

KMI / TMA

TVORBA MOBILNÍCH APLIKACÍ

3. SEMINÁŘ | 30.9.2020
ZS 2020/2021 | STŘEDA 15:00-17:30

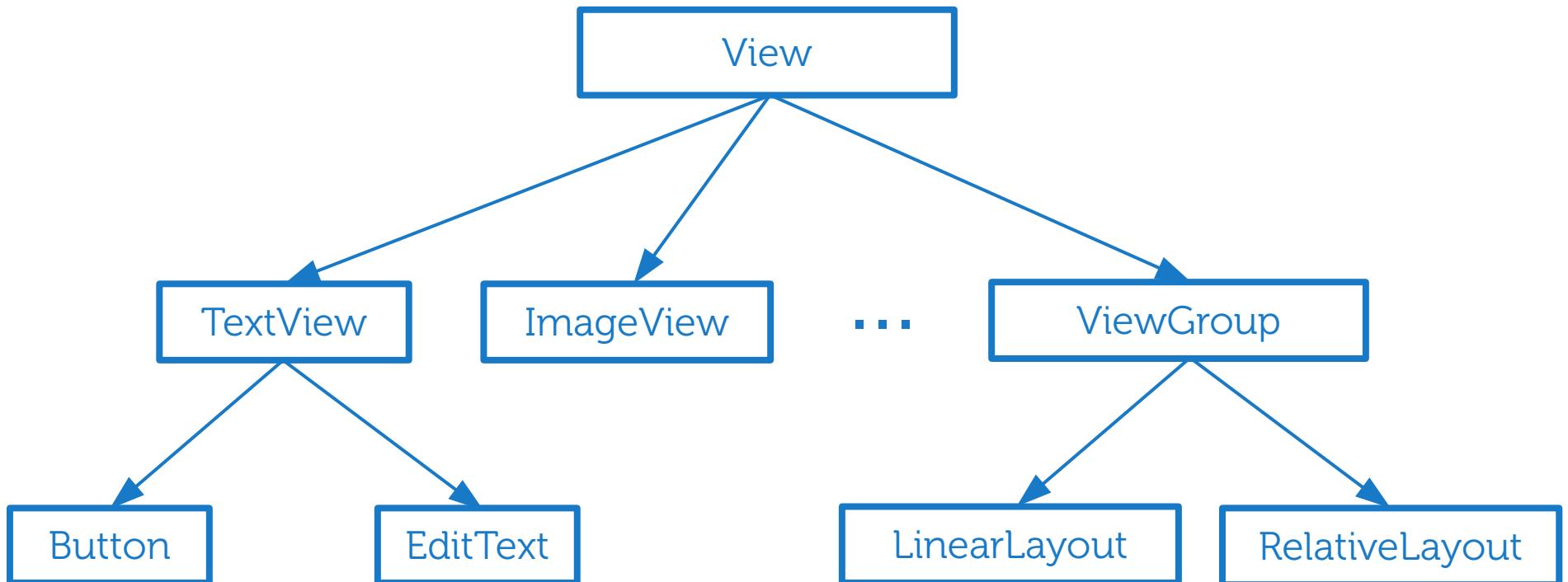
OBSAH SEMINÁŘE

**VZTAH AKTIVIT A LAYOUTŮ,
VIEWS A LAYOUTY PODROBNĚ,
PŘIZPŮSOBENÍ SE HW**

HIERARCHIE VIEWS

CO VŠECHNO MŮŽEME PŘIDAT DO LAYOUTŮ?

- » potomky `android.view.View`, základního stavebního bloku uživatelských rozhraní



VIEWS

VNOŘENÍ VIEWS

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp">
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"/>
```

```
</RelativeLayout>
```

VIEWS

ZÁKLADNÍ ATRIBUTY

- » **xmlns:android**
 - » definice namespace **android**, povinná v kořenovém prvku
- » **android:id**
 - » jednoznačný identifikátor **View** v rámci layoutu (nikoliv aplikace!)
 - » pro získání instance **View** v kódu
 - » pro definici vztahů v layoutu

INTERMEZZO

JEDNOTKY

- » **mm** (milimetry), **in** (palce), **pt** (typ. body)
 - » nepoužívat pls
- » **px** (pixely)
 - » pixely mohou být na každém zařízení jinak velké
 - » také nepoužívat
- » **dp** (density-independent pixels)
 - » yay!

INTERMEZZO

JEDNOTKY

- » **dp** (density-independent pixels)
 - » virtuální jednotka, která se přepočítává dle hustoty obrazových bodů (pixelů)
 - » hustota obrazových bodů závisí na rozlišení a úhlopříčce
 - » přepočty např. na <http://dpi.lv/>
 - » jsou zavedeny kategorie hustot pixelů: ldpi (0.75x), **mdpi**, hdpi (1.5x), xhdpi (2x), xxhdpi (3x), xxxhdpi (4x),...

INTERMEZZO

JEDNOTKY

- » **dp** (density-independent pixels)
 - » pro mdpi (~160dpi) platí **dp = px**
 - » pro ostatní kategorie se virtuální **dp** na skutečné **px** přepočítávají při běhu
 - » <https://pixplicity.com/dp-px-converter/>
 - » **dp** řeší problém se zobrazením stejných rozměrů prvků na různých hustotách displejů
 - » **dp** se používá napříč SDK
 - » **sp** = dp, které zohledňuje zvětšení písma v nastavení syst., používat pro velikosti písma

VIEWS

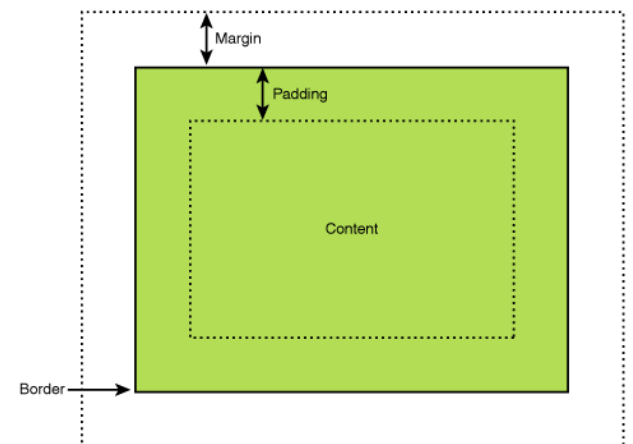
ZÁKLADNÍ ATRIBUTY

- » **android:layout_width** a **layout_height**
 - » šířka/výška prvku
 - » může být určena pevnou konstantou společně s jednotkou (dp, px, ...) nebo
 - » může být použita jedna z konstant
 - » **match_parent**
 - » velikost stejná jako rodič
 - » **wrap_content**
 - » velikost dle obsahu

VIEWS

ZÁKLADNÍ ATRIBUTY

- » `android:padding*` a `padding`
 - » vnitřní okraj prvku
- » `android:layout_margin*` a `layout_margin`
 - » vnější okraj prvku
- » kde * je bottom, top, left, right
- » hodnota + jednotka
- » atribut bez suffixu nastavuje všechny strany



VIEWS

ATRIBUTY JEN PRO NĚKTERÉ VIEW

- » **android:text**
 - » řetězec nebo identifikátor řetězce
- » **android:textColor**
 - » barva písma v hexadecimálním formátu nebo identifikátor barvy
- » **android:background**
 - » barva pozadí v hexadecimálním formátu nebo identifikátor barvy
- » ... (v layoutu **ctrl+space** pro zobrazení dalších atributů, případně použít designer)

ZÁKLADNÍ VIEWS

TEXTVIEW

- » základní textové pole
- » zápis:
 - » v layoutu: `<TextView>`
 - » v kódu: `android.widget.TextView`
- » speciální atributy
 - » **text**: obsah – řetězec
 - » **textColor**: barva textu – hexadecimální zápis
 - » **textSize**: velikost textu – hodnota + jednotka

ZÁKLADNÍ VIEWS

EDITTEXT

- » textové pole pro editaci (input)
- » zápis:
 - » v layoutu: `<EditText>`
 - » v kódu: `android.widget.EditText`
- » speciální atributy
 - » třída dědí z `TextView`, dědí i atributy
 - » `inputType`: typ vkládaného textu, konstanty jako `text`, `number`, `textPassword`, ...

ZÁKLADNÍ VIEWS

BUTTON

- » tlačítko
- » zápis:
 - » v layoutu: `<Button>`
 - » v třídě: `android.widget.Button`
- » speciální atributy
 - » třída dědí z `TextView`, dědí i atributy
 - » nemá definované žádné vlastní atributy

ZÁKLADNÍ VIEWS

IMAGEVIEW

- » obrázek
- » zápis:
 - » v layoutu: `<ImageView>`
 - » v třídě: `android.widget.ImageView`
- » speciální atributy
 - » **src**: zdroj obrázku
 - » **scaleType**: pravidlo jak se zdroj zobrazí v ImageView

ZÁKLADNÍ VIEWS

ABSOLUTELAYOUT

- » prvky v layoutu jsou zobrazeny podle pevných hodnot/souřadnic

ZÁKLADNÍ VIEWS

ABSOLUTELAYOUT

- » prvky v layoutu jsou zřetelzovány podle pevných souřadnic

nepoužívejte to!

ZÁKLADNÍ VIEWS

LINEARLAYOUT

- » prvky v layoutu jsou poskládány za sebou nebo vedle sebe
- » **android:orientation**
 - » směr pokládání nových Views
 - » může nabývat hodnot **horizontal** nebo **vertical**

ZÁKLADNÍ VIEWS

RELATIVELAYOUT

- » prvky v layoutu mají definovány vztahy mezi sebou
- » prvkům je možné přiřadit atributy, např:
 - » **alignParentLeft**
 - » zda je prvek připnut k levé hraně layoutu, true|false
 - » **toLeftOf**
 - » „nalevo od“, identifikátor prvku
 - » **alignLeft**
 - » levá hrana bude stejná jako levá hrana jiného prvku, identifikátor prvku

AKTIVITY VS LAYOUTY

JAKÝ MAJÍ VZTAH?

- » nikde není před kompilací definováno jaký layout používá která aktivita
- » **Activity setContentView(int)**
 - » načte layout, vytvoří instance Views a přidá je do těla aktivity
 - » reference jednotlivých Views získáme pomocí metody **Activity.findViewById(int)**

AKTIVITY VS LAYOUTY

WTF IS R?

- » vždy při změně resources (layouts, ...) se automaticky generuje třída **R**, která se v kódu používá pro referenci na resources
- » např. při vytvoření nového layoutu ve složce res/layout se do **R** přidala datová složka **R.layout.{název layoutu}**
 - » tuto hodnotu přijímá **setContentView (int layoutResId)**

AKTIVITY VS LAYOUTY

WTF IS R?

- » např. při přiřazení identifikátoru `android:id="@+id/nazev"` prvku v layoutu se do R přidá `R.id.{nazev}`
 - » tuto hodnotu přijímá `findViewById(int)`
- » hodnoty z R jsou přístupné i z XML
 - » `R.id.{nazev}` → `@id/nazev`
 - » `R.layout.{nazev}` → `@layout/nazev`

AKTIVITY VS LAYOUTY

ZÍSKÁNÍ REFERENCÍ NA VIEW

- » po zavolání `Activity.setContentView(int)` můžeme získat reference vytvořených View pomocí metody `Activity.findViewById(int)`
- » metoda prostupuje layoutem, dokud nenajde **View** s daným identifikátorem
- » pokud ho najde, vrátí ho, jinak **null**
- » existují jiná/lepší řešení získání referencí, např. knihovna ButterKnife/Data binding/Kotlin

AKTIVITY VS LAYOUTY

ATRIBUTY VS METODY

- » většina atributů View z XML jsou po získání instance dostupná i z Javy, např.
 - » **android:text** → `TextView.setText(String)`
- » ne však všechny atributy, např.
 - » **android:textColor="#FFFFFF"**
 - ~~`TextView.setTextColor(String)`~~
- » nejsou pro to žádná pravidla, nutné se spolehnout na dokumentaci

PODPORA HW KONFIGURACÍ

RŮZNÉ LAYOUTY

- » co když potřebujeme zobrazit jiný layout pro zobrazení na výšku a na šířku?
- » jednotlivé adresáře ve složce **res/** mohou být rozděleny na adresáře dle parametrů, které jsou známy při běhu aplikace
- » např. můžeme vytvořit adresáře **layout-port** a **layout-land**, ve kterých jsou layouty rozdělené právě podle orientace

PODPORA HW KONFIGURACÍ

RŮZNÉ LAYOUTY

- » pro layouty
`res/layout-port/activity_main.xml`
a `res/layout-land/activity_main.xml`
je vytvořena v `R` pouze jedna hodnota
`R.layout.activity_main`
- » při běhu aplikace a volání `setContentView`
je vybrán layout, který lépe splňuje aktuální
konfiguraci (zde orientaci zařízení)

PODPORA HW KONFIGURACÍ

DALŠÍ KONFIGURACE

- » rozdělení dle
 - » hustoty obrazových bodů
 - » `layout-mdpi`, `layout-hdpi`, ...
 - » systémového jazyka
 - » `layout-cs`, `layout-en`
 - » a dalších
- » jsou možné kombinace
 - » např. `layout-cs-port-hdpi`
(pozor na pořadí!)

ÚKOL 3. SEMINÁŘE

PŘIDÁNÍ TODO POLOŽKY

- 1) Vytvořit novou aktivitu **InsertActivity** přístupnou přes nové tlačítko z **MainActivity**.
- 2) Do nové aktivity vložit nový layout **activity_insert** s formulářovými prvky pro **title** a **content** (label + edit) a tlačítkem.
- 3) Layout musí reagovat na orientaci zařízení, tj. na výšku bude label a edit pod sebou, na šířku vedle sebe.
- 4) Po kliku na tlačítko vypsát do konzole zapsané hodnoty.

OTÁZKY PTEJTE SE!

