

Week06.  
오디오 재생



안드로이드 앱

# 개발환경 구축 절차

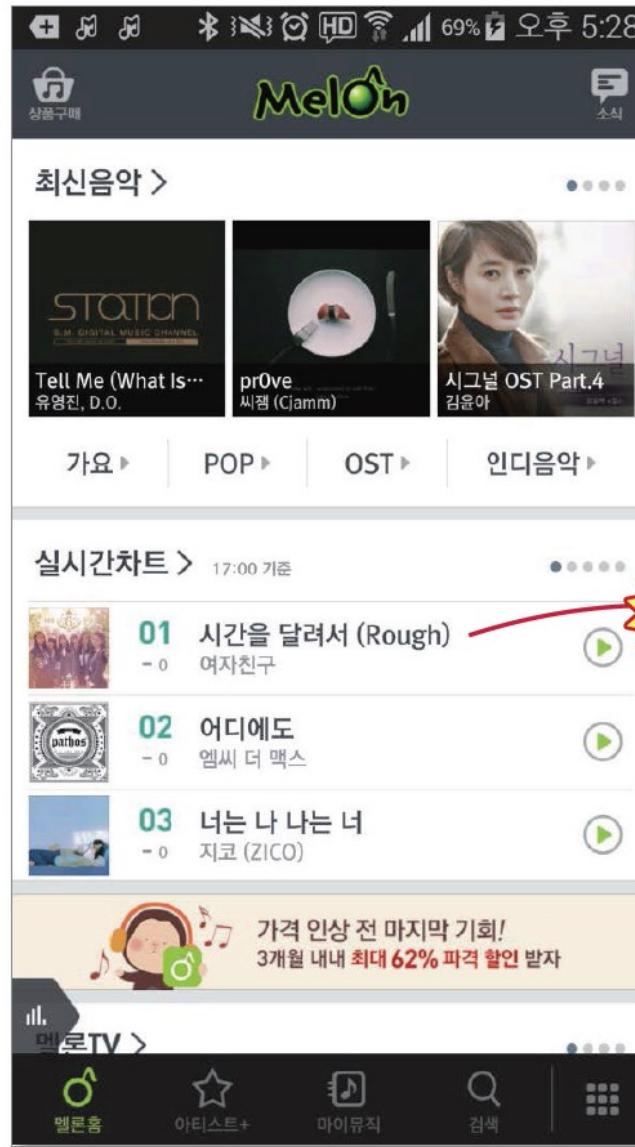
2

주 차	수업 내용
1	수업 소개
2	개발 환경 구축과 맛보기 프로젝트
3	텍스트 출력과 레이아웃
4	이미지의 출력
5	이벤트 처리와 액티비티 간 이동
6	<b>오디오 재생</b>
7	비디오 재생
8	<b>중간고사</b>
9	애니메이션
10	사물인터넷과 센서 – 터치 센서, 모션 센서
11	사물인터넷과 센서 – 위치 센서, 환경 센서
12	NFC 활용
13	공공 DB 오픈 API 활용
14	구글 맵과 위치 추적
15	<b>기말 고사</b>



[http://thinkdata.co.kr/wordpress/?page\\_id=29](http://thinkdata.co.kr/wordpress/?page_id=29)

# 오디오 출력 앱의 예

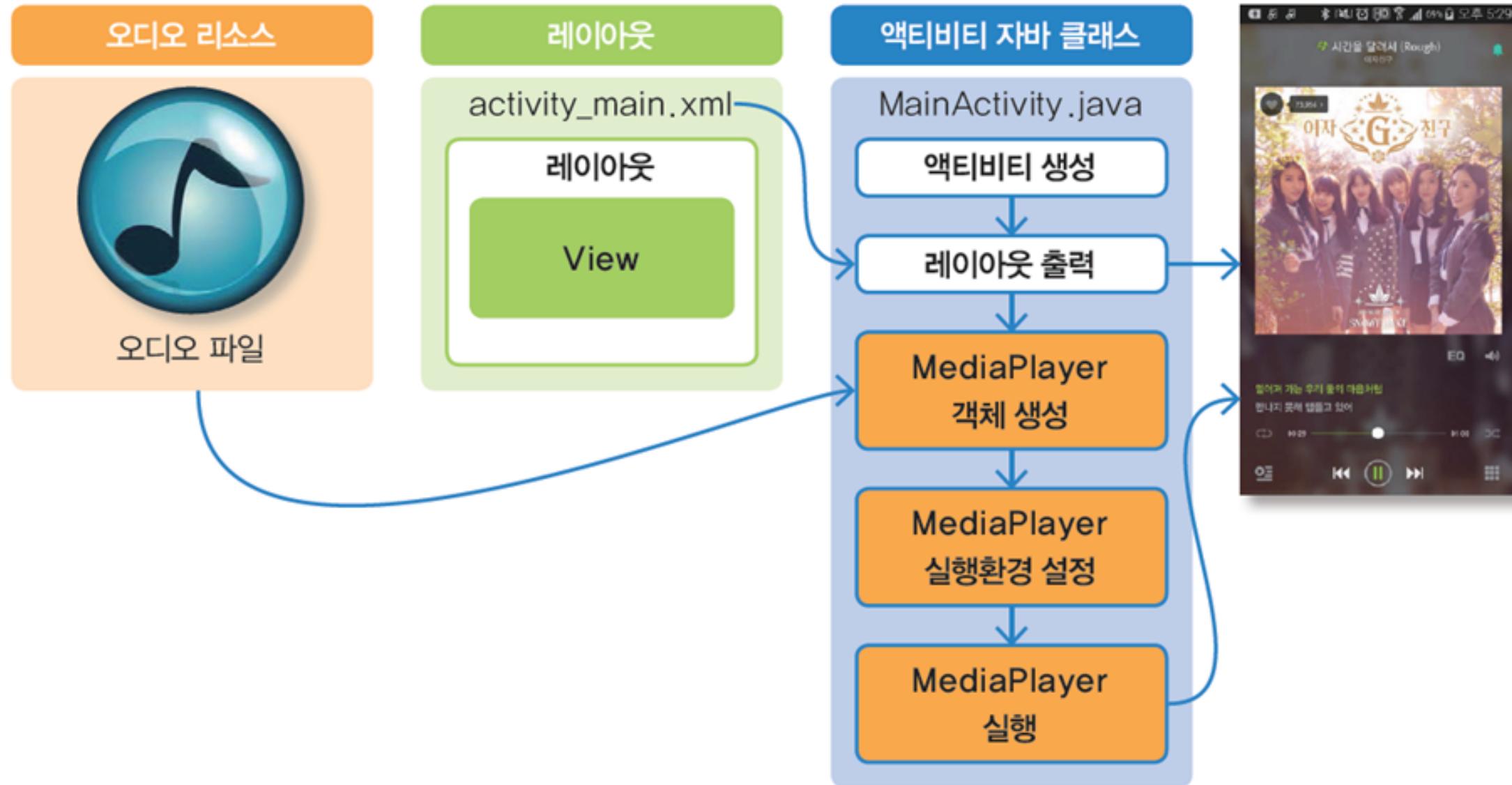


(a) 초기 화면



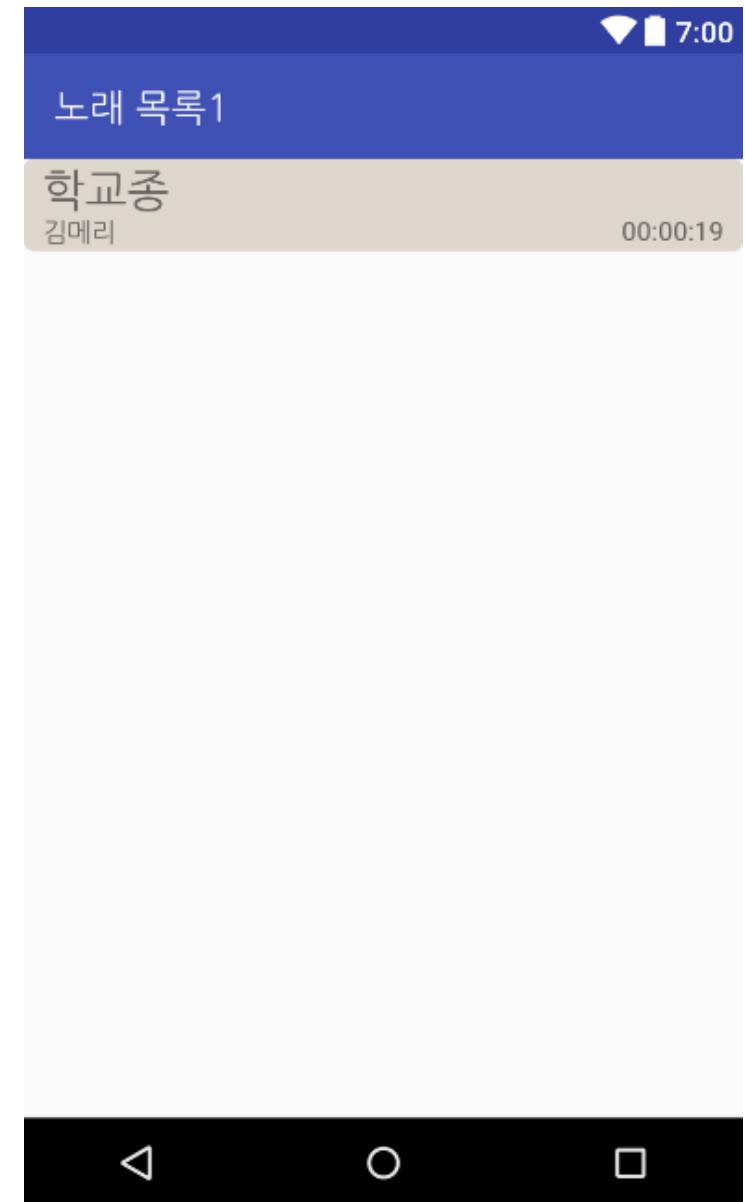
(b) 노래 재생

# 오디오 재생 원리



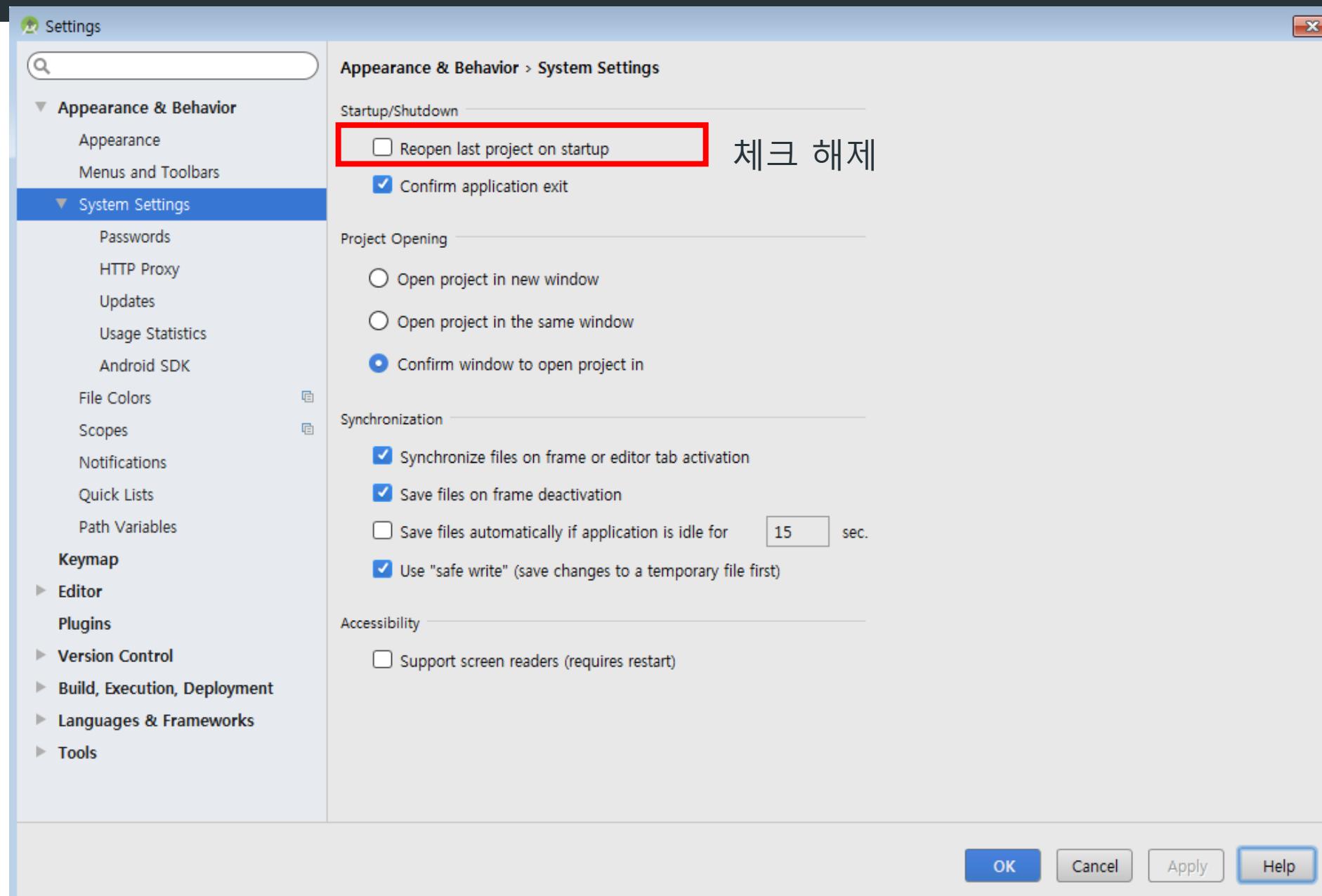
# Step 0. 프로젝트 개요

6



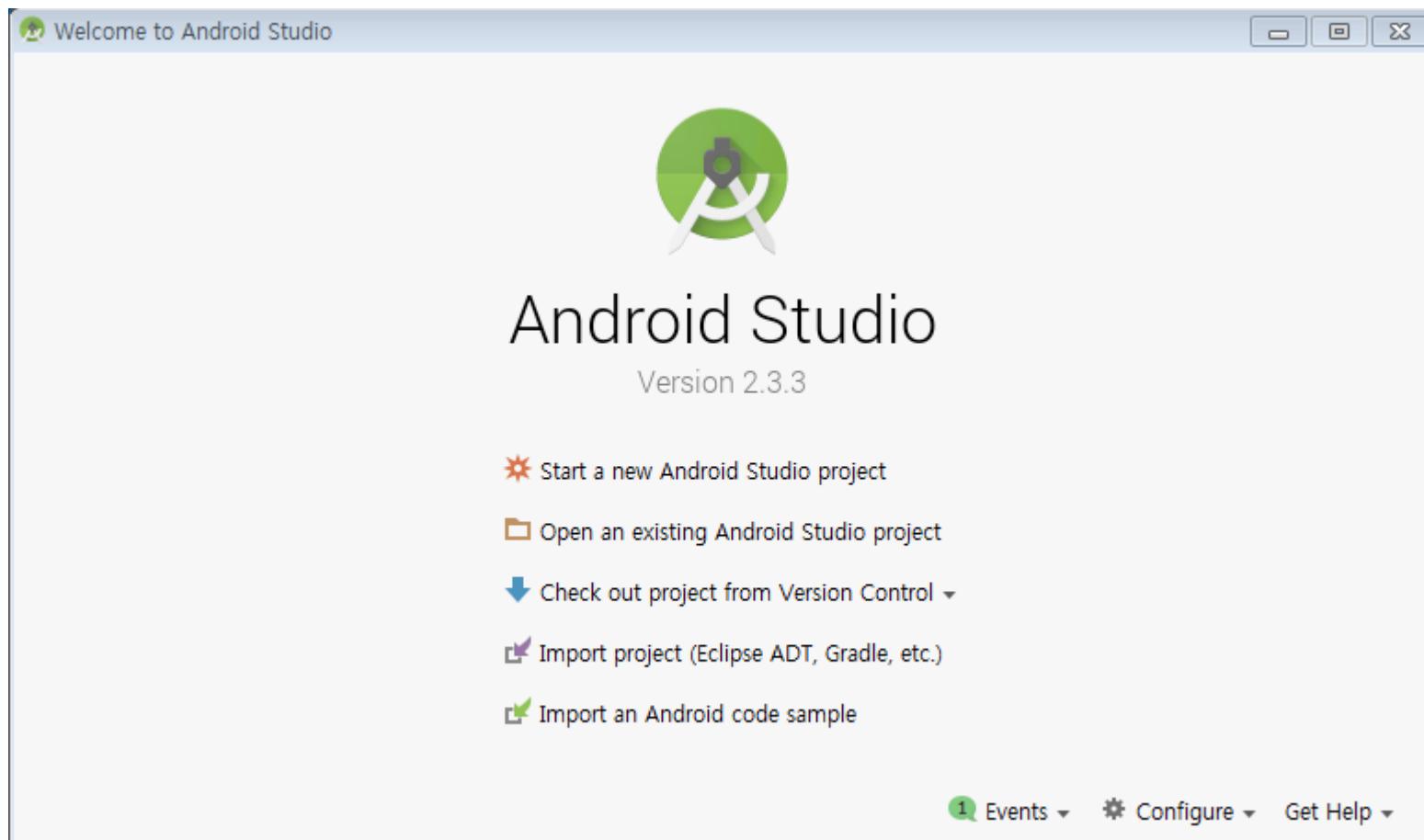
# 초기 화면 변경-유저가 프로젝트 선택

7



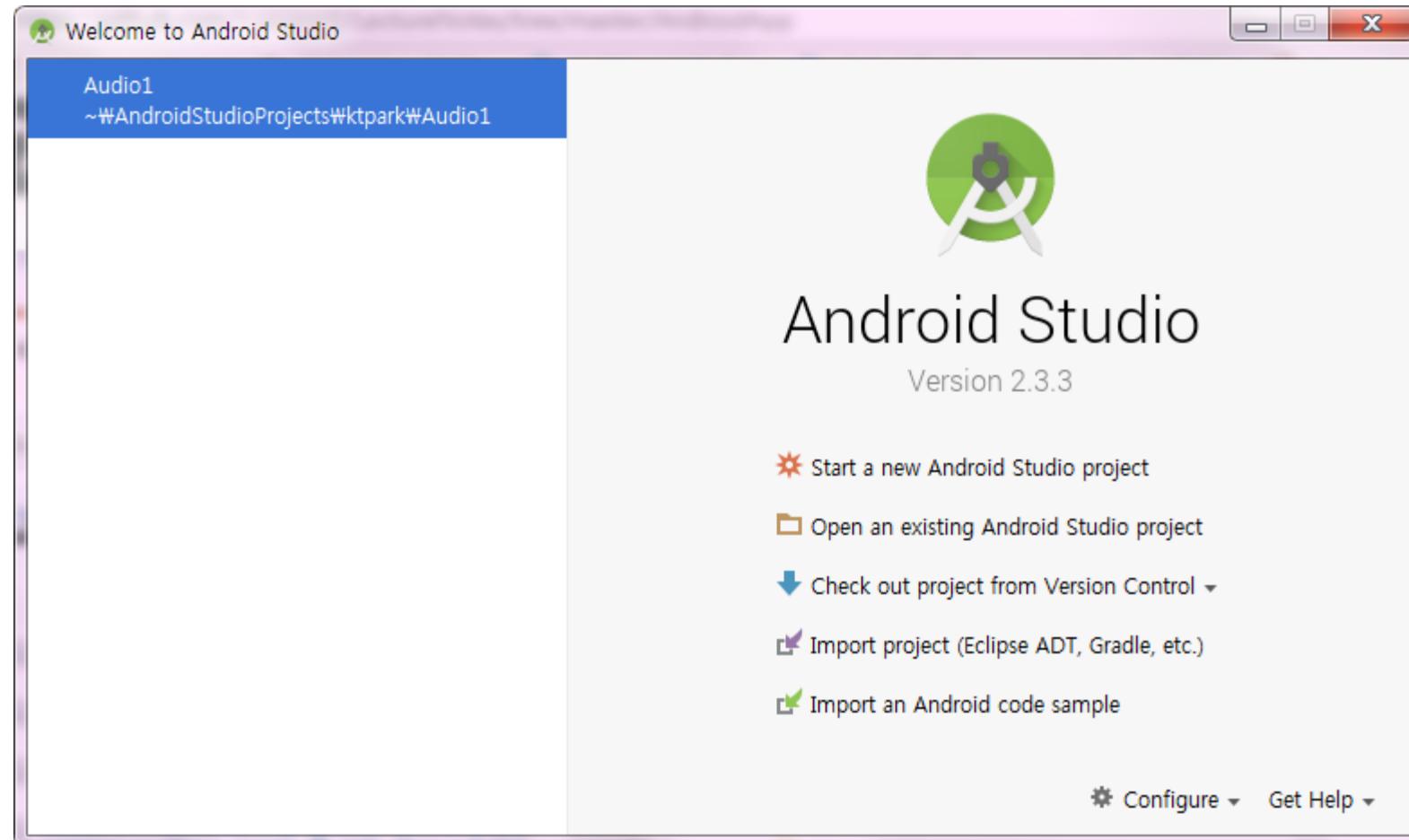
# Start a new Android Studio project-type1

8



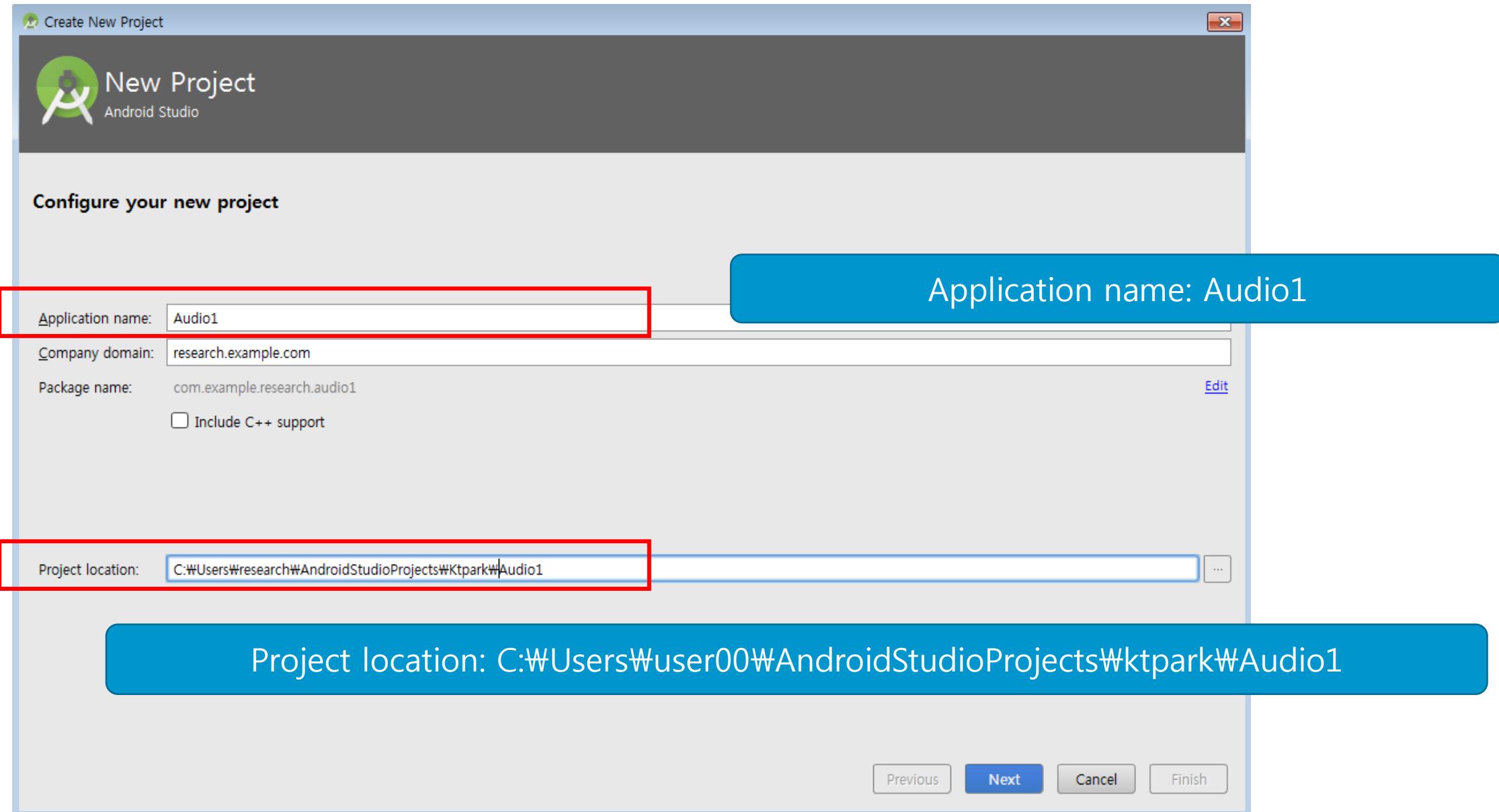
# Start a new Android Studio project-type2

9



# Configure your new project

10



# Configure your new project

11

Application name:

Company domain:

Package name: com.example.research.audio1 [Edit](#)

Include C++ support

Project location:  [...](#)

# Step 1. 프로젝트 생성

12

절차	내용
① 프로젝트 시작	메뉴에서 ‘File → New Project’ 클릭
② 프로젝트 구성	Application Name: <b>Audio1</b> Company Domain: 사용자계정.example.com(디폴트 사용) <b>Project location:</b> ~\user00\AndroidStudioProjects\ktpark\Audio1
③ 제품 형태	Phone and Tablet(사용할 안드로이드 버전 지정: <b>Android 7.0 Nougat</b> )
④ 액티비티 유형	Empty Activity
⑤ 파일 옵션	Activity Name: <b>MainActivity</b> (디폴트 사용) Layout Name: <b>activity_main</b> (디폴트 사용)

# Step 2. 파일 편집

13

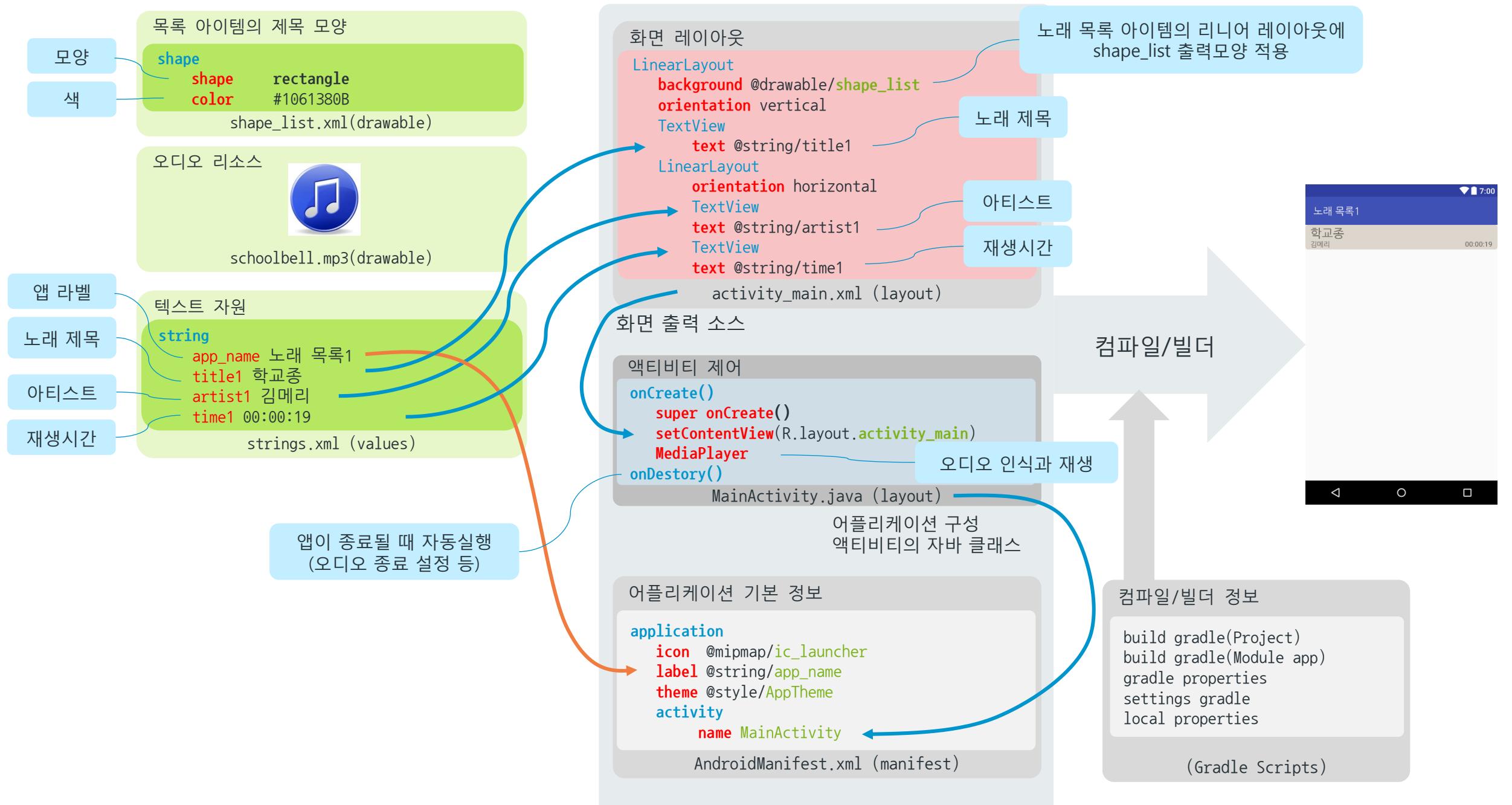
모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.kyungtae.audio1	MainActivity.java	<ul style="list-style-type: none"> <li>• 오디오 목록 출력과 오디오 자동 재생</li> </ul>
res	drawable	shape_list.xml	<ul style="list-style-type: none"> <li>• 목록 아이템에 대한 출력 스타일 설계 (테두리, 패딩, 모서리)</li> </ul> 
	layout	activity_main.xml	<ul style="list-style-type: none"> <li>• 노래 목록의 화면 배치</li> <li>• 목록 아이템에 출력 모양 적용 (shape_list.xml)</li> </ul>
	mipmap	ic_launcher.png	
	raw	schoolbell.mp3	<ul style="list-style-type: none"> <li>• 노래 오디오 파일</li> </ul>
	values	colors.xml	
		dimens.xml	
		strings.xml	<ul style="list-style-type: none"> <li>• 어플리케이션 라벨("노래 목록1")</li> <li>• 노래에 대한 제목, 작사/작곡자, 재생 시간에 대한 텍스트 리소스 정의</li> </ul>
		styles.xml	

# Step 2.1 오디오 파일 복사

14

- res 폴더에 있는 **raw** 폴더에 schoolbell.mp3 파일 저장

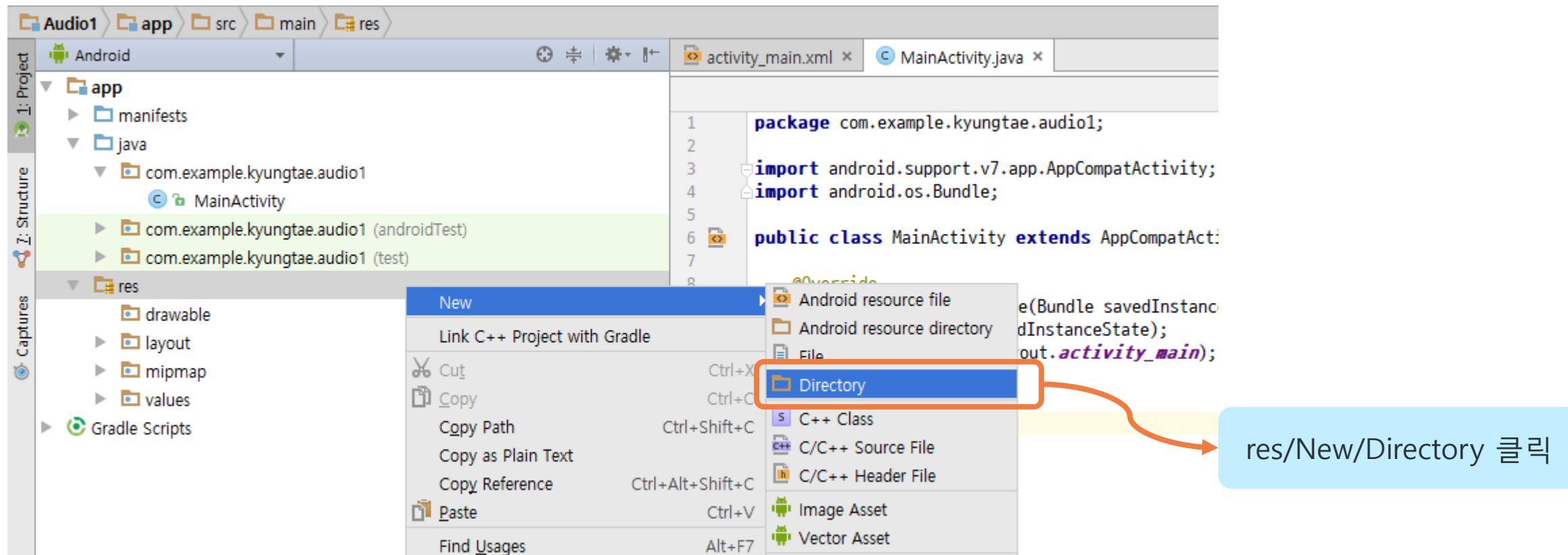
모듈	폴더	소스 파일	내용
res	raw	schoolbell.mp3	 학교종 노래



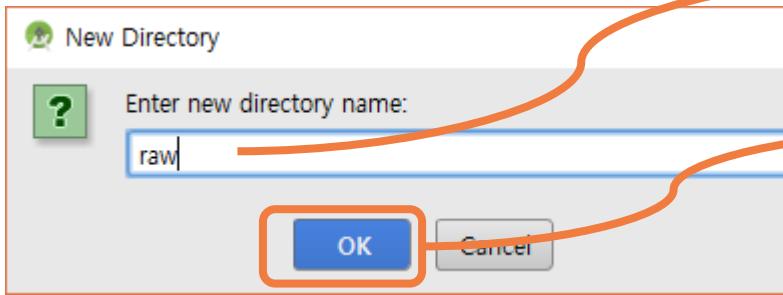


# drawable/raw 폴더에 오디오 파일 추가하기

- app → res → New → Directory 클릭



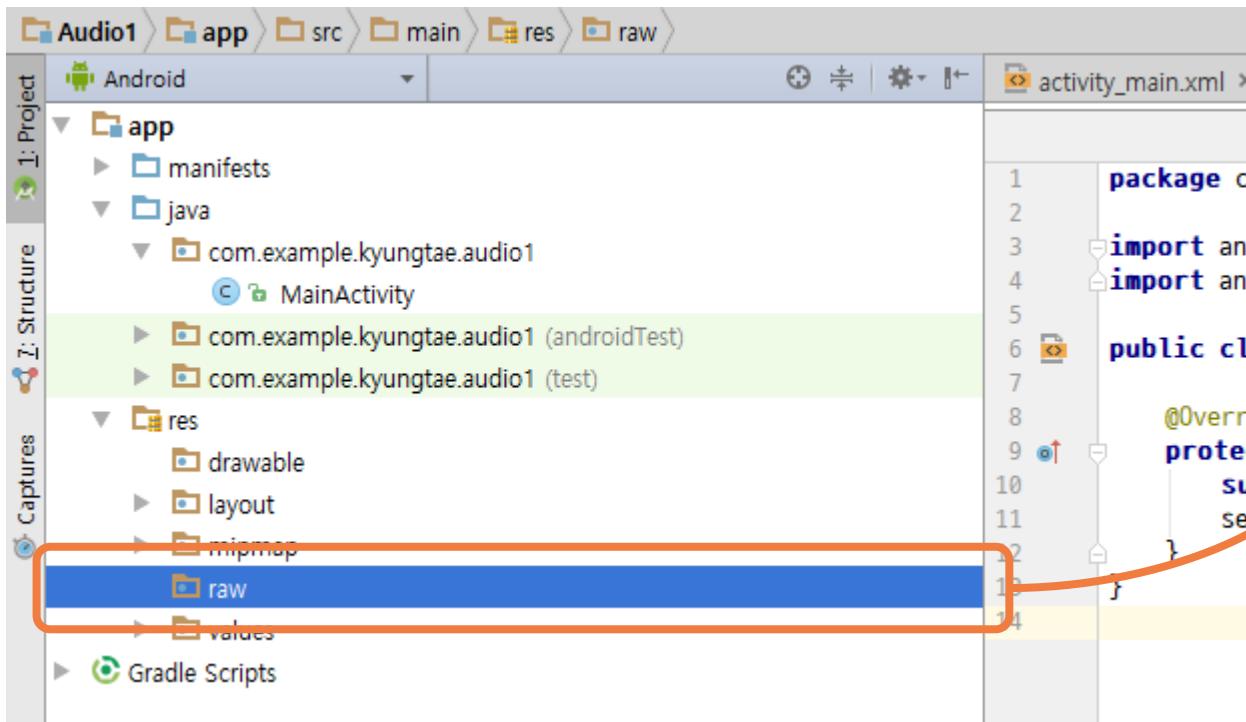
## • 폴더 이름 작성



폴더 이름: raw

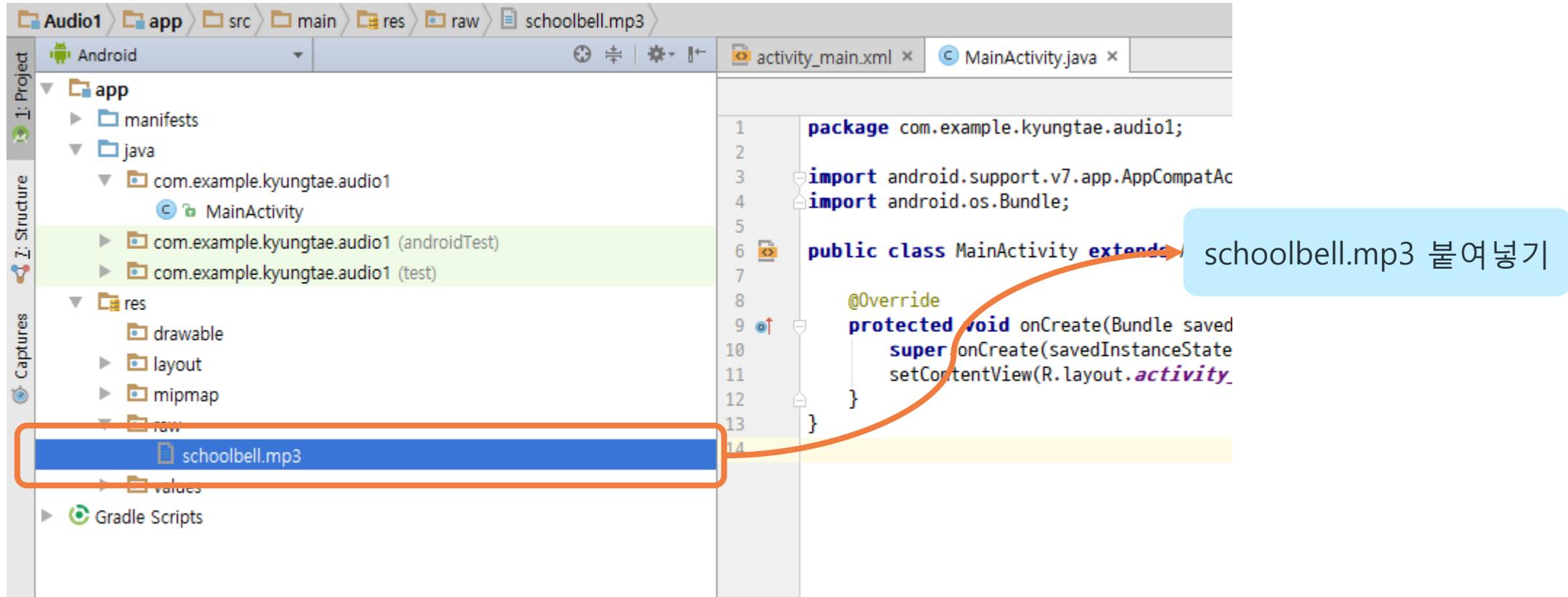
OK 클릭

## • 실행 결과



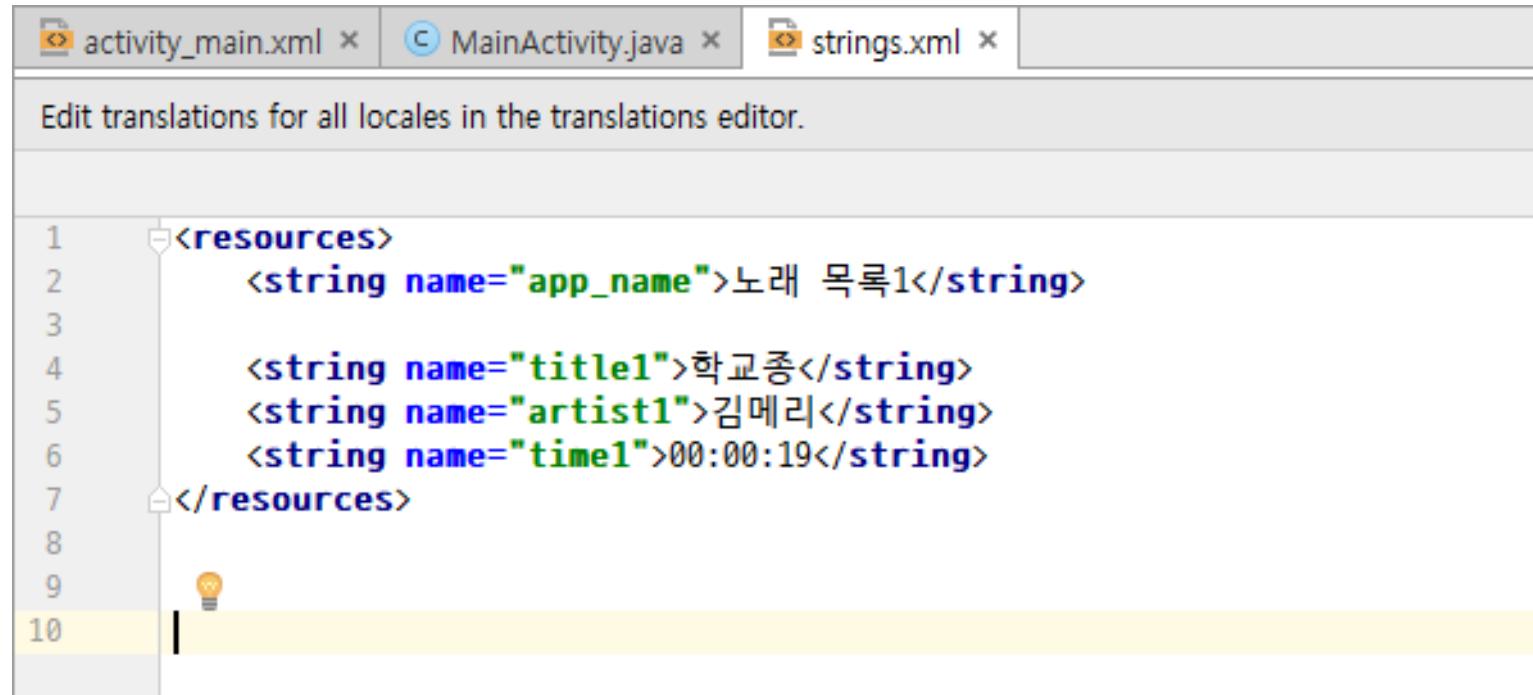
raw 폴더 생성됨

## • 오디오 파일(schoolbell.mp3) 복사하기



# Step 2.2 텍스트 자원의 편집

- strings.xml



The screenshot shows the Android Studio interface with the tabs "activity\_main.xml", "MainActivity.java", and "strings.xml" visible at the top. The "strings.xml" tab is active. The editor displays the following XML code:

```
1 <resources>
2     <string name="app_name">노래 목록1</string>
3
4     <string name="title1">학교종</string>
5     <string name="artist1">김메리</string>
6     <string name="time1">00:00:19</string>
7 </resources>
8
9
10 |
```

The code defines three string resources: "app\_name" with the value "노래 목록1", "title1" with the value "학교종", and "artist1" with the value "김메리". A fourth string resource, "time1", is also defined with the value "00:00:19". The XML structure is shown with indentation and line numbers. A yellow lightbulb icon is located near the bottom left of the code area.

# Step 2.3 Drawable Resource 추가 및 편집

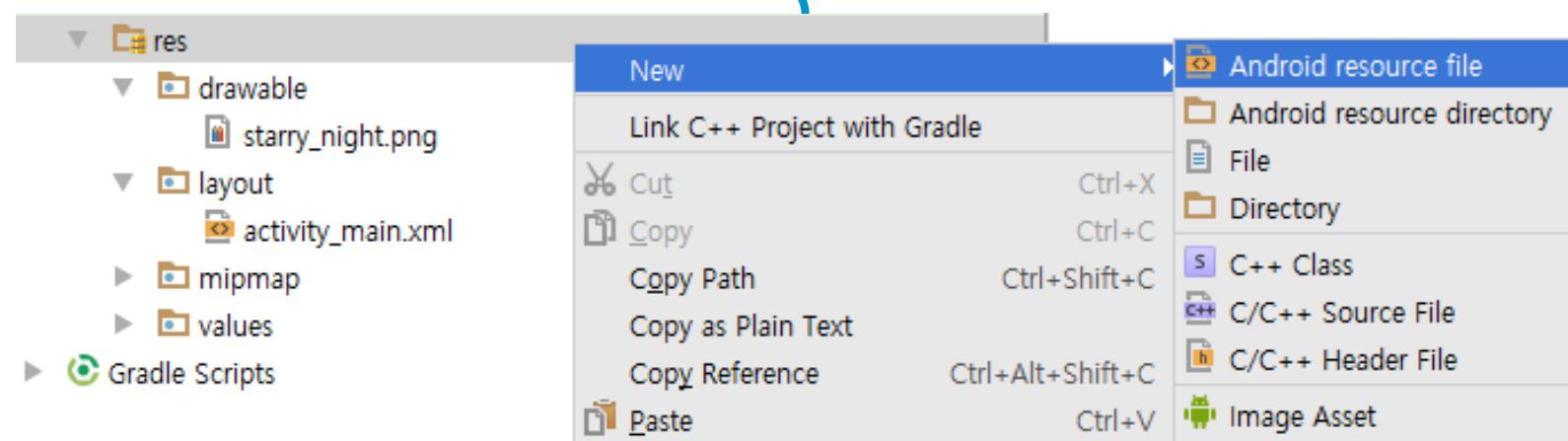
21

- **shape\_list.xml** 생성(res/drawable 폴더)

- drawable resource를 이용한 그림 출력
- 노래 제목(title)에 대한 출력 모양을 지정
- **drawable 폴더에는 화면에 그릴 수 있는 요소(도형)를 XML로 정의**
- **android:shape** 속성을 사용하여 다른 XML 리소스에 적용할 수 있는 그래픽에 대한 일반적인 개념



XML 파일 생성



- Set New Resource File

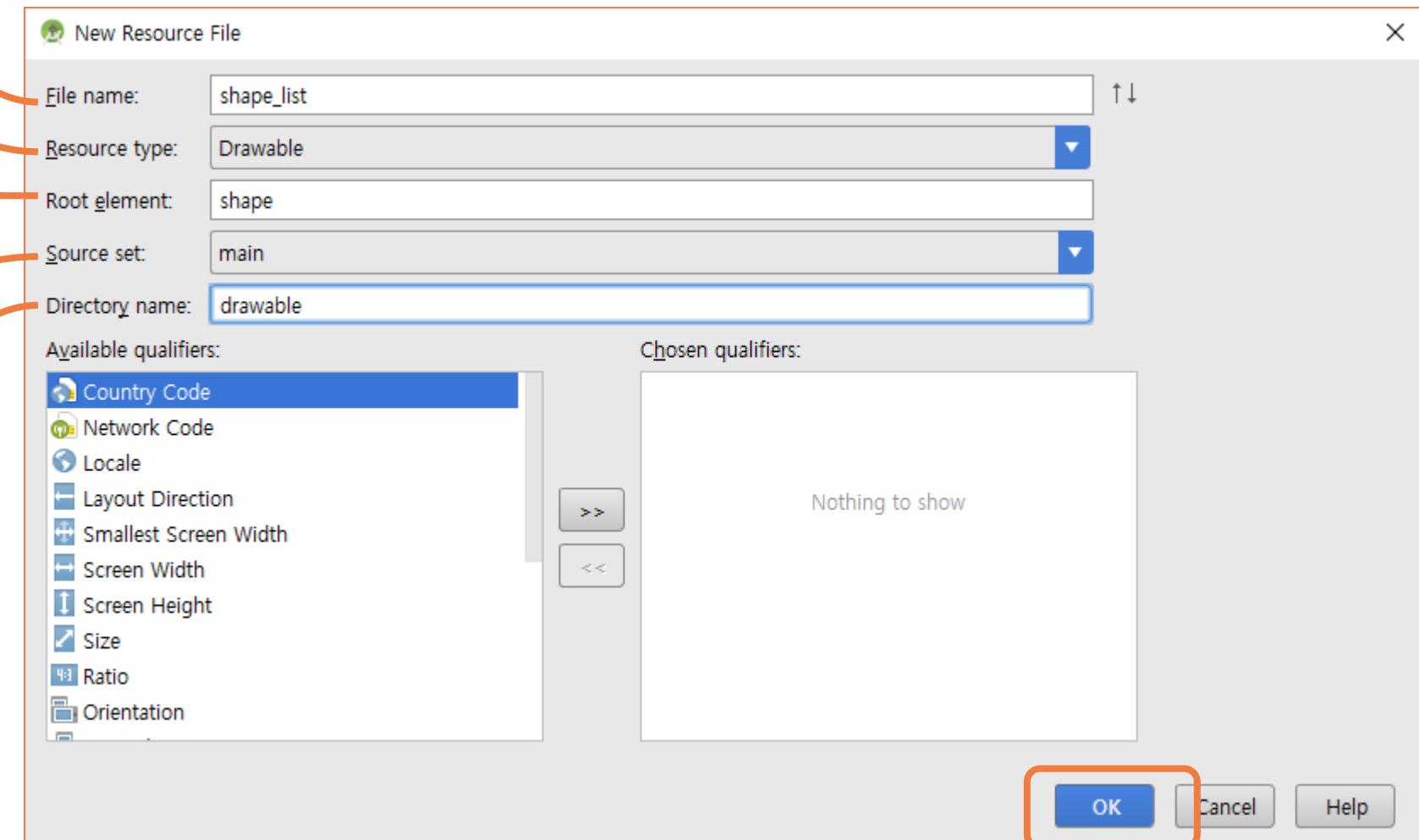
File name: shape\_list

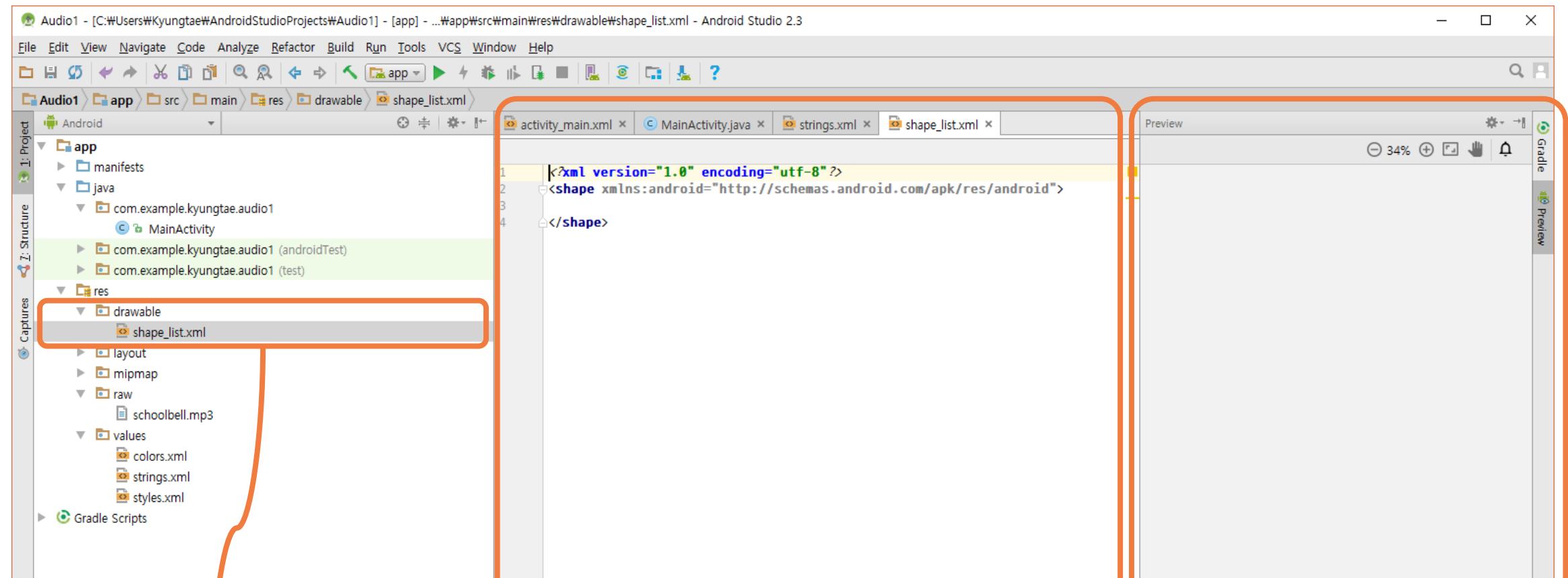
Resource type: Drawable

Root element: shape

Source set: main

Directory name: drawable





shape\_list.xml 파일

shape\_list.xml 파일의  
텍스트 코딩 영역

shape\_list 파일에 의한  
shape 미리보기 영역

## • shape\_list.xml 소스

The screenshot shows the Android Studio interface with the `shape_list.xml` file open. The code defines a rectangle shape with a light brown fill, 2dp top/bottom padding, 10dp left/right padding, and 5dp rounded corners.

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">

    <solid android:color="#3061380B"/>

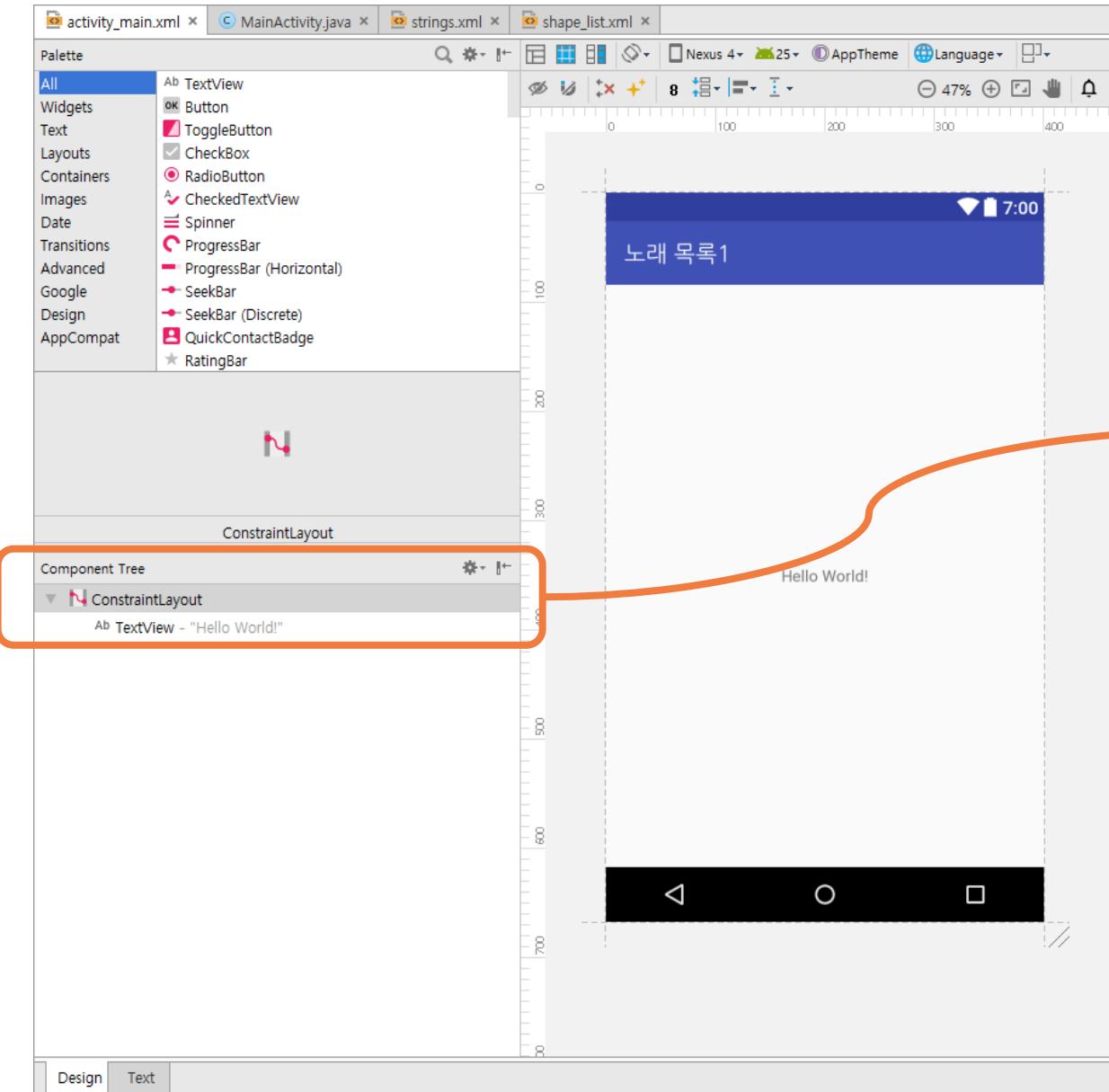
    <padding
        android:top="2dp"
        android:bottom="2dp"
        android:left="10dp"
        android:right="10dp">
    </padding>

    <corners android:radius="5dp"></corners>
</shape>
```

Annotations from the image:

- 출력모양을 사각형으로 지정
- 출력모양을 내부의 색
- 내부 패딩 정보
- 출력모양 모서리를 둥근 모양  
으로 지정(반지름은 5dp)

# 2.4 화면 설계



ConstraintLayout →  
LinearLayout으로 변경

Text 에디터에서 수정

activity\_main.xml x MainActivity.java x strings.xml x shape\_list.xml x

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.example.kyungtae.audio1.MainActivity">
8
9     <TextView
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="Hello World!"
13        app:layout_constraintBottom_toBottomOf="parent"
14        app:layout_constraintLeft_toLeftOf="parent"
15        app:layout_constraintRight_toRightOf="parent"
16        app:layout_constraintTop_toTopOf="parent" />
17
18 </android.support.constraint.ConstraintLayout>
19

```

activity\_main.xml x MainActivity.java x strings.xml x shape\_list.xml x

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.example.kyungtae.audio1.MainActivity">
8
9     <TextView
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="Hello World!"
13        app:layout_constraintBottom_toBottomOf="parent"
14        app:layout_constraintLeft_toLeftOf="parent"
15        app:layout_constraintRight_toRightOf="parent"
16        app:layout_constraintTop_toTopOf="parent" />
17
18 </LinearLayout>
19

```

삭제

activity\_main.xml x MainActivity.java x

Preview

Palette

Nexus 4 24 AppTheme Language

53% 0 100 200 300 400

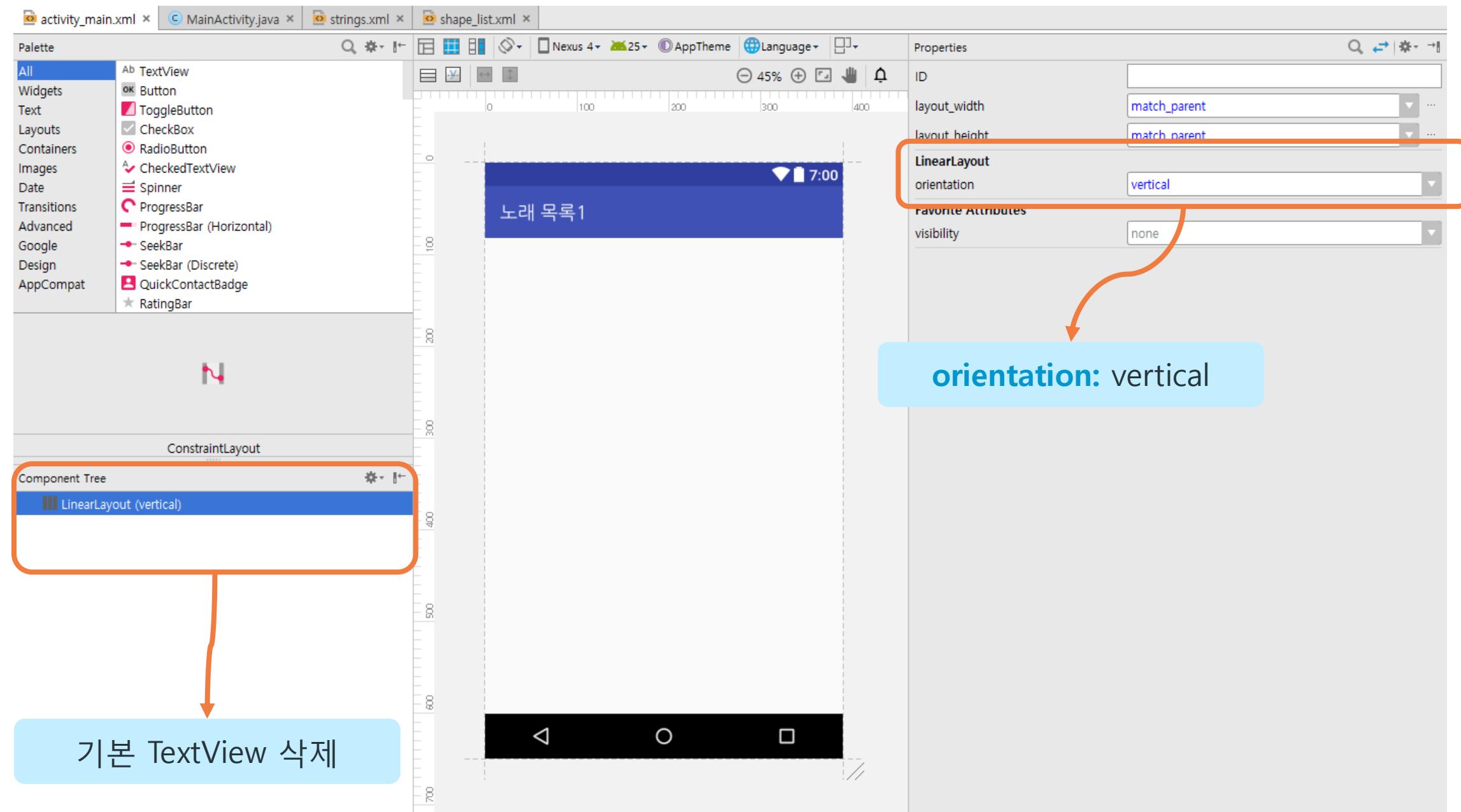
0 100 200 300 400 500 600

노래 목록1 7:00

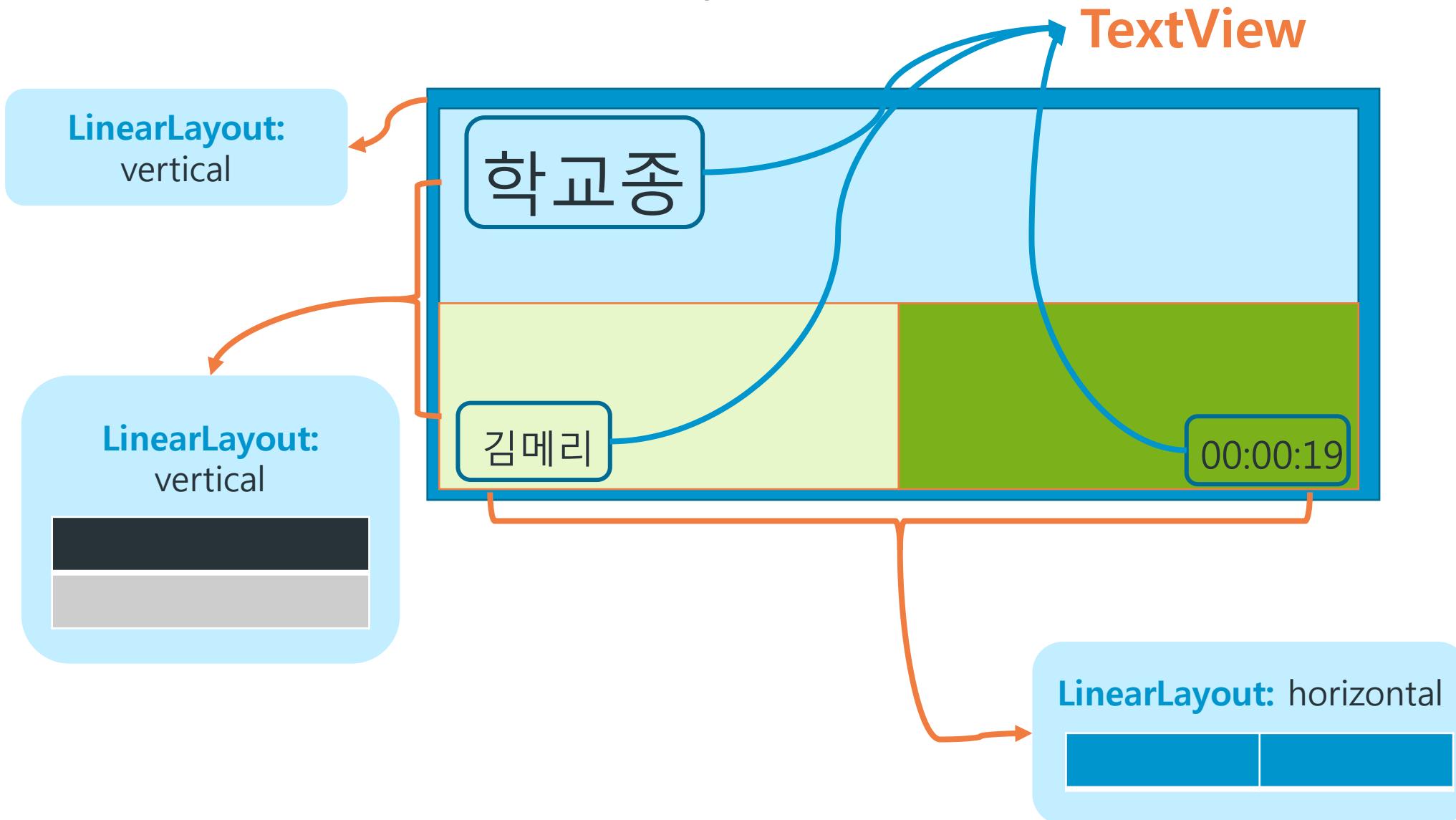
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    tools:context="com.example.research.audio1.MainActivity">  
  
</LinearLayout>

# Layout 변경 결과 화면

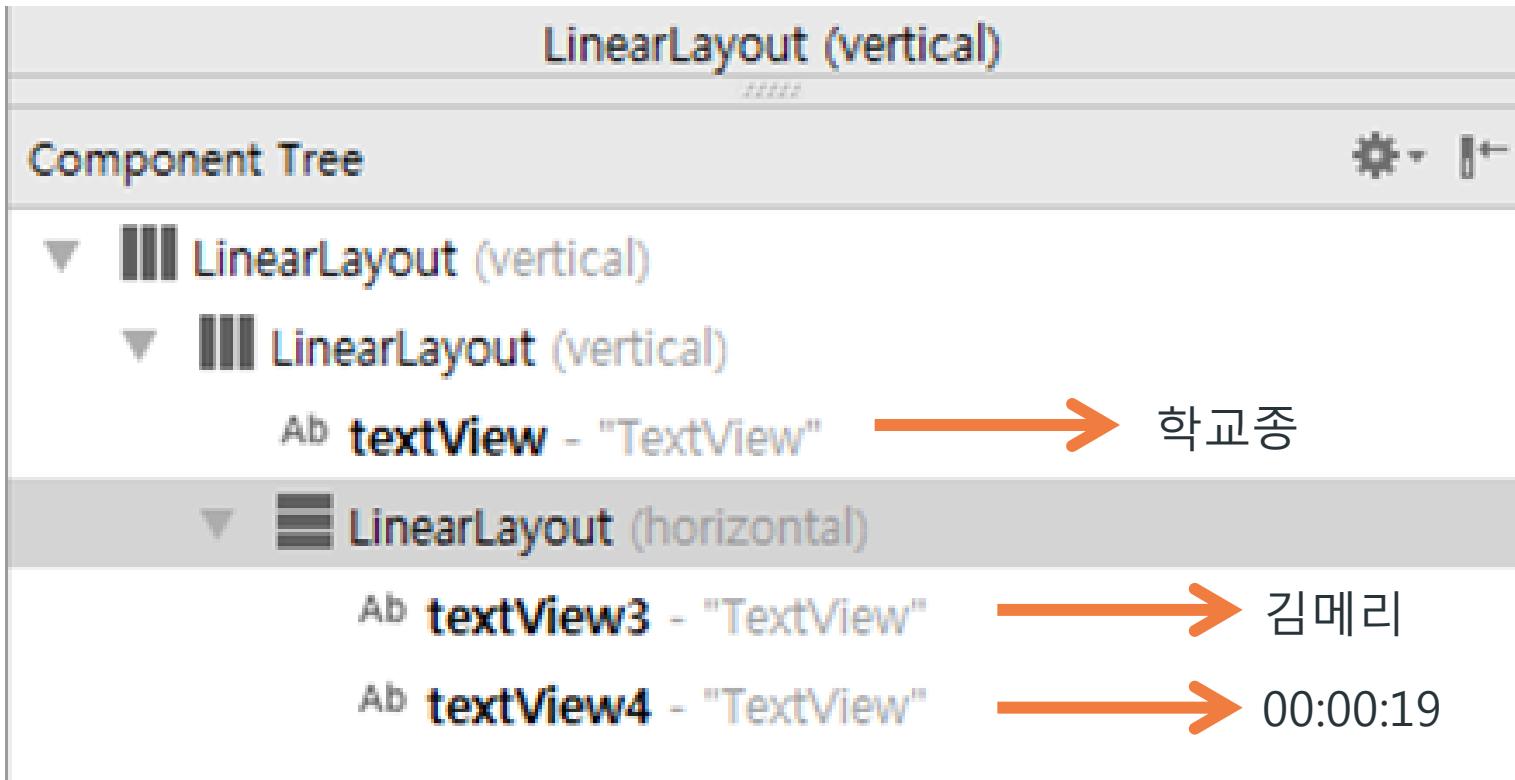
# • Layout 초기화 설정 - LinearLayout중 vertical 방향



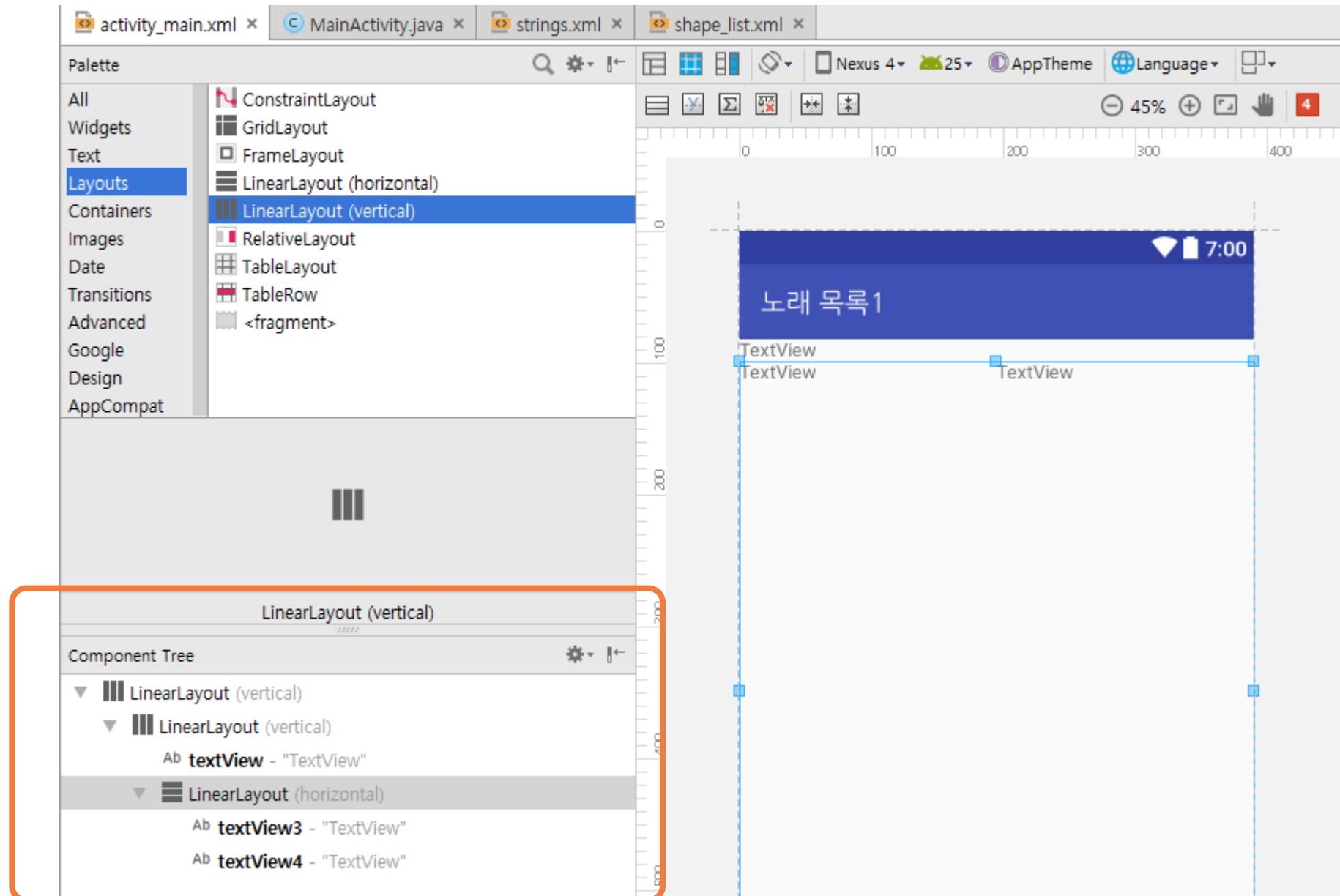
- 노래 목록 표시를 위한 Layout 구조

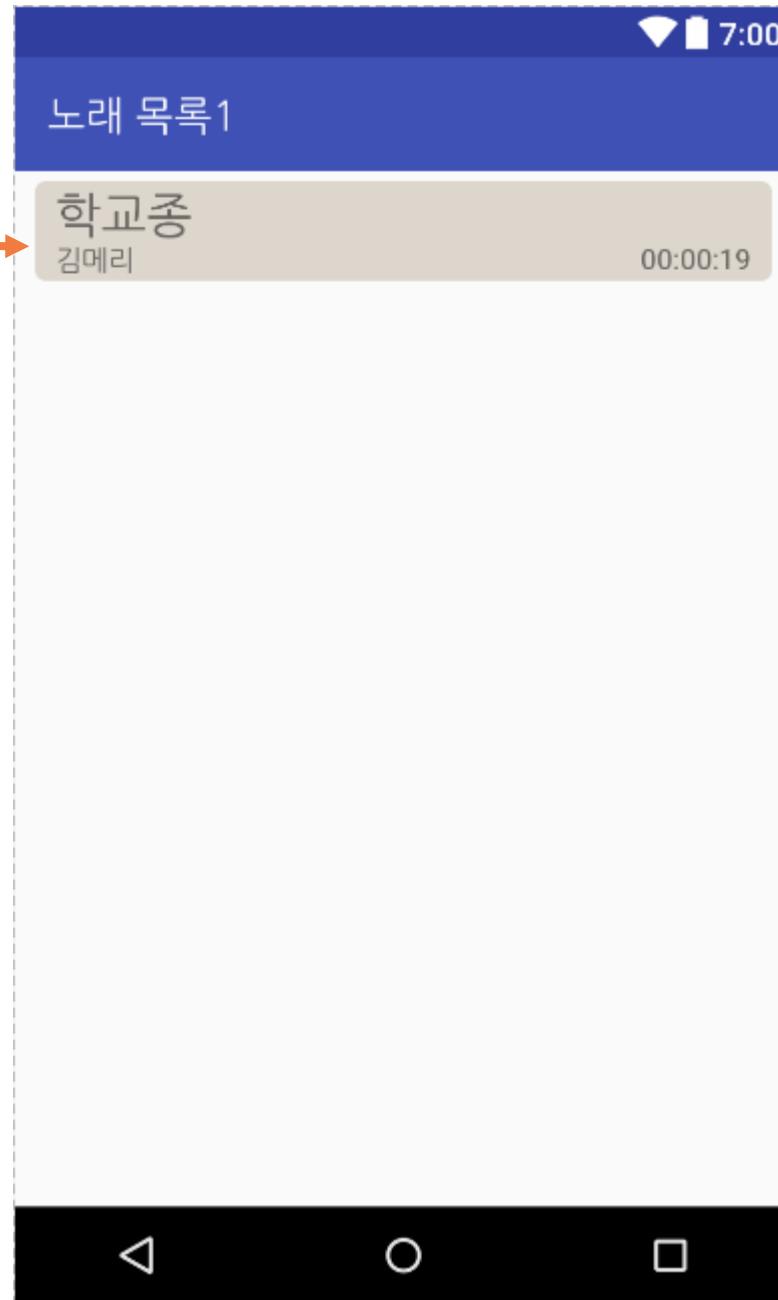
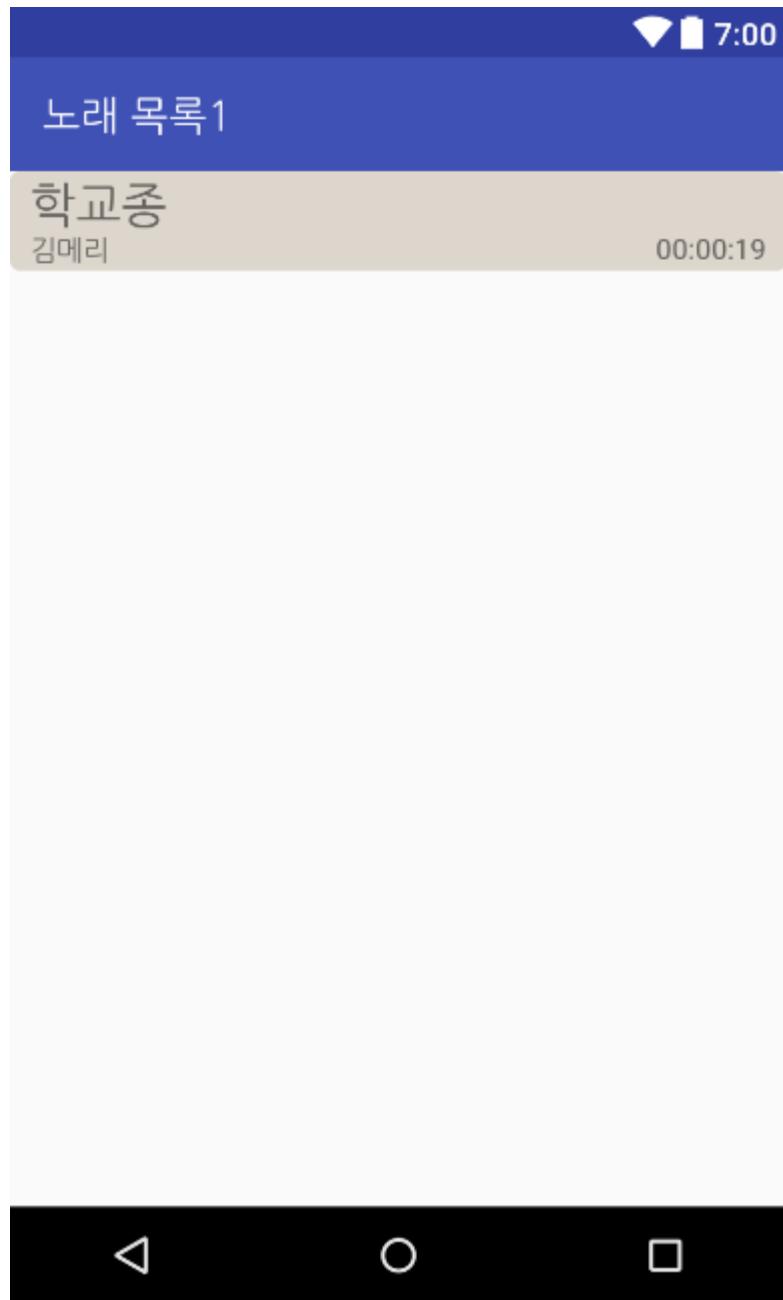


- 노래 목록 표시를 위한 Layout 구조-Component Tree

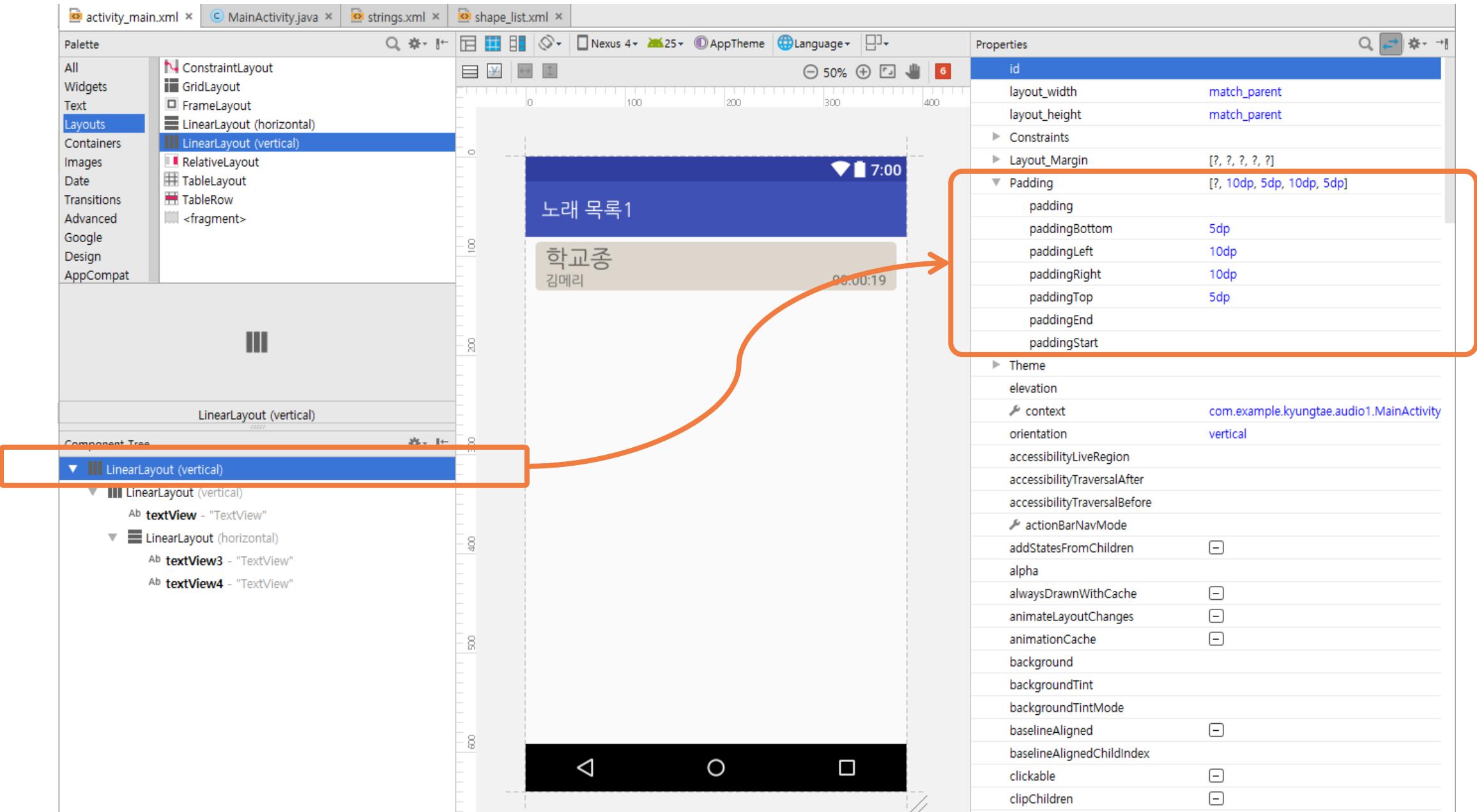


## • 노래 목록 표시 Layout





# • padding 속성을 이용한 여백 설정



# 2.5 Activity 제어

34

- 오디오 리소스에 대한 MediaPlayer를 생성하고 재생

```
activity_main.xml * MainActivity.java * strings.xml * shape_list.xml *
```

```
1 package com.example.kyungtae.audio1;
2
3 import android.media.MediaPlayer;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6
7 public class MainActivity extends AppCompatActivity {
8     MediaPlayer mp = new MediaPlayer(); // 미디어 플레이어 객체 생성
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14
15        // raw 폴더의 schoolbell 오디오 파일에 대한 미디어 플레이어 설정
16        mp = MediaPlayer.create(this, R.raw.schoolbell);
17        mp.setLooping(false); // 반복 재생하지 않음
18        mp.start();
19
20    }
21
22 }
23
24 }
```

미디어 플레이어 객체 생성

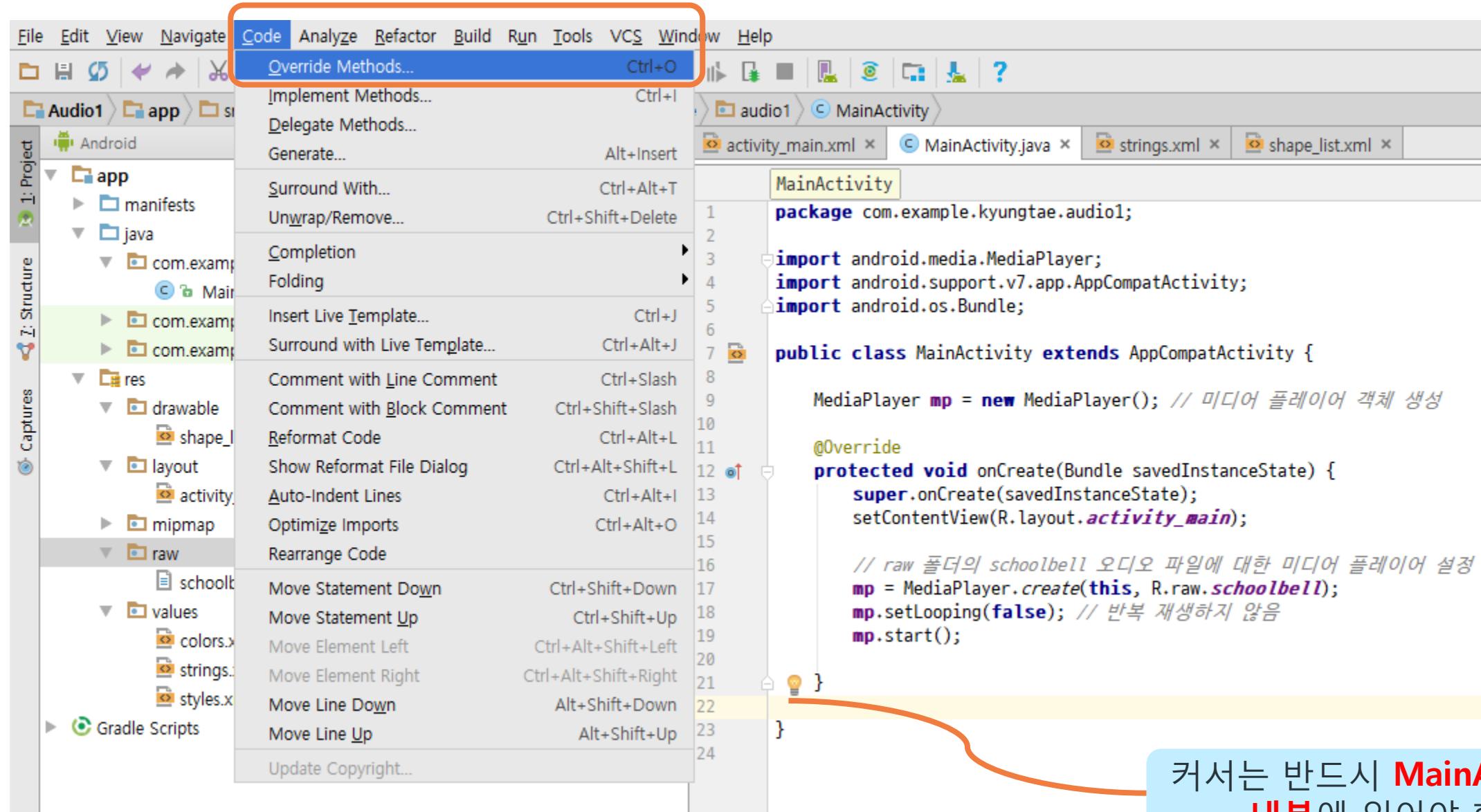
raw폴더의 schoolbell 오디오 파일에 대한 미디어 플레이어 설정

반복 재생하지 않음

미디어 플레이어 실행

# • Activity 소멸될 때 호출되는 메소드(destroy()) 추가

35



- `onDestroy()`는 수퍼 클래스에 정의 되어 있으므로 Override 함

The screenshot shows the Android Studio interface with two main windows:

- Select Methods to Override/Implement Dialog:** This window is titled "Select Methods to Override/Implement". It lists methods from the superclass `android.support.v7.app.AppCompatActivity`. The method `onDestroy():void` is highlighted with a blue selection bar and is connected by a red arrow to its implementation in the code editor.
- MainActivity.java Code Editor:** This window contains the Java code for the `MainActivity`. The `onDestroy()` method is implemented as follows:
 

```

package com.example.kyungtae.audiol;

import android.media.MediaPlayer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    MediaPlayer mp = new MediaPlayer(); // 미디어 플레이어 객체 생성

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

        // raw 폴더의 schoolbell 오디오 파일에 대한 미디어 플레이어 설정
        mp = MediaPlayer.create(this, R.raw.schoolbell);
        mp.setLooping(false); // 반복 재생하지 않음
        mp.start();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
    }
}
      
```

```
activity_main.xml x MainActivity.java x strings.xml x shape_list.xml x

1 package com.example.kyungtae.audio1;
2
3 import android.media.MediaPlayer;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6
7 public class MainActivity extends AppCompatActivity {
8
9     MediaPlayer mp = new MediaPlayer(); // 미디어 플레이어 객체 생성
10
11    @Override
12    protected void onCreate(Bundle savedInstanceState) {
13        super.onCreate(savedInstanceState);
14        setContentView(R.layout.activity_main);
15
16        // raw 폴더의 schoolbell 오디오 파일에 대한 미디어 플레이어 설정
17        mp = MediaPlayer.create(this, R.raw.schoolbell);
18        mp.setLooping(false); // 반복 재생하지 않음
19        mp.start();
20    }
21
22
23    @Override
24    protected void onDestroy() {
25        mp.stop();
26        mp.release();
27        super.onDestroy();
28    }
29
30 }
```

Activity가 소멸될 때 호출되는 메소드  
(Activity 클래스에 정의된 메소드를 재정의함)

미디어 플레이어 실행 중지

미디어 플레이어에 할당된 자원을 해제

Activity 클래스에 정의된  
onDestroy() 메소드로 Activity 종료

# 클래스와 속성/메소드

## • 클래스

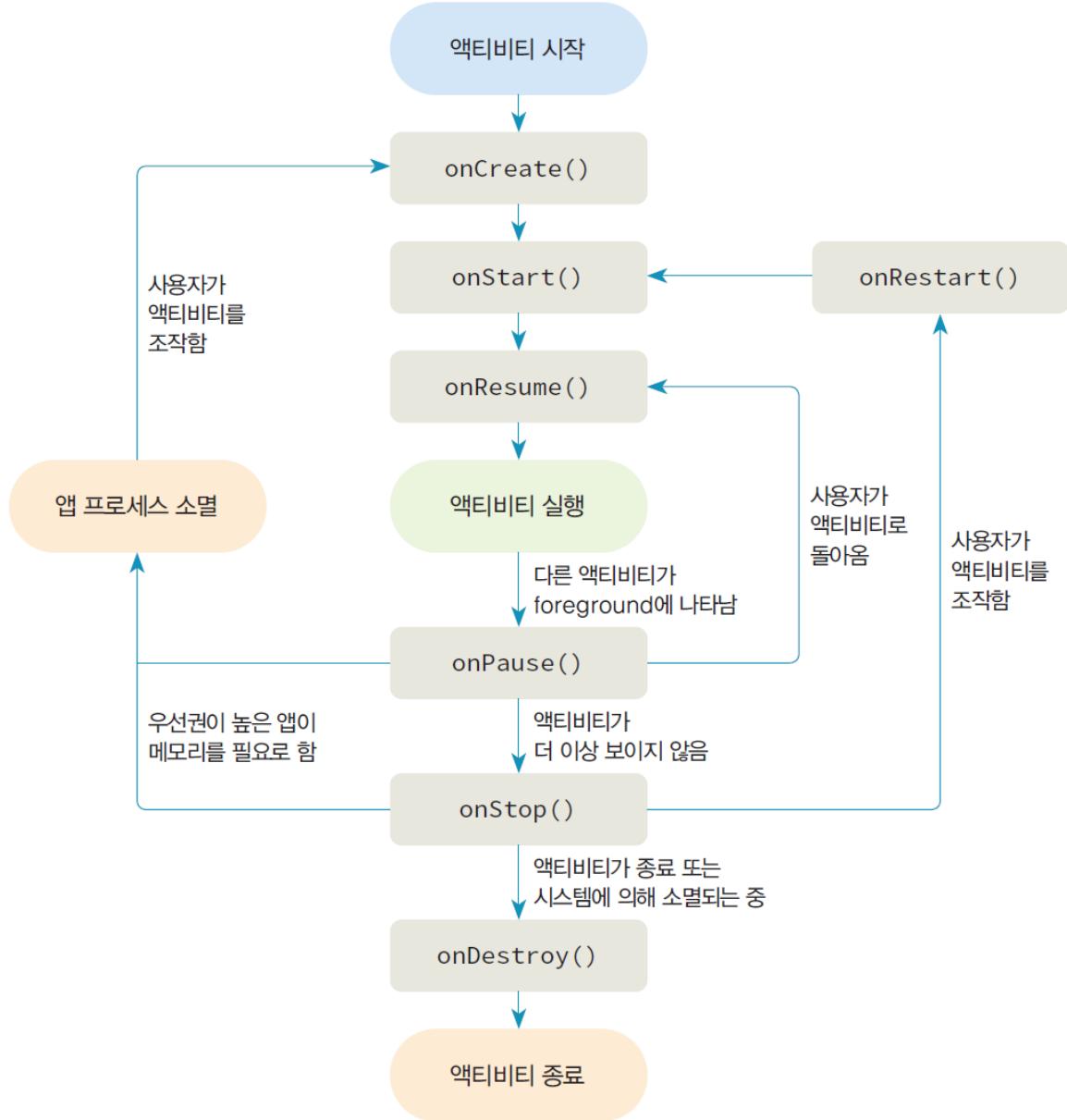
클래스	설명
MediaPlayer	오디오/비디오 재생 및 제어를 위해 사용됨.

## • 메소드

클래스	메소드	설명
Activity	void onDestroy()	Activity가 소멸될 때 불려짐
MediaPlayer	static MediaPlayer.create(Context context, int resid)	주어진 리소스 ID에 대한 MediaPlayer를 생성함
	MediaPlayer()	MediaPlayer 클래스의 생성자
	void setLooping(boolean looping)	미디어 재생을 반복(true) 또는 비반복(false)으로 정함
	void start()	미디어를 재생함
	void stop()	미디어 재생을 중지함
	void release()	MediaPlayer 객체에 할당된 자원을 해제함

# Activity의 생명주기

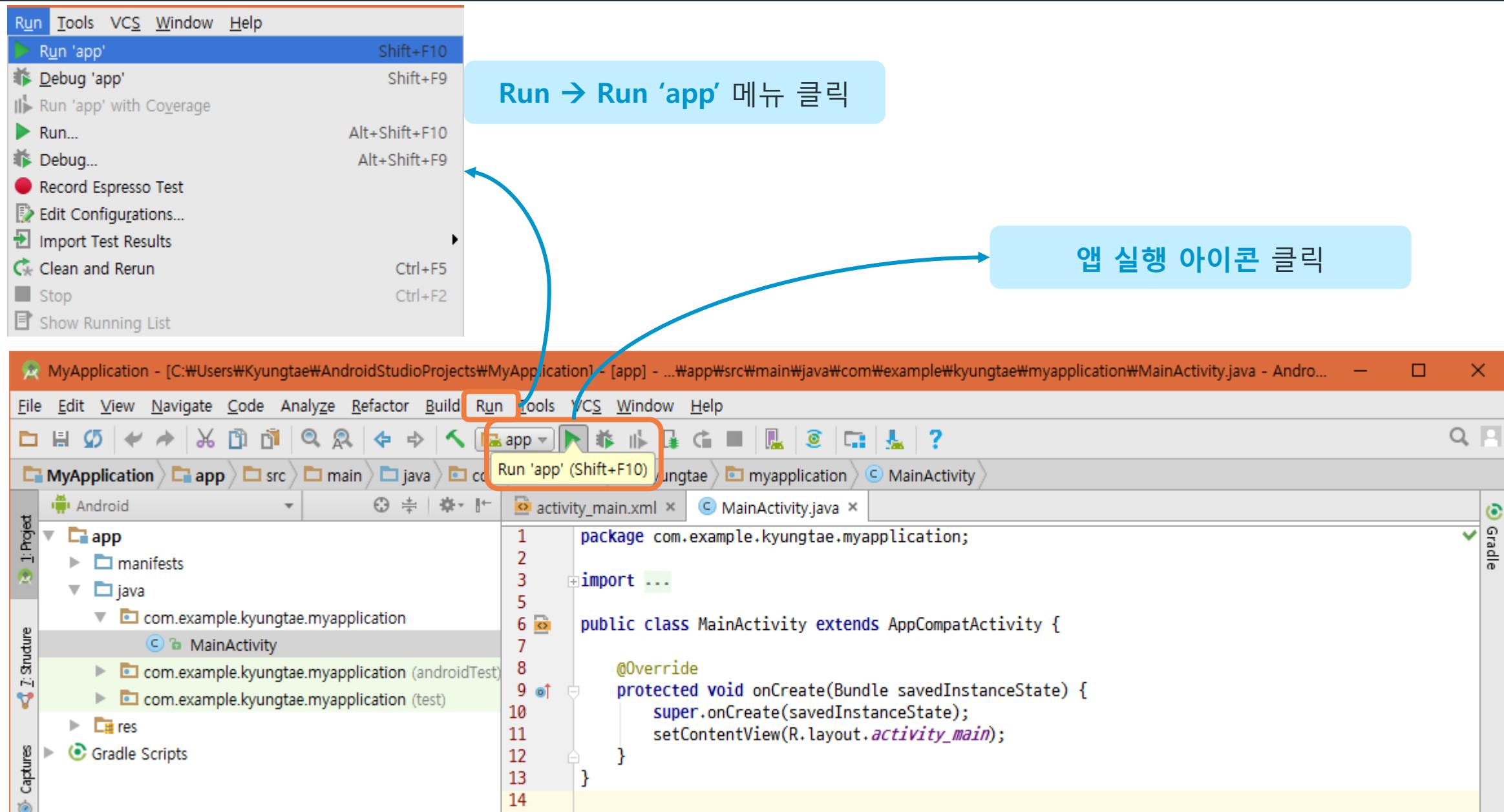
39



메소드	설명
onCreate()	액티비티가 생성될 때
onRestart()	액티비티가 중지되었다가 다시 시작하기 전
onStart()	액티비티가 사용자가 보여질 때
onResume()	액티비티가 사용자와 상호작용할 때
onPause()	다른 액티비티를 시작할 때
onStop()	액티비티가 사용자에게 더 이상 보여지지 않을 때
onDestroy()	액티비티가 소멸될 때

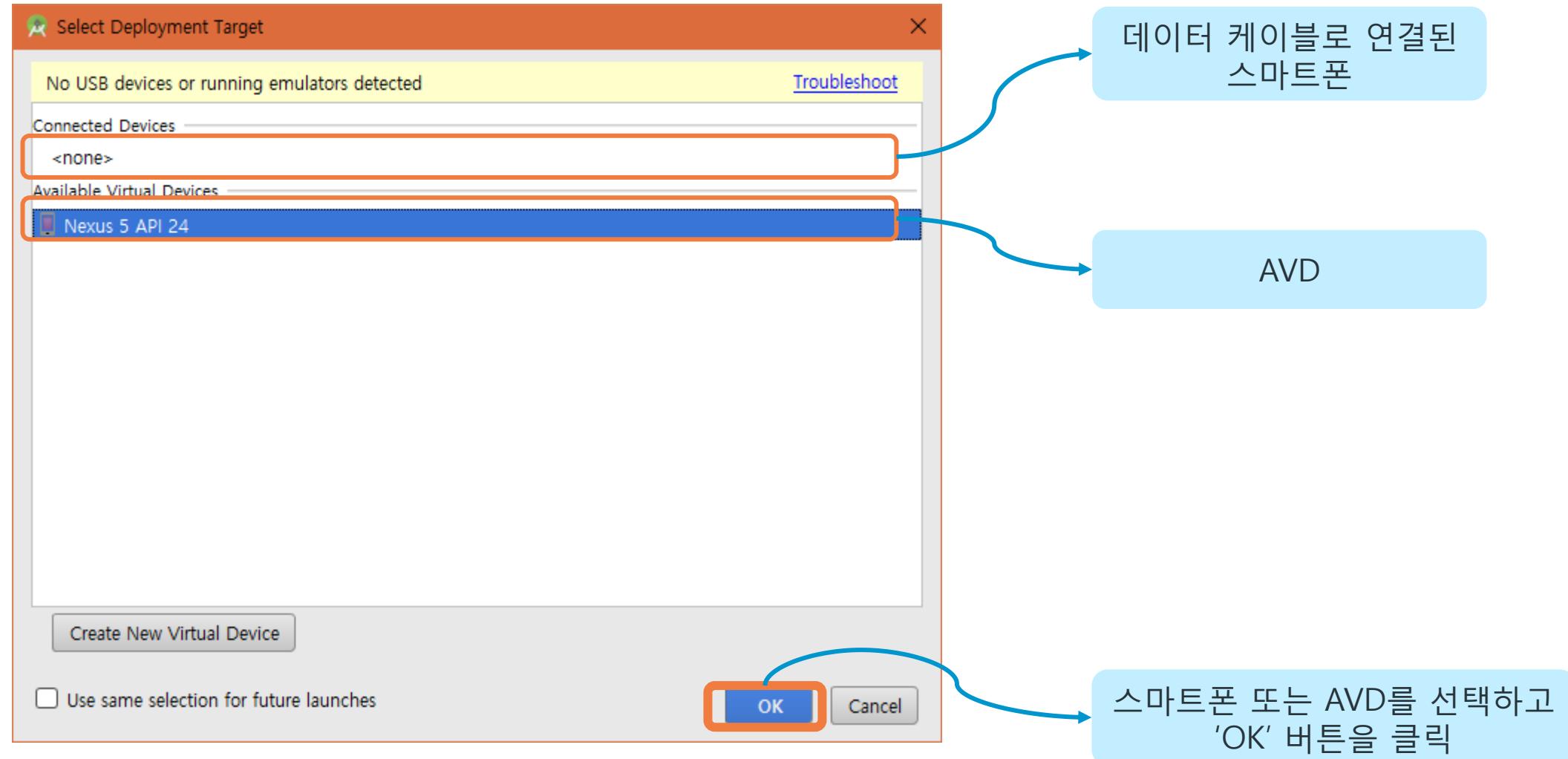
# Step 3. 프로젝트 실행

40

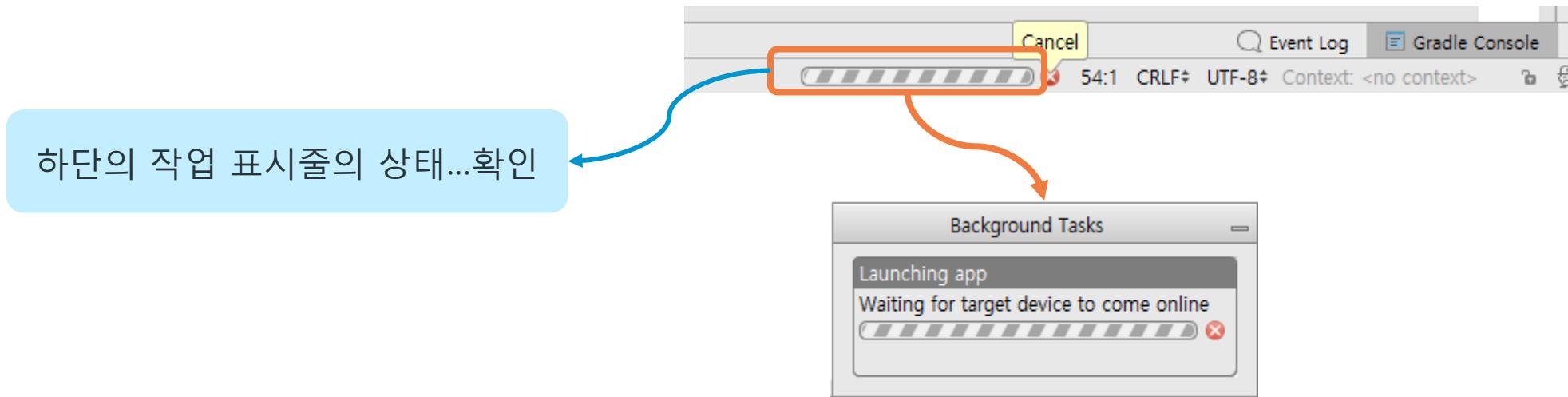


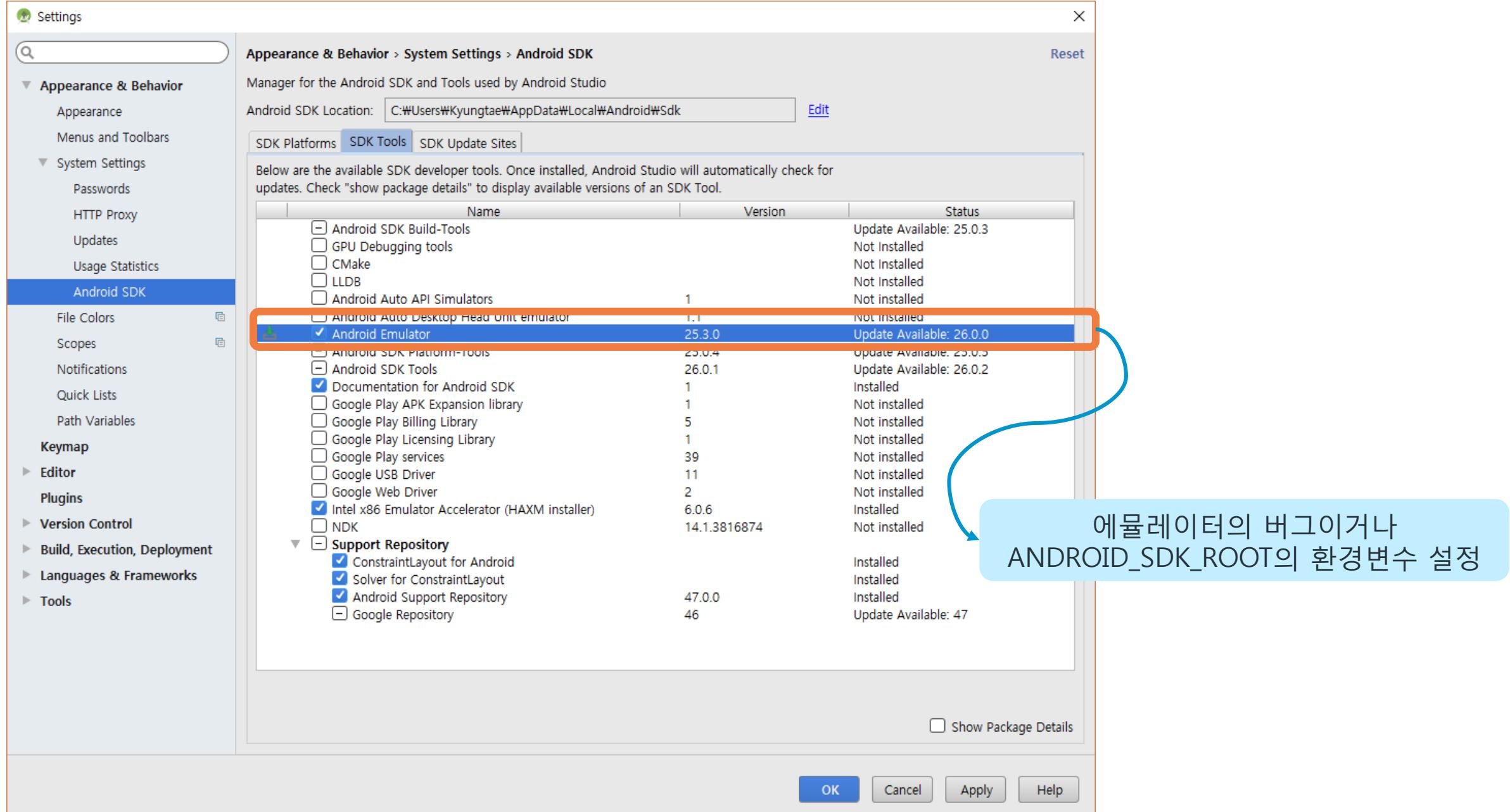
## • AVD 장비 선택하기

41

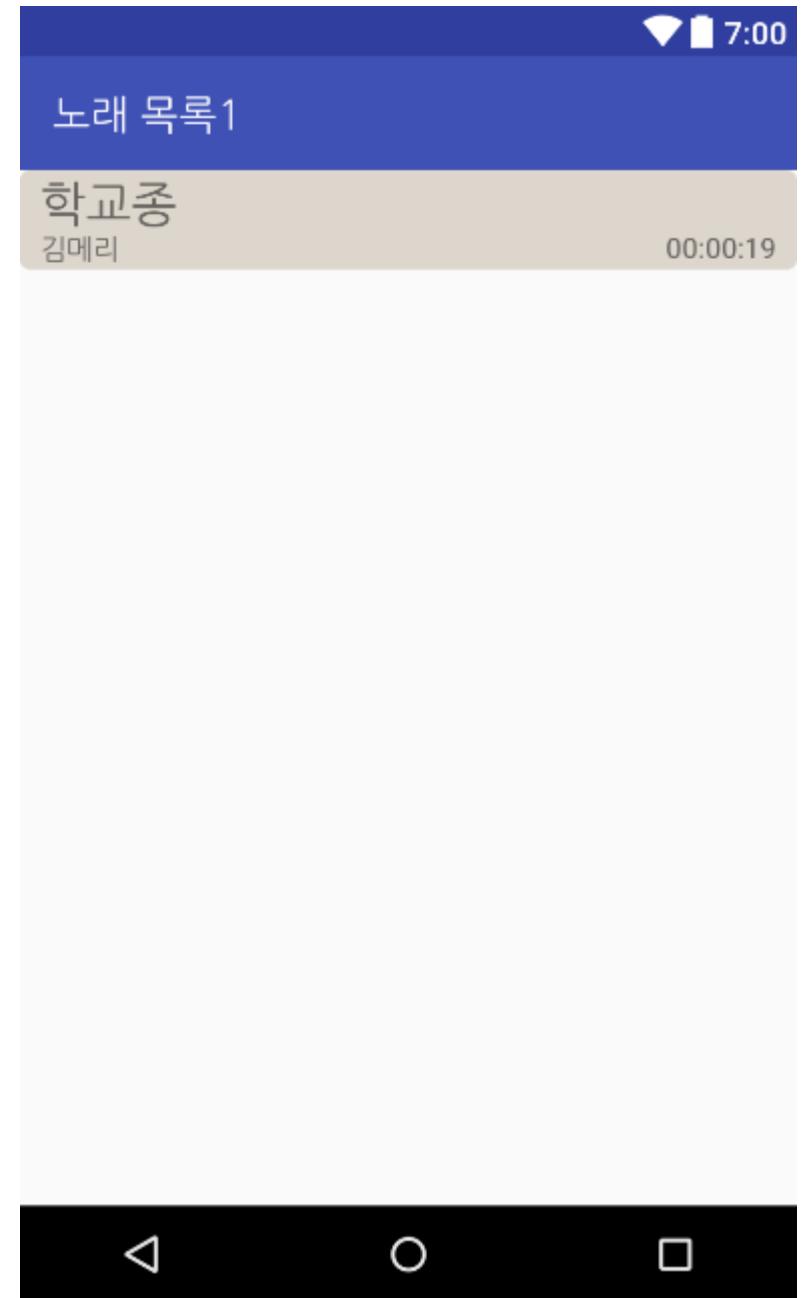


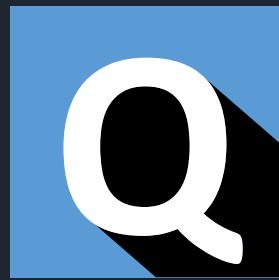
- 에뮬레이터는 실행되지만 앱이 실행되지 않을 경우





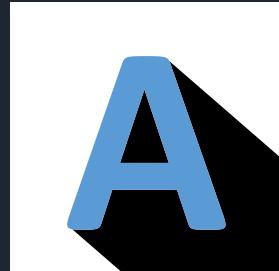
# O outputs





question

&



answer

