



Aplikacje mobilne

Pliki zasobów, grafiki, menu, podpinanie zdarzeń

dr Tomasz Jach
Instytut Informatyki,
Uniwersytet Śląski



Pliki zasobów

- Jak już robiliśmy wcześniej – sporo rzeczy w Androidzie umieszczamy w plikach xml (np. strings.xml).
- Jest to dobra praktyka, w plikach zasobów możemy umieszczać sporo rzeczy, do których chcemy mieć łatwy dostęp ze wszystkich miejsc w aplikacji



Strings.xml

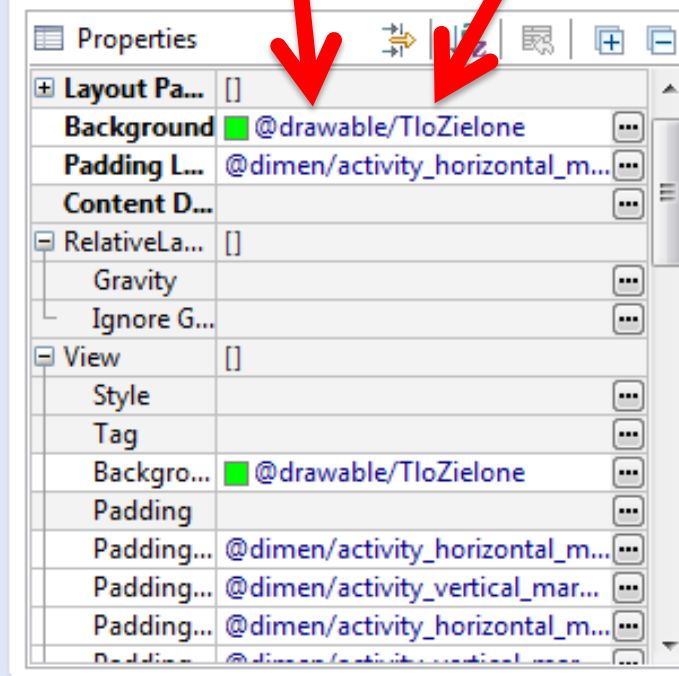
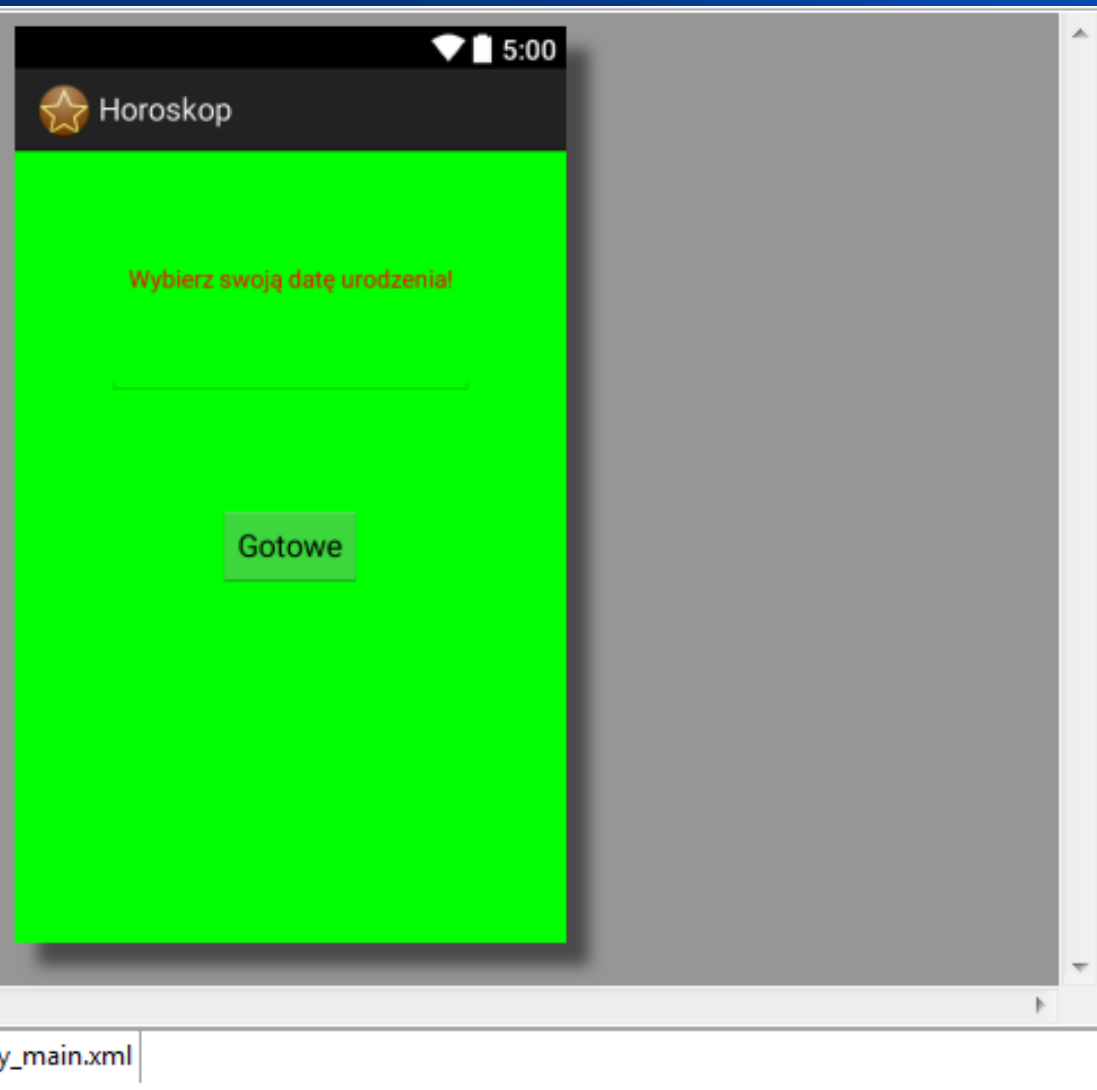
Elementy:

- Stringi
- Tablice stringów
- Tablice liczb
- Kolory
- Rozmiary
- Kolory tła

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

    <string name="app_name">Horoskop</string>
    <string name="wybor">Wybierz swoją datę urodzenia!</string>
    <string name="action_settings">Settings</string>
    <string name="dzialaj">Gotowe</string>
    <string-array name="znaki">
        <item >Baran</item>
        <item >Byk</item>
    </string-array>
    <color name="czerwony">#FF0000</color>
    <dimen name="Maly12px">12dp</dimen>
    <drawable name="TloZielone">#00FF00</drawable>

</resources>
```





Styles.xml

- W tym pliku ustawiamy definicje stylów
- Style stosowane są do wszystkich elementów (nie tylko tekstu)
- Jest to odpowiednik plików CSS z HTMLa



Styles.xml

```
<resources>
  <style name="AppBaseTheme" parent="android:Theme.Light">

  </style>

  <!-- Application theme. -->
  <style name="AppTheme" parent="AppBaseTheme">
    <!-- All customizations that are NOT specific to a particular API-level can go
  </style>
  <style name="wielki_napis">
    <item name="android:textSize">50sp</item>
    <item name="android:textColor">#0000FF</item>
    <item name="android:tex
  </style>
  <style name="maly_na
    <item name="android:
    <item name="android:
  </style>

</resources>
```

- ⓐ android:text
- ⓐ android:textAlignment
- ⓐ android:textAllCaps
- ⓐ android:textAppearance
- ⓐ android:textColor
- ⓐ android:textColorHighlight
- ⓐ android:textColorHint
- ⓐ android:textColorLink
- ⓐ android:textCursorDrawable
- ⓐ android:textDirection
- ⓐ android:textEditNoPasteWindowLayout
- ⓐ android:textEditPasteWindowLayout

Text color for links. [color, reference]



Styles.xml

Możemy też dodawać obrazki. W tym celu:

1. Kopiujemy obraz do katalogu res/drawable-XXX (XXX – rozdzielczość ekranu na który piszemy)
2. W pliku zasobów – dodaje ten plik bez rozszerzenia

Java - Horoskop/res/layout/activity_main.xml - Eclipse

File Edit Refactor Source Navigate Search Project Run Window Help

Package Explorer

- Horoskop
 - Android 4.4.2
 - Android Private Libraries
 - src
 - com.tjach.horoskop
 - MainActivity.java
 - gen [Generated Java Files]
 - assets
 - bin
 - libs
 - res
 - drawable-hdpi
 - background.jpg
 - ic_launcher.png
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi

MainActivity.java *activity_main.xml strings.xml styles.xml dir

```
<RelativeLayout 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:background="@drawable/background"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.tjach.horoskop.MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:labelFor="@+id/editText1"
        android:text="@string/wybor"
```

Ćwiczenie 1



- Wykonaj prostą aplikację typu horoskop.
- Do pliku strings.xml dodaj tablicę znaków zodiaku.
- Na formie dodaj element odpytujący użytkownika o datę urodzenia. Za pomocą logiki aplikacji sprawdź jego znak zodiaku.
- Wyświetl mu horoskop dobrany dla jego znaku zodiaku
- Dodatkowo, zdefiniuj w pliku zasobów odpowiednie tła dla każdego znaku zodiaku.
- Użyj komponentu DatePicker z dodatkową właściwością: `android:calendarViewShown="false"`



Podpinanie zdarzeń - 1

```
11 public class MainActivity extends Activity {
12
13     Button b1;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         b1=(Button) findViewById(R.id.button1);
19         OnClickListener l = new OnClickListener() {
20
21             @Override
22             public void onClick(View arg0) {
23                 Toast.makeText(getApplicationContext(),"Om nom nom", Toast.LENGTH_SHORT).show();
24             }
25         };
26         b1.setOnClickListener(l);
27     }
28 }
29
30
31
32 }
```

Tworzymy nowy listener, przeciążamy jego metodę `onClick()` i całość podpinamy za pomocą `findViewById()` do odpowiedniego przycisku.

Podpinanie zdarzeń - 2



```
11 public class MainActivity extends Activity {
12
13     Button b1;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         b1=(Button) findViewById(R.id.button1);
19         b1.setOnClickListener(new OnClickListener() {
20
21             @Override
22             public void onClick(View v) {
23                 Toast.makeText(getApplicationContext(),"Om nom nom", Toast.LENGTH_SHORT).show();
24             }
25         });|
26     }
27
28
29 }
```

Jak wcześniej, ale tu stworzymy listener niejako „w locie”. Mniejsza czytelność, ale mniej kodu.



Podpinanie zdarzeń - 3

Tworzymy w kodzie metodę, która nic nie zwraca (void), jako jedyny parametr przyjmuje View v (obiekt, który wywołał zdarzenie – po View dziedziczą praktycznie wszystkie elementy na formie):

```
public void reakcjaKlikniecia(View v) {  
    Toast.makeText(getApplicationContext(), "Cos kliknales", Toast.LENGTH_SHORT).show();  
}
```

W pliku xml aktywności dodajemy:

android:onClick=„<NAZWA FUNKCJI>”

np.

android:onClick="reakcjaKlikniecia,,

Można podpinąć jedną funkcję do wielu przycisków.

```
MainActivity... activity_mai... strings.xml styles.xml  
android:layout_marginTop="37dp"  
android:calendarViewShown="false"  
android:measureAllChildren="true" />  
  
<Button  
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginBottom="50dp"  
    android:text="@string/dzialaj"  
    android:onClick="reakcjaKlikniecia"/>  
  
<Button  
    android:id="@+id/button2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignBaseline="@+id/button1"  
    android:layout_alignLeft="@+id/datePicker1"  
    android:text="@string/dzialaj"  
    android:onClick="reakcjaKlikniecia"/>  
  
<Button  
    android:id="@+id/button3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignBaseline="@+id/button1"  
    android:layout_alignBottom="@+id/button1"  
    android:layout_alignRight="@+id/datePicker1"  
    android:text="@string/dzialaj"  
    android:onClick="reakcjaKlikniecia"/>  
</RelativeLayout>
```

Jedna funkcja – wiele działań



- Do każdego z przycisków dokładam w xmlu: `android:tag=„<Tu nazwa/int/cokolwiek>„`

- W funkcji odpowiedzialnej za reakcję na kliknięcie:

```
public void reakcjaKlikniecia(View v) {  
    Toast.makeText(getApplicationContext(), "Kliknales: " + v.getTag(), Toast.LENGTH_SHORT).show();  
}
```

- **Nie zmieniamy ID, zmieniamy sobie wartość tag.**



ImageView

- Jest to komponent do wyświetlania obrazków na formie.
- Obrazki przechowujemy w plikach zasobów



```
<ImageView  
    android:id="@+id/imageView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="74dp"  
    android:src="@drawable/ic_launcher" />
```

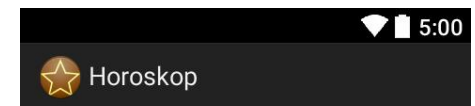
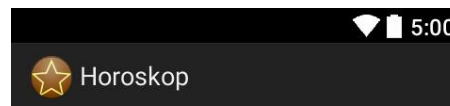
ImageView

- Wewnątrz kodu możemy zmieniać obrazek wyświetlany w ImageView:

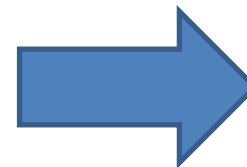
```
}
```

```
public void reakcjaKlikniecia(View v) {  
    ImageView im = (ImageView) findViewById(R.id.imageView1);  
    im.setImageResource(R.drawable.background);  
}
```

```
}
```



Button



Ćwiczenie 2



- Oprogramuj grę w kółko i krzyżyk dla dwóch osób.
- W grze ma występować plansza oraz naprzemienne ruchy graczy. Po kliknięciu na dane pole – powinno ono się zmienić na kółko lub krzyżyk odpowiedniego gracza
- Gra powinna zliczać liczbę wygranych każdego gracza



Menu aplikacji

Android Menu

Menu Elements I G Az

- I action_settings (Item) Add... Remove... Up Down

Attributes for action_settings (Item)

Base attributes that are available to all Item objects.

Id	@+id/action	Browse...
Menu category		
Order in category	100	
Title	@string/actio	Browse...
Title condensed		Browse...
Icon		
Alphabetic shortcut		Browse...
Numeric shortcut		Browse...
Checkable		
Checked		
Visible		
Enabled		
On click		Browse...
Show as action	never	Select...
Action layout		Browse...
Action view class		Browse...

Package Explorer

- Horoskop
 - Android 4.4.2
 - Android Private Libraries
 - src
 - com.tjach.horoskop
 - MainActivity.java
 - gen [Generated Java Files]
 - assets
 - bin
 - libs
 - res
 - drawable-hdpi
 - background.jpg
 - ic_launcher.png
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - menu
 - main.xml
 - values
 - dims.xml
 - strings.xml
 - styles.xml
 - values-v11
 - values-v14
 - values-w820dp
 - AndroidManifest.xml
 - ic_launcher-web.png
 - proguard-project.txt

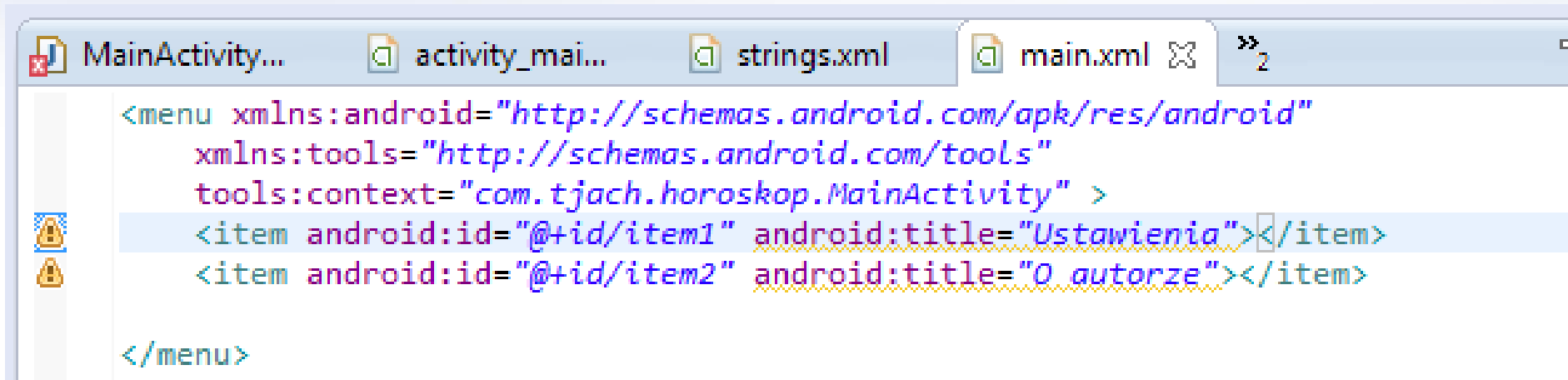
Dodajemy elementy podobnie jak stringi do xmla, ale tu jest więcej opcji (kolejności, itp.)

W tym pliku definiujemy elementu menu



Menu aplikacji

- Bezpośrednio z kodu też można:



```
MainActivity... activity_mai... strings.xml main.xml » 2
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="com.tjach.horoskop.MainActivity" >
  <item android:id="@+id/item1" android:title="Ustawienia"></item>
  <item android:id="@+id/item2" android:title="O autorze"></item>
</menu>
```



Menu aplikacji

- A odwołujemy się do tego tak:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    switch(id) {
        case R.id.item1:
            Toast.makeText(getApplicationContext(), "Opcja 1 - Ustawienia", Toa
            break;
        case R.id.item2:
            Toast.makeText(getApplicationContext(), "Opcja 2 - o Autorze", Toas
            break;
        default:
            Toast.makeText(getApplicationContext(), "Siakis blad", Toast.LENGTH
            break;
    }

    return super.onOptionsItemSelected(item);
}
}
```

Ćwiczenie 3



- Napisz prosty konwerter jednostek
- W menu użytkownik wybiera rodzaj konwersji, w wyniku włączana jest nowa aktywność
- Konwersje to: temperatura, odległość, pojemność
- Na każdym ekranie zadбай o schludny wygląd

Ćwiczenie 4



- Napisz grę polegającą na jak najszybszym rozwiązywaniu zadań matematycznych (dodawanie, odejmowanie, mnożenie i dzielenie dwóch liczb)
- W menu wybieraj poziomy trudności (wielkość liczb)
- Znajdź sposób na mierzenie czasu jaki użytkownik potrzebuje na wykonanie zadania
- Po wykonaniu 10 działań – wyświetl łączny czas.
- Dodaj okienko „HighScores”. Zadbaj o graficzny wygląd, w tym okienko z fajerwerkami na końcu.

Ćwiczenie 5



- Przygotuj aplikację-quiz na wybrany przez siebie temat.
- Aplikacja ma wyświetlać zdjęcie wybranego obiektu, a użytkownik ma wpisać co to za przedmiot (lub wybrać z kilku opcji zaproponowanych przez Ciebie).
- Tematyka dowolna: znane osoby, marki zabawek, loga firm, itp.
- Przy wpisywaniu tekstu – zadbaj o to, aby małe/wielkie litery nie powodowały braku uznania odpowiedzi.