

Курс:

Программирование под Android с нуля

Тренер:

Пономарев Алексей

oleksii_android@web-academy.com.ua



Лекция 15

Views, localization, orientation

Custom view

- **Разметка** – layout с разметкой для view. Разметки может не быть, если это расширение стандартных компонентов.
- **Дополнительные атрибуты** – дополнительные свойства для новой view, которые будут прописываться в разметке при добавлении этой view.
- **Код** – класс с реализацией.

RequiredEditText, атрибуты

```
//res/values/attrs.xml
```

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

```
<resources>
```

```
    <declare-styleable name="RequiredEditText">
```

```
        <attr name="required" format="boolean"/>
```

```
    </declare-styleable>
```

```
</resources>
```

RequiredEditText, код

```
public class RequiredEditText extends EditText {
    private boolean required = false;

    public RequiredEditText(Context context, AttributeSet attrs) {
        super(context, attrs);

        TypedArray a = //Получение атрибутов
            context.obtainStyledAttributes(attrs, R.styleable.RequiredEditText, 0, 0);
        this.required = a.getBoolean(R.styleable.RequiredEditText_required, false);

        a.recycle();
    }

    public boolean getRequired() {
        return this.required;
    }

    public boolean validate() {
        if (required && getText().toString().length() == 0) {
            setError("Field is required"); //Установка текста ошибки
            return false;
        }

        return true;
    }
}
```

RequiredEditText, подключение

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    //Подключение атрибутов
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ua.com.webacademy.lection12.RequiredEditText
        android:id="@+id/requiredEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        //Атрибут
        app:required="true"/>

</RelativeLayout>
```

RequiredEditText, использование

```
RequiredEditText requiredEditText =  
    (RequiredEditText) findViewById(R.id.requiredEditText);  
  
if(requiredEditText.validate()) {  
    Toast.makeText(this,  
        requiredEditText.getText(),  
        Toast.LENGTH_SHORT).show();  
}
```

MyButton, разметка

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="100dp"
    android:layout_height="80dp">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="Button"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:src="@mipmap/ic_launcher"
        android:scaleType="center"
        android:paddingBottom="20dp"/>

</RelativeLayout>
```

MyButton, атрибуты

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="RequiredEditText">
        <attr name="required" format="boolean"/>
    </declare-styleable>

    <declare-styleable name="MyButton">
        <attr name="text" format="string|reference"/>
        <attr name="src" format="reference"/>
    </declare-styleable>
</resources>
```


MyButton, код

```
public class MyButton extends RelativeLayout {
    private TextView textView;
    private ImageView imageView;

    private ClickInterface clickInterfaceListener;

    public MyButton(Context context) {
        super(context);
        init(context, null);
    }

    public MyButton(Context context, AttributeSet attrs) {
        super(context, attrs);
        init(context, attrs);
    }

    public void init(Context context, AttributeSet attrs) {
        //Инициализация
    }

    public void setOnClickListener(ClickInterface listener) {
        clickInterfaceListener = listener;
    }

    //Обработка нажатий
    public interface ClickInterface {
        public void onClick();
    }
}
```

MyButton, подключение

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    //Подключение атрибутов
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ua.com.webacademy.lection12.MyButton
        android:id="@+id/view4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        //Атрибуты
        app:src="@mipmap/ic_launcher"
        app:text="Save"/>
</RelativeLayout>
```

MyButton, использование

```
MyButton myButton = (MyButton) findViewById(R.id.view3);

myButton.setOnClickListener(new MyButton.ClickInterface() {
    @Override
    public void onClick() {

    }

});
```

StudentView, разметка

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

    <ua.com.webacademy.lection12.RequiredEditText
        android:id="@+id/firstName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="First name"           //Подсказки
        app:required="true"/>

    <ua.com.webacademy.lection12.RequiredEditText
        android:id="@+id/lastName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Last name"
        app:required="true"/>

    <ua.com.webacademy.lection12.RequiredEditText
        android:id="@+id/age"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:hint="Age"
        android:inputType="number"         //Тип ввода
        app:required="true"/>

</LinearLayout>
```

StudentView, код

```
public class StudentView extends LinearLayout {  
    ...  
    public StudentView(Context context) {  
        super(context);  
        init(context);  
    }  
  
    public StudentView(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        init(context);  
    }  
  
    public void init(Context context) {  
        //Инициализация  
    }  
  
    public void setStudent(Student student) {  
        //Установка студента  
    }  
  
    public Student getStudent() {  
        //Возврат студента  
    }  
  
    public boolean validate() {  
        //Проверка что все поля заполнены  
    }  
}
```

StudentView, подключение

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    tools:context=".MainActivity">

    <ua.com.webacademy.lection12.StudentView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/view6"/>

</RelativeLayout>
```

StudentView, использование

```
MyButton myButton = (MyButton) findViewById(R.id.view3);
MyButton myButton2 = (MyButton) findViewById(R.id.view4);
StudentView studentView = (StudentView) findViewById(R.id.view6);

myButton.setOnClickListener(new MyButton.ClickInterface() {
    @Override
    public void onClick() {
        Student student = new Student("Ivanov", "Ivan", 22);
        studentView.setStudent(student);
    }
});

myButton2.setOnClickListener(new MyButton.ClickInterface() {
    @Override
    public void onClick() {
        if (studentView.validate()) {
            Student student = studentView.getStudent();

            Toast.makeText(
                MainActivity.this,
                student.FirstName,
                Toast.LENGTH_SHORT).show();
        }
    }
});
```

Localization

res
 values
 strings.xml
 values-ru
 strings.xml
 values-de
 strings.xml

```
Locale localeRU = new Locale("ru");
Locale.setDefault(localeRU);

Configuration configRU = new Configuration();
configRU.locale = localeRU;

getResources().updateConfiguration(configRU,
    getResources().getDisplayMetrics());
```


Help

app

java

java code

res

resources

assets

files

```
WebView webViewHelp = (WebView) findViewById(R.id.helpView);  
webViewHelp.setWebViewClient(new WebViewClient());
```

```
WebSettings settings = webViewHelp.getSettings();  
settings.setDefaultTextEncodingName("utf-8");
```

```
webViewHelp.loadUrl("file:///android_asset/help/ru/index.html");
```

Orientation

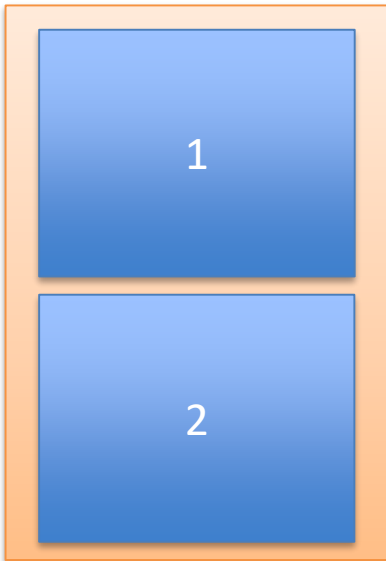
res

layout

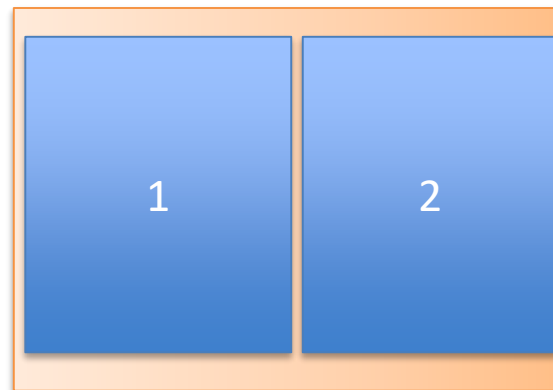
activity_main.xml

layout-land

activity_main.xml



Landscape layout



Localization

```
@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    savedInstanceState.putInt (STATE_LEVEL, mCurrentLevel);

    super.onSaveInstanceState (savedInstanceState);
}
```

Сохранение данных

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate (savedInstanceState);

    if (savedInstanceState != null) {
        mCurrentLevel = savedInstanceState.getInt (STATE_LEVEL);
    }
    ...
}
```

Получение данных здесь

```
public void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState (savedInstanceState);

    mCurrentLevel = savedInstanceState.getInt (STATE_LEVEL);
}
```

Или здесь

Спасибо за внимание

Тренер:

Пономарев Алексей

oleksii_android@web-academy.com.ua

Лекция 15

Views, localization, orientation

