## Layout’ы и View

### Цель работы

Познакомится с основными визуальными элементами приложения и приемами работы с ними

### Задания для выполнения

1. Внимательно прочтите и повторите в своем проекте методические указания.
2. Попробуйте креативно подойти к размещению и оформлению элементов. Экспериментируйте.

Решение заданий для выполнения совмещены с решениями дополнительных заданий.

### Контрольные вопросы

1. Какие основные элементы управления Вы изучили?

Button, EditText, ListView.

1. Как происходит группировка элементов управления на странице?

Группировка элементов происходит с помощью элементов Layout, например, ConstraintLayout, LinearLayout и т. д. Они задают порядок расположения внутренних элементов. Так же элементы могут быть сгруппированы в список с помощью элемента ListView.

1. Зачем разделять XML и .java файлы?

Файлы XML и .java разделяют, чтобы разделить бизнес-логику и статические данные: представления (Layout), ресурсы (строки, массивы, цвета и т. д.). Это дает несколько преимуществ. Во-первых, XML файлами могут пользоваться сразу несколько программ, и, если появится необходимость что-то исправить, то это нужно будет сделать только в одном месте — в XML. Во-вторых, это повышает читаемость. Было бы непросто читать код, в котором помимо бизнес-логики реализованы и шаблоны. Да и реализация шаблонов на языке Java, очевидно, была бы менее удобной, чем на XML.

1. Где в проекте размещаются строковые ресурсы?

Расположение файлов — res/values/filename.xml, в элементах <string>. filename — любое имя файла. Чаще всего это файл res/values/strings.xml

1. Что такое событийно-ориентированное программирование?

Парадигма программирования, в которой выполнение программы определяется событиями — действиями пользователя (клавиатура, мышь), сообщениями других программ и потоков, событиями операционной системы.

1. Какие события вы использовали в своем приложении?

Событие onClick у Button, onItemClick у AdapterView.

1. Какие события еще Вы знаете?

Событие создания onCreate у Activity, событие изменения прогресса OnSeekBarChange у ProgressBar, событие создания меню onCreateOptionsMenu у Activity, событие выбора элемента меню onOptionsItemSelected у Activity, событие получения результат от другой активности onActivityResult у Activity и др.

### Дополнительные задания

1. Реализуйте интерактивный список с автоматической сортировкой и удалением дубликатов элементов.
2. Реализуйте возможность удаления выделенного элемента списка.

**MainActivity.java**

package com.example.layoutandviews;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.util.SparseBooleanArray;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.Collections;  
  
public class MainActivity extends AppCompatActivity implements View.OnClickListener,  
 AdapterView.OnItemClickListener {  
  
 TextView mainTextView;  
 Button mainButton;  
 EditText mainEditText;  
 ListView mainListView;  
 ArrayAdapter mArrayAdapter;  
 ArrayList mNameList = new ArrayList();  
  
 Button ok\_btn, cnc\_btn;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mainTextView = findViewById(R.id.*main\_textview*);  
 mainButton = findViewById(R.id.*main\_button*);  
 mainEditText = findViewById(R.id.*main\_edittext*);  
 mainListView = findViewById(R.id.*main\_listview*);  
  
 mainTextView.setText("Set in Java!");  
  
 mainButton.setOnClickListener(this);  
  
 mArrayAdapter = new ArrayAdapter(this,  
 android.R.layout.*simple\_list\_item\_multiple\_choice*,  
 mNameList);  
 mainListView.setAdapter(mArrayAdapter);  
  
 mainListView.setOnItemClickListener(this);  
  
 ok\_btn = findViewById(R.id.*ok\_btn*);  
 cnc\_btn = findViewById(R.id.*cnc\_btn*);  
  
 View.OnClickListener oclBtn = new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 switch (v.getId()) {  
 case R.id.*ok\_btn*:  
 mainTextView.setText("Нажата кнопка ОК");  
 Toast.*makeText*(getApplicationContext(), "Нажата кнопка ОК", Toast.*LENGTH\_LONG*).show();  
 SparseBooleanArray selected = mainListView.getCheckedItemPositions();  
 for (int i = 0; i < mNameList.size(); i++)  
 if (selected.get(i)) {  
 mNameList.remove(i);  
 mArrayAdapter.notifyDataSetChanged();  
 break;  
 }  
 break;  
 case R.id.*cnc\_btn*:  
 mainTextView.setText("Нажата кнопка Cancel");  
 Toast.*makeText*(getApplicationContext(), "Нажата кнопка CANCEL", Toast.*LENGTH\_LONG*).show();  
 break;  
 }  
 mainListView.requestLayout();  
 mainListView.clearChoices();  
 mArrayAdapter.notifyDataSetChanged();  
 }  
 };  
  
 ok\_btn.setOnClickListener(oclBtn);  
 cnc\_btn.setOnClickListener(oclBtn);  
 }  
  
 @Override  
 public void onClick(View view) {  
 mainTextView.setText(mainEditText.getText().toString()  
 + " is learning Android development!");  
 String item = mainEditText.getText().toString();  
 if (!mNameList.contains(item)) {  
 mNameList.add(item);  
 Collections.*sort*(mNameList);  
 mArrayAdapter.notifyDataSetChanged();  
 }  
 }  
  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {  
 mainTextView.setText(mNameList.get(position).toString()  
 + " is learning Android development!");  
 }  
}

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 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:orientation="vertical"  
 android:layout\_margin="20dp"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/main\_textview"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/greetings"  
 android:textSize="24dp" />  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/main\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="@string/button" />  
  
 <EditText  
 android:id="@+id/main\_edittext"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:hint="@string/hint" />  
  
 </LinearLayout>  
  
 <ListView  
 android:choiceMode="multipleChoice"  
 android:id="@+id/main\_listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="400dp"  
 android:layout\_marginTop="20dp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/ok\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="OK" />  
  
 <Button  
 android:id="@+id/cnc\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="cancel" />  
 </LinearLayout>  
  
  
  
</LinearLayout>

**strings.xml**

<resources>  
 <string name="app\_name">HelloAndroid</string>  
 <string name="greetings">Hello from XML</string>  
 <string name="button">Update The TextView</string>  
 <string name="hint">A name</string>  
</resources>











