Programowanie urządzeń mobilnych dr inż. Andrzej Grosser na podstawie wykładu dr inż. Juliusza Mikody

Wprowadzenie

- Android 1.0 (API level 1) 2008
- Android 1.5 Cupcake (API level 3) 2009
- Android 2.0 Eclair (API level 5) 2009
- Android 2.3 Gingerbread (API level 9) 2010
- Android 3.0 Honeycomb (API level 11) 2011

Wprowadzenie

- Android 4.0 Ice Crea Sandwich (API level 14) 2011
- Android 4.1 Jelly Bean (API level 16) 2012
- Android 4.4 (API level 19) KitKat 2013
- Android 4.4w (API level 20) KitKat 2014
- Android 5.0 (API level 21) Lollipop 2014
- Android 6.0 (API level 23) Marshmallow 2015

Wprowadzenie

- Android 7.0 (API level 24) Nougat 2016
- Android 8.0 (API level 26) Oreo 2017

Warstwy Androida

- Jądro systemu Linux + sterowniki
- Warstwa abstrakcji sprzętu (HAL)
- Biblioteki uruchomieniowe (Dalvik/ART)
- Biblioteki (Webkit, OpenGL ES, Media Framework)
- Szkielet aplikacji (Application framework)
- Aplikacje (Email, Kalendarz)

Składniki aplikacji

- Aktywności (ang. Activities)
- Serwisy (ang. Services)
- Menadżery treści (ang. Content providers)
- Odbiorniki powiadomień (ang. Broadcast receivers)

Aktywności

- Podstawowy składnik każdej aplikacji.
- Reprezentują pojedynczy ekran interfejsu użytkownika.
- Aktywności są klasami pochodnymi od Activity.

Serwisy

- Służą do pracy w tle systemu operacyjnego mogą pracować nawet gdy użytkownik przełączy się na inną aplikację.
- Są wykorzystywane do wykonywania długotrwałych operacji lub pracy ze zdalnymi procesami (połączenia sieciowe).
- Są klasami pochodnymi od Service

Dostawcy treści

- Są warstwą ułatwiającą dostęp do danych zapisanych w urządzeniu.
- Pozwalają na bezpieczne współdzielenie danych aplikacji.
- Dziedziczą po klasie ContentProvider

Odbiornicy powiadomień

- Ich zadaniem umożliwia komunikację pomiędzy składnikami systemu.
- Pozwalają na nasłuchiwanie i odpowiedź na określone zdarzenie.
- Dziedziczą po klasie BroadcastReceiver

Instalacja środowiska

- Środowisko programowania
 - Android Studio

http://developer.android.com/sdk/installing/studio.html

Instalacja środowiska

- ADT jest dostarczane w komplecie z Eclipse i SDK dla Androida – instalacja polega na rozpakowaniu archiwum.
- Android Studio jest dostarczane razem z instalatorem – wystarczy uruchomić go i postępować zgodnie z zaleceniami.

Konfiguracja środowiska 1

SDK Path: /Applications/!programy/android-sdk-macosx/

Packages

Ŵ	Name API Rev. Status			
$\mathbf{\overline{v}}$	Tools			
	Android 4.1 (API 16)			
	Android 4.0.3 (API 15)			
	Android 4.0 (API 14)			
	Android 3.2 (API 13)			
	Android 3.1 (API 12)			
	Android 3.0 (API 11)			
Ø	Android 2.3.3 (API 10)			
	Android 2.2 (API 8)			
	Android 2.1 (API 7)			
	Android 1.6 (API 4)			
	Android 1.5 (API 3)			
	🕨 🧰 Extras			

Konfiguracja środowiska 2

00	Android SI	DK Manager	
K Path: /Applicatio	ns/!programy/adt/sdk		
ckages			
I Name		ΔPI Rev	Status
			Status
Android 4.	3 (API 18)		
► ► Android 4.	2.2 (API 17)		
Android 4.	1.2 (API 16)		
Android 4.	0.3 (API 15)		
🛛 🕨 🔂 Android 4.	0 (API 14)		
] 🕨 🔁 Android 3.	2 (API 13)		
] 🕨 🔂 Android 3.	1 (API 12)		
📔 🕨 🔂 Android 3.	0 (API 11)		
🗴 🕨 🔁 Android 2.	3.3 (API 10)		
) 🕨 🔁 Android 2.	2 (API 8)		
Android 2.	1 (API 7)		
Android 1.	6 (API 4)		
Android 1.	5 (API 3)		
Extras			
now: 🗹 Updates,	/New 🗹 Installed 🛛 🗌 Ob	osolete Select New or Update	es Install package
-	~		(
ort by: 💽 API level	Repository	Deselect All	Delete 16 packa

File > New > Projekt... Android > Android Project

	new Android Application		
New Android Applica	ation		
Creates a new Android Application			
Application Name	e: 0 HelloWord		
Project Name	e: 0 HelloWord		
Package Name	e: 0 pl_example.helloword		
Minimum Required SDR	(: API 10: Android 2.3.3 (Gingerbread)		
Target SD	(: 1) API 10: Android 2.3.3 (Gingerbread) +		
Compile With	n: 1 API 10: Android 2.3.3 (Gingerbread) +		
Them	2: 0 None +		
• The package name	e must be a unique identifier for your application.		
0			

000	New Android Application		
New Andro	oid Application		
Configure I	Project		
Create c	ustom launcher icon		
Create activity			
Mark this project as a library			
Create P	Project in Workspace		
Location:	/Users/mikoda/Documents/workspace/HelloWord Browse		
Working s	sets		
🗌 Add j	project to working sets		
Working	sets: \$ Select		
?	< Back Next > Cancel Finish		

000	New Android Application		
Configure Launch Configure the attri	er lcon butes of the icon set		
Foreground: Foreground Scaling: Shape Background Color:	Image Clipart Text Image File: launcher_icon Image File: launcher_icon Image Trim Surrounding Blank Space Additional Padding: Image Clipart Text Image File: launcher_icon Image File: launcher_icon <	Browse 0%	Preview: mdpi: hdpi: whdpi: whdpi: whdpi: xxhdpi:
?	< Back Next >	Ca	ncel Finish

000 New Android Application Create Activity Select whether to create an activity, and if so, what kind of activity. Create Activity **Blank Activity** Fullscreen Activity Master/Detail Flow ~~~~~ **Blank Activity** Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

< Back

Next >

Cancel

Finish

?

000 New Android Application Create Activity Select whether to create an activity, and if so, what kind of activity. Create Activity **Blank Activity** Fullscreen Activity Master/Detail Flow ~~~~~ **Blank Activity** Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

< Back

Next >

Cancel

Finish

?

000	New Android Application	
Blank Activity Creates a new bla	nk activity, with an action bar and optional navigational elements such as tabs or	
horizontal swipe.		
Activity Name 0	MainActivity	
Layout Name 🛙	activity_main	
Navigation Type	<pre>/ None Fixed Tabs + Swipe Scrollable Tabs + Swipe Dropdown</pre>	
♀ The name of the	activity class to create	
?	Sack Next > Cancel Finite	sh

Pierwszy program – konstrukcja

- Zawartość utworzonego projektu:
- src katalog pakietów kodu programu
- gen dane generowane przez kompilator na podstawie zasobów
- assest dodatki katalog plików niekompilowalnych
- res zasoby
- AndroidManifest.xml plik konfiguracyjny Aplikacji android



Pierwszy program – zasoby

res / value

- łańcuchy tekstowe
- definicje kolorów, elementów rysowanych
- definicja rozmiarów
- definicja styli

۰.

. . .

definicja tablic tekstowych / liczbowych



☐ strings.xml 🔀	
🚔 Android Resources (default)	
Resources Elements S O S S I Az S hello (String) Add Add Remove S app_name (String) Up Down	Attributes for hello (String) Strings, with optional simple formatting, can be stored and retrieved as resources. You can add formatting to your string by using three standard HTML tags: b, i, and u. If you use an apostrophe or a quote in your string, you must either escape it or enclose the whole string in the other kind of enclosing quotes. Name* hello Value* Hello World, HelloAndroid!
Resources strings.xml	

Pierwszy program – układ

layout
 activity_main.xml
 menu
 activity_main.xml

<RelativeLayout xmlns:android= "http://schemas_android_com/ank//

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

android:layout_width="match_parent"
android:layout_height="match_parent" >

<TextView

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:text="@string/hello_world"
tools:context=".MainActivity" />

</RelativeLayout>

- Elementy pozycjonujące
 - TableLayout
 - LinearLayout
 - RelativeLayout
 - AbsoluteLayout
 - FrameLayout
- Kontrolki
 - TextView
 - EditText
 - Button
 - AnalogClock
 - DigitalClock
 - a ...

Pierwszy program – R.java

🗁 res 🔻 🗁 drawable-hdpi 🙀 ic_launcher.png 🗁 drawable-ldpi 🔻 🗁 drawable-mdpi 🙀 ic_launcher.png 🔻 🗁 drawable-xhdpi 🙀 ic_launcher.png 🔻 🗁 drawable-xxhdpi 🐼 ic_launcher.png

package pl.example.helloword; public final class R { public static final class drawable { public static final int ic_action_search=0x7f02000; public static final int ic_launcher=0x7f020001; public static final class id { public static final int menu settings=0x7f070000; public static final class layout { public static final int activity main=0x7f030000; public static final class menu { public static final int activity_main=0x7f060000; public static final class string { public static final int app name=0x7f040000; public static final int hello_world=0x7f040001; public static final int menu settings=0x7f040002;

public static final int title_activity_main=0x7f040003;

Pierwszy program – Start

†	🗭 🗊 📶 💈 14:57
MainActivity	

package pl.example.helloword;

Import

public class MainActivity extends Activity {

@Override

public void onCreate(Bundle savedInstanceState) {

```
super.onCreate(savedInstanceState);
```

setContentView(R.layout.activity_main);

```
Hello world!
```

@Override

}

public boolean onCreateOptionsMenu(Menu menu) {
 getMenuInflater().inflate(R.menu.activity_main,
 menu);

return true;

worze	enie wyglądu okna
The second sec	🖸 🔹 🔲 Nexus One 🔹 🔄 🔹 🛧 AppTheme 🔹 🕞 MainActivity 🔹 🌎 🔹 🚔 18 🔹
🗁 Form Widgets	
OK Button	
CheckBox	HelloWord
CheckedTextView	
Ab Large Text	
Ab Medium Text	
ProgressBar (Horizon)	
ProgressBar (Large)	
ProgressBar (Normal)	
ProgressBar (Small)	
QuickContactBadge	
RadioButton	Lielle weddi
🖹 RadioGroup	Hello World:
🚖 RatingBar	
🦲 Text Fields	
🗀 Layouts	
Composite	
🗀 Images & Media	
🗀 Time & Date	
C Transitions	
C Advanced	
🗀 Other	
🗀 Custom & Library Vi	Structure ===

- View klasa bazowa wszystkich kontrolek interfejsu użytkownika systemu android.
- Zajmuje prostokątny wycinek obrazu ekranu.
- Klasa zajmuje się także obsługa podstawowych zdarzeń.
- Znane bezpośrednie podklasy: AnalogClock, ImageView, KeyboardView, MediaRouteButton, ProgressBar, Space, SurfaceView, TextView, TextureView, ViewGroup, ViewStub.

Przydatne metody chronione:

- onMeasure(int, int) określenie wymagań wielkości elementu.
- onLayout(boolean, int, int, int, int) przypisanie rozmiarów elementów i jego dzieci.
- onDraw(android.graphics.Canvas) przerysowanie zawartości okna.
- onKeyDown(int, KeyEvent) wciśnięcie klawisza systemowego.
- onKeyUp(int, KeyEvent) zwolnienie klawisza systemowego
- onTrackballEvent(MotionEvent) zarządzanie ruchem trackballa.
- onTouchEvent(MotionEvent) zarządzanie ekranem dotykowym.
- onWindowFocusChanged(boolean) zmiana focusa ekranu.
- onWindowVisibilityChanged(int) zmiana widoczności kontrolki.

Przypisanie słuchaczy ogólnych:

- setOnClickListener (View.OnClickListener I) ustawienie słuchacza zdarzenia kliknięcia elementu,
- setOnFocusChangeListener (View.OnFocusChangeListener I) ustawienie słuchacza zmiana focusa kontrolki,
- setOnKeyListener (View.OnKeyListener I) ustawienie słuchacza zdarzenia wciśnięcia klawisza,
- setOnLongClickListener (View.OnLongClickListener I) ustawienie słuchacza przedłużonego kliknięcia kontrolki,
- setOnTouchListener (View.OnTouchListener I) ustawienie słuchacza zdarzeń ekranu dotykowego,

Przydatne atrybuty:

- android:background kolor tła kontrolki
- android:clickable, android:longClickable aktywacja zdarzenia kliknięcia elementu
- android:focusable umożliwia przejmowanie fokusa przez kontrolkę
- android:id identyfikator kontrolki
- android:layout_width, android:layout_height szerokość i wysokość kontrolki
- android:onClick przypisanie nazwy metody odpowiedzialnej za zdarzenie kliknięcia na kontrolce

Przydatne atrybuty:

- android:padding, android:paddingBottom, android:paddingLeft, android:paddingRight, android:paddingTop – rozmiar wewnętrznej ramki kontrolki,
- android:tag atrybut zawierający dodatkową zmienną (obiekt klasy) powiązany z kontrolką,
- android:visibility widoczność kontrolki (gone, invisible, visible),
- android:layout_width, android:layout_height szerokość i wysokość kontrolki,
- android:layout_margin, android:layout_marginBottom, android:layout_marginLeft, android:layout_marginRight, android:layout_marginTop – margines zewnętrzny kontrolki.

Kontrolki TextView

Kontrolka TextView pozwala na wyświetlanie tekstu lecz nie zezwala na jego edycję. Klasa bazowa dla: Button, CheckedTextView, Chronometer, DigitalClock, EditText, TextClock

<TextView android:id="@+id/label" android:layout_width ="fill_parent" android:layout_height ="wrap_content" android:text



="Wprowadzony tekst" />

Kontrolki TextView

Przydatne atrybuty:

- android:autolink rozpoznawanie linków webowych,
- android:drawableBottom, android:drawableLeft, android:drawableRight, android:drawableTop - dodatkowy obraz w kontrolce tekstowej
- android:drawablePadding odległość obrazka,
- android:editable ustawienie jako kontrolka edytowalna,
- android: ellipsize skracanie tekstu po przekroczeniu dostępnego rozmiaru kontrolki,
- android:gravity pozycja tekstu wewnątrz kontrolki,
- android:hint wskazówka, jeśli tekst jest pusty,
- android:inputType typ wprowadzanej wartości,

Kontrolki TextView

Przydatne atrybuty:

- android:lines liczba linii tekstu,
- android:maxLines maksymalna liczba linii tekstu,
- android:maxLength maksymalna długość tekstu,
- android:singleLine ustawienie pojedynczej linii tekstu,
- android:text tekst wyświetlany przez kontrolkę,
- android:textColor kolor tekstu,
- android:textColorHighlight kolor tekstu po kliknięciu,
- android:textColorHint kolor podpowiedzi,
- android:textColorLink kolor linku webowego,
- android:textSize rozmiar tekstu,
- android:textStyle styl tekstu,
- android:typeface czcionka wyświetlanego tekstu.

Kontrolki EditText

Kontrolka EditText pozwala na pobranie i edycję tekstu przez użytkownika. Klasa bazowa dla: AutoCompleteTextView, ExtractEditText



<EditText android:id="@+id/pole1" android:layout_height="wrap_content" android:layout_width="fill_parent" android:text="Tekst wyswietlany na oknie" />

Kontrolki EditText

Przydatne atrybuty:

android:inputType – formatowania danych wejściowych

(none, text, textCapCharacters, textCapWords, textCapSentences, textAutoCorrect, textAutoComplete, textMultiLine, textImeMultiLine, textNoSuggestions, textUri, textEmailAddress, textEmailSubject, textShortMessage, textLongMessage, textPersonName, textPostalAddress, textPassword, textVisiblePassword, textWebEditText, textFilter, textPhonetic, textWebEmailAddress, textWebPassword, number, numberSigned, numberDecimal, numberPassword, phone, datetime, date, time)

Metody:

- Editable getText()
- setText(CharSequence text, TextView.BufferType type)

Kontrolka Button pozwala na umieszczenie przycisku na ekranie urządzenia. Znane podklasy: CheckBox, RadioButton, Switch, ToggleButton

<Button
android:text="Przycisk 1"
android:id="@+id/Button01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin ="10px" />



Obsługa naciśnięcia przycisku:

public class MainActivity extends Activity implements

```
View.OnClickListener {
```

protected void onCreate(Bundle icicle) {
 super.onCreate(icicle);
 setContentView(R.layout.main);
 // pobranie obiektu kontrolki z widoku
 final Button button = (Button) findViewById(R.id.Button01);
 // dołączenie słuchacza powiązanego z kontrolką
 button.setOnClickListener(this);

```
}
public void onClick(View v) {
    // action on click
```

Obsługa naciśnięcia przycisku:

public class MainActivity extends Activity { protected void onCreate(Bundle icicle) { super.onCreate(icicle); setContentView(R.layout.main); // pobranie obiektu kontrolki z widoku final Button button = (Button) findViewById(R.id.Button01); // dołączenie słuchacza powiązanego z kontrolką button.setOnClickListener(new View.OnClickListener() { public void onClick(View v) { // action on click });

Obsługa naciśnięcia przycisku – sposób alternatywny (od wersji 1.6):

<Button

android:layout_height="wrap_content" android:layout_width="wrap_content" android:text="@string/self_destruct" android:onClick="selfDestruct" />

W programie:

public void selfDestruct(View view) {
 // action on click

Układ okna – LinearLayout

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

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

android:layout_width="fill_parent,, android:layout_height="fill_parent,, android:orientation="vertical" >

</LinearLayout>



Układ okna – LinearLayout

Rozmieszczenie elementów podrzędnych w poziomie lub w pionie. Dostępne atrybuty:

- android:gravity orientacja elementów podrzędnych,
- android:orientation kierunek układania elementów podrzędnych (vertical, horizontal).
- Atrybuty elementu podrzędnego
- android:layout_width rozmiar poziomy elementu podrzędnego:
 - ill_parent / mach_parent w pełni wypełniona zawartość,
 - wrap_conten wypełnienie dopasowane do zawartości,
 - 10px, 5dp ustalony rozmiar,
- android:layout_height rozmiar pionowy elementu podrzędnego
- android:layout_gravity położenie komponentu na oknie

Układ okna – TableLayout



- Rozmieszczenie elementów podrzędnych w postaci tabeli
- android:collapseColumns
 - liczba elementów w kolejnych wierszach
- android:stretchColumns="0,1,2"
 - równomierne rozciągnięcie kolumn
- Element xml'a <TableRow> ...
 </TableRow> określa jeden wiersz rozmieszczenia na ekranie

Układ okna – TableLayout

<?xml version="1.0" encoding="utf-8"?> <TableLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_height="wrap_content" android:layout_width="fill_parent"> <EditText android:text="Bez Row" /> <TableRow> <TextView android:text="Nazwisko" /> <EditText android:text="N1" /> <EditText android:text="N2" /> </TableRow> <TableRow> <TextView android:text="Pesel" /> <EditText android:text="Text" /> </TableRow> <TableRow> <EditText android:text="N1" android:layout_height="150px"/> <EditText android:text="N2" /> <EditText android:text="N3" android:layout gravity="bottom"/> <EditText android:text="N4" /> <EditText android:text="N5" /> </TableRow> </TableLayout>

Układ okna – FrameLayout

Pozwala wyświetlić elementy, gdzie bazą dla każdego elementu będzie lewy, górny róg ekranu.



<FrameLayout

android:layout_width="fill_parent,, android:layout_height="fill_parent"> <ImageView

android:layout_width="fill_parent" android:layout_height="fill_parent" android:scaleType="center" android:src

="@drawable/golden_gate" />

<TextView

android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_gravity

="center_horizontal|bottom" android:layout_marginBottom="20dip" android:padding="12dip" android:background="#AA000000" android:textColor="#ffffffff" android:text="Golden Gate" /> </FrameLayout>

Układ okna – AbsoluteLayout

Pozwala wyświetlić elementy w określonych współrzędnych ekranu. Elementy mogą zachodzić na siebie.



<?xml version="1.0" encoding="utf-8"?> <AbsoluteLayout xmlns:android ="http://schemas.android.com/apk/res/android" android:layout_width="wrap_content" android:layout_height="wrap_content"> <EditText android:layout height="wrap content" android:layout_width="wrap_content" android:layout_x="20px" android:layout_y="20px" android:text="20x20" /> <EditText android:layout height="wrap content" android:layout_width="wrap_content" android:layout_y="80px" android:layout x="40px" android:text="40x80" /> <EditText android:layout height="wrap content" android:layout_width="wrap_content" android:layout_x="60px" android:layout_y="50px" android:text="60x50" /> </AbsoluteLayout>

Rozmieszczenie elementów podrzędnych relatywnie - względem innych elementów lub układu nadrzędnego



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

xmlns:android="http://schemas.android.com/ apk/res/android" android:layout_width="wrap_content" android:layout_height="wrap_content">

<TextView android:id="@+id/label" android:layout_width="fill_parent" android:layout_height="wrap_content" android:layout_marginTop="10px" android:layout_marginLeft="10px" android:text="Wprowadź tekst:" />

<EditText android:id="@+id/entry" android:layout_width="fill_parent" android:layout_height="wrap_content" android:layout_margin="10px" android:layout_below="@id/label" />



<Button android:id="@+id/cancel"

android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_below="@id/entry" android:layout_alignParentRight="true" android:layout_marginLeft="10px" android:layout_marginRight="10px" android:text="Wycofaj" />

<Button android:id="@+id/ok" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_toLeftOf="@+id/cancel" android:layout_alignTop="@+id/cancel" android:text="Zapisz" />

</RelativeLayout>

Pozycje konfiguracyjne przyjmujące Id elementu:

- android:layout_above umieszczenie nad elementem o danym id
- android:layout_alignBaseline powiązanie z liniami bazowymi elementów
- android:layout_alignBottom powiązanie dolnych linii
- android:layout_alignLeft powiązanie lewych krawędzi
- android:layout_alignRight powiązanie prawych krawędzi
- android:layout_alignTop powiązanie górnych krawędzi
- android:layout_below pod elementem
- android:layout_toLeftOf po lewej od elementu
- android:layout_toRightOf po prawej od elementu

Właściwości modyfikacji elementów typu boolean:

- android:layout_alignParentBottom dowiązanie do dolnej linii rodzica
- android:layout_alignParentLeft dowiązanie do lewej linii rodzica
- android:layout_alignParentRight dowiązanie do prawej linii rodzica
- android:layout_alignParentTop dowiązanie do górnej linii rodzica
- android:layout_centerHorizontal wycentrowanie poziome
- android:layout_centerInParent wycentrowanie w stosunku do rodzica
- android:layout_centerVertical wycentrowanie pionowe