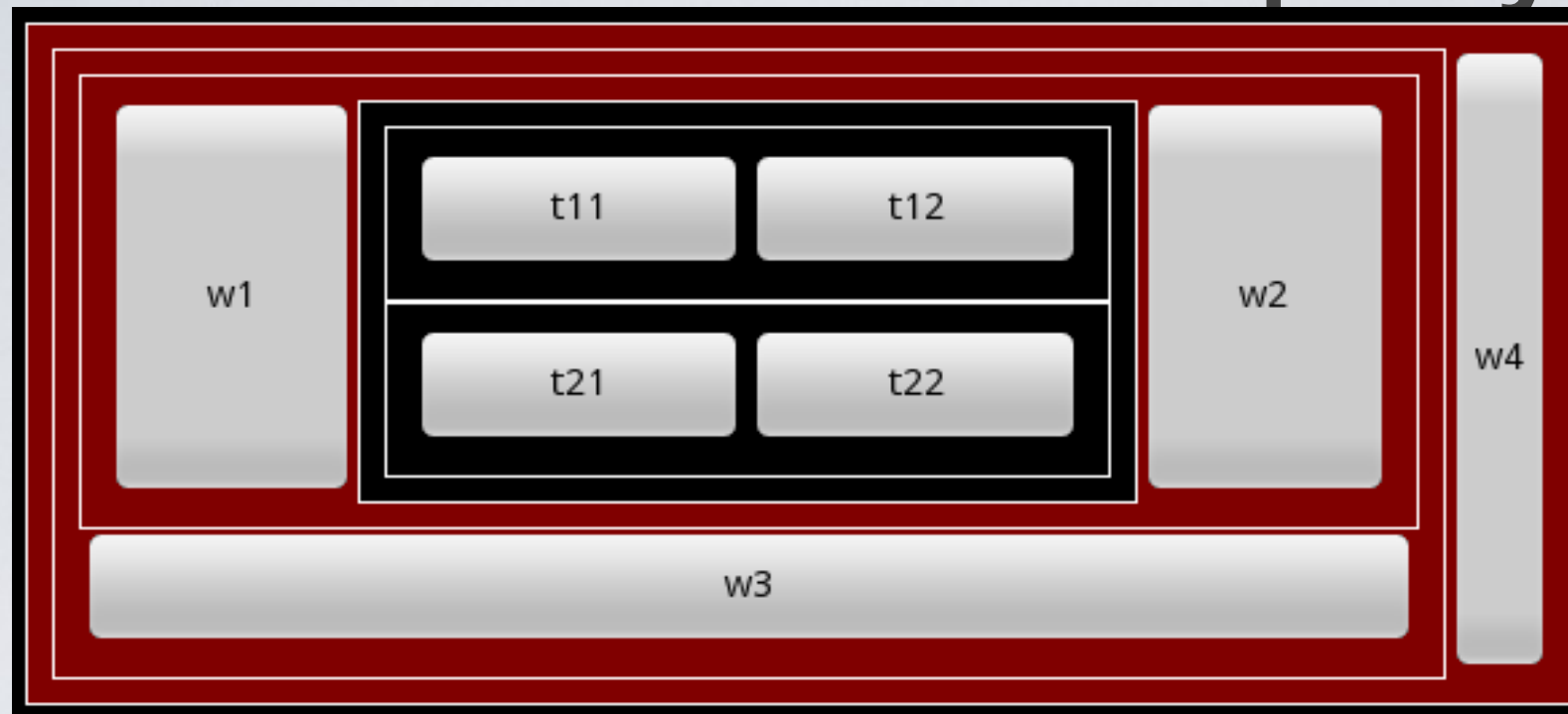
```
<LinearLayout android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_weight="1"
    android:orientation="vertical">
    <LinearLayout>
        <!-- ..... -->
    </LinearLayout>
    <Button android:text="w3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

# Rozmieszczenie - przykład



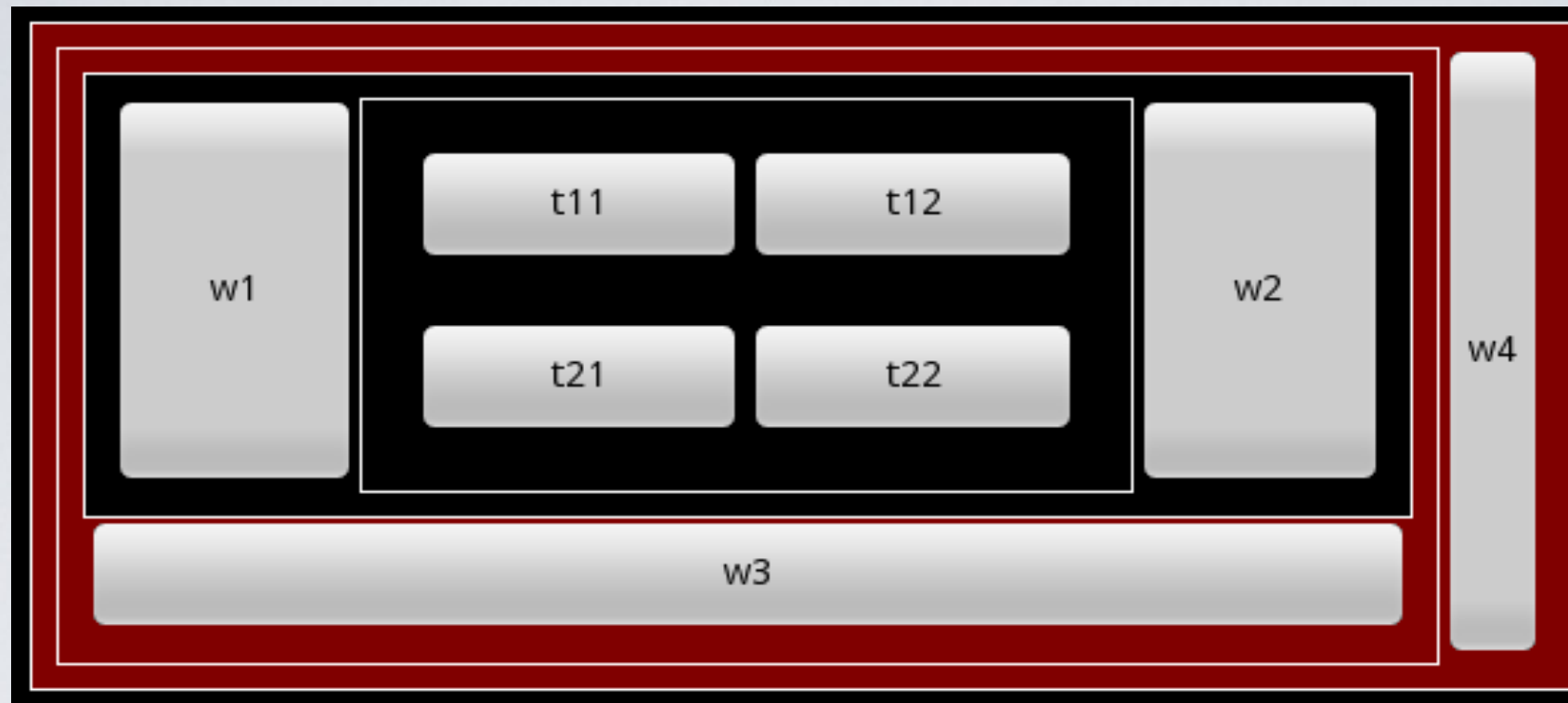
```
<LinearLayout android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:orientation="horizontal" >
    <Button android:text="w1" android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"/>
    <TableLayout>
        <!-- ..... -->
    </TableLayout>
    <Button android:text="w2" android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"/>
</LinearLayout>
```

# Rozmieszczenie - przykład



```
<TableLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="3" >
    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TableRow>
    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TableRow>
</TableLayout>
```

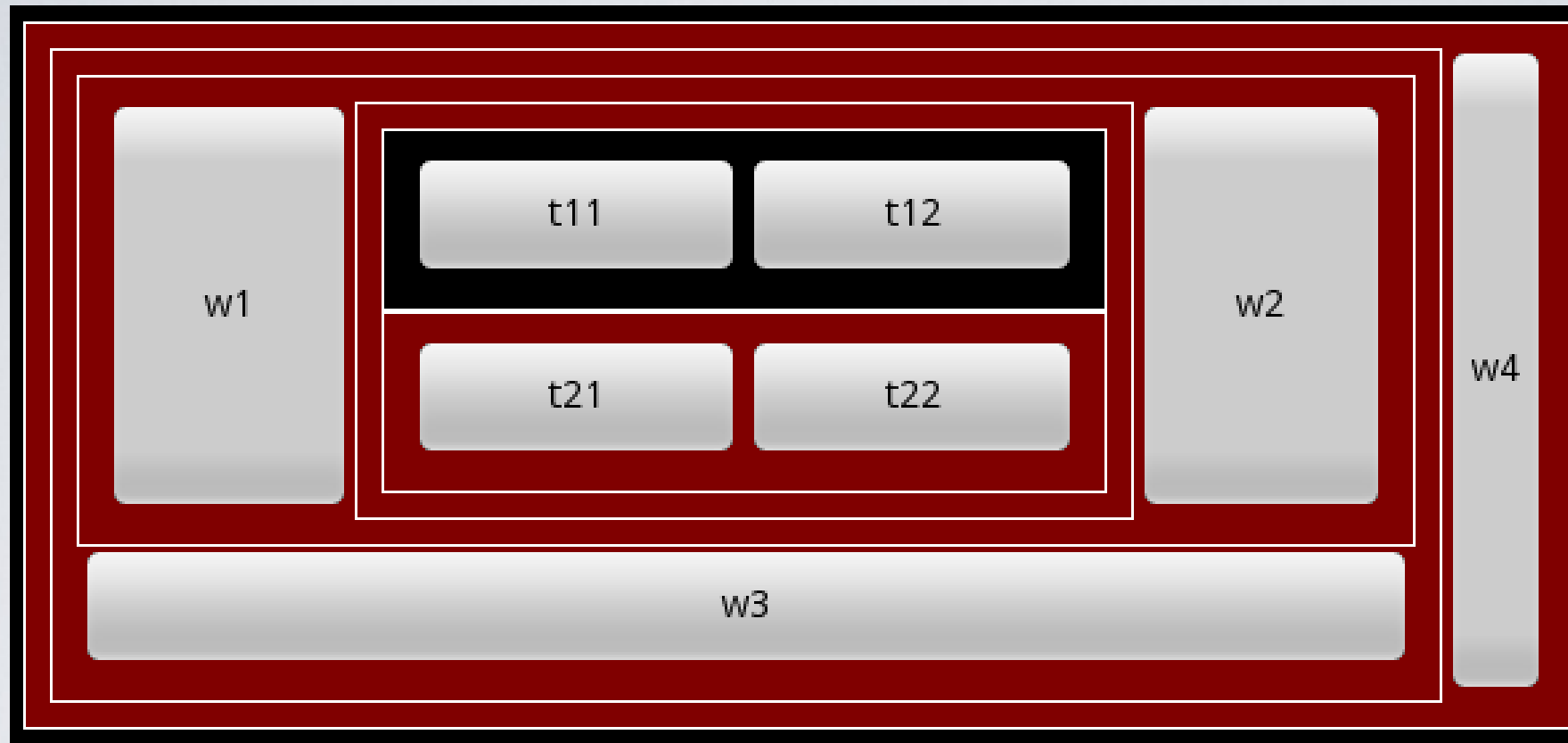
# Rozmieszczenie - przykład



```
<LinearLayout android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:orientation="horizontal" >
    <Button android:text="w1" android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"/>
    <TableLayout>
        <!-- ..... -->
    </TableLayout>
    <Button android:text="w2" android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"/>
</LinearLayout>
```



# Rozmieszczenie - przykład



```
<TableRow android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button android:text="t11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" />
    <Button android:text="t12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" />
    </TableRow>
```



# Rozmiary – dimension

- px – piksele ekranowe,
- in – cale,
- mm – milimetry,
- pt - punkty  $\frac{1}{72}$  cala,
- dp (dip) – punkty na cal (160dp na cal) – wielkość uzależniona od upakowania pikseli na ekranie,
- sp – piksele niezależne od skali – jednostka do ustawiania rozmiaru czcionek ekranowych, uzależniona od upakowania pikseli oraz ustawień użytkownika

# Rozmiary – dimension

- Zawarty w bloku zasobów /res/values, może być umieszczony w dowolnym pliku xml w tym katalogu
- Służy do zapisu informacji zbiorczej o rozmiarach elementów:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
    <dimen name="sz_przycisk">200dp</dimen>
```

```
    <dimen name="wy_przycisk">100dp</dimen>
```

```
    <dimen name="font_przycisk">25sp</dimen>
```

```
</resources>
```

```
<Button android:text="Wybierz" android:id="@+id/select"
```

```
    android:layout_width="@dimen/sz_przycisk"
```

```
    android:layout_height="@dimen/wy_przycisk"
```

```
    android:textSize="@dimen/font_przycisk" />
```

# Zasoby – kolory

- Predefiniowany kolor:  
`android:drawable="@color/green"`
- Określenie barwy :
  - `#RGB`    `#ARGB`    : 0 - F
  - `#RRGGBB`   `#AARRGGBB` : 00 - FF

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="darkred">#200000</color>
    <color name="darkblue">#000020</color>
    <color name="red">#f00</color>
    <color name="translucent_red">#80ff0000</color>
</resources>

<Button
    android:textColor="@color/red"
    android:textColorHighlight="@color/translucent_red" />
```

# Zasoby – wypełniacze

- Elementy pozwalające rysować elementy w dawnym kolorze (color-drawable) odpowiadające narysowaniu prostokąta w jednym kolorze:

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>
```

```
    <color name="red">#f00</color>
```

```
    <drawable  
name="red_rectangle">@color/red</drawable>  
    <drawable name="blue_rectangle">#00f</drawable>  
</resources>
```

```
<Button  
    android:background="@drawable/blue_rectangle"  
>
```





# Zasoby – wypełniacze

- Niezależny obiekt w katalogu /res/drawable
- Jeden obiekt w danym pliku:  
nazwa - identyfikator
- Kształt: "rectangle", "oval", "line", "ring"
- Pozwala zaokrąglić brzeg rysowanego obszaru:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<shape
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:shape="rectangle" >
```

```
<solid android:color="#80800000" />
```

```
<corners android:radius="8dp" />
```

```
</shape>
```



Wybierz

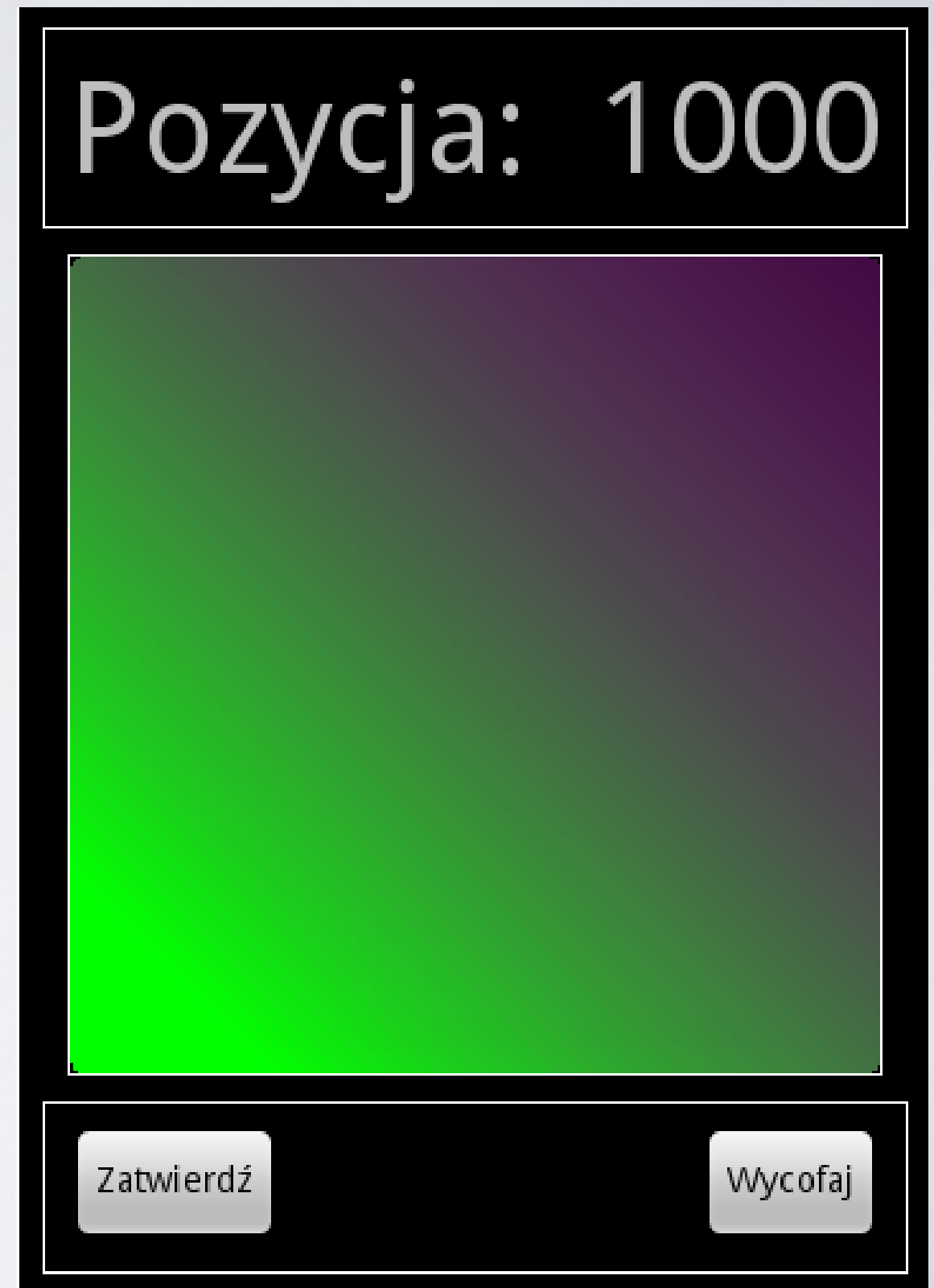
# Zasoby – wypełniacze

- Zasób shape pozwala także tworzyć nierównomierne wypełniane obszary:



```
<?xml version="1.0"
    encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <gradient
        android:startColor="#FF00FF00"
        android:endColor="#30FF00FF"
        android:angle="45"/>
    <corners android:radius="8dp" />
</shape>
```

# Rozmieszczenie - przykład



# Rozmieszczenie - przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <LinearLayout
        android:layout_height="wrap_content"
        android:layout_width="fill_parent">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Pozycja:"
            android:textSize="50sp" />
        <TextView android:id="@+id/nrpos"
            android:layout_height="wrap_content"
            android:layout_width="fill_parent"
            android:gravity="right"
            android:textSize="50sp"
            android:text="1000" />
        </LinearLayout>
    <ListView android:id="@+id/List"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/round_rect"
        android:layout_weight="1"
        android:layout_margin="10dp"
    </ListView>
```



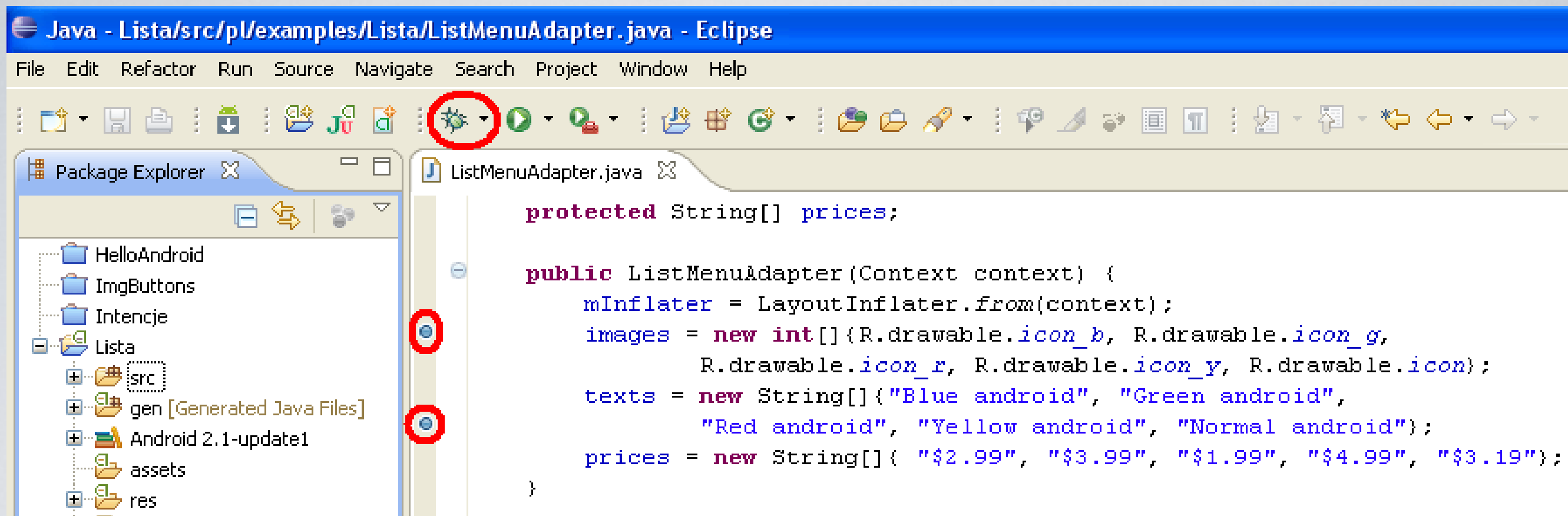


# Rozmieszczenie - przykład

```
<LinearLayout android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button android:id="@+id/ok"><
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Zatwierdź" />
    <View android:layout_width="wrap_content"
        android:layout_weight="1"
        android:layout_height="1px" />
    <Button android:id="@+id/cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Wycofaj" />
</LinearLayout>
</LinearLayout>
```

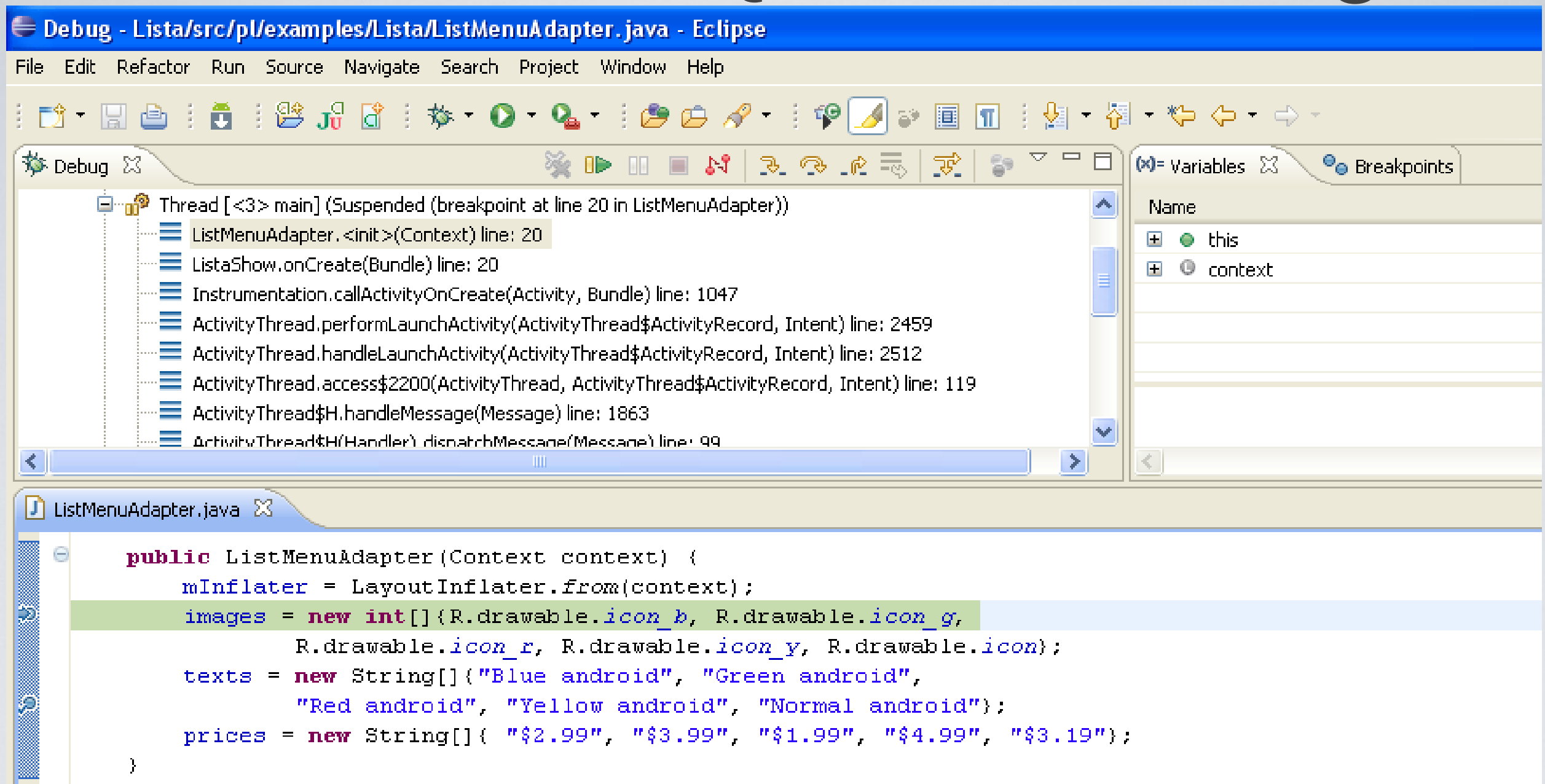


# Szukanie błędów – Debug



- Wstawianie pułapek: Ctrl+Shift+B (2 x klik na pasek)
- Run : Debug As : Android Application
- Powoduje zmianę wyglądu środowiska (Wersja Debug)

# Szukanie błędów – Debug



- F5 – Step Into – wejście do instrukcji złożonej
- F6 – Step Over – następna komenda
- F7 – Step return – wyjście z procedury
- F8 – Resume – dalsze przetwarzanie programu



# Intencje – co chcę zrobić

---

Intencją będziemy nazywać akcję  
oraz powiązane z nią dane

Intencje między innym służą do:

- Wywołania aplikacji z poziomu innej aplikacji.
- Wywołania wewnętrznych lub zewnętrznych składników aplikacji.
- Można generować zdarzenia na które inne aplikacje mogą odpowiadać.

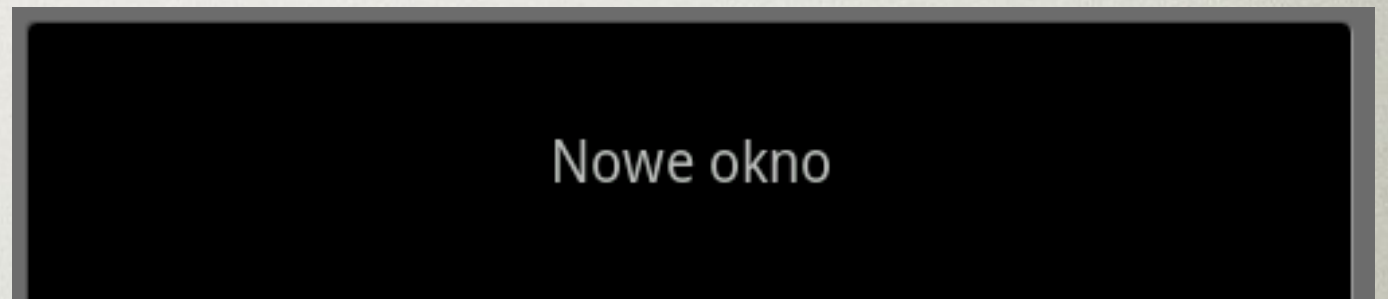


# Wyświetl moje okno

---

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

    <TextView android:layout_margin="20dp"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:gravity="center_horizontal"
        android:text="Nowe okno"/>
</LinearLayout>
```



```
public class NoweOkno extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.nowe_okno);
    }
}
```

# Wyświetl moje okno

The screenshot shows the Android Studio interface with the 'Application Nodes' tree on the left and the 'Attributes for .NoweOkno (Activity)' panel on the right. The 'Application Nodes' tree shows a package structure with an Activity named '.NoweOkno' and its associated Intent Filter and Action/Category. The 'Attributes for .NoweOkno (Activity)' panel shows the configuration for the activity, including Name, Theme, Label, Icon, Launch mode, Screen orientation, Config changes, Permission, and Multiprocess. The 'Manifest' tab is selected at the bottom.

**Application Nodes**

- .Intencje (Activity)
  - .NoweOkno (Activity)
    - Intent Filter
      - pl.example.Intencje.action.NoweOkno (Action)
      - android.intent.category.DEFAULT (Category)

**Attributes for .NoweOkno (Activity)**

The "activity" tag declares an android.app.Activity class that is available as part of the package's application components, implementing a part of the application's user interface.

Name*	.NoweOkno	Browse...
Theme	@android:style/Theme.NoTitleBar	Browse...
Label	Test nowego widoku	Browse...
Icon		Browse...
Launch mode		▼
Screen orientation		▼
Config changes		Select...
Permission		▼
Multiprocess		▼

Manifest Application Permissions Instrumentation AndroidManifest.xml

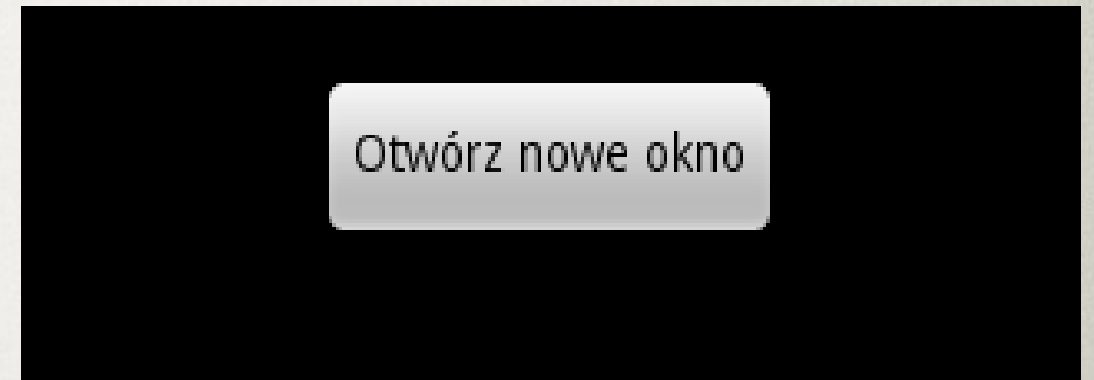
```
<manifest ...
  <application ...
    <activity android:name=".NoweOkno"
      android:label="Test nowego widoku"
      android:theme="@android:style/Theme.NoTitleBar.Fullscreen"> // pełne okno - bez nagłówka
      <intent-filter>
        <action android:name="pl.example.Intencje.action.NoweOkno"/>
        <category android:name="android.intent.category.DEFAULT"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```



# Wyświetl moje okno

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android
    ="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:text="Otwórz nowe okno"
        android:onClick="nacisnietoPrzycisk" />
</LinearLayout>
```



```
public class Intencje extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
    public void nacisnietoPrzycisk(View view) {
        String url = "pl.example.Intencje.action.NoweOkno";
        Intent zamiar = new Intent(url);
        // Intent zamiar = new Intent(this, NoweOkno.class);
        startActivity(zamiar);
    }
}
```



# Intencje dostępne w systemie

---

<http://developer.android.com/guide/appendix/g-app-intents.html>

- Przeglądarka stron internetowych

```
Intent openwww = new Intent(Intent.ACTION_VIEW);
openwww.setData(Uri.parse("http://www.onet.pl"));
startActivity(openwww);
```

```
startActivity(new Intent(Intent.ACTION_VIEW,
    Uri.parse("http://www.onet.pl")));
```

- Wyszukiwarka stron

```
startActivity(new Intent(Intent.ACTION_WEB_SEARCH));
```

- Nawiązanie połączenia telefonicznego

```
startActivity(new Intent(Intent.ACTION_DIAL,
    Uri.parse("tel:555-555-1234")));
```

```
startActivity(new Intent(Intent.ACTION_CALL,
    Uri.parse("tel:555-555-1234")));
```

```
<uses-permission
```

```
    android:name="android.permission.CALL_PHONE"/>
```



# Intencje – przesyłane typy dane

---

- Intent      putExtra(String name, long value)
  - boolean, byte, char, short, int, float, double
- Intent      putExtra(String name, long[] value)
  - boolean[], byte[], char[], short[], int[], float[], double[]
- Intent      putExtra(String name, String value)
- Intent      putExtra(String name, String[] value)
- Intent      putExtra(String name, Serializable value)
- Intent      putExtra(String name, Bundle value)
- Intent      putIntegerArrayListExtra(String name, ArrayList<Integer> value)
- Intent      putStringArrayListExtra(String name, ArrayList<String> value)
- Intent      putExtras(Bundle extras)



# Intencje – przesyłane typy dane

---

- `String[]`    `getStringArrayExtra(String name)`
- `ArrayList<String>`    `getStringArrayListExtra(String name)`
- `String`    `getStringExtra(String name)`
- `Bundle`    `getExtras()`
  - `double`    `getDouble(String key)`
  - `long`    `getLong(String key)`
  - `String`    `getString(String key)`
  - `String[]`    `getStringArray(String key)`



# Intencje a dane

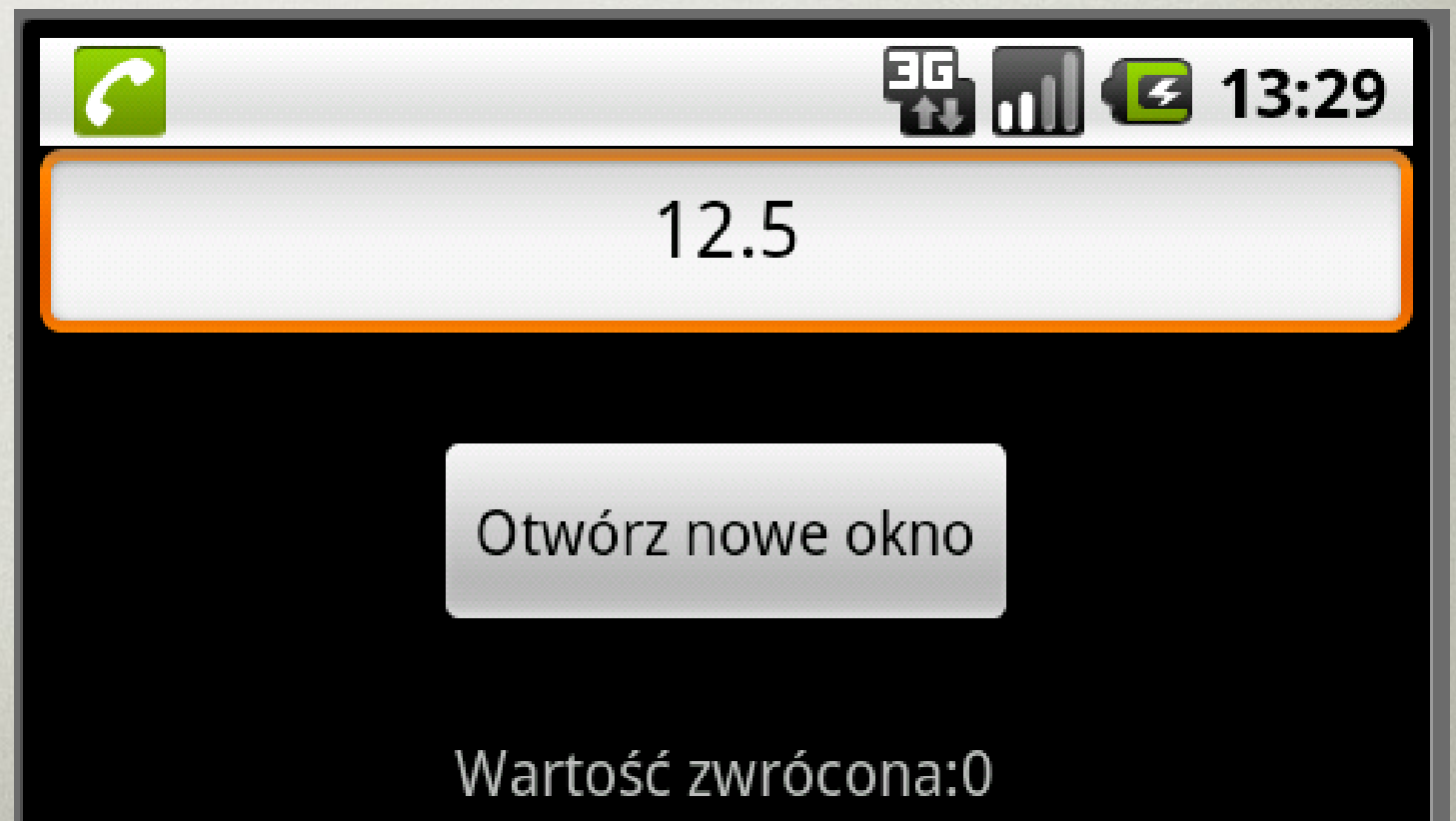
---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <EditText android:layout_height="wrap_content"
        android:id="@+id/SendValue"
        android:inputType="numberDecimal"
        android:text="0"
        android:layout_width="fill_parent"
        android:gravity="center_horizontal"/>
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:text="Otwórz nowe okno"
        android:onClick="nacisnietoPrzycisk" />
    <LinearLayout android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal">
        <TextView android:layout_height="wrap_content"
            android:layout_width="fill_parent"
            android:text="Wartość zwrócona:" />
        <TextView android:text="0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/TextResult"/>
    </LinearLayout>
</LinearLayout>
```



# Intencje a dane

```
public class Intencje extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        requestWindowFeature(Window.FEATURE_NO_TITLE);  
        setContentView(R.layout.main);  
    }  
    public void nacisnietoPrzycisk(View view) {  
        EditText et = (EditText) findViewById(R.id.SendValue);  
        double wartosc = Double.parseDouble(et.getText().toString());  
        Intent zamiar = new Intent(this, NoweOkno.class);  
        zamiar.putExtra("typ", "Promień koła");  
        zamiar.putExtra("wartosc", wartosc);  
        startActivityForResult(zamiar, 1);  
    }  
    // .....  
}
```





# Intencje a dane

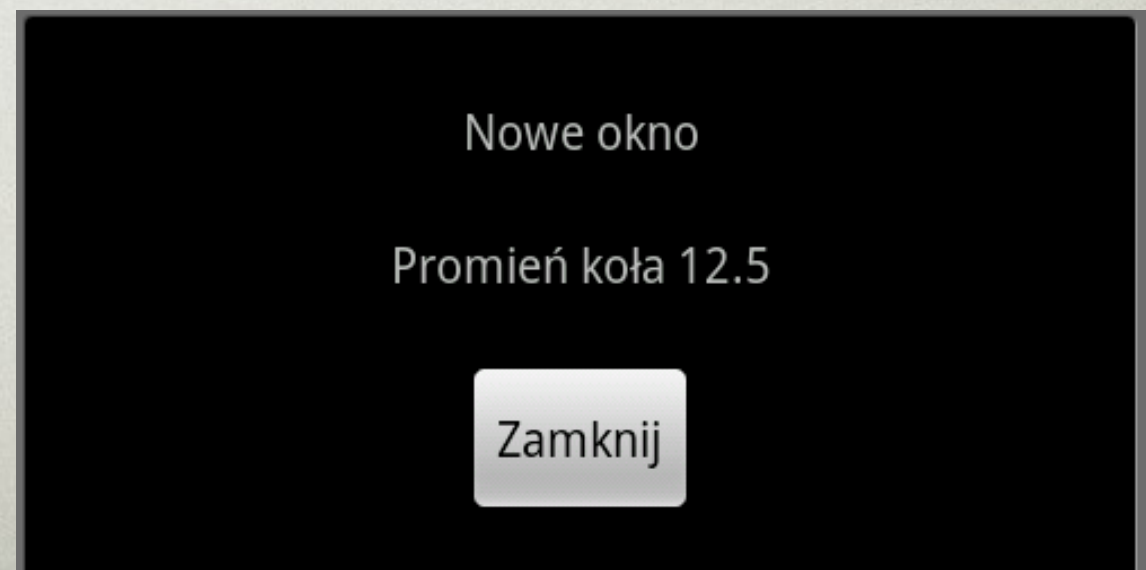
---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:orientation="vertical">
    <TextView android:layout_margin_top="20dp"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:gravity="center_horizontal"
        android:text="Nowe okno"/>
    <TextView android:id="@+id/Text"
        android:layout_margin_top="20dp"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:gravity="center_horizontal"
        android:text="Przekazane"/>
    <Button android:id="@+id/Close"
        android:layout_margin_top="20dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Zamknij"
        android:onClick="zamknijOkno"/>
</LinearLayout>
```



# Intencje a dane

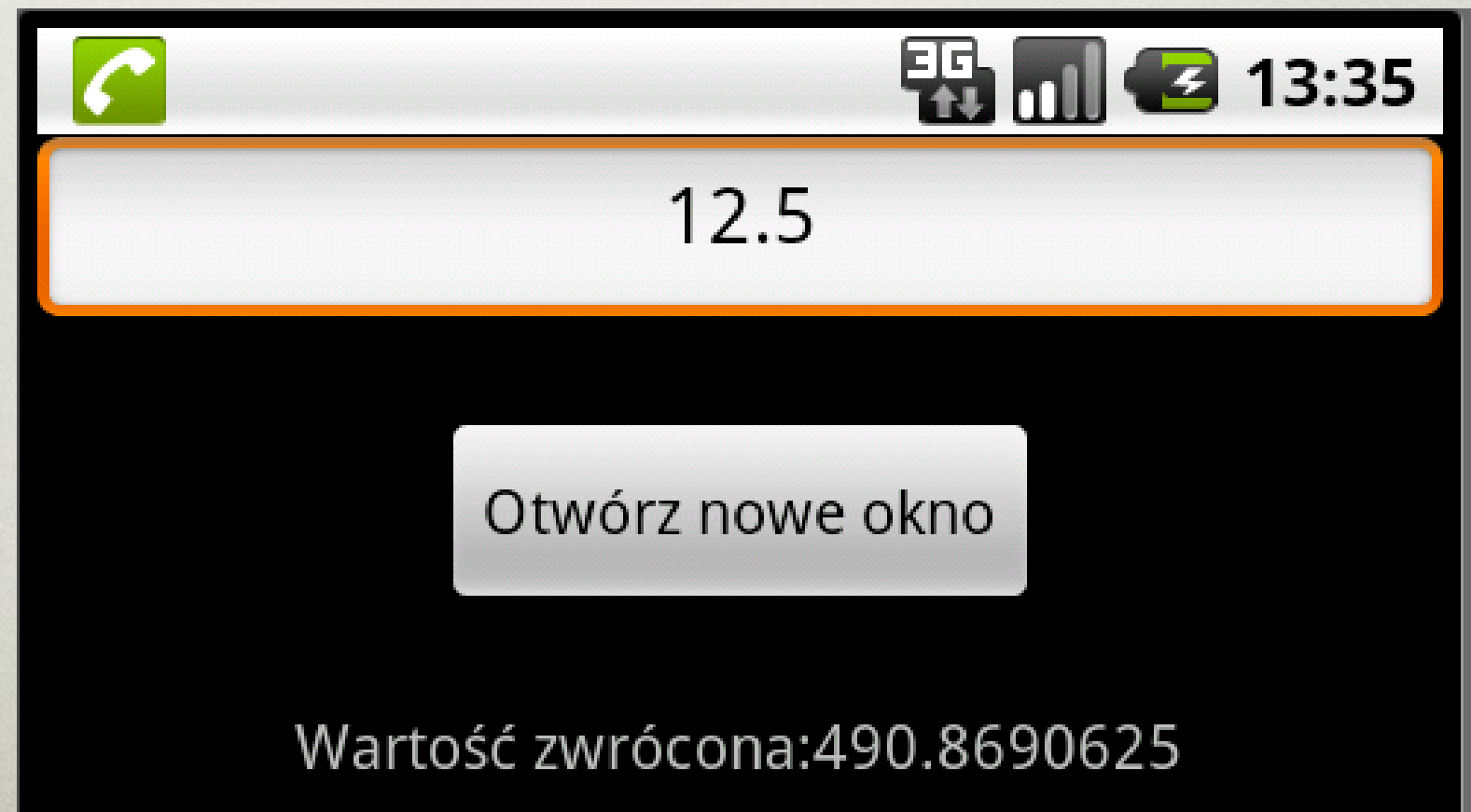
```
public class NoweOkno extends Activity {
    private double promien;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.nowe_okno);
        String typ = "";
        promien = 0;
        Bundle b = getIntent().getExtras();
        if (b.containsKey("typ")) typ = b.getString("typ");
        if (b.containsKey("wartosc")) promien = b.getDouble("wartosc");
        TextView tv = (TextView) findViewById(R.id.Text);
        tv.setText(typ + " " + promien);
    }
    public void zamknijOkno(View view) {
        Intent resultIntent = new Intent();
        resultIntent.putExtra("wynik", promien * promien * 3.141562);
        setResult(RESULT_OK, resultIntent);
        finish();
    }
}
```





# Intencje a dane

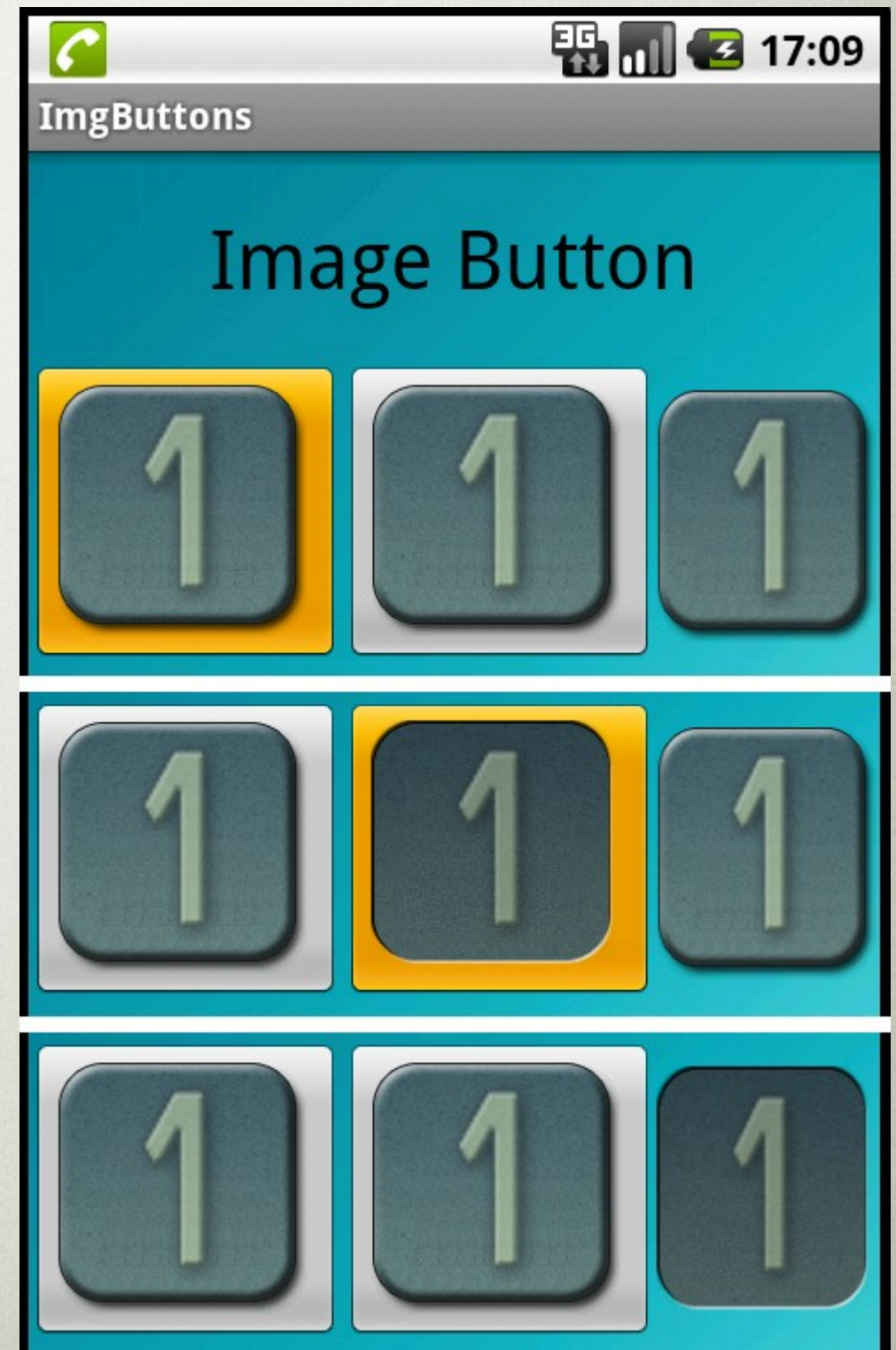
```
public class Intencje extends Activity {  
    //.....  
    protected void onActivityResult(int requestCode, int resultCode,  
        Intent data) {  
        super.onActivityResult(requestCode, resultCode, data);  
        if(resultCode == RESULT_OK && requestCode == 1){  
            if (data.getExtras().containsKey("wynik")) {  
                TextView tv = (TextView) findViewById(R.id.TextResult);  
                double msg = data.getExtras().getDouble("wynik");  
                tv.setText(Double.toString(msg));  
            }  
        }  
    }  
}
```





# Kontrolki - Przyciski obrazkowe

- Elementy interakcji z użytkownikiem pozwalające wprowadzić obraz zamiast napisu.
- Udostępnia metodę **setImageResource()** oraz atrybut **android:src**
- Możliwość rozszerzenia funkcjonalności elementu graficznego na skutek zaznaczenia / wciśnięcia przycisku.





# Kontrolki – Przyciski obrazkowe

---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:background="@drawable/back"
    android:layout_width="fill_parent" android:layout_height="fill_parent" >
<TextView
    android:layout_width="fill_parent" android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" android:gravity="center_horizontal"
    android:text="@string/hello" android:textSize="30dp"
    android:textColor="#000000" android:layout_margin="20dp" />
<LinearLayout android:orientation="horizontal"
    android:layout_height="wrap_content" android:layout_width="fill_parent"
    android:gravity="center_vertical|center_horizontal">

    <ImageButton android:src="@drawable/one_up"
        android:layout_width="wrap_content" android:layout_height=...
        android:layout_gravity="center_vertical|center_horizontal" />

    <ImageButton android:src="@drawable/btn_one"
        android:layout_width="wrap_content" android:layout_height=...
        android:layout_gravity="center_vertical|center_horizontal" />

    <ImageButton android:background="@drawable/btn_one"
        android:layout_width="wrap_content" android:layout_height=...
        android:layout_gravity="center_vertical|center_horizontal" />
</LinearLayout>
</LinearLayout>
```



# Kontrolki - selector



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

    <item android:state_pressed="true"
        android:drawable="@drawable/one_down" />

    <item android:state_focused="true"
        android:drawable="@drawable/one_down" />

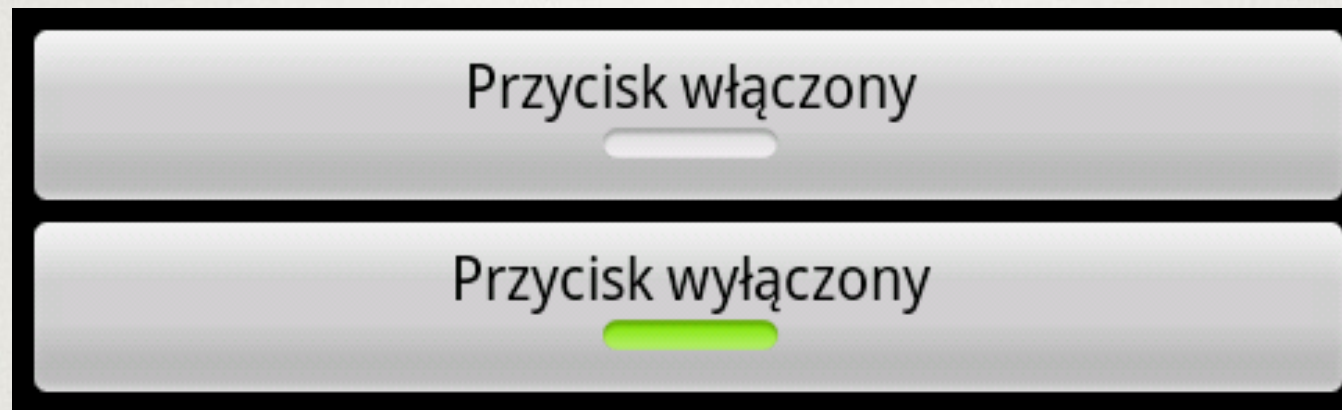
    <item android:drawable="@drawable/one_up" />
</selector>
```

- **android:state\_pressed**
- **android:state\_focused**
- **android:state\_selected**
- **android:state\_checkable**
- **android:state\_checked**
- **Android:state\_enabled**
- **android:state\_window\_focused**



# Kontrolki – przyciski przełączane

---

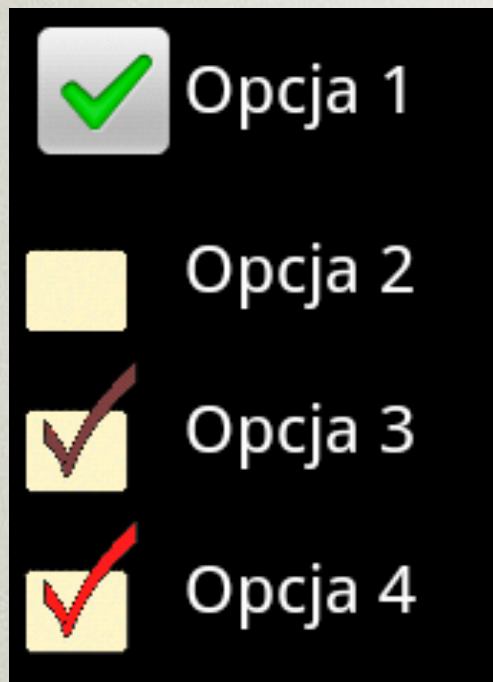


```
<ToggleButton
    android:layout_height="wrap_content"
    android:text="Przycisk przełączany"
    android:layout_width="fill_parent"
    android:textOff="Przycisk włączony"
    android:textOn="Przycisk wyłączony" />
```

```
void      setChecked(boolean checked)
boolean   isChecked()
void      setTextOff(CharSequence textOff)
void      setTextOn(CharSequence textOn)
```



# Kontrolki – Zaznaczamy opcje



<CheckBox

```
    android:layout_height="wrap_content"  
    android:layout_width="fill_parent"
```

```
    android:text="Opcja 3"  
    android:checked="true"
```

```
    android:button="@drawable/btn_check" />
```

```
// drawable/btn_check.xml  
<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  
    <item android:state_checked="true"  
        android:drawable="@drawable/checkbox1" />  
  
    <item android:state_focused="true"  
        android:drawable="@drawable/checkbox1" />  
  
    <item android:drawable="@drawable/checkbox2" />  
  
</selector>
```



# Kontrolka – wyboru jednej opcji

```
<RadioGroup android:id="@+id/RadioGr"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content">  
    <RadioButton android:id="@+id/Opcja1"  
        android:text="Opcja 1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content" />  
    <RadioButton android:id="@+id/Opcja2"  
        android:text="Opcja 2"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content" />  
    <RadioButton android:id="@+id/Opcja3"  
        android:text="Opcja 3"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content" />  
</RadioGroup>
```

**interface**

**RadioGroup.OnCheckedChangeListener**

**void onCheckedChanged(**

**RadioGroup group, int checkedId)**

