



# Multimedia





# 학습 목표

## 교육 목표

- ❖ Android Multimedia
- ❖ Android Audio 재생 구현
- ❖ 실습 I : Audio Player 구현





# Android Multimedia (1)

## ■ Android 지원 Audio 형식

- ❖ WAV (PCM 압축되지 않음)
- ❖ AAC (iPod 포맷, 보호 안됨)
- ❖ MP3 (MPEG-3)
  - ◆ 음성은 모노, 음악은 스테레오
- ❖ WMA (Windows Media Audio)
- ❖ AMR (음성 코덱)
- ❖ OGG (Ogg Vorbis)
  - ◆ 게임 음향 효과와 같은 짧은 클립에 적합
- ❖ MIDI (악기)
- ❖ 대부분의 에뮬레이터에서 제대로 동작하는 포맷
  - ◆ OGG, WAV, MP3 포맷
- ❖ 기본 오디오 포맷은 44.1 KHZ, 16 비트 스테레오 오디오





# Android Multimedia (2)

## ■ MediaPlayer 객체 사용 과정

① import android.media.MediaPlayer;

② MediaPlayer 객체 생성

MP = new MediaPlayer(); // MediaPlayer 객체 생성

③ setDataSource()를 이용하여 audio 파일 Load

MP.setDataSource(File\_Path);

④ prepare() 메소드를 이용하여 사용할 준비

MP.prepare(); // Audio File 준비

⑤ start() 메소드로 재생

MP.start();

⑥ MediaPlayer의 각종 메소드로 재생 컨트롤

MP.pause(); // Pause

⑦ MediaPlayer 객체 소멸 (Garbage collection)

MP.release(); // MediaPlayer 객체 Release



# Android Multimedia (3)

## ■ Android 지원 Video 형식

- ❖ MP4 (MPEG-4 낮은 비트 속도)
- ❖ H.263
- ❖ H.264 (AVC)
- ❖ 윈도우 SDK 상에서는 MP4 만이 안정적으로 동작

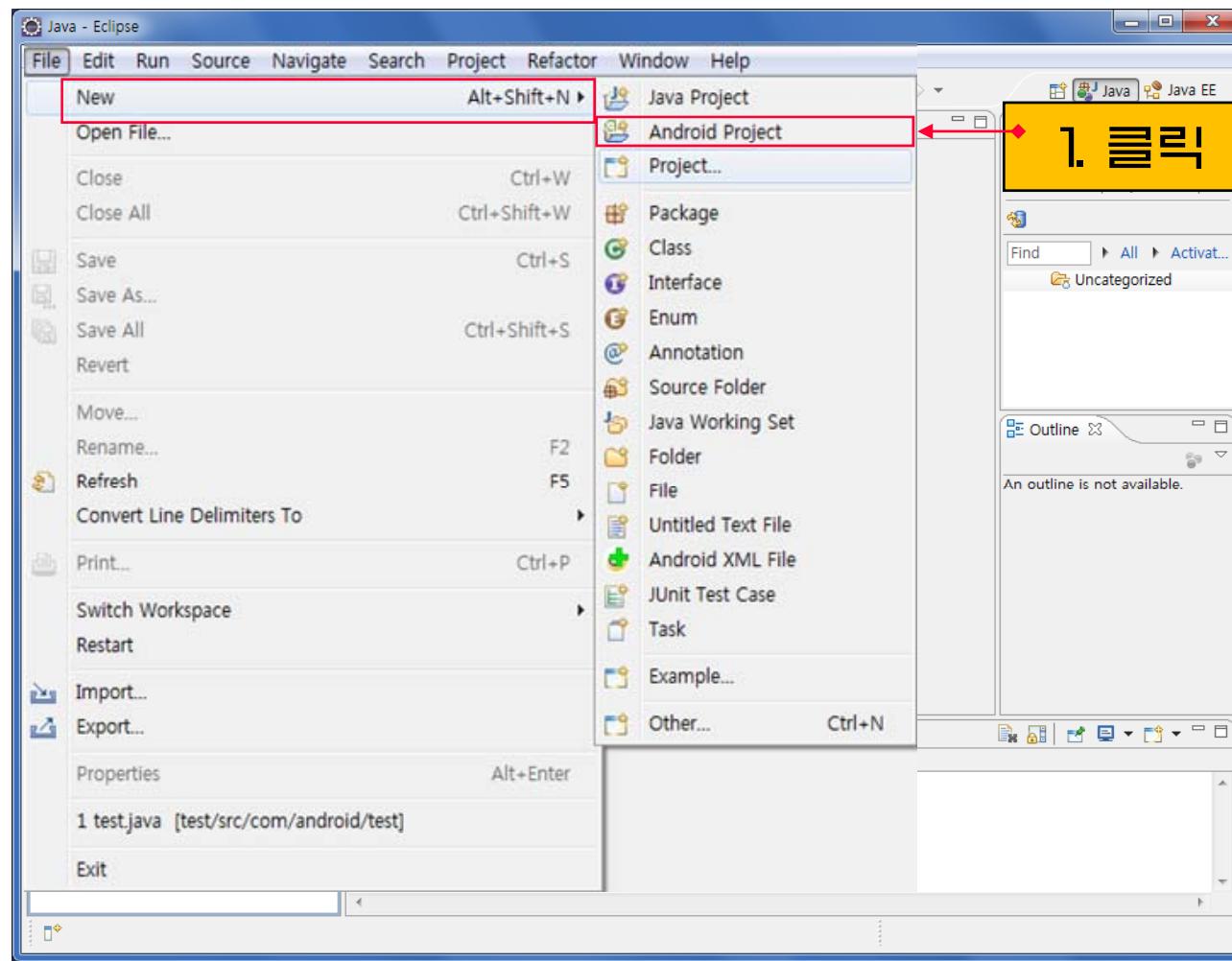




# Android Audio 재생 구현 (1)

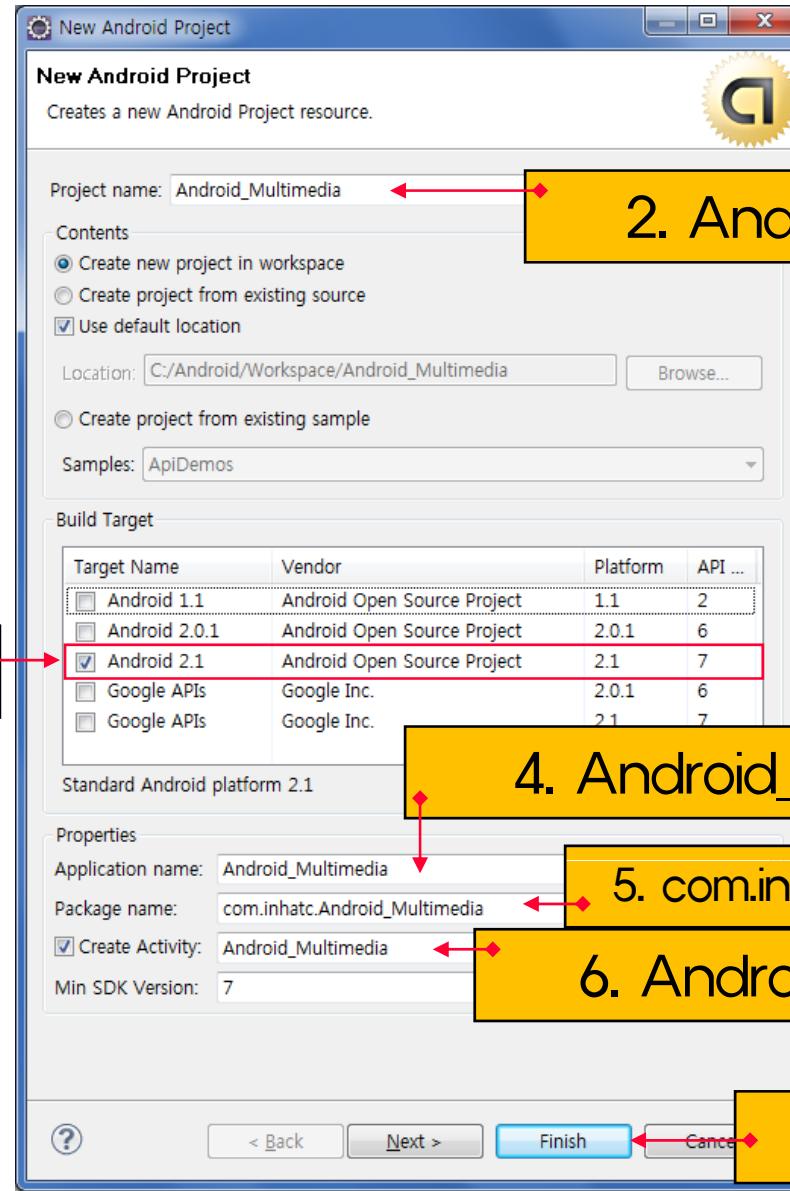
## ■ Android 프로젝트 생성

❖ 프로젝트 명 : Android\_Multimedia





# Android Audio 재생 구현 (2)



| Target Name                                     | Vendor                      | Platform | API ... |
|---|-----------------------------|----------|---------|
| <input type="checkbox"/> Android 1.1            | Android Open Source Project | 1.1      | 2       |
| <input type="checkbox"/> Android 2.0.1          | Android Open Source Project | 2.0.1    | 6       |
| <input checked="" type="checkbox"/> Android 2.1 | Android Open Source Project | 2.1      | 7       |
| <input type="checkbox"/> Google APIs            | Google Inc.                 | 2.0.1    | 6       |
| <input type="checkbox"/> Google APIs            | Google Inc.                 | 2.1      | 7       |

Standard Android platform 2.1

Properties

Application name: Android\_Multimedia

Package name: com.inhatc.Android\_Multimedia

Create Activity: Android\_Multimedia

Min SDK Version: 7

2. Android\_Multimedia 입력

3. 클릭

4. Android\_Multimedia 입력

5. com.inhatc.android\_Multimedia 입력

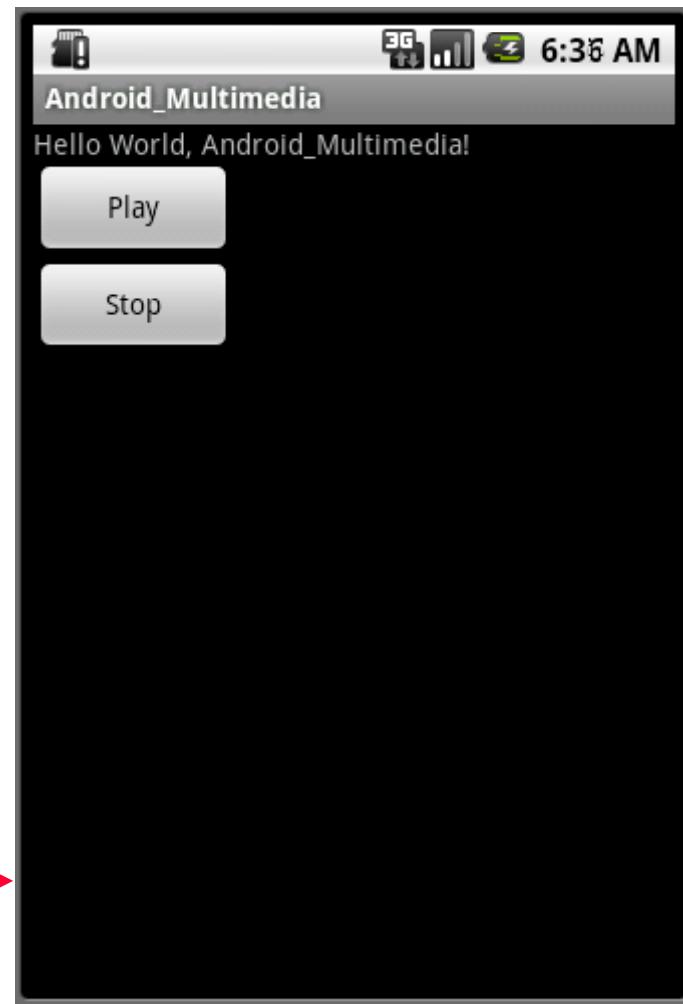
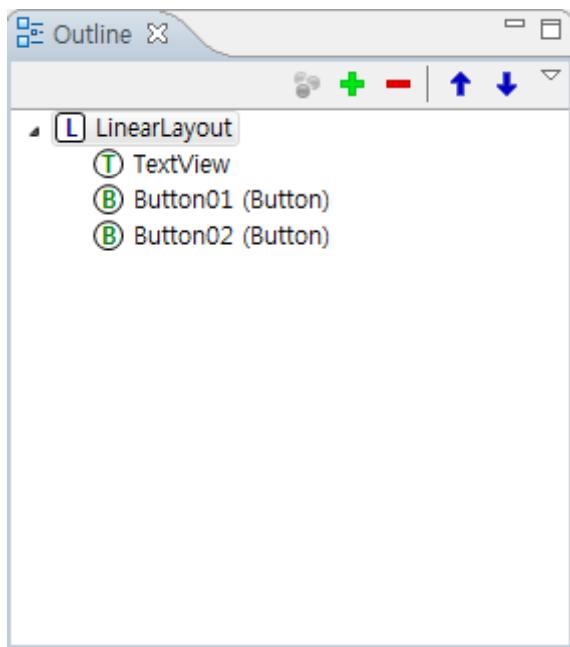
6. Android\_Multimedia 입력

7. 클릭



# Android Audio 재생 구현 (3)

## ■ UI 설계



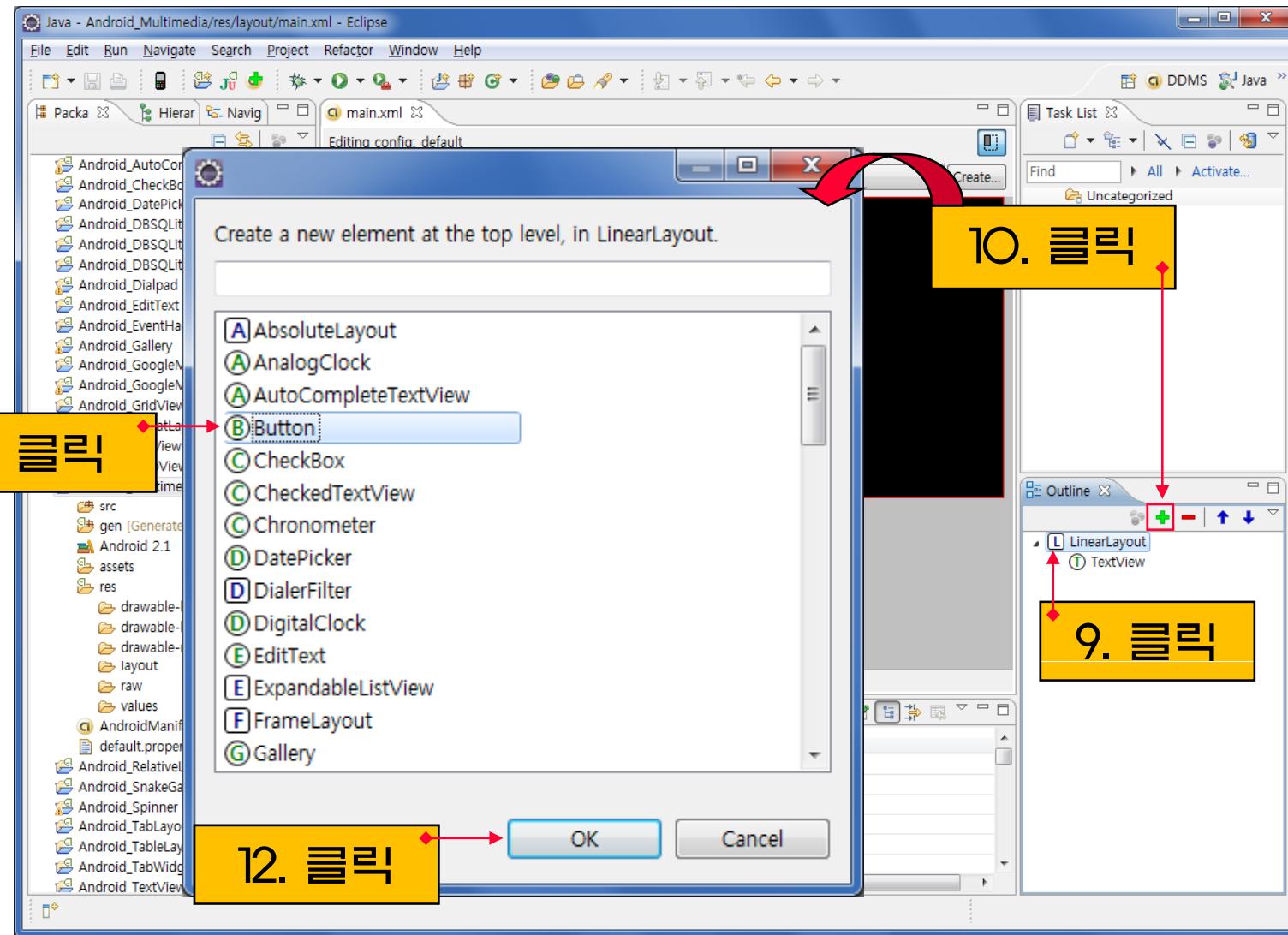
8. UI 설계 및 속성 지정





# Android Audio 재생 구현 (4)

## ❖ Button 추가





# Android Audio 재생 구현 (5)

## ❖ Button 추가 확인

Java - Android\_Multimedia/res/layout/main.xml - Eclipse

File Edit Run Navigate Search Project Refactor Window Help

Packa Hierar Navig

\*main.xml

Editing config: default

Devices ADP1 Config Landscape, clk Locale Theme Create...

Layouts

- A AbsoluteLayout
- D DialerFilter
- E ExpandableList...
- F FrameLayout
- G GridView
- H HorizontalScro...
- I ImageSwitcher
- L LinearLayout
- M ScrollView

Views

- G GestureOverlay...
- S SurfaceView
- V View
- V ViewStub
- A AnalogClock
- A AutoComplete...
- B Button
- C CheckBox

Layout main.xml

Problems Javadoc Properties Console

Property Value

|                          |  |
|--------------------------|--|
| LinearLayout             |  |
| Add states from children |  |
| Always drawn with cache  |  |
| Animation cache          |  |
| Background               |  |
| Baseline aligned         |  |

Task List Find All Activate... Uncategorized

Outline L LinearLayout  
T TextView  
B Button01 (Button)  
B Button02 (Button)

13. Button 추가 확인



# Android Audio 재생 구현 (6)

## ❖ TextView01, Button 01, 02 속성 지정

| 컨트롤        | 속성 지정   |
|------------|---|
| TextView01 | <ul style="list-style-type: none"><li>• android:layout_width="fill_parent"</li><li>• android:layout_height="wrap_content"</li><li>• android:text="@string/hello"</li></ul>  |
| Button01   | <ul style="list-style-type: none"><li>• android:id="@+id/Button01"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:text="Play"</li><li>• android:minWidth="100dip"</li></ul> |
| Button02   | <ul style="list-style-type: none"><li>• android:id="@+id/Button02"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:text="Stop"</li><li>• android:minWidth="100dip"</li></ul> |





# Android Audio 재생 구현 (7)

## Main.xml 수정

```
1<?xml version="1.0" encoding="utf-8"?>
2<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3    android:orientation="vertical"
4    android:layout_width="fill_parent"
5    android:layout_height="fill_parent"
6    >
7    <TextView
8        android:layout_width="fill_parent"
9        android:layout_height="wrap_content"
10       android:text="@string/hello"
11    />
12    <Button
13        android:id="@+id/Button01"
14        android:layout_width="wrap_content"
15        android:layout_height="wrap_content"
16        android:text="Play"
17        android:minWidth="100dip">
18    </Button>
19    <Button
20        android:id="@+id/Button02"
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:text="Stop"
24        android:minWidth="100dip">
25    </Button>
26</LinearLayout>
```

14. XML code 수정 확인



# Android Audio 재생 구현 (8)

## ❖ Main.xml 수정 확인

The screenshot shows the Eclipse IDE interface for an Android project named "Android\_Multimedia". The left sidebar displays the project structure with packages like "Android\_AutoComplete", "Android\_CheckBoxRadioButton", and "Android\_Multimedia". The main workspace shows the "main.xml" layout editor. The layout contains a black rectangular view with the text "Hello World, Android Multimedia!" at the top, followed by two buttons labeled "Play" and "Stop". A red box highlights this area. Below the layout editor, a yellow box contains the text "16. 속성 지정 확인". Red arrows point from the text "15. 클릭" in a yellow box below to the "Layout" tab in the bottom-left and to the "Property" table in the bottom-right.

Java - Android\_Multimedia/res/layout/main.xml - Eclipse

File Edit Run Navigate Search Project Refactor Window Help

Packa X Hierar X Navig X

Devices ADP1 Config Landscape, clc Locale Theme Create...

Editing config: default

Layouts

- A AbsoluteLayout
- D DialerFilter
- E ExpandableList...
- F FrameLayout
- G GridView
- H HorizontalScro...
- I ImageSwitcher
- L LinearLayout

Views

- G GestureOverlay...
- S SurfaceView
- V View
- V ViewStub
- A AnalogClock
- A AutoComplete...
- B Button
- C CheckBox

main.xml

Hello World, Android Multimedia!

Play

Stop

16. 속성 지정 확인

Layout main.xml

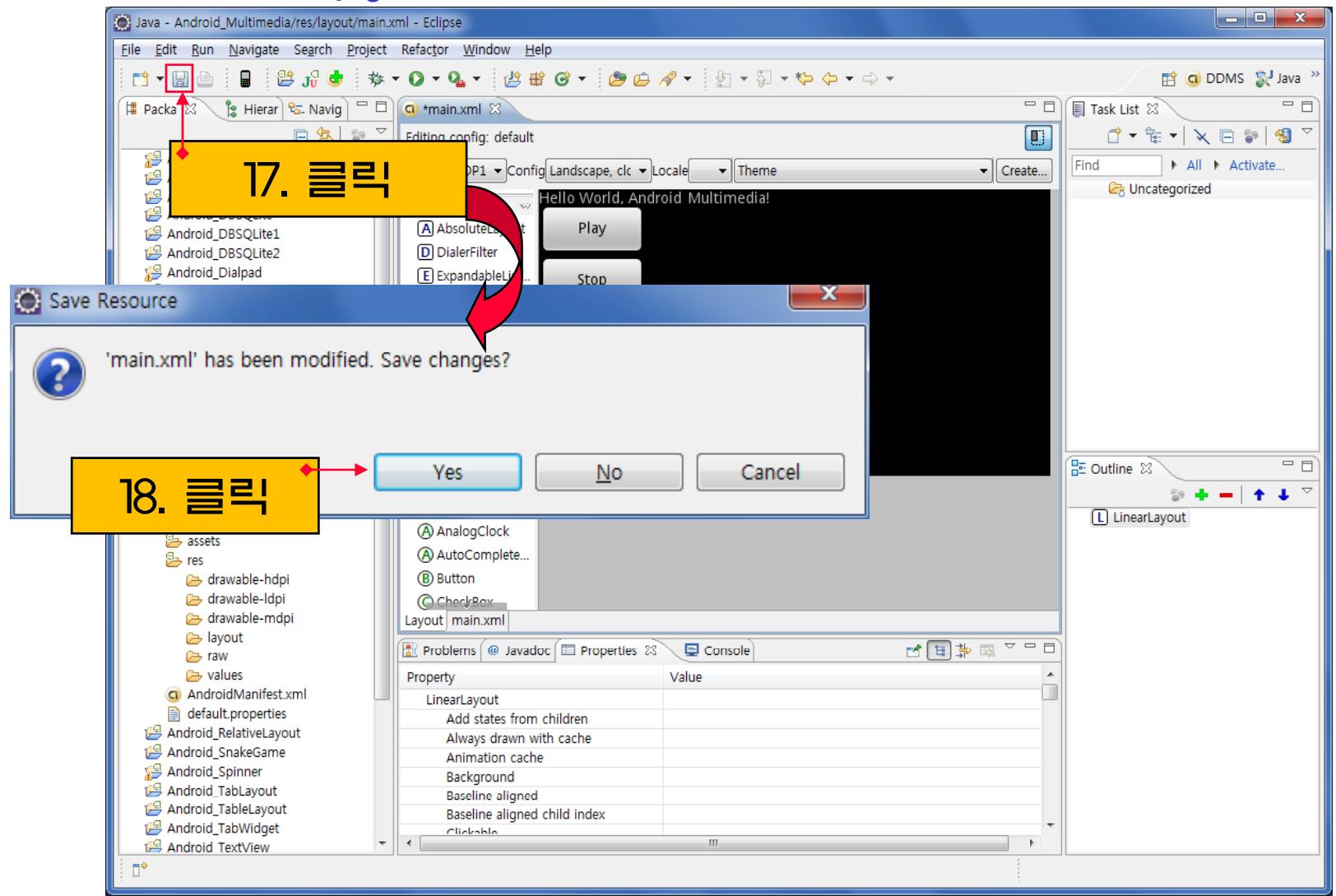
Property Value

|              |                              |
|--------------|------------------------------|
| LinearLayout | Add states from children     |
|              | Always drawn with cache      |
|              | Animation cache              |
|              | Background                   |
|              | Baseline aligned             |
|              | Baseline aligned child index |
|              | Clickable                    |



# Android Audio 재생 구현 (9)

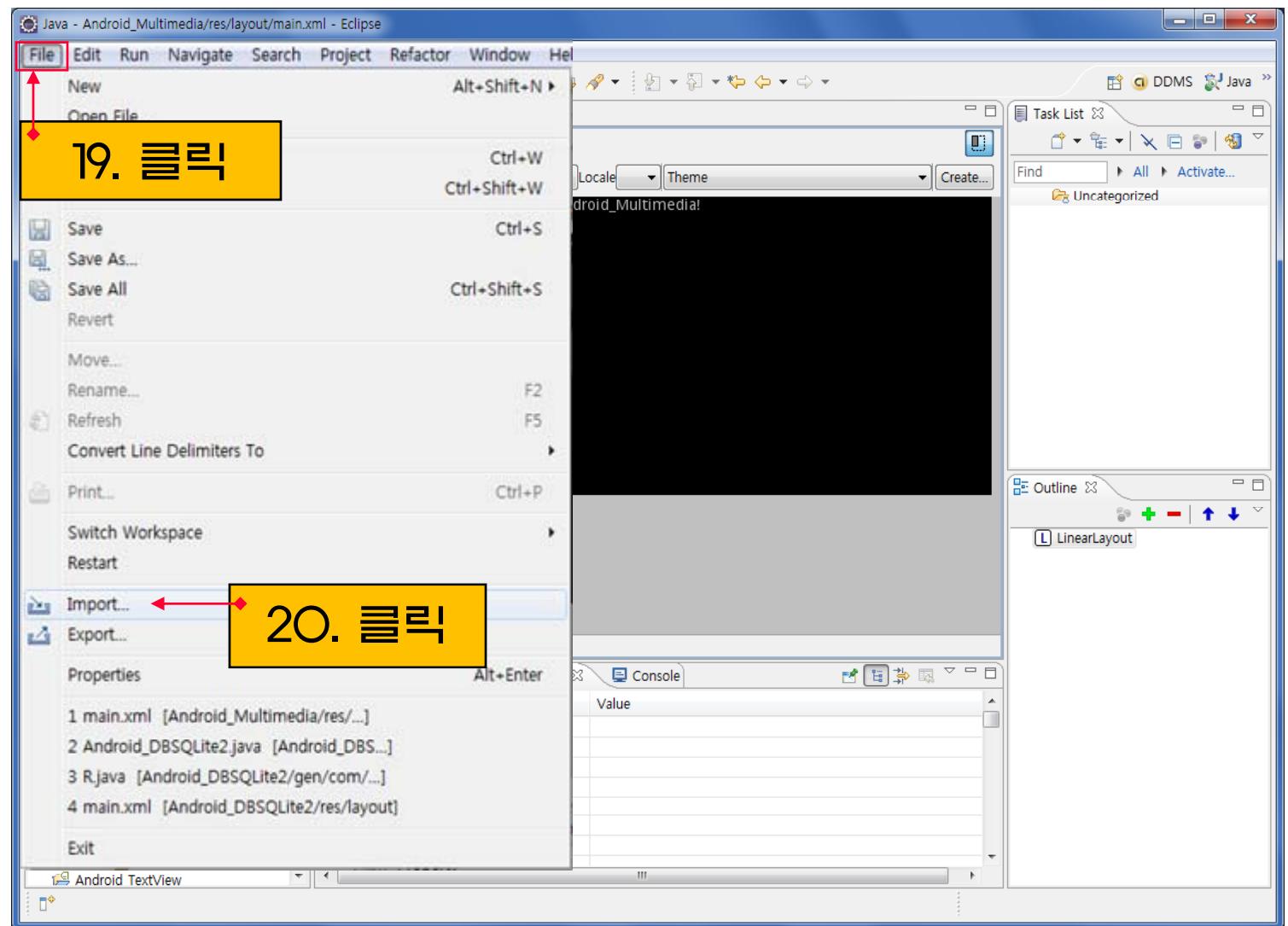
## ❖ Main.xml 저장





# Android Audio 재생 구현 (10)

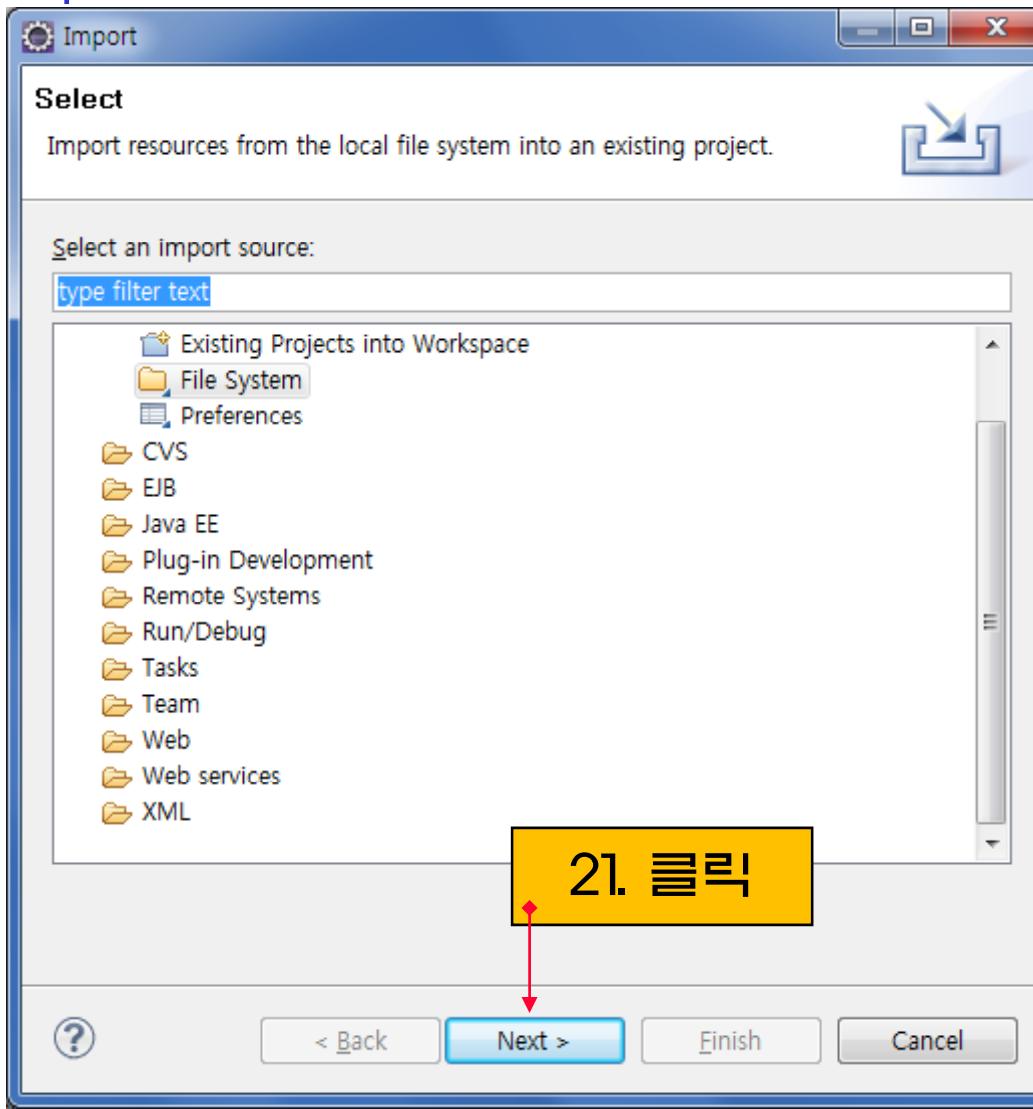
## ❖ MP3 파일 import





# Android Audio 재생 구현 (11)

## ❖ Import 화면





# Android Audio 재생 구현 (12)

Import

File system  
Source must not be empty.

From directory:  Browse...

Filter Types... Select All Deselect All

Into folder: Android\_Multimedia/res

Options  
 Overwrite existing resources without warning  
 Create complete folder structure  
 Create selected folders only

22. 클릭

Import from directory  
Select a directory to import from.

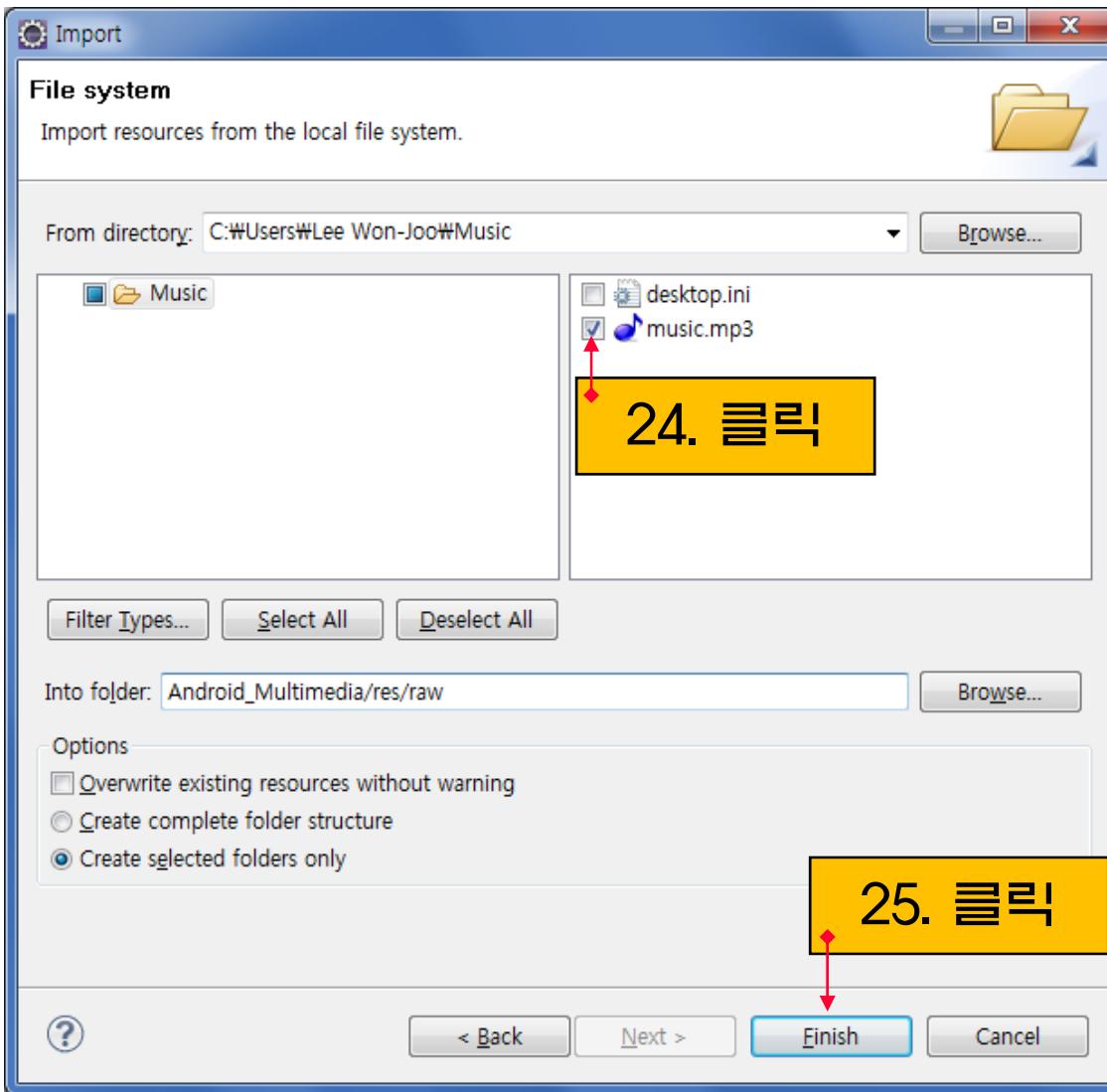
Lee Won-Joo  
  ▶ .android  
  ▶ AppData  
  ▶ EurekaLog  
  ▶ Office Genuine Advantage  
  ▶ Tracing  
  ▶ workspace  
  ▶ 검색  
  ▶ 내 문서  
  ▶ 내 비디오  
  ▶ 내 사진  
  ▶ 내 음악  
    ▶ 재생 목록  
  ▶ 다운로드  
  ▶ 링크  
  ▶ 바탕 화면  
  ▶ 예약  
폴더(F): 내 음악  
새 폴더 만들기(A) 확인 취소

23. 클릭



# Android Audio 재생 구현 (13)

## ❖ MP3 파일 선택





# Android Audio 재생 구현 (14)

## ❖ MP3 파일 import 결과 확인

The screenshot shows the Eclipse IDE interface for an Android project named "Android\_Multimedia". The left pane displays the project structure with the "raw" folder highlighted by a red box and an arrow pointing to a yellow box containing the text "26. 확인". The "raw" folder contains a file named "music.mp3". The center pane shows the XML layout file "main.xml" with two buttons labeled "Play" and "Stop". The right pane shows the "Console" output window with the following log entries:

| Value         |   |
|---------------|---|
| editable      | false                                   |
| last modified | true                                    |
| linked        | 2010년 3월 1일 오전 10:55:09                 |
| location      | false                                   |
| name          | C:\Android\workspace\Android_Multimedia |
| path          | Android_Multimedia                      |



# Android Audio 재생 구현 (15)

## ❖ R.java



```
* R.java X
1 /* AUTO-GENERATED FILE. DO NOT MODIFY. */
2
3 package com.inhatc.Android_Multimedia;
4
5 public final class R {
6     public static final class attr {
7     }
8     public static final class drawable {
9         public static final int icon=0x7f020000;
10    }
11    public static final class id {
12        public static final int Button01=0x7f060000;
13        public static final int Button02=0x7f060001;
14    }
15    public static final class layout {
16        public static final int main=0x7f030000;
17    }
18    public static final class raw {
19        public static final int music=0x7f040000;
20    }
21    public static final class string {
22        public static final int app_name=0x7f050001;
23        public static final int hello=0x7f050000;
24    }
25 }
26 }
```

27. id 클래스 변수 추가 확인





# Android Audio 재생 구현 (16)

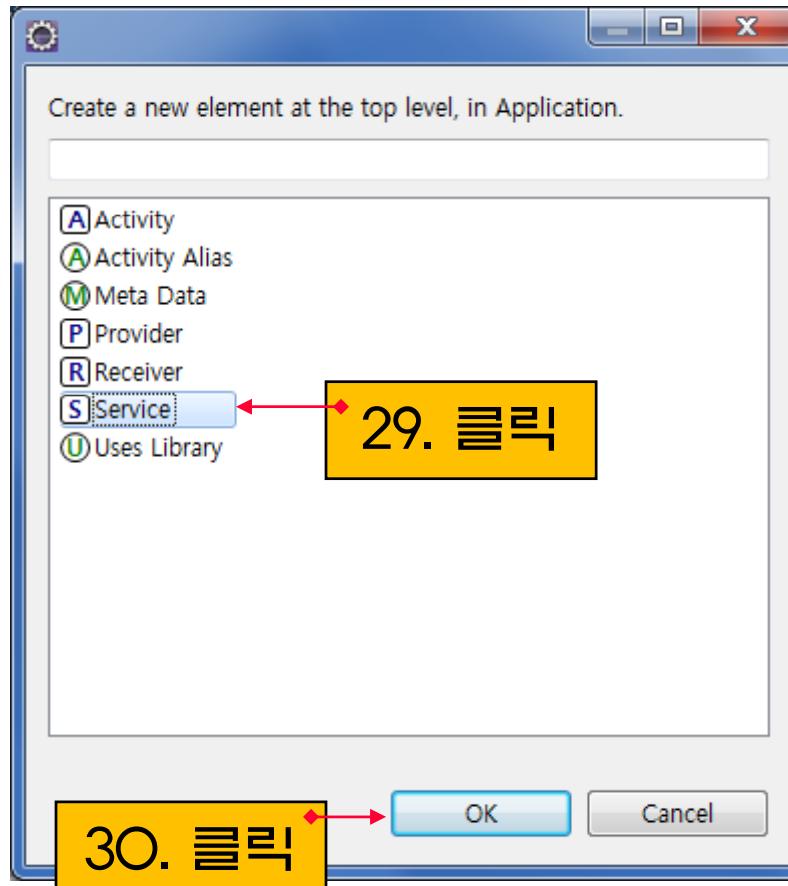
## ■ AndroidManifest.xml 수정

The screenshot shows the Eclipse IDE interface with the title bar "Java - Android\_Multimedia/AndroidManifest.xml - Eclipse". The left sidebar displays the project structure for "Android\_Multimedia" with files like Android\_AutoComplete, Android\_CheckBoxRadioButton, and Android\_Multimedia. The main central area is titled "Android Manifest Application" under "Application Toggle". It contains sections for "Application Attributes" (Name, Theme, Label, Icon, Description, Permission, Process, Task affinity, Allow task reparenting, Has code) and "Application Nodes" (listing ".Android\_Multimedia (Activity)" with buttons for Add..., Remove..., Up, and Down). Below these are tabs for Manifest, Application, Permissions, Instrumentation, and AndroidManifest.xml. The right side of the interface includes toolbars for DDMS and Java, and windows for Task List, Outline, Problems, Javadoc, Properties, and Console.



# Android Audio 재생 구현 (17)

## ❖ Service 추가





# Android Audio 재생 구현 (18)

## ❖ AndroidManifest.xml 수정

The screenshot shows the Eclipse IDE interface with the title "Java - Android\_Multimedia/AndroidManifest.xml - Eclipse". The left sidebar displays the project structure with packages like "Android\_AutoComplete", "Android\_CheckBoxRadioButton", etc., and files like "Android\_Multimedia.java", "MP3\_Service.java", and "AndroidManifest.xml". The main editor area is titled "Android Manifest Application" and shows an error message: "The application tag describes application-level components contained in the package, as well as general application attributes. Define an <application> tag in the AndroidManifest.xml". Below this, the "Application Attributes" section is visible, containing fields for Name, Theme, Label, Icon, Description, Permission, Process, Task affinity, Allow task reparenting, and Has code. A yellow callout box with the text "31. MP3\_Service 입력" points to the "Name" field in the "Attributes for" section, which currently contains "MP3\_Service". A red arrow also points to this same field. The bottom of the editor shows tabs for Manifest, Application, Permissions, Instrumentation, and AndroidManifest.xml.



# Android Audio 재생 구현 (19)

## ❖ AndroidManifest.xml 수정 확인

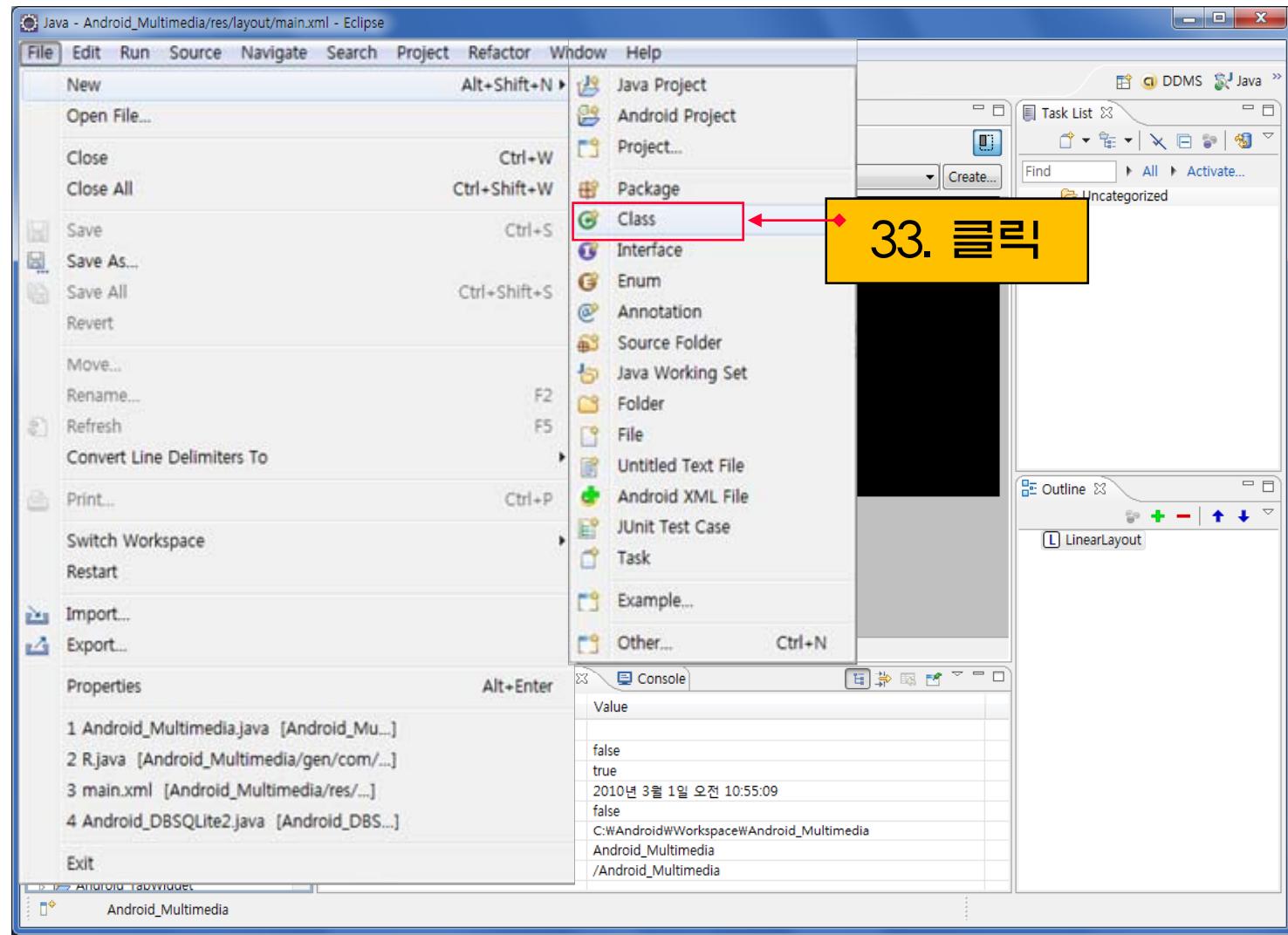
```
*Android_Multimedia Manifest X
1<?xml version="1.0" encoding="utf-8"?>
2<manifest xmlns:android="http://schemas.android.com/apk/res/android"
3    package="com.inhatc.Android_Multimedia"
4    android:versionCode="1"
5    android:versionName="1.0">
6    <application android:icon="@drawable/icon" android:label="@string/app_name">
7        <activity android:name=".Android_Multimedia"
8            android:label="@string/app_name">
9            <intent-filter>
10               <action android:name="android.intent.action.MAIN" />
11               <category android:name="android.intent.category.LAUNCHER" />
12            </intent-filter>
13        </activity>
14        <service android:name="MP3_Service"></service>
15    </application>
16    <uses-sdk android:minSdkVersion="7" />
17
18
19</manifest>
```

32. MP3\_Service 입력 확인



# Android Audio 재생 구현 (20)

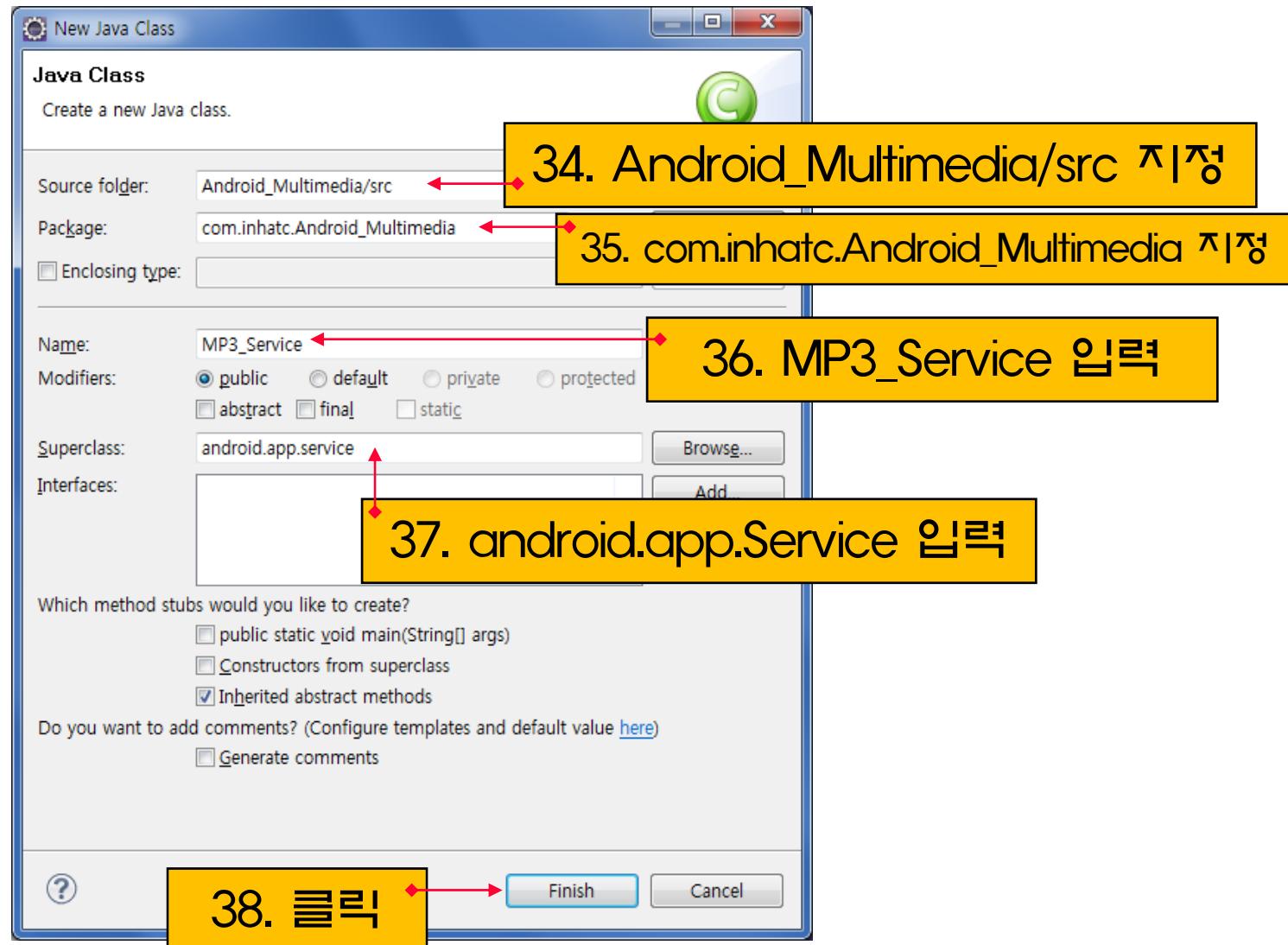
## ❖ Class 추가





# Android Audio 재생 구현 (21)

## ❖ MP3\_Service Class 추가





# Android Audio 재생 구현 (22)

## MP3\_Service.java



```
*MP3_Service.java X
1 package com.inhatc.Android_Multimedia;
2
3 import android.app.Service;
4 import android.content.Intent;
5 import android.media.MediaPlayer;
6 import android.os.IBinder;
7
8 public class MP3_Service extends Service {
9     private MediaPlayer objMP;      //MediaPlayer 객체 선언
10    @Override
11    public IBinder onBind(Intent arg0) {
12        return null;
13    }
14
15    @Override
16    public void onStart(Intent intent, int startId) {
17        try{
18            objMP = MediaPlayer.create(this, R.raw.music);
19            objMP.start();
20        } catch(IllegalStateException e) {
21
22        }
23        super.onStart(intent, startId);
24    }
25
26    @Override
27    public void onDestroy() {
28        objMP.stop();
29        super.onDestroy();
30    }
31 }
```



39. Coding



# Android Audio 재생 구현 (23)

## ■ Android\_MultiMedia.java



```
+Android_Multimedia.java X
1 package com.inhatc.Android_Multimedia;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8
9 public class Android_Multimedia extends Activity {
10
11     Button btnPlay, btnStop;          //Button 객체 선언
12
13     /** Called when the activity is first created. */
14     @Override
15     public void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.main);
```

40. Coding



# Android Audio 재생 구현 (24)

## ❖ Button Event Handler

```
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
```

`btnPlay = (Button)this.findViewById(R.id.Button01);
btnStop = (Button)this.findViewById(R.id.Button02);

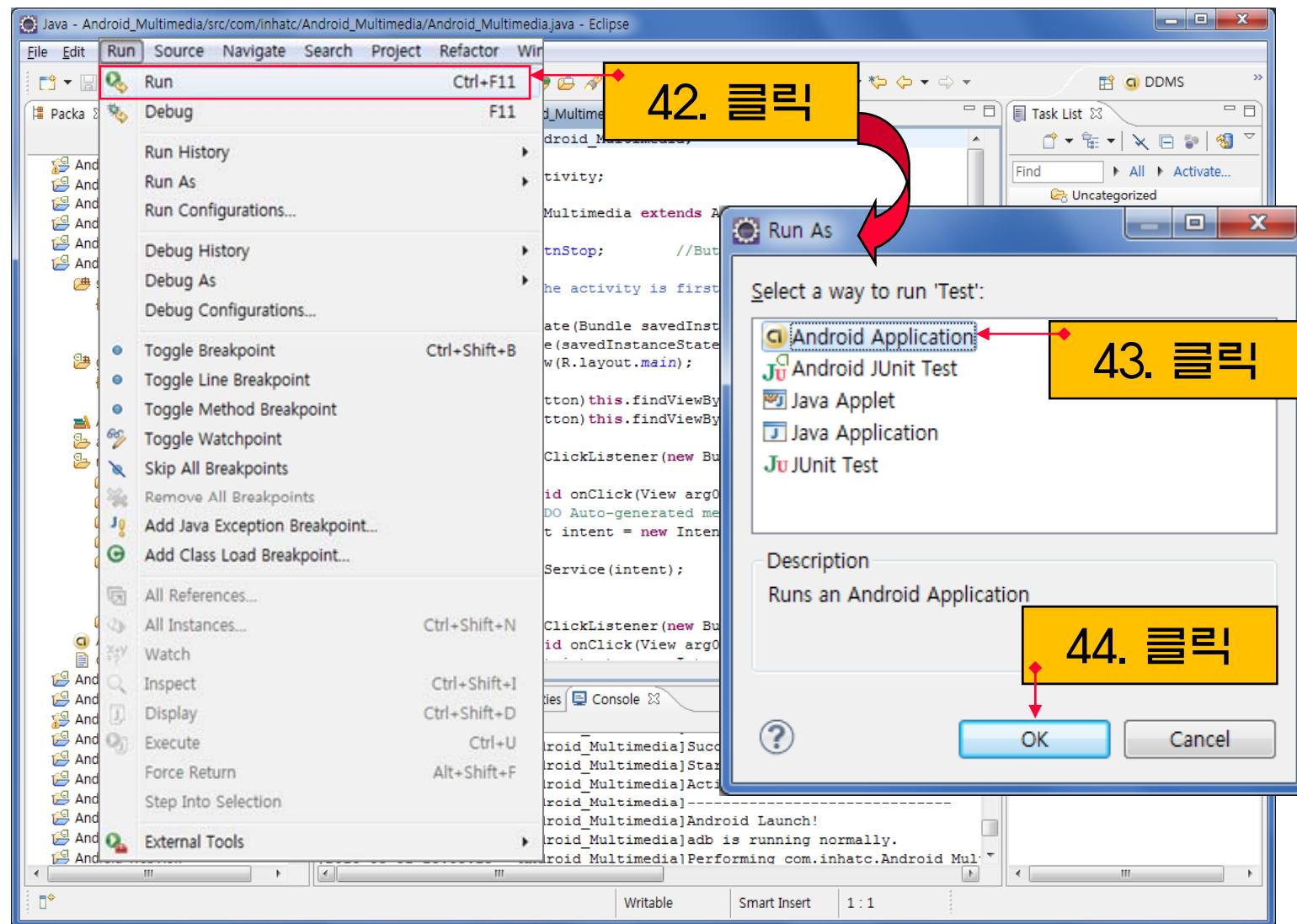
btnPlay.setOnClickListener(new Button.OnClickListener() {
 @Override
 public void onClick(View arg0) {
 // TODO Auto-generated method stub
 Intent intent = new Intent(Android_Multimedia.this,
 MP3_Service.class);
 startService(intent);
 }
});
btnStop.setOnClickListener(new Button.OnClickListener() {
 public void onClick(View arg0){
 Intent intent = new Intent(Android_Multimedia.this,
 MP3_Service.class);
 startService(intent);
 }
});`

41. Coding



# Android Audio 재생 구현 (25)

## ❖ Android 프로젝트 실행





# Android Audio 재생 구현 (26)

## ❖ 실행 결과





# 실습 1 : Audio Player 구현

## ■ Android\_AudioPlayer (실습 시간 : 30분)

❖ 아래 그림과 같이 AudioPlayer를 구현하시오.

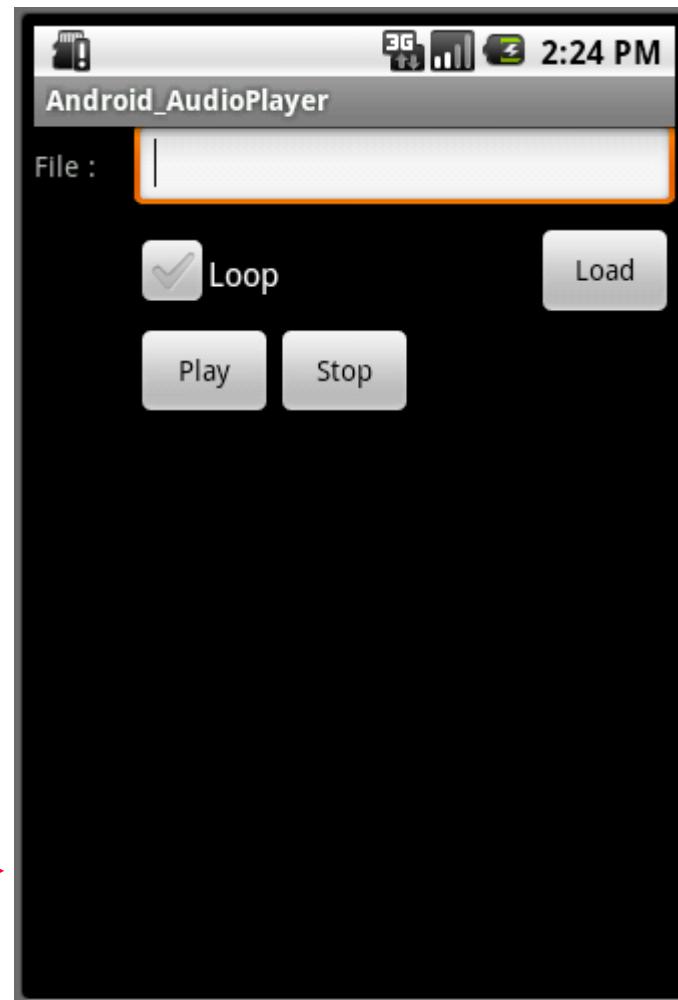
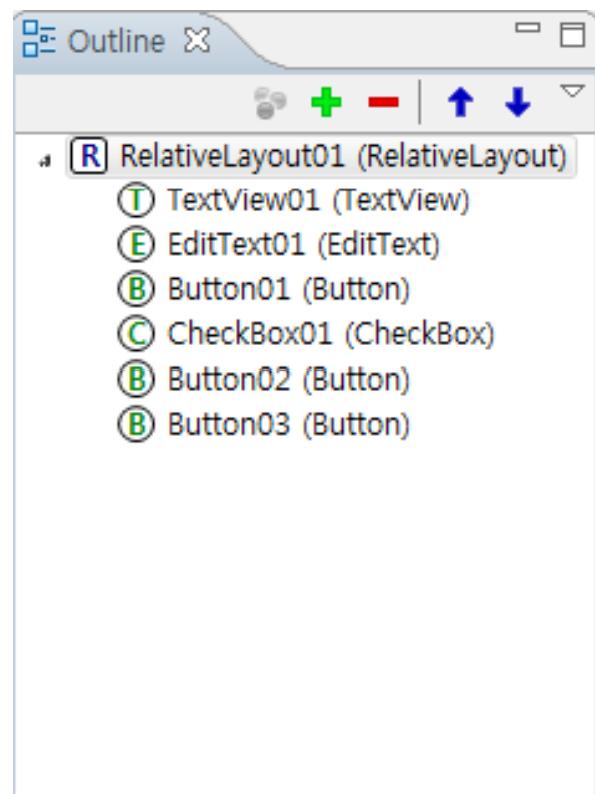
◆ Project Name : Android\_AudioPlayer





# 실습 1 : Audio Player 구현 (1)

## UI 설계



1. UI 설계 및 속성 지정



# 실습 1 : Audio Player 구현 (2)

## ❖ TextView01, EditText01, Button 01 속성 지정

| 컨트롤        | 속성 지정  |
|------------|--|
| TextView01 | <ul style="list-style-type: none"><li>• android:id="@+id/txtFLocation"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:text="File : "</li><li>• android:layout_marginTop="10dip"</li></ul>  |
| EditText01 | <ul style="list-style-type: none"><li>• android:id="@+id/edtSrcFile"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:layout_alignBaseline="@+id/txtFLocation"</li><li>• android:layout_marginLeft="50dip"</li><li>• android:minWidth="270dip"</li></ul> |
| Button01   | <ul style="list-style-type: none"><li>• android:id="@+id/btnLoad"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:text="Load"</li><li>• android:minWidth="70dip"</li><li>• android:layout_marginLeft="250dip"</li></ul>                                 |



# 실습 1 : Audio Player 구현 (2)

## ❖ CheckBox01, Button 02 속성 지정

| 컨트롤        | 속성 지정   |
|------------|---|
| CheckBox01 | <ul style="list-style-type: none"><li>• android:id="@+id/chkLoop"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:layout_marginTop="50dip"</li><li>• android:layout_marginLeft="50dip"</li><li>• android:text="Loop"</li><li>• android:checked="false"</li></ul>   |
| Button02   | <ul style="list-style-type: none"><li>• android:id="@+id	btnPlay"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:layout_marginTop="100dip"</li><li>• android:layout_marginLeft="50dip"</li><li>• android:text="Play"</li><li>• android:minWidth="70dip"</li></ul> |



# 실습 1 : Audio Player 구현 (3)

## ❖ Button 03 속성 지정

| 컨트롤      | 속성 지정  |
|----------|--|
| Button03 | <ul style="list-style-type: none"><li>• android:id="@+id	btnStop"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:layout_alignBaseline="@+id	btnPlay"</li><li>• android:layout_marginTop="100dip"</li><li>• android:text="Stop"</li><li>• android:minWidth="70dip"</li><li>• android:layout_marginLeft="130dip"</li></ul> |





# 실습 1 : Audio Player 구현 (4)

## ❖ Main.xml 수정 확인

Eclipse IDE interface showing the Android project structure and the main.xml layout editor.

The main.xml layout contains the following components:

- A **EditText** labeled `txtSrcFile` for file input.
- A **CheckBox** labeled `chkLoop` for loop playback.
- A **Button** labeled `btnLoad` for loading files.
- A **Button** labeled `btnPlay` for playing audio.
- A **Button** labeled `btnStop` for stopping audio.

The **Properties** panel shows the `Layout gravity` is set to `fill_parent`.

Annotations in the screenshot:

- 2. 블럭**: Points to the `Android_EventHandler` item in the package explorer.
- 3. 속성 지정 확인**: Points to the properties panel.

Java - Android\_AudioPlayer/res/layout/main.xml - Eclipse

File Edit Run Navigate Search Project Refactor Window Help

Packag Hierar Navig

Devices ADP1 Config Landscape Locale Theme Create...

Editing config: default

File : main.xml

File : main.xml

Layouts

- AbsoluteLayout
- DialerFilter
- ExpandableList...
- FrameLayout
- GridView
- HorizontalScro...
- ImageSwitcher
- LinearLayout

Views

- GestureOverlay...
- SurfaceView
- View
- ViewStub
- AnalogClock
- AutoComplete...
- Button
- CheckBox

Task List Outline

RelativeLayout01 (RelativeLayout)

- txtSrcFile (EditText)
- btnLoad (Button)
- chkLoop (CheckBox)
- btnPlay (Button)
- btnStop (Button)

Problems Javadoc Properties Console

Property Value

Misc

- Layout gravity fill\_parent
- Layout height
- Layout margin
- Layout margin bottom
- Layout margin left
- Layout margin right
- Layout margin top



# 실습 1 : Audio Player 구현 (5)

## Main.xml 수정



```
*main.xml x
30    <CheckBox
31        android:id="@+id/chkLoop"
32        android:layout_width="wrap_content"
33        android:layout_height="wrap_content"
34        android:layout_marginTop="50dip"
35        android:text="Loop"
36        android:checked="false"
37        android:layout_marginLeft="50dip">
38    </CheckBox>
39    <Button
40        android:id="@+id	btnPlay"
41        android:layout_width="wrap_content"
42        android:layout_height="wrap_content"
43        android:layout_marginTop="100dip"
44        android:text="Play"
45        android:minWidth="70dip"
46        android:layout_marginLeft="50dip">
47    </Button>
48    <Button
49        android:id="@+id	btnStop"
50        android:layout_width="wrap_content"
51        android:layout_height="wrap_content"
52        android:layout_alignBaseline="@+id	btnPlay"
53        android:layout_marginTop="100dip"
54        android:text="Stop"
55        android:minWidth="70dip"
56        android:layout_marginLeft="120dip">
57    </Button>
58</RelativeLayout>
```

Layout main.xml

4. XML code 수정 확인



# 실습 1 : Audio Player 구현 (6)

## R.java



```
/* AUTO-GENERATED FILE. DO NOT MODIFY. */

package com.inhatc.Android_AudioPlayer;

public final class R {
    public static final class attr {
    }

    public static final class drawable {
        public static final int icon=0x7f020000;
    }

    public static final class id {
        public static final int RelativeLayout01=0x7f050000;
        public static final int btnLoad=0x7f050003;
        public static final int btnPlay=0x7f050005;
        public static final int btnStop=0x7f050006;
        public static final int chkLoop=0x7f050004;
        public static final int edtSrcFile=0x7f050002;
        public static final int txtFLocation=0x7f050001;
    }

    public static final class layout {
        public static final int main=0x7f030000;
    }

    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

5. id class 추가 확인



# 실습 1 : Audio Player 구현 (7)

## ■ Android\_AudioPlayer.java

```
J *Android_AudioPlayer.java X
1 package com.inhatc.Android_AudioPlayer;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.view.View.OnClickListener;
7 import android.widget.Button;
8 import android.widget.CheckBox;
9 import android.widget.EditText;
10 import android.widget.Toast;
11 import android.media.MediaPlayer;
12
13 public class Android_AudioPlayer extends Activity {
14     private MediaPlayer objMP; // MediaPlayer 객체 선언
15
16     @Override
17     public void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.main);
20
21         final EditText audioSrcFile = (EditText)findViewById(R.id.edtSrcFile);
22         final CheckBox chkLoopCTRL = (CheckBox)findViewById(R.id.chkLoop);
23         final Button btnLoad = (Button)findViewById(R.id.btnLoad);
24         final Button btnPlay = (Button)findViewById(R.id.btnPlay);
25         final Button btnStop = (Button)findViewById(R.id.btnStop);
```

6. Coding



# 실습 1 : Audio Player 구현 (8)

## ❖ Load Button Event Handler 구현



```
*Android_AudioPlayer.java
27
28    //Load Button Event Handler
29    btnLoad.setOnClickListener(new OnClickListener() {
30        public void onClick(View v){
31            //Audio File Load
32            if(!LoadAudioFile(audioSrcFile.getText().toString())){
33                Toast //Audio File Load Fail
34                    .makeText(getApplicationContext(), "Audio File Load Fail !",
35                        Toast.LENGTH_LONG)
36                    .show();
37                    return;
38            }
39
40            audioSrcFile.setEnabled(false); // 파일 Path 입력란을 비활성화합니다.
41            btnPlay.setEnabled(true);
42            btnStop.setEnabled(true);
43            chkLoopCTRL.setEnabled(true);
44            btnLoad.setEnabled(false);
45            Toast
46                .makeText(getApplicationContext(), "File : " +
47                    audioSrcFile.getText().toString() +
48                    " Load Success !", Toast.LENGTH_LONG)
49                .show();
50        }
51    });
}
```



7. Coding



# 실습 1 : Audio Player 구현 (9)

## ❖ Loop CheckBox / Play Button Event Handler 구현



The screenshot shows the Android Studio IDE with the file `*Android_AudioPlayer.java` open. The code implements two event handlers: one for a checkbox and one for a play button.

```
// Loop CheckBox Event Handler
chkLoopCTRL.setOnCheckedChangeListener(new OnCheckedChangeListener() {
    public void onClick(View v) {
        if(chkLoopCTRL.isChecked()) { //chkLoopCTRL.Checked = true
            objMP.setLooping(true); //Loop Active
            Toast.makeText(getApplicationContext(), "Loop Active Status",
                    Toast.LENGTH_SHORT)
                    .show();
        }else{
            objMP.setLooping(false); // 반복을 비활성화합니다.
            Toast.makeText(getApplicationContext(), "Loop UnActive Status",
                    Toast.LENGTH_SHORT)
                    .show();
        }
    }
});
```

```
// Play Button Event Handler
btnPlay.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        if(PlayPauseAudio()==0){ //Play
            btnPlay.setText("Pause"); //btnPaly.Text = Pause
        }else{ // 일시정지
            btnPlay.setText("Play"); //btnPaly.Text = Play
        }
    }
});
```

8. Coding



# 실습 1 : Audio Player 구현 (10)

## ❖ Stop Button Event Handler 구현

The screenshot shows a Java code editor window titled "\*Android\_AudioPlayer.java". The code is part of a class definition, likely for an Activity. A red rectangular box highlights the following code block:

```
81
82     //Stop Button Event Handler
83     btnStop.setOnClickListener(new OnClickListener(){ //Press Stop button
84         public void onClick(View v){
85             objMP.stop(); //Stop
86             audioSrcFile.setEnabled(true); //audioSrcFile Active
87             btnPlay.setText("Play");
88             chkLoopCTRL.setChecked(false);
89             btnPlay.setEnabled(false);
90             btnStop.setEnabled(false);
91             chkLoopCTRL.setEnabled(false);
92             btnLoad.setEnabled(true);
93         }
94     });
95 }
```

9. Coding





# 실습 1 : Audio Player 구현 (1)

## ❖ PlayPauseAudio() / LoadAudioFile() Method 구현

```
+Android_AudioPlayer.java x
96    private int PlayPauseAudio(){
97        if(!objMP.isPlaying()){
98            objMP.start();
99            Toast.makeText(getApplicationContext(), "Play",
100                           Toast.LENGTH_SHORT).show();
101           return 0;
102        }else{ // 재생중이라면
103            objMP.pause(); // Pause
104            Toast.makeText(getApplicationContext(), "Pause",
105                           Toast.LENGTH_SHORT).show();
106           return 1;
107        }
108    }
109
110   private boolean LoadAudioFile(String path){ //Audio File Load
111       objMP = new MediaPlayer(); // MediaPlayer 객체 생성
112       try{
113
114           objMP.setDataSource(path);
115           objMP.prepare(); // Audio File 준비
116           return true;
117       }catch(Exception e){ //Audio File Load Fail
118           Toast.makeText(getApplicationContext(), e.getMessage(),
119                           Toast.LENGTH_SHORT).show();
120           return false;
121       }
122       objMP = MediaPlayer.create(this, R.raw.music);
123       return true;
124   }
```

10. Coding



# 실습 1 : Audio Player 구현 (12)

## ❖ onDestroy() Method 구현

```
*Android_AudioPlayer.java X
125
126  public void onDestroy(){ //Activity Destroy
127      super.onDestroy();
128      if(objMP != null) objMP.release(); // MediaPlayer 객체 Release
129      objMP = null;
130  }
131 }
```

A screenshot of an IDE showing Java code for an Android application. The code defines an `onDestroy()` method. It first calls `super.onDestroy()`, then checks if `objMP` is not null and releases it using `release()`. Finally, it sets `objMP` to null. The code is annotated with Korean comments: //Activity Destroy, // MediaPlayer 객체 Release. The code block from line 126 to 130 is highlighted with a red rectangle.

11. Coding





# 실습 1 : Audio Player 구현 (8)

## ❖ 실행 결과





# 학습 요약

- MediaPlayer 객체 사용 방법
- Android Audio 재생 구현
- 실습 I : Audio Player 구현

