CONTENTS

Android Project Structure

Application components

Create activity

Listview
Mobile Programming

Android

Windows Phone

Fire Phone

Tizen
Architecture
Android Project Structure
App source structure

- Manifests
  - Declare android components
  - HW/SW features (camera...)
  - Permissions

- Java
  - Main java source codes

- Res
  - Resources: picture, graphic, sound, style, string, text...
“Gradle”
When we use it?

• Kind of build configure file

• Build.gradle(Project)
  – Including Jar or other java library
  – Other dependencies

• Build.gradle(Module)
  – SDK level
  – Including additional android library

• gradle.properties
  – Build options : JVM memory size...
Android application components
Application components

• **Essential building block**
  – Define the processes of an application.

• **Activity**
  – Most basic component in application.
  – One *user interface screen*.
  – Each activities are independence, but make user experiment together.

• **Service**
  – Executed in *background*.
  – Long time executed process & remote process.

• **Broadcast receivers**
  – Listen *broadcast event* in system level.
  – For communicating with other components.

• **Content provider**
  – Manage and share *in-app data*. 
Activity

- **Activity and screen**
  - One activities express one user interface screen.
  - Screen can be designed in layout.

- **Activity is an entry point**
  - As a main function in C, one of activity in application can be an entry point of process.
  - Under various environment, such entry point activity can be changed.

- **Activity stack**
  - When a new activity is executed, previous activity is saved in back stack.
  - When user quit current activity, top activity in back stack is popped and executed.
  - Activity life cycle.
Activity Life cycle

Activity launched
- onCreate()
- onStart()
- onResume()

Activity running
- onRestart()
- onPause()
- onStop()
- onDestroy()

App process killed
- User navigates to the activity
- Another activity comes into the foreground
- User returns to the activity

Apps with higher priority need memory
- The activity is no longer visible
- User navigates to the activity
- The activity is finishing or being destroyed by the system
- Activity shut down
Create activity
Design layout
Constraint Layout

- **Constraint**
  - Set square space

- **Bias**
  - Relative position in constraint square

- **Margin**
  - Just margin
Constraint Layout

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

    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintVertical_bias="0.653"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintHorizontal_bias="0.26" />
```

- **Constraint**
  - Set square space

- **Bias**
  - Relative position in constraint square

- **Margin**
  - Just margin
Create activity

• There is no main function. (*Event driven process*)

• When an activity is run, ‘onCreate()’ function is executed.

• `setContentView()` function appoint layout of this activity. And it should be in `onCreate()` function

• According to activity life cycle, callback function is exist at each steps.

• Let’s run!
Create activity
Modify layout file
Connect activity with Intent

```java
package com.example.sky.myapplication;

import ...

public class MainActivity extends AppCompatActivity {
    private Context mContext = this;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btn = (Button)findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intentToActivity2 = new Intent(mContext, Main2Activity.class);
                startActivity(intentToActivity2);
            }
        });
    }
}
```
Activity back stack

• When a new activity called, previous activity goes to back stack.

• Various methods are exist in activity stack policy. That policies can be set on intent. `addFlags()`

ListView – custom item
Coding process

• Design listview in layout file.

• Make item class which store image and name information. And design it.
  – Save image file in drawable.
  – Design item layout in layout.

• Make custom listview adapter.
  – Adapter is a kind of data manager of listview.
  – Define “how to show data”.
  – Normal adapter just show string, but we also want to show image.

• Link listview and adapter in activity.

• Set item click event listeners.
  – Open new activity.
  – Get data from previous activity.
<xml version="1.0" encoding="utf-8">
    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="com.example.sky.mypatitude.MainActivity">
    <ListView
        android:layout_width="368dp"
        android:layout_height="495dp"
        android:layout_marginLeft="8dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_marginBottom="8dp"/>
</android.support.constraint.ConstraintLayout>
Listview

```java
package com.example.sky.myapplication;

/**
 * Created by sky on 2017-04-12.
 */

public class MusicItem {
    private int icon;
    private String name;

    public MusicItem(int icon, String name){
        this.icon = icon;
        this.name = name;
    }

    public int getIcon() {
        return icon;
    }

    public String getName() {
        return name;
    }
}
```
Listview

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@android:drawable/screen_background_light_transparent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="music name" />

</LinearLayout>
```
Listview
public class MusicListAdapter extends BaseAdapter {
    private LayoutInflater inflater;
    private ArrayList<MusicItem> musics;

    public MusicListAdapter(Context context, ArrayList<MusicItem> musics) {
        inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        this.musics = musics;
    }

    @Override
    public int getCount() {
        return musics.size();
    }

    @Override
    public MusicItem getItem(int position) {
        return musics.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        if (convertView == null) {
            convertView = inflater.inflate(R.layout.music_item, parent, false);
        }

        MusicItem music = musics.get(position);

        ImageView imageView = (ImageView) convertView.findViewById(R.id.imageView);
        TextView textView = (TextView) convertView.findViewById(R.id.textView);

        imageView.setImageResource(music.getIcon());
        textView.setText(music.getName());

        return convertView;
    }
}
Listview

```java
package com.example.sky.myapplication;

import ...

public class MainActivity extends AppCompatActivity {
    private Context mContext = this;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ArrayList<MusicItem> musics = new ArrayList<>();
        musics.add(new MusicItem(R.drawable.music0, getString(R.string.music0)));
        musics.add(new MusicItem(R.drawable.music1, getString(R.string.music1)));
        musics.add(new MusicItem(R.drawable.music2, getString(R.string.music2)));
        musics.add(new MusicItem(R.drawable.music3, getString(R.string.music3)));
        musics.add(new MusicItem(R.drawable.music4, getString(R.string.music4)));
        musics.add(new MusicItem(R.drawable.music5, getString(R.string.music5)));
        musics.add(new MusicItem(R.drawable.music6, getString(R.string.music6)));

        ListView listView = (ListView)findViewById(R.id.listView);
        MusicListviewAdapter mAdapter = new MusicListviewAdapter(mContext, musics);
        listView.setAdapter(mAdapter);
    }
```
Listview
THANK YOU!!!