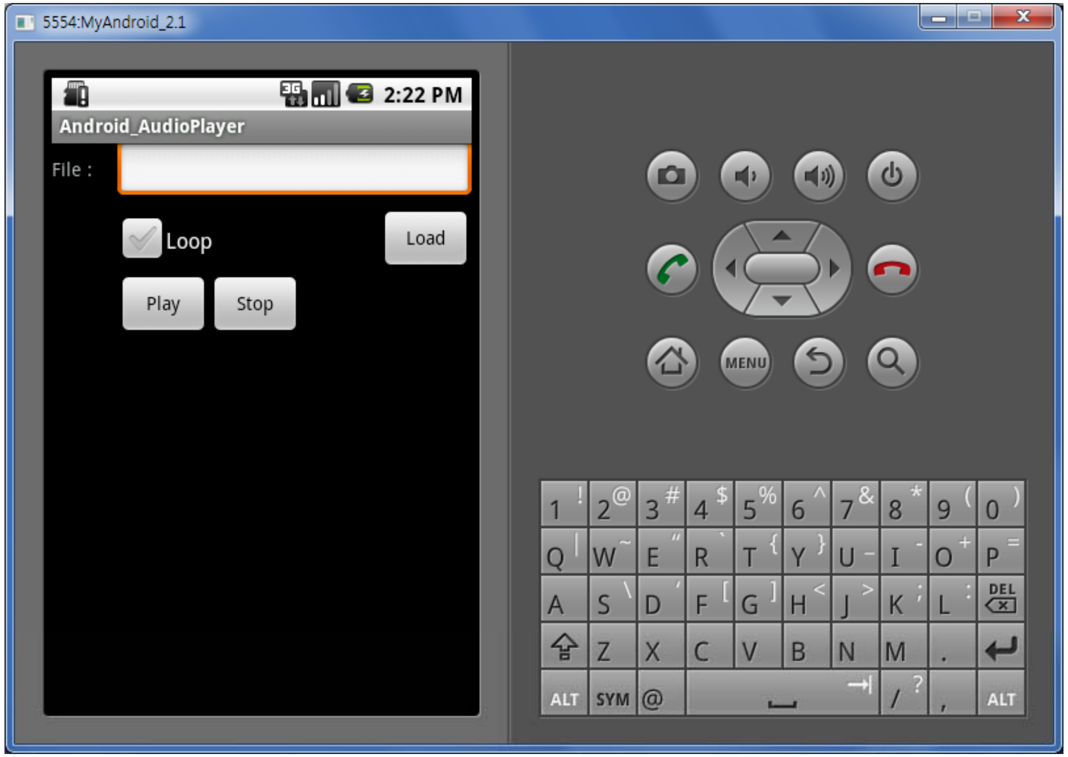
ANDROID DEVELOPER CHALLENGE

Multimedia



학습 목표

■ 교육 목표

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

Navigation voice
Traffic view
Street View

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 사용하여 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)

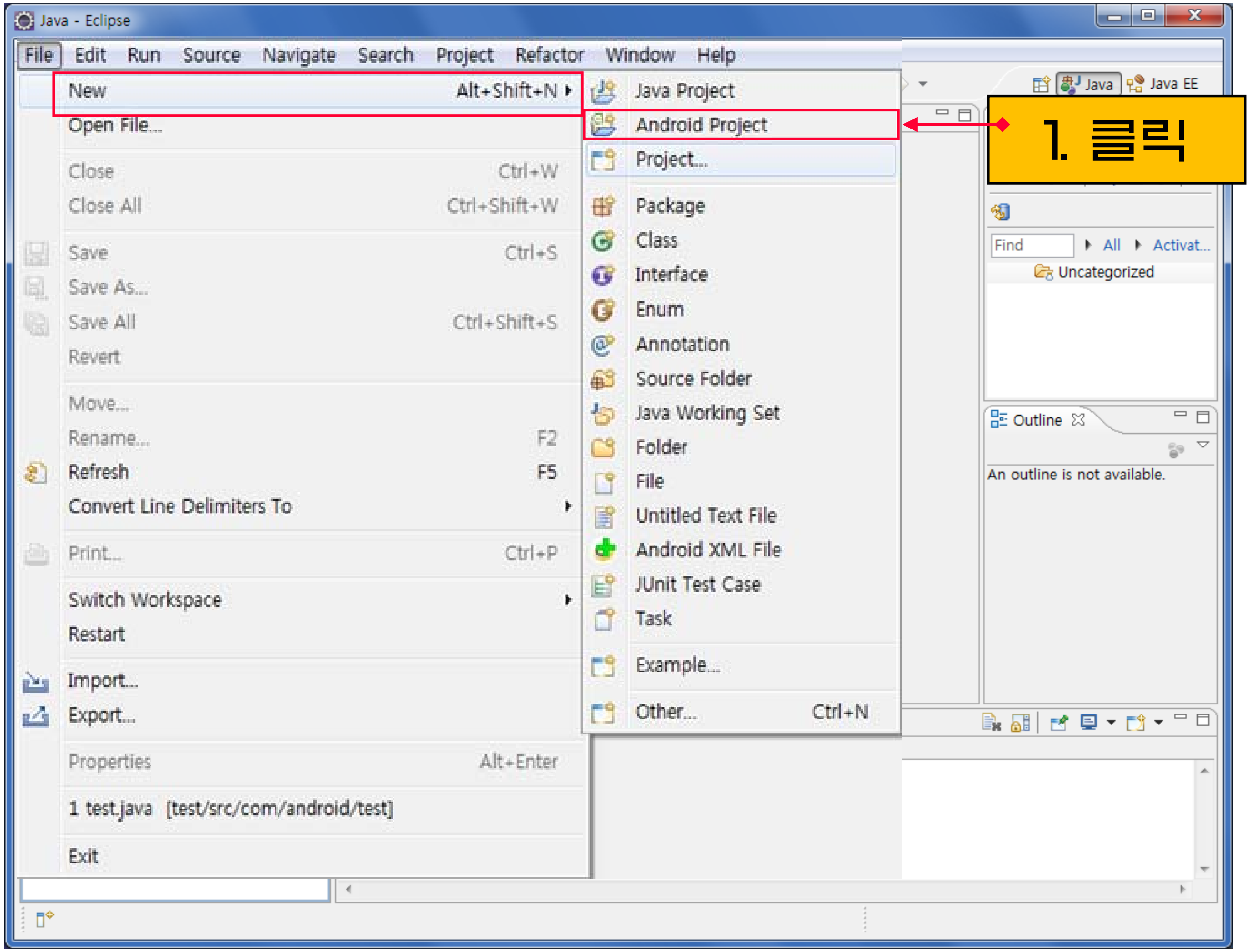
Android 지원 Video 형식

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

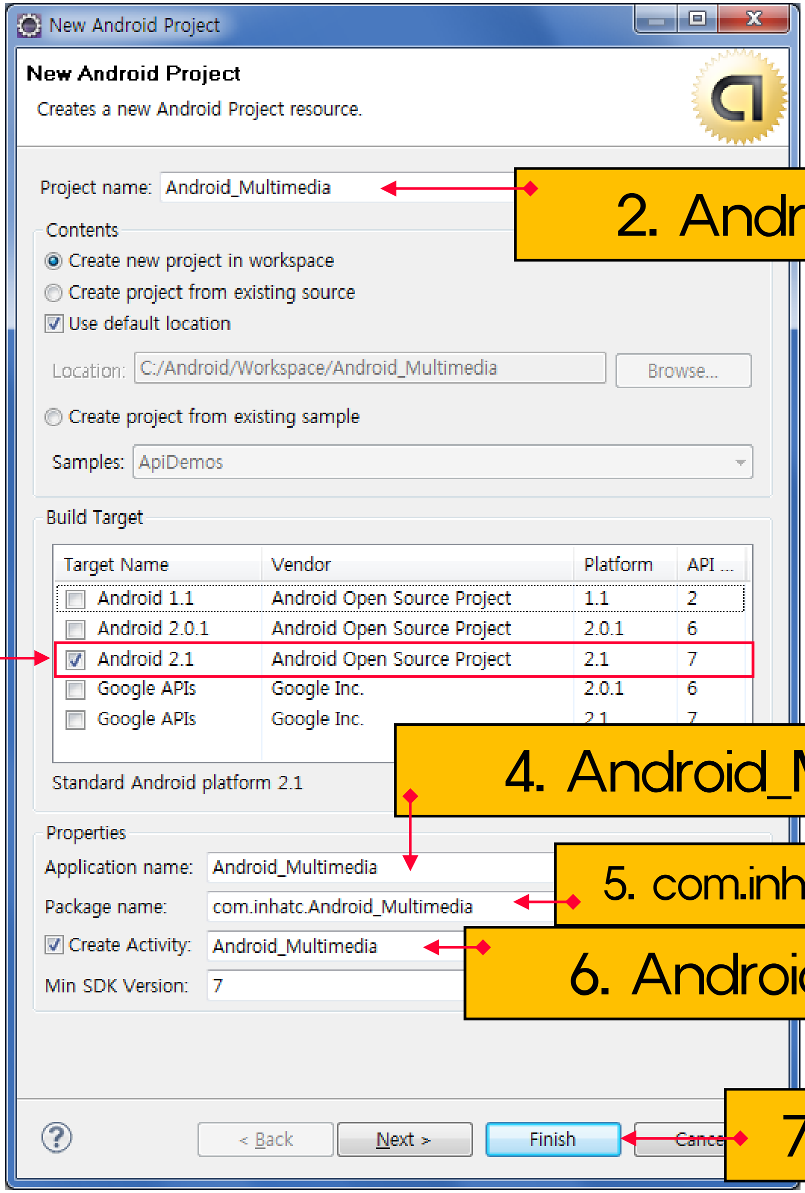
Android Audio 재생 구현 (1)

Android 프로젝트 생성

- 프로젝트 명 : Android_Multimedia



Android Audio 재생 구현 (2)



New Android Project

New Android Project

Creates a new Android Project resource.

Project name: Android_Multimedia

Contents

● Create new project in workspace

○ Create project from existing source

☑ Use default location

Location: C:/Android/Workspace/Android_Multimedia

○ Create project from existing sample

Samples: ApiDemos

Build Target

Target Name	Vendor	Platform	API ...
☐ Android 1.1	Android Open Source Project	1.1	2
☐ Android 2.0.1	Android Open Source Project	2.0.1	6
☑ Android 2.1	Android Open Source Project	2.1	7
☐ Google APIs	Google Inc.	2.0.1	6
☐ 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

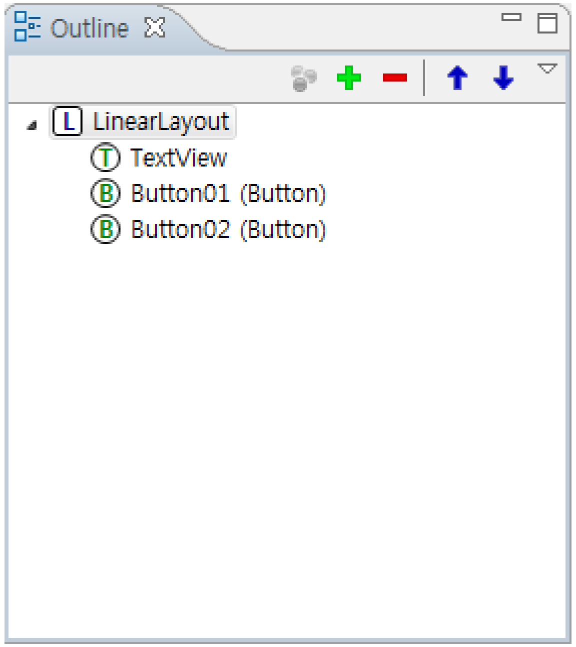
5. com.inhatc.android_Multimedia 입력

6. Android_Multimedia 입력

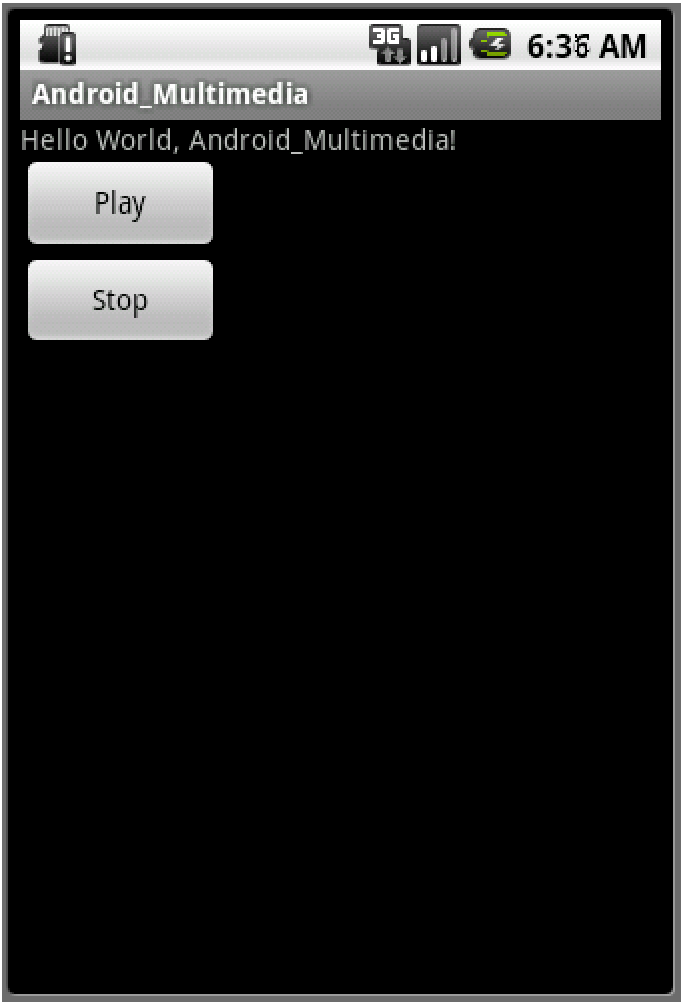
7. 클릭

Android Audio 재생 구현 (3)

UI 설계


Outline

- LinearLayout
  - TextView
  - Button01 (Button)
  - Button02 (Button)



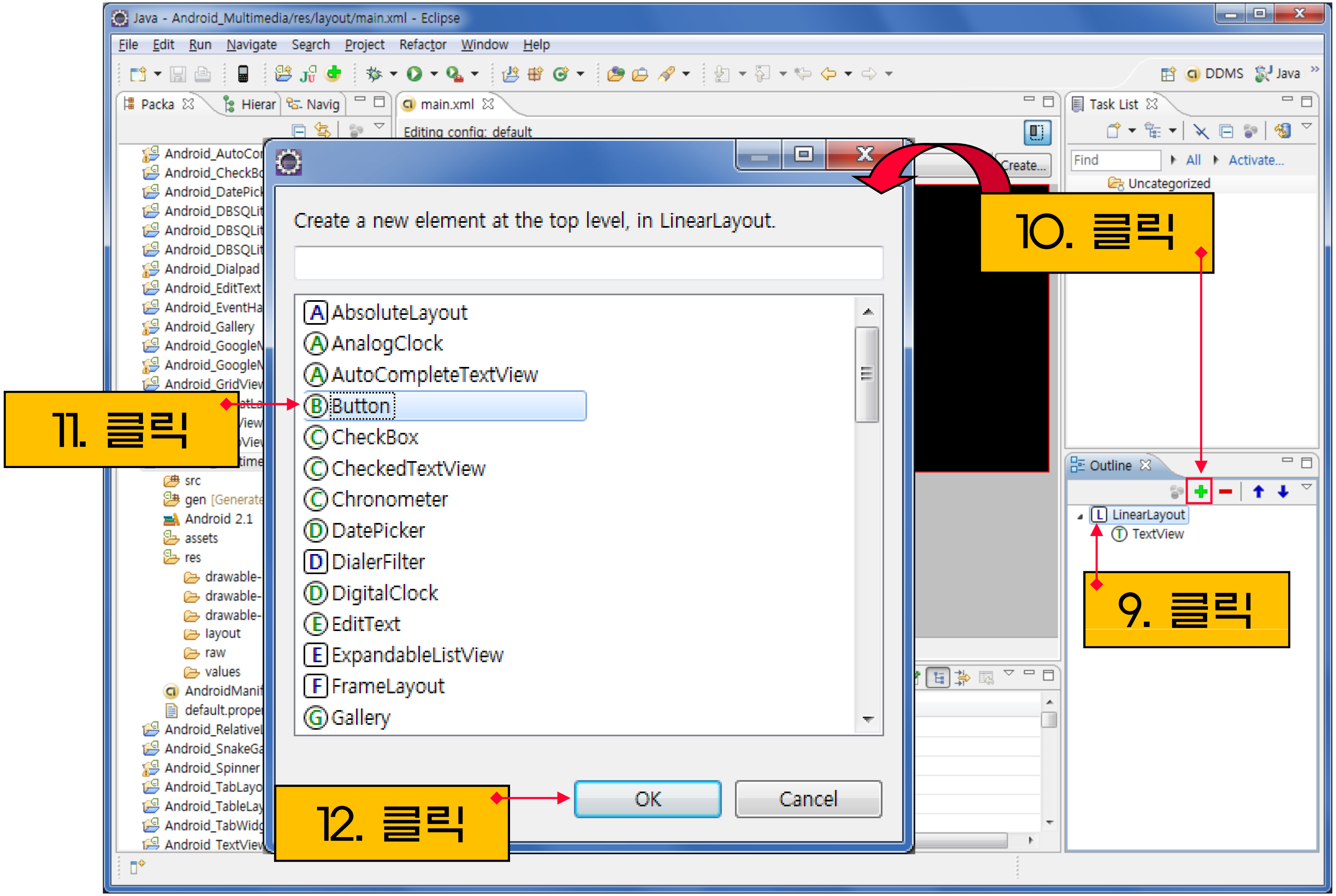
Android_Multimedia

Hello World, Android_Multimedia!

8. UI 설계 및 속성 지정

Android Audio 재생 구현 (4)

❖ Button 추가



Create a new element at the top level, in LinearLayout.

9. 클릭

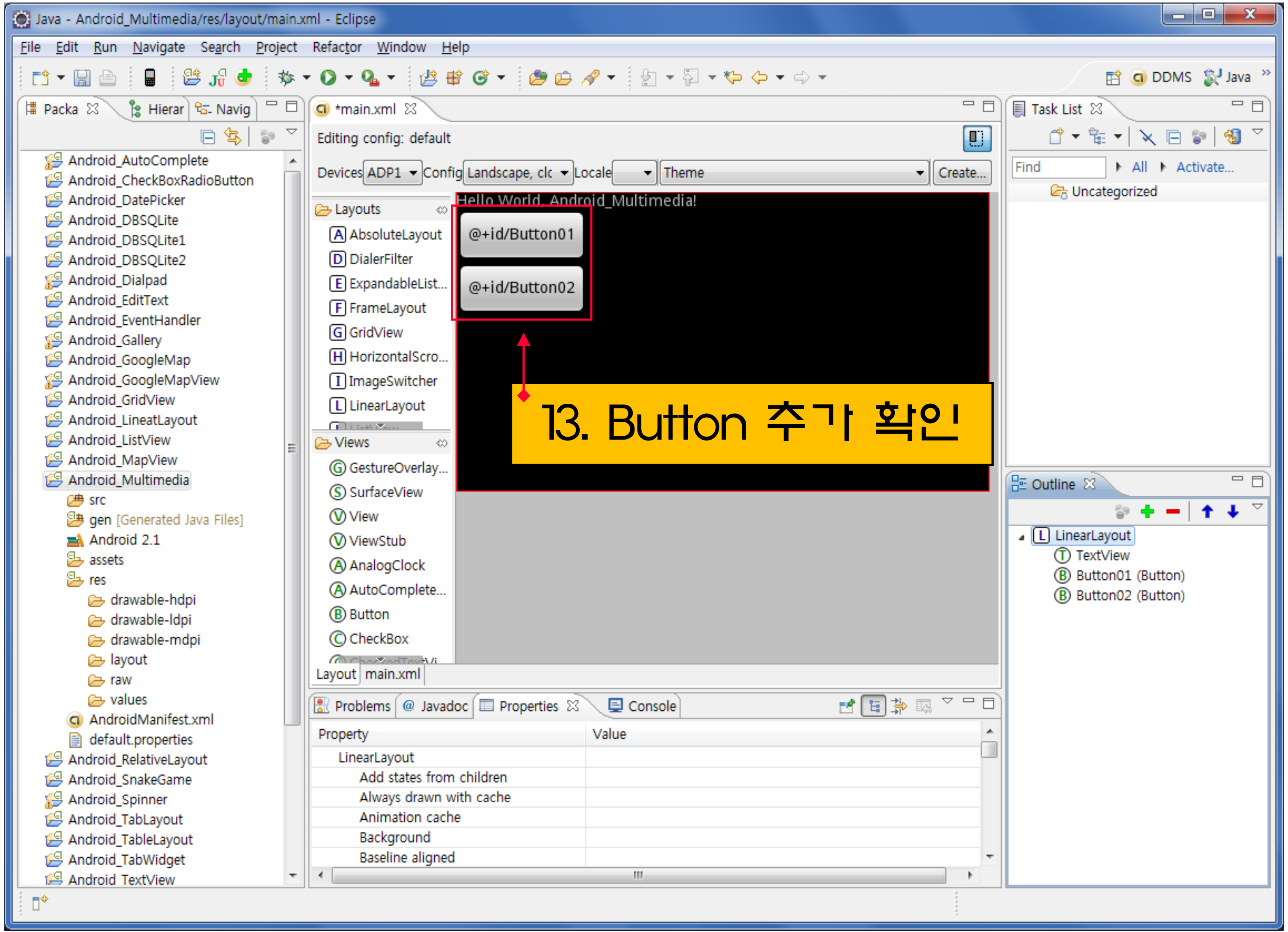
10. 클릭

11. 클릭

12. 클릭

Android Audio 재생 구현 (5)

❖ Button 추가 확인



13. Button 추가 확인

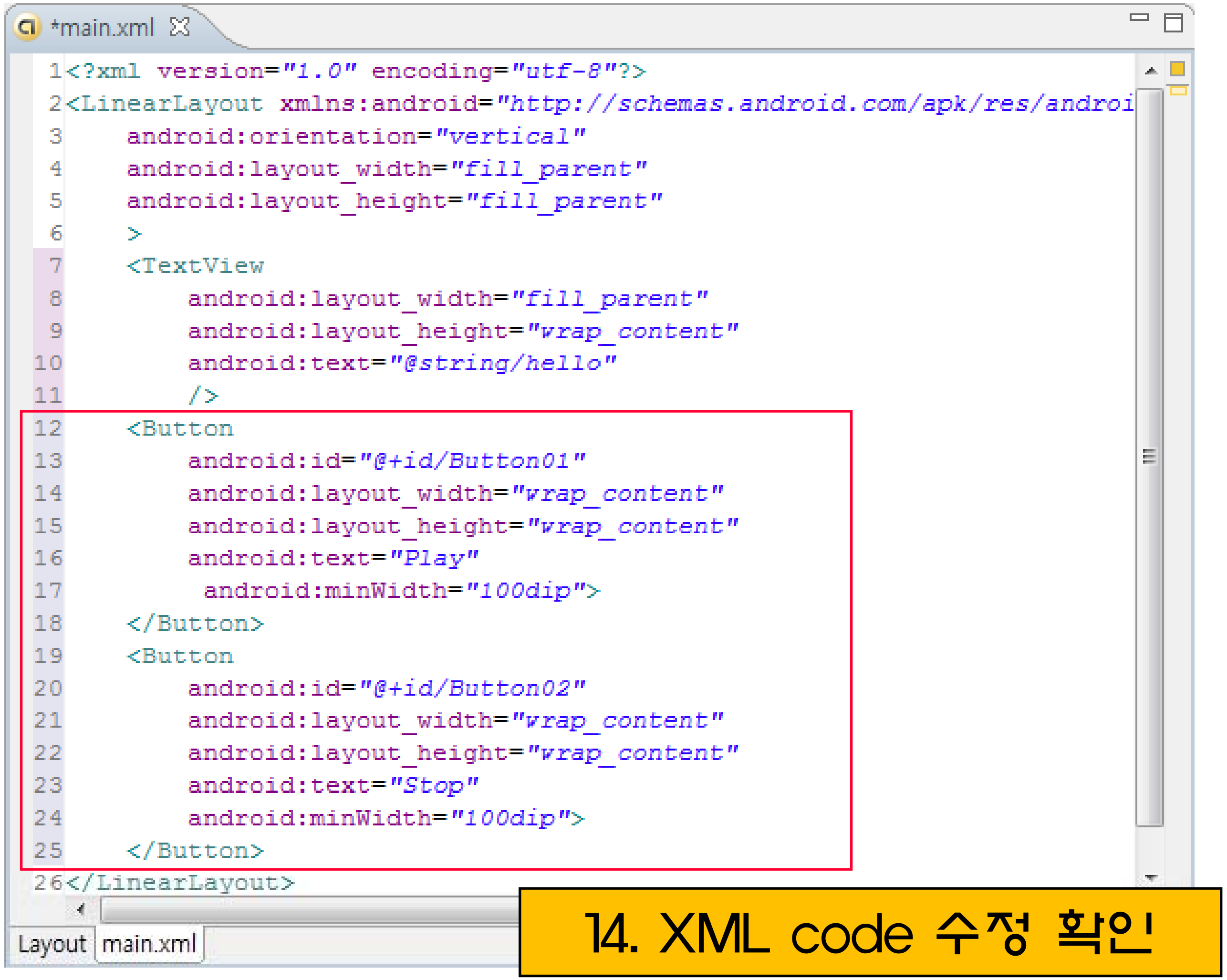
Android Audio 재생 구현 (6)

- TextView01, Button 01, 02 속성 지정

컨트롤	속성 지정
TextView01	• android:layout_width="fill_parent" • android:layout_height="wrap_content" • android:text="@string/hello"
Button01	• android:id="@+id/Button01" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:text="Play" • android:minWidth="100dip"
Button02	• android:id="@+id/Button02" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:text="Stop" • android:minWidth="100dip"

Android Audio 재생 구현 (7)

Main.xml 수정

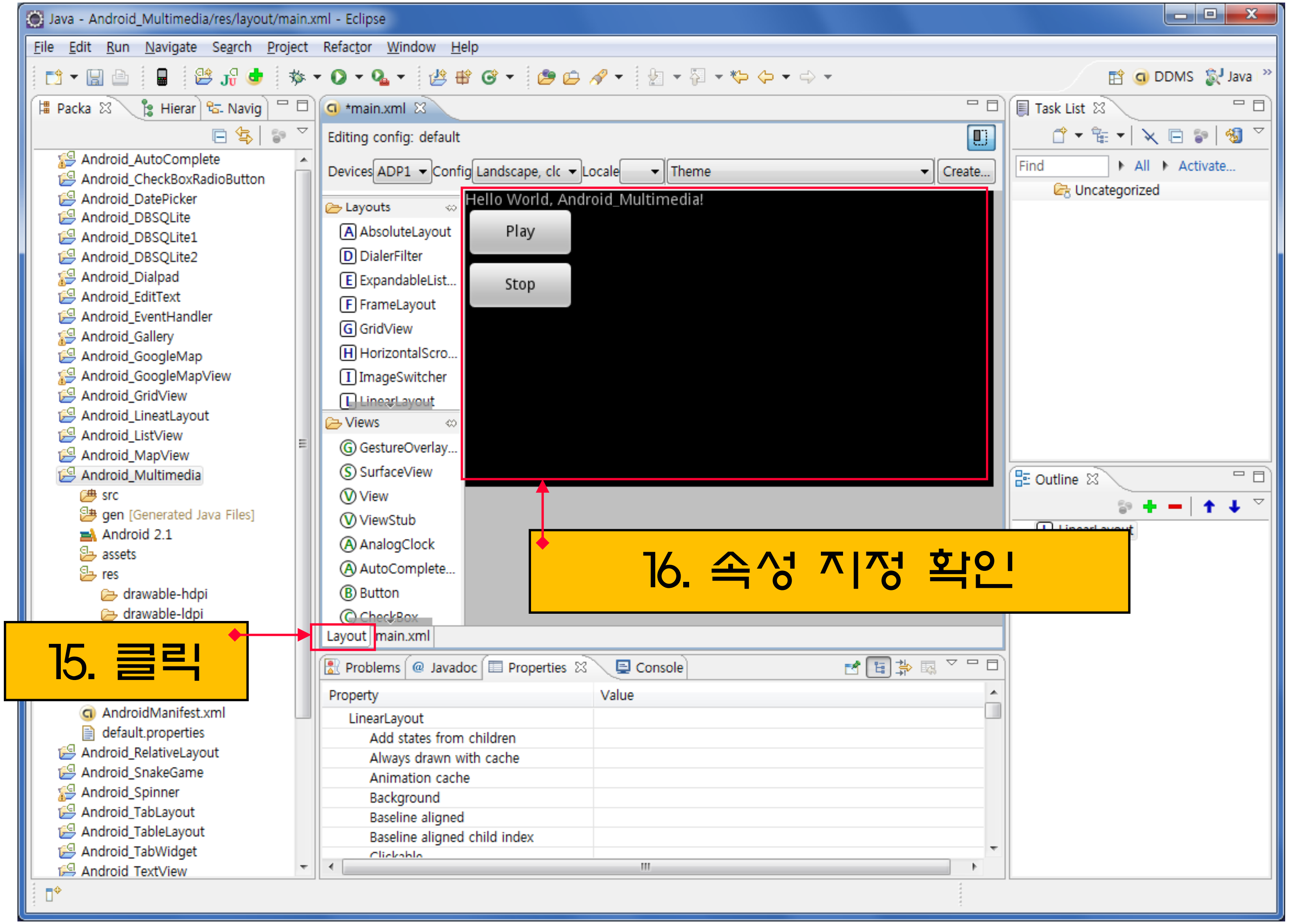


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/androi
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        />
    <Button
        android:id="@+id/Button01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play"
         android:minWidth="100dip">
    </Button>
    <Button
        android:id="@+id/Button02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Stop"
        android:minWidth="100dip">
    </Button>
</LinearLayout>
```

14. XML code 수정 확인!

Android Audio 재생 구현 (8)

❖ Main.xml 수정 확인!



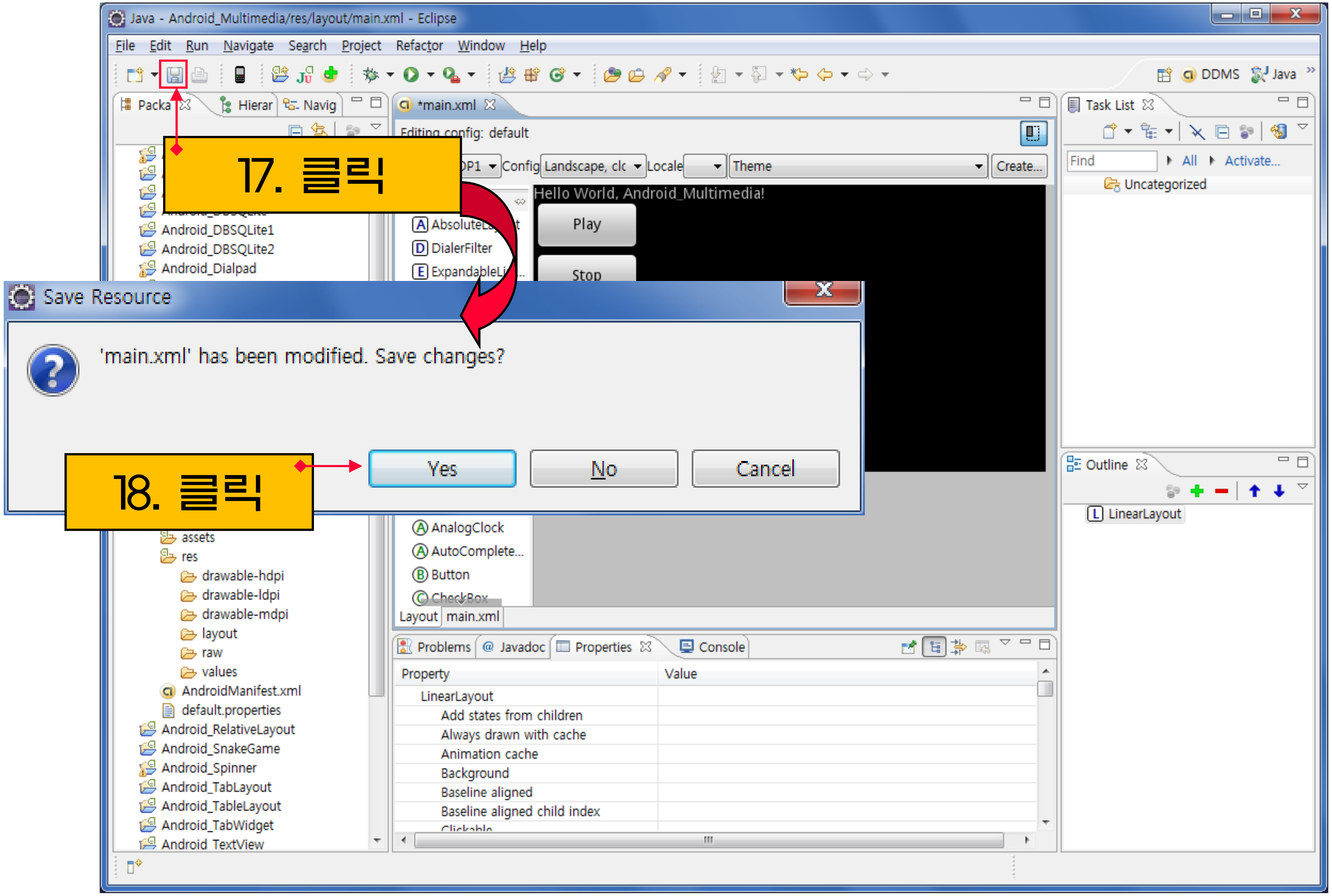
Property	Value
LinearLayout	
Add states from children	
Always drawn with cache	
Animation cache	
Background	
Baseline aligned	
Baseline aligned child index	

15. 클릭

16. 속성 지정 확인!

Android Audio 재생 구현 (9)

❖ Main.xml 저장



17. 클릭

18. 클릭

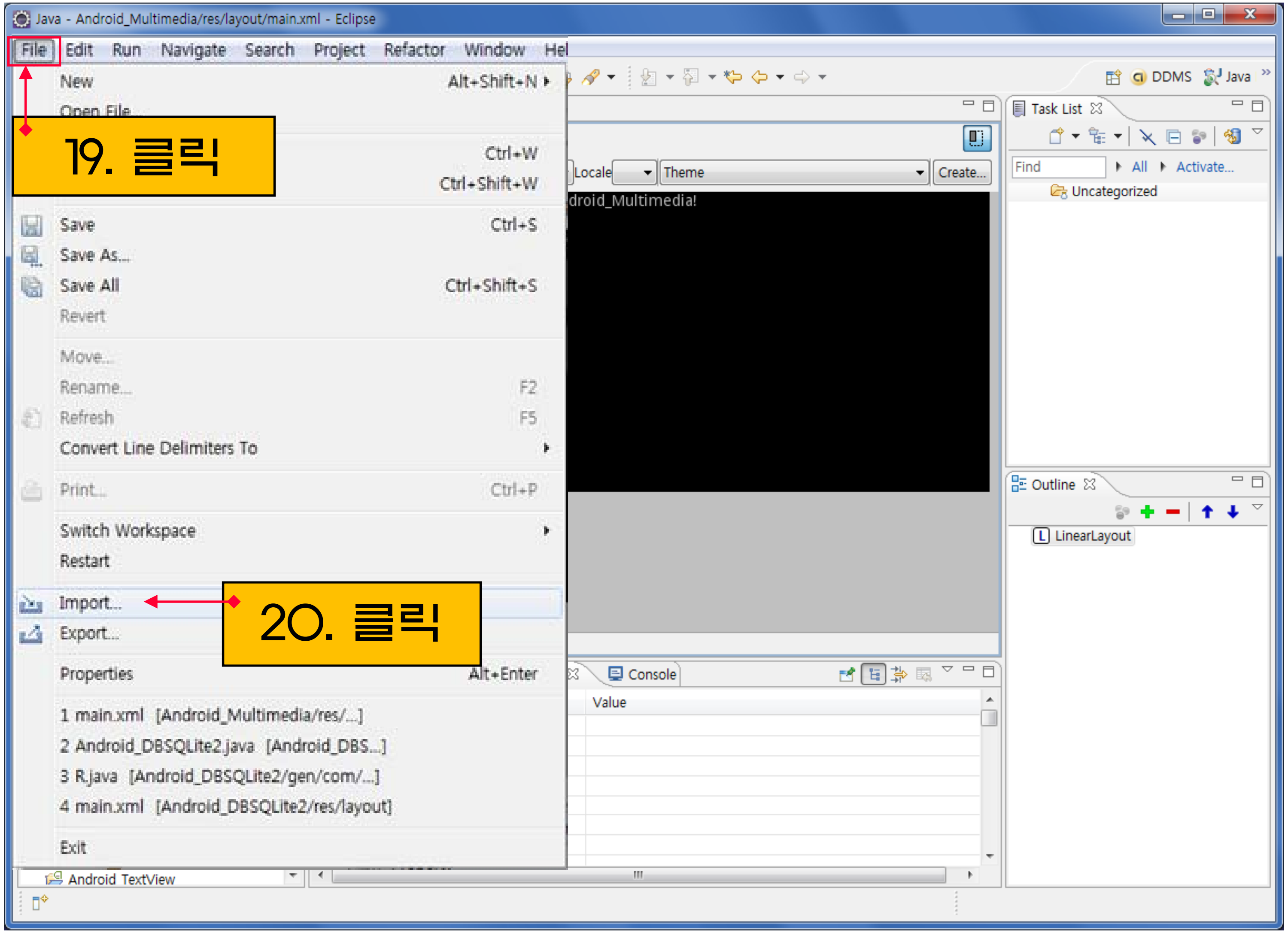
Save Resource

'main.xml' has been modified. Save changes?

Java - Android_Multimedia/res/layout/main.xml - Eclipse
File Edit Run Navigate Search Project Refactor Window Help
*main.xml
Editing config: default
Hello World, Android_Multimedia!
Play
Stop
Yes
No
Cancel
Task List
Outline
LinearLayout
Problems
Javadoc
Properties
Console
Property
Value

Android Audio 재생 구현 (10)

- MP3 파일 import

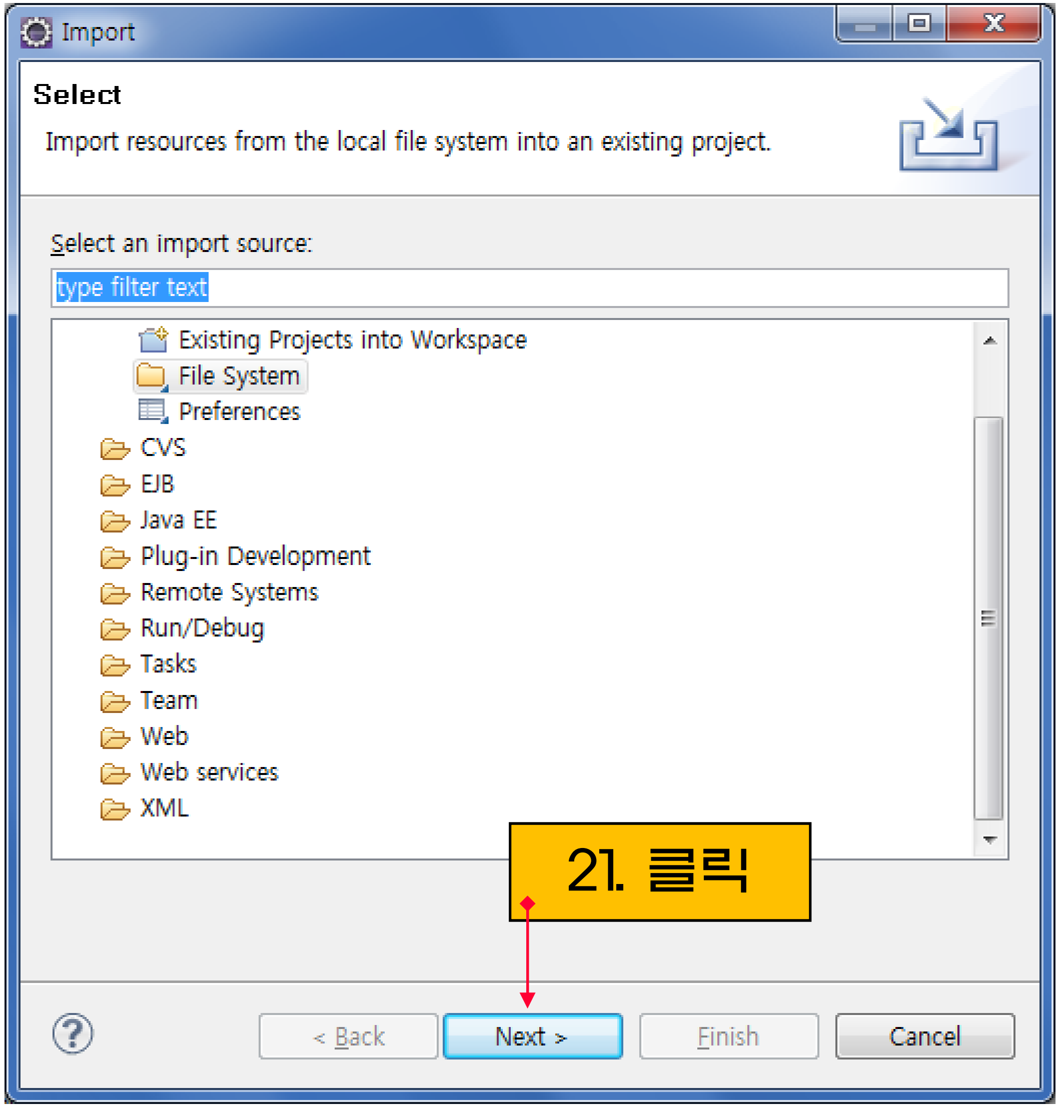


19. 클릭

20. 클릭

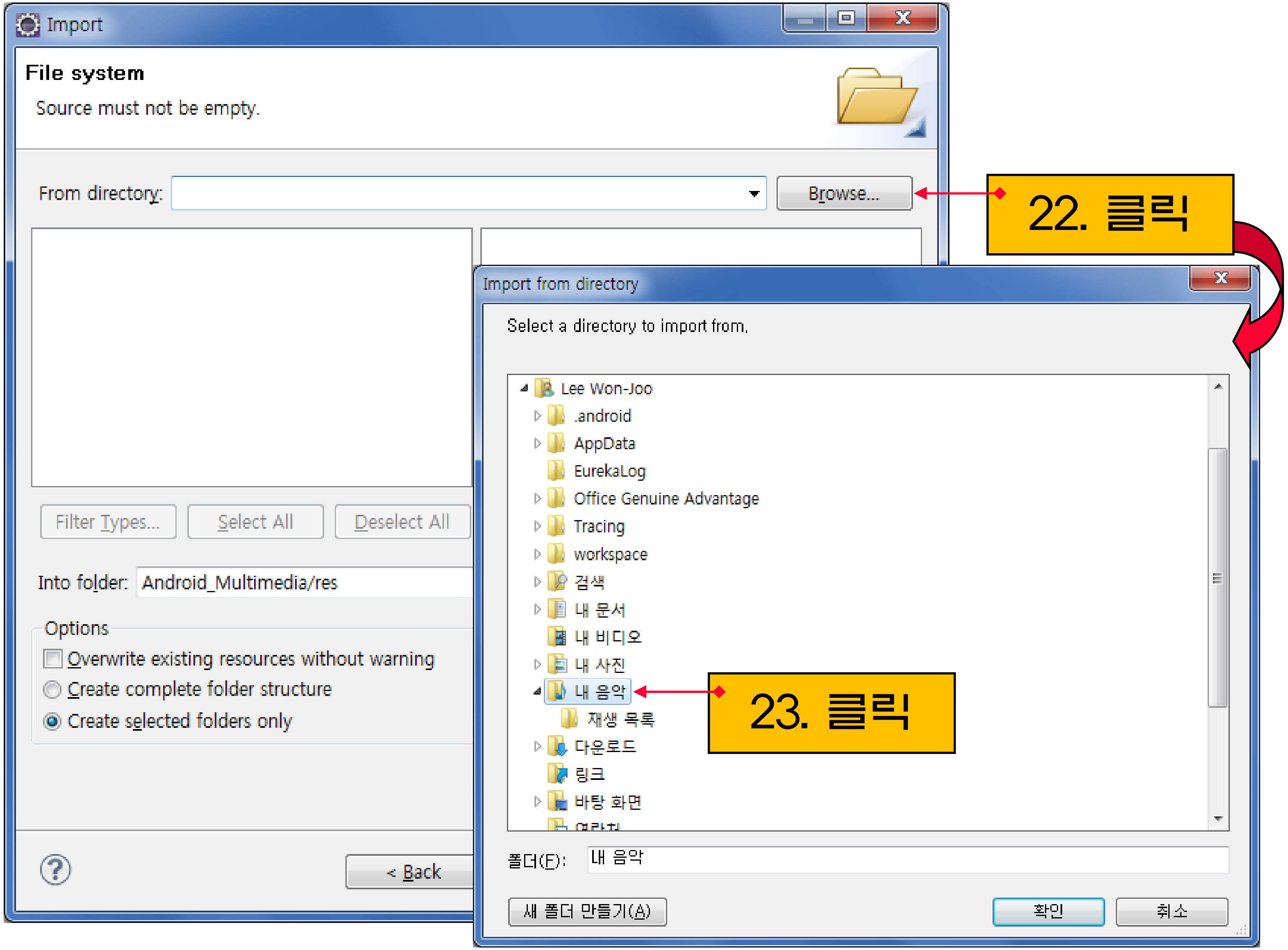
Android Audio 재생 구현 (11)

- Import 화면



21. 클릭

Android Audio 재생 구현 (12)



Import

File system

Source must not be empty.

From directory:

Into folder: Android_Multimedia/res

Options

☐ Overwrite existing resources without warning
○ Create complete folder structure
● Create selected folders only

Import from directory

Select a directory to import from.

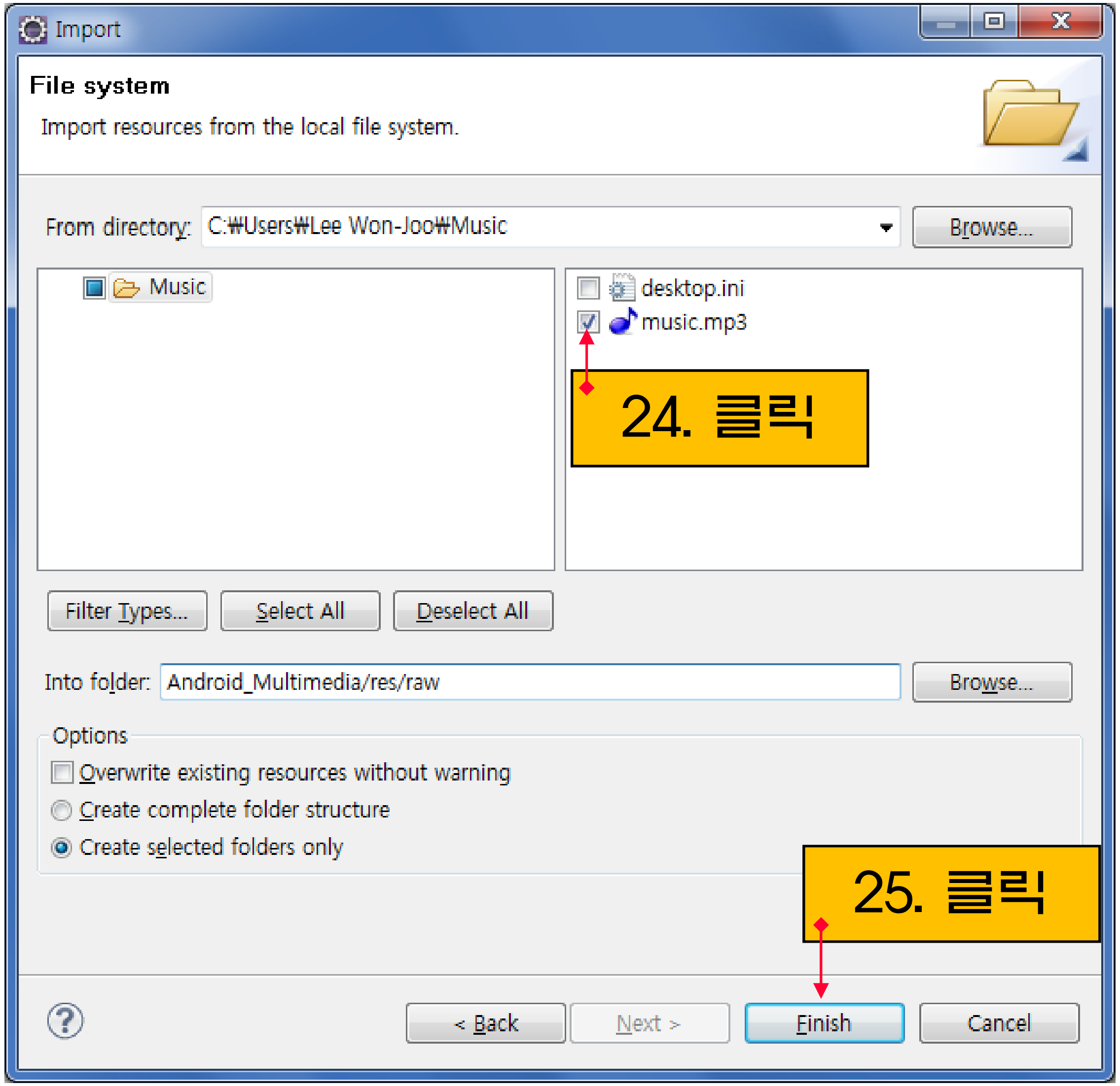
폴더(F): 내 음악

22. 클릭

23. 클릭

Android Audio 재생 구현 (13)

❖ MP3 파일 선택



Import

File system

Import resources from the local file system.

From directory: C:₩Users₩Lee Won-Joo₩Music

☐ desktop.ini

☑ music.mp3

Into folder: Android_Multimedia/res/raw

Options

☐ Overwrite existing resources without warning

○ Create complete folder structure

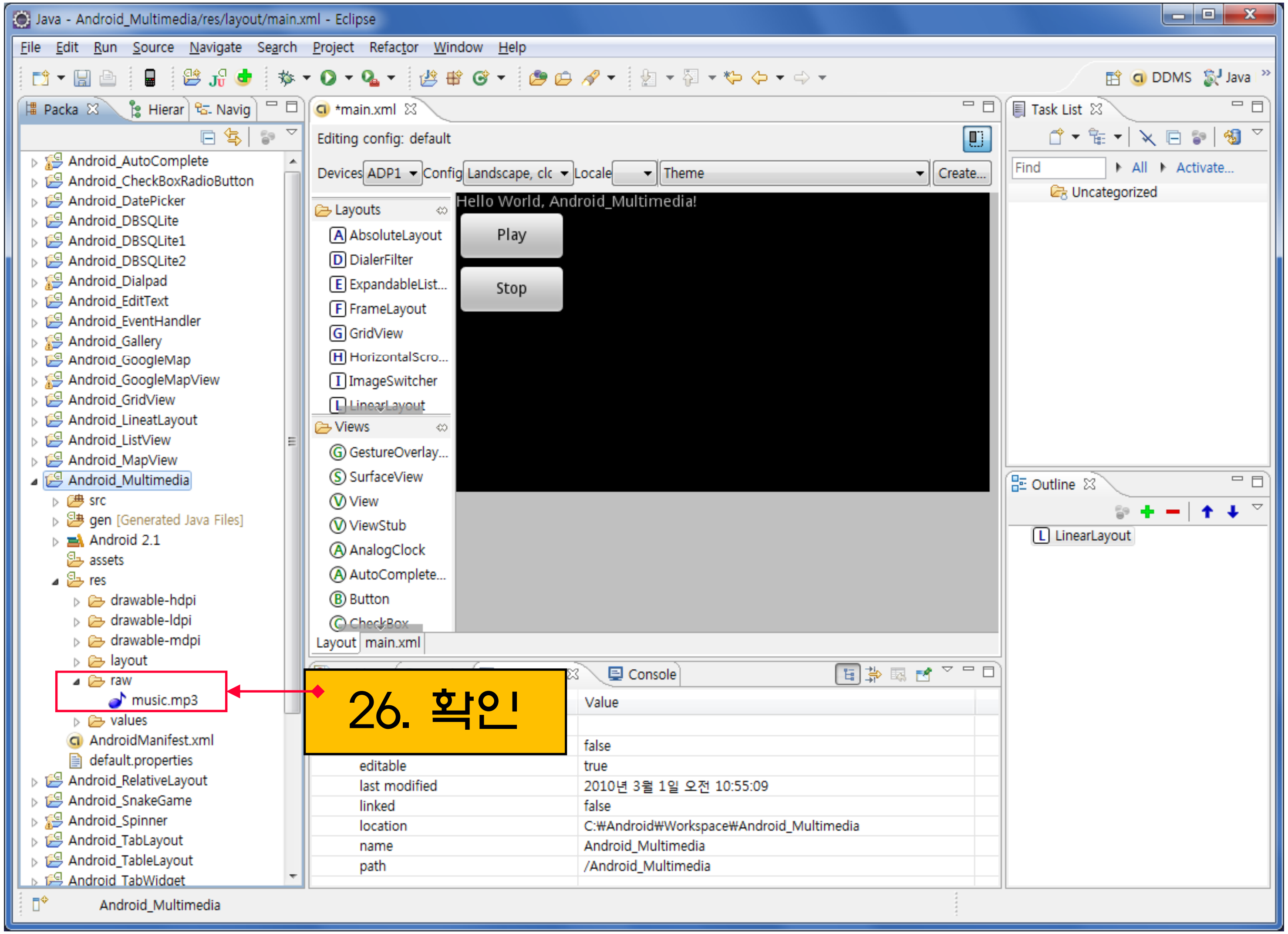
● Create selected folders only

24. 클릭

25. 클릭

Android Audio 재생 구현 (14)

❖ MP3 파일 import 결과 확인!



Editing config: default

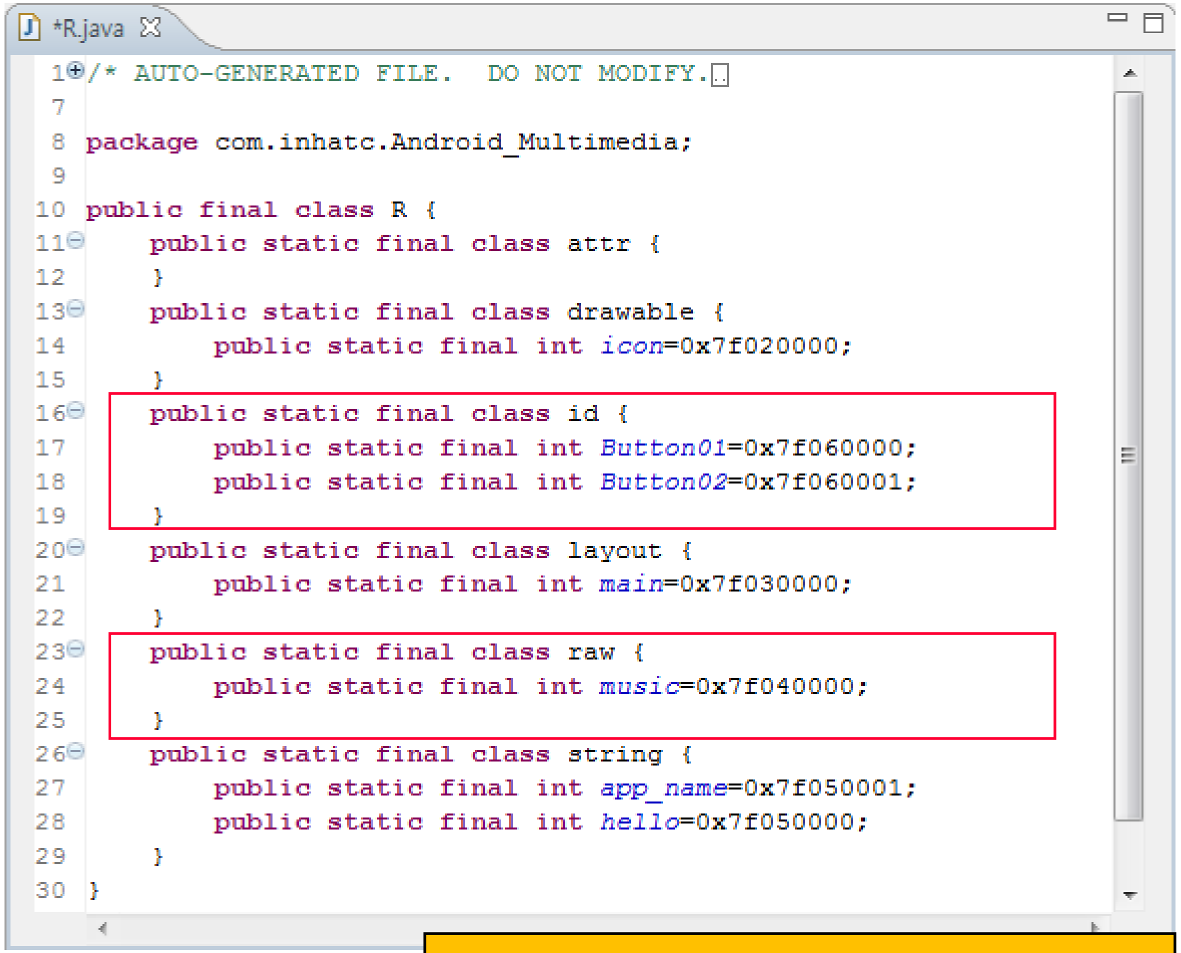
Devices ADP1 Config Landscape, clc Locale Theme Create...

	Value
	false
editable	true
last modified	2010년 3월 1일 오전 10:55:09
linked	false
location	C:₩Android₩Workspace₩Android_Multimedia
name	Android_Multimedia
path	/Android_Multimedia

26. 확인!

Android Audio 재생 구현 (15)

❖ R.java

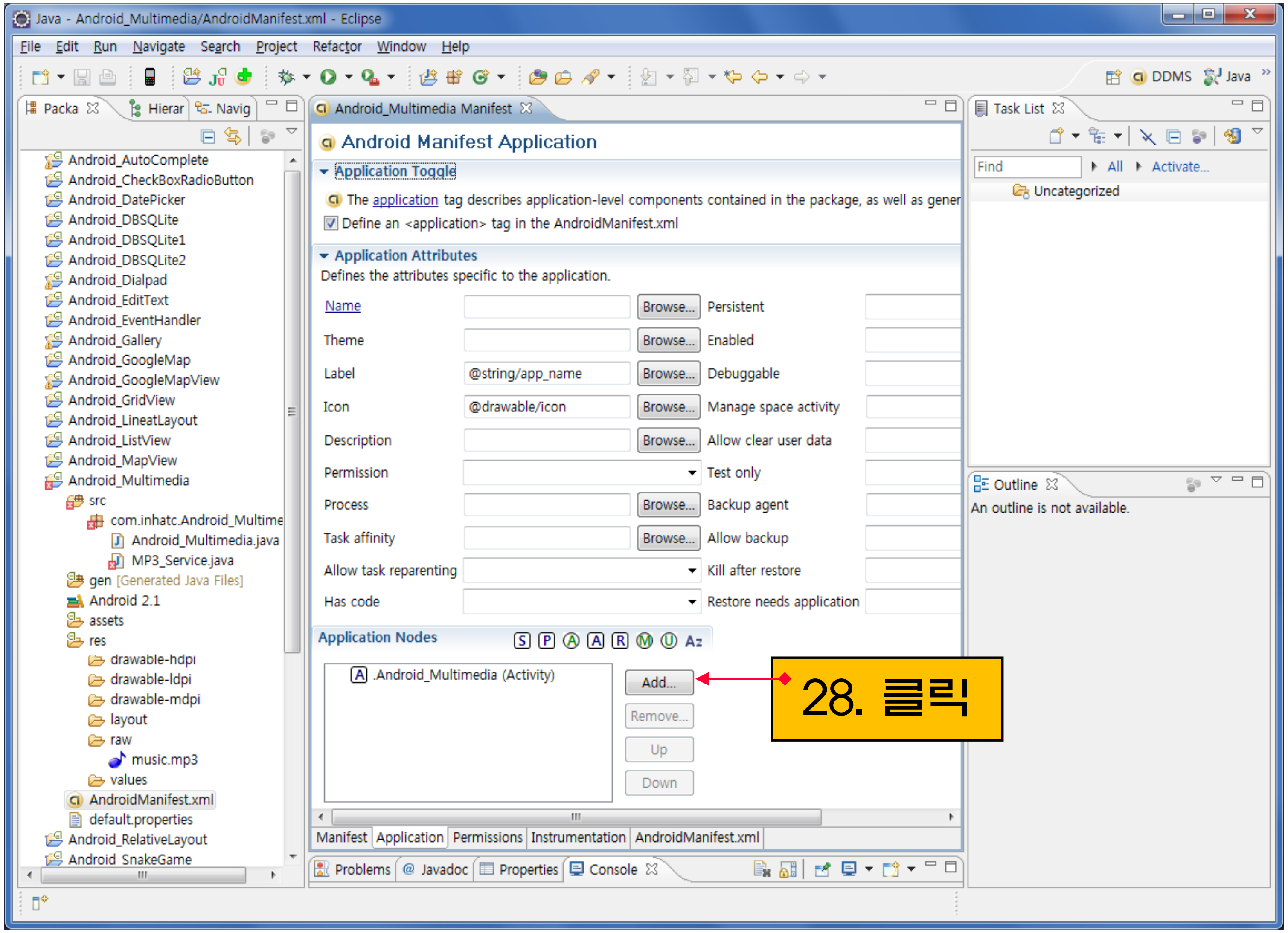


```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.

package com.inhatc.Android_Multimedia;

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 Button01=0x7f060000;
        public static final int Button02=0x7f060001;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class raw {
        public static final int music=0x7f040000;
    }
    public static final class string {
        public static final int app_name=0x7f050001;
        public static final int hello=0x7f050000;
    }
}
```

27. id 클래스 변수 추가 확인

Android Audio 재생 구현 (16)

- AndroidManifest.xml 수정



Java - Android_Multimedia/AndroidManifest.xml - Eclipse

Android Manifest Application

Application Toggle

The application tag describes application-level components contained in the package, as well as gener

☑ Define an <application> tag in the AndroidManifest.xml

Application Attributes

Defines the attributes specific to the application.

Label: @string/app_name

Icon: @drawable/icon

Application Nodes

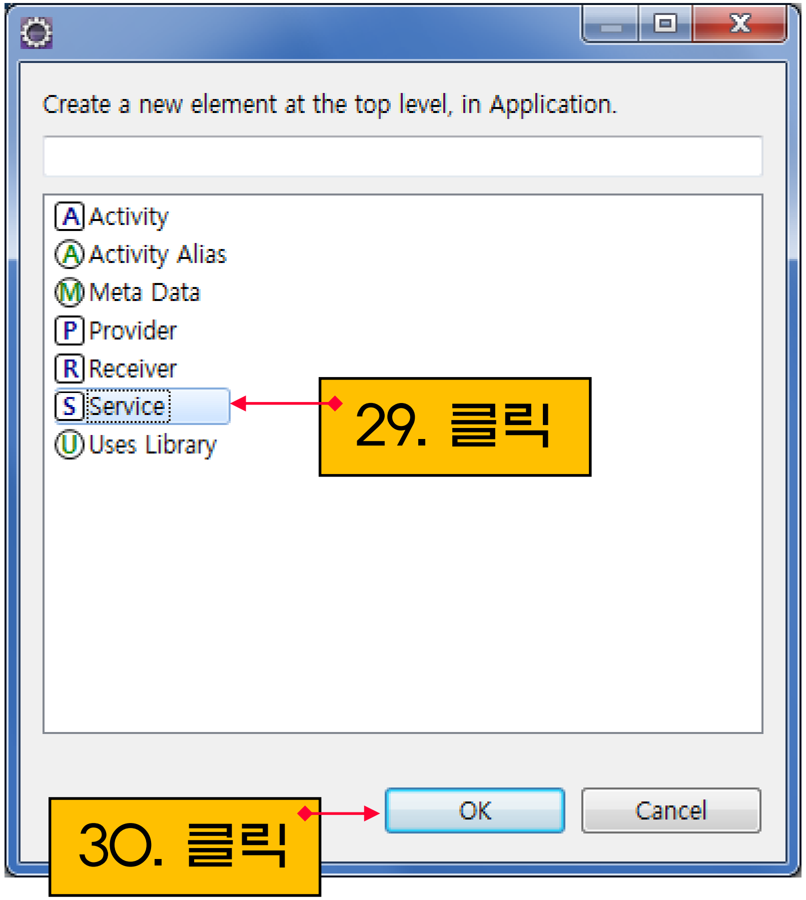
A Android_Multimedia (Activity)

An outline is not available.

28. 클릭

Android Audio 재생 구현 (17)

❖ Service 추가



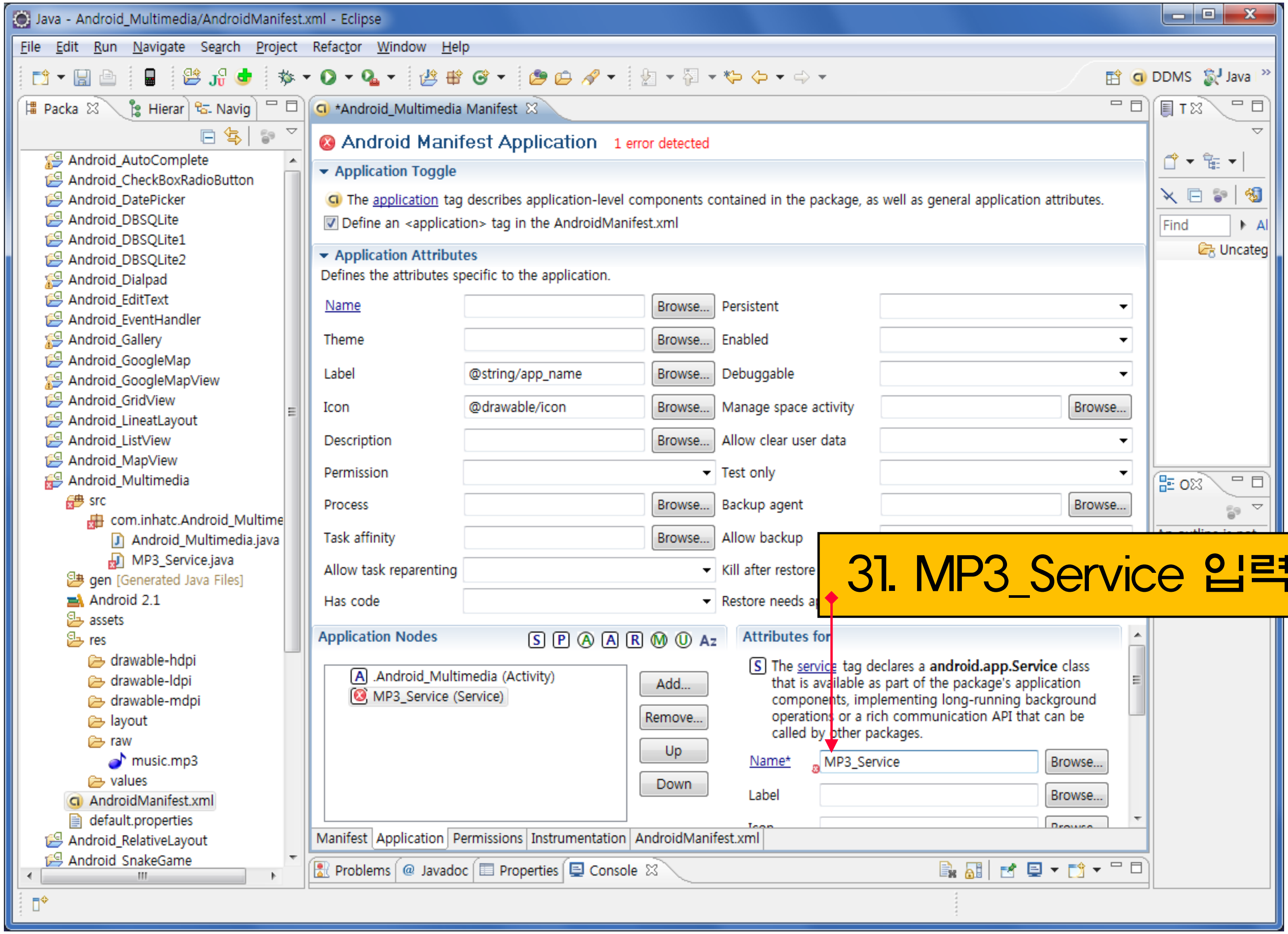
Create a new element at the top level, in Application.

29. 클릭

30. 클릭

Android Audio 재생 구현 (18)

- AndroidManifest.xml 수정



Java - Android_Multimedia/AndroidManifest.xml - Eclipse

*Android_Multimedia Manifest

Android Manifest Application 1 error detected

Application Toggle

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

Application Attributes

Defines the attributes specific to the application.

Label: @string/app_name

Icon: @drawable/icon

Application Nodes

Android_Multimedia (Activity)

MP3_Service (Service)

Attributes for

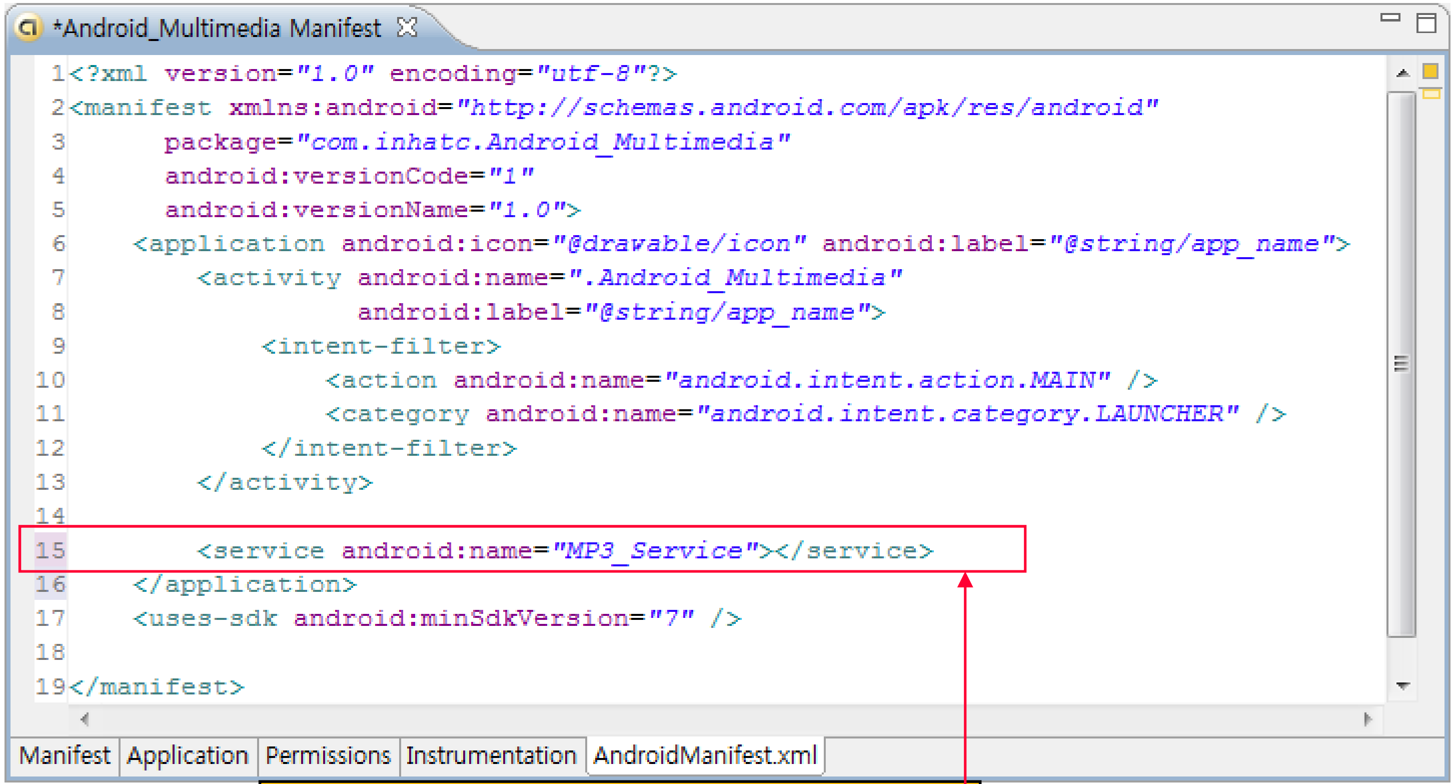
The service tag declares a **android.app.Service** class that is available as part of the package's application components, implementing long-running background operations or a rich communication API that can be called by other packages.

Name*: MP3_Service

31. MP3_Service 입력

Android Audio 재생 구현 (19)

❖ AndroidManifest.xml 수정 확인!

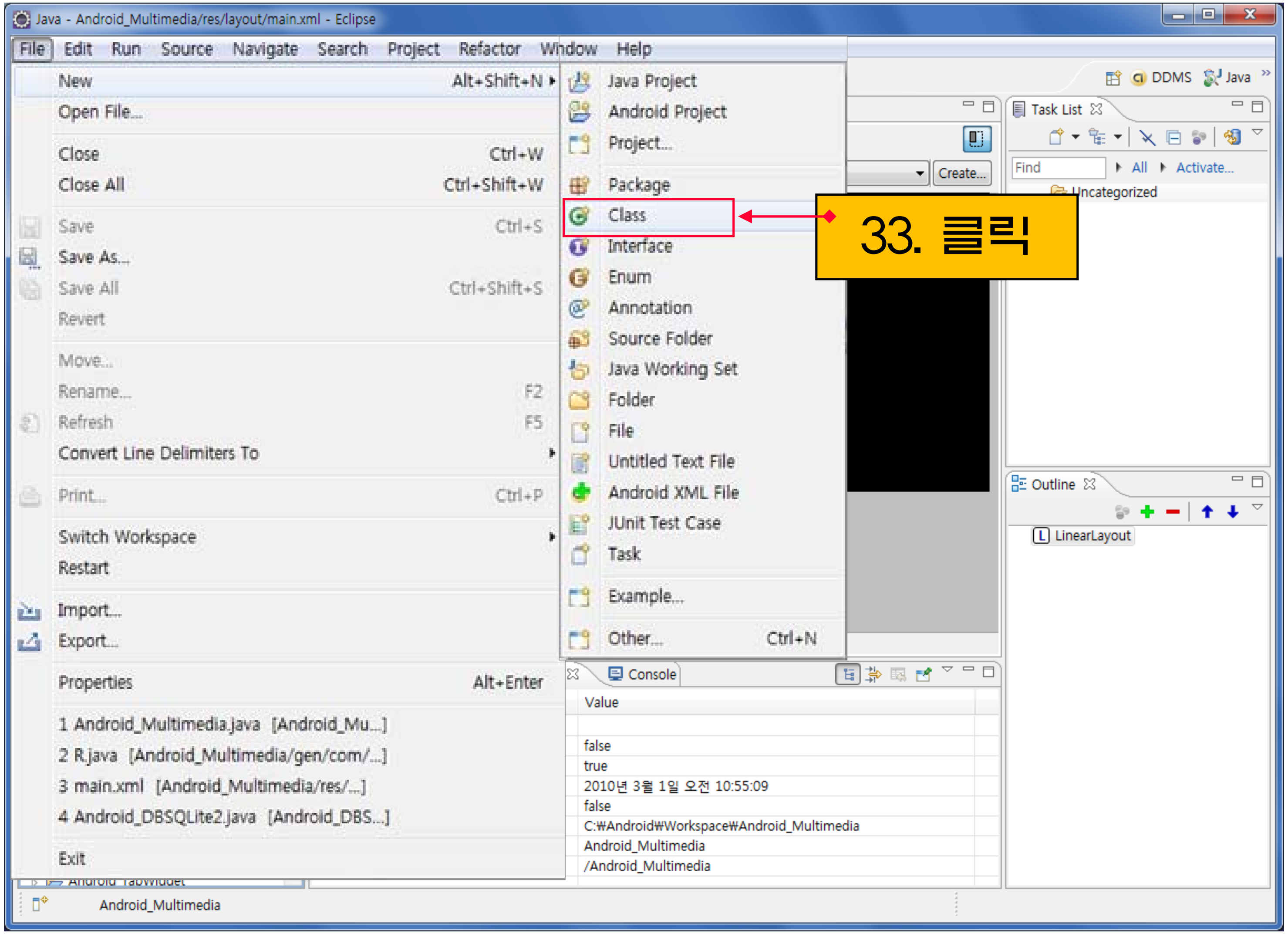


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.inhatc.Android_Multimedia"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Android_Multimedia"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service android:name="MP3_Service"></service>
    </application>
    <uses-sdk android:minSdkVersion="7" />

</manifest>
```

32. MP3_Service 입력 확인!

Android Audio 재생 구현 (20)

❖ Class 추가

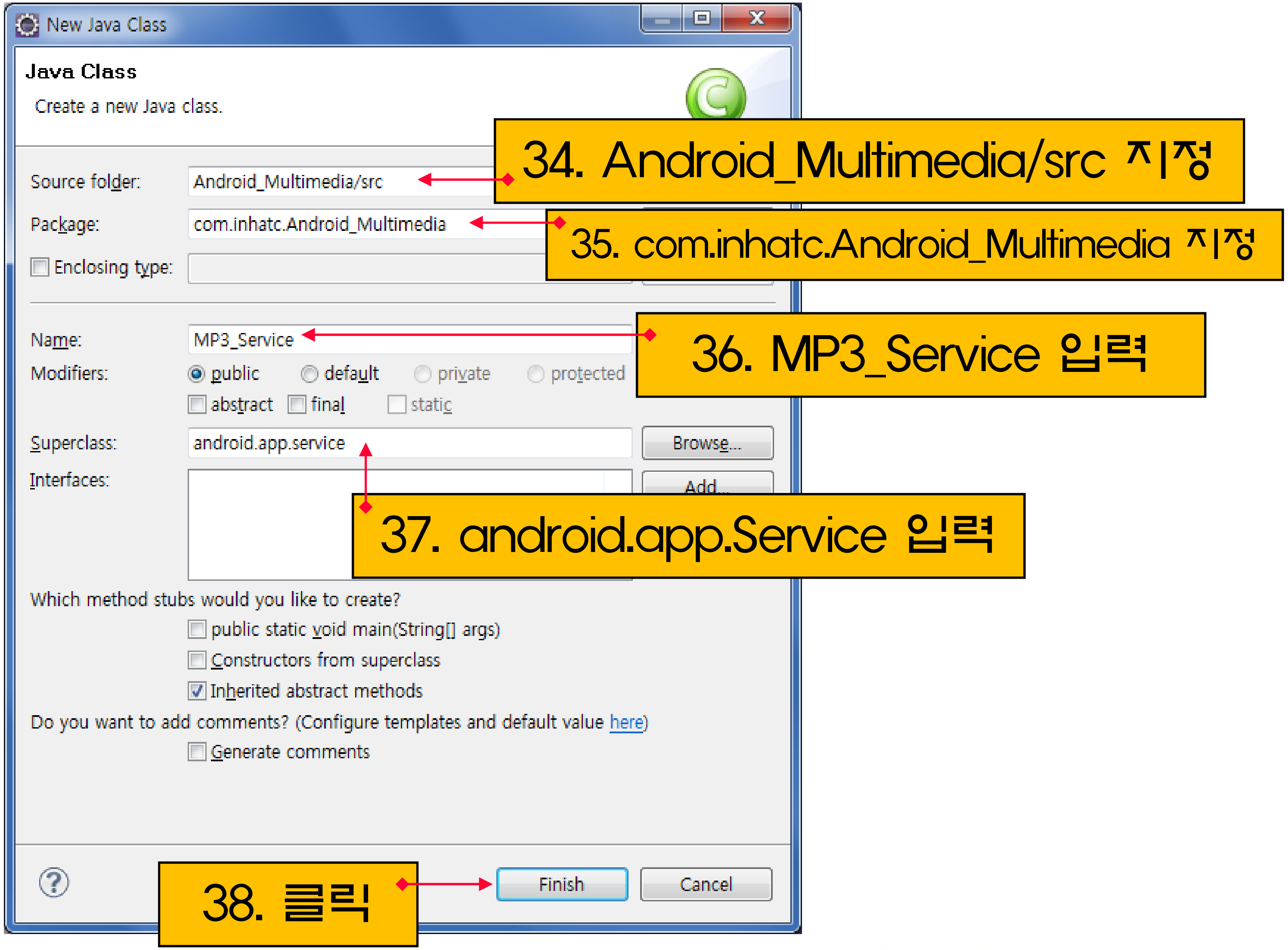


```
false
true
2010년 3월 1일 오전 10:55:09
false
C:₩Android₩Workspace₩Android_Multimedia
Android_Multimedia
/Android_Multimedia
```

33. 클릭

Android Audio 재생 구현 (21)

❖ MP3_Service Class 추가



Java Class

Create a new Java class.

Source folder: Android_Multimedia/src

Package: com.inhatc.Android_Multimedia

☐ Enclosing type:

Name: MP3_Service

Modifiers: ● public ○ default ○ private ○ protected

☐ abstract ☐ final ☐ static

Superclass: android.app.service

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

34. Android_Multimedia/src 지정

35. com.inhatc.Android_Multimedia 지정

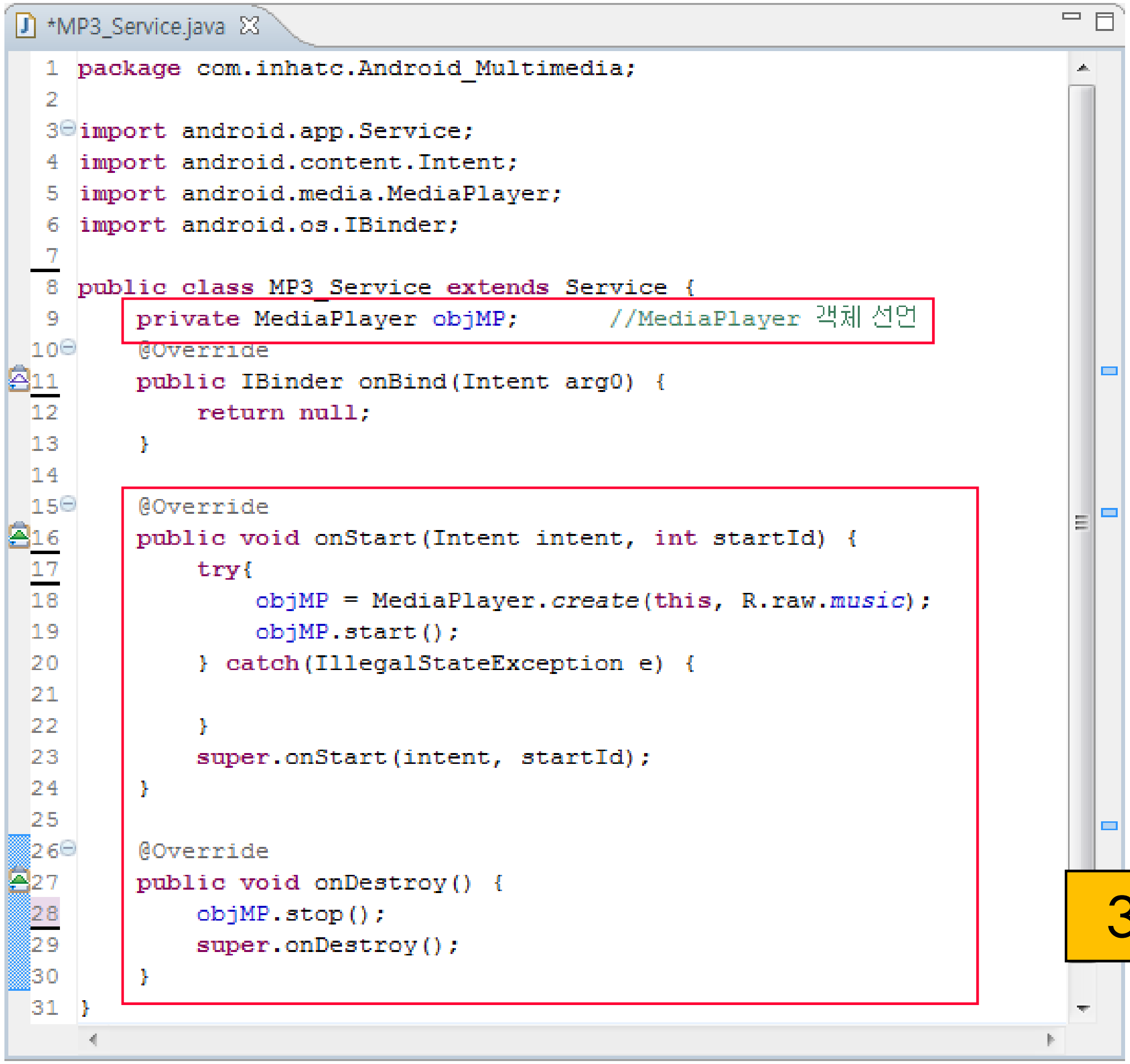
36. MP3_Service 입력

37. android.app.Service 입력

38. 클릭

Android Audio 재생 구현 (22)

MP3_Service.java

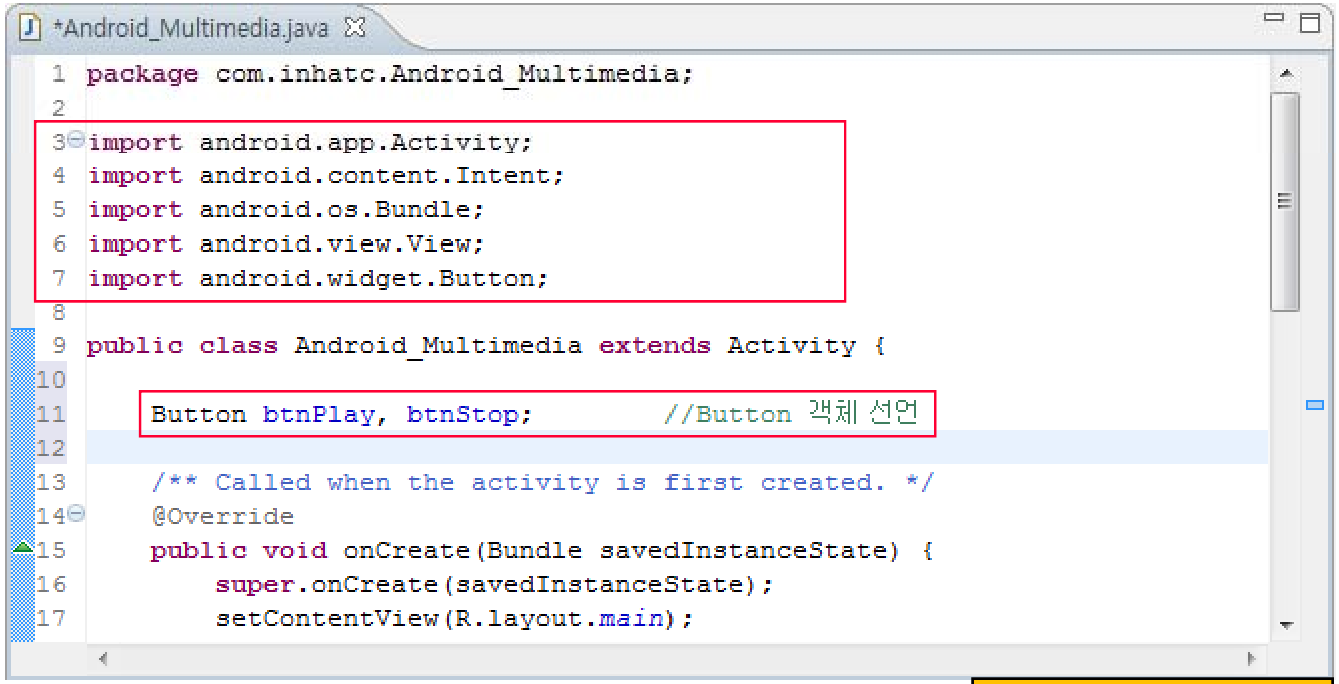


```
package com.inhatc.Android_Multimedia;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;

public class MP3_Service extends Service {
    private MediaPlayer objMP;       //MediaPlayer 객체 선언
    @Override
    public IBinder onBind(Intent arg0) {
        return null;
    }

    @Override
    public void onStart(Intent intent, int startId) {
        try{
            objMP = MediaPlayer.create(this, R.raw.music);
            objMP.start();
        } catch(IllegalStateException e) {

        }
        super.onStart(intent, startId);
    }

    @Override
    public void onDestroy() {
        objMP.stop();
        super.onDestroy();
    }
}
```

39. Coding

Android Audio 재생 구현 (23)

Android_MultiMedia.java

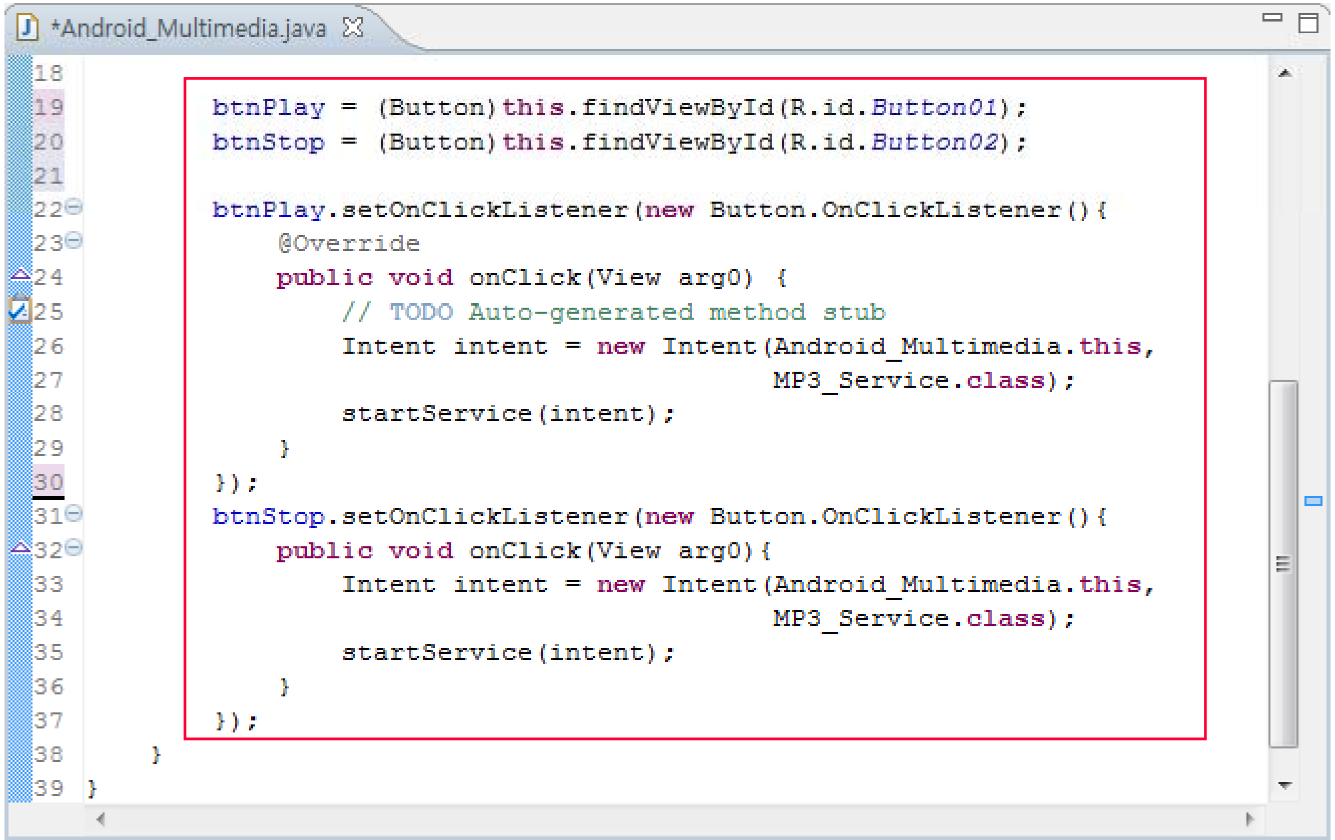


```
package com.inhatc.Android_Multimedia;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Android_Multimedia extends Activity {

    Button btnPlay, btnStop;        //Button 객체 선언

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
```

40. Coding

Android Audio 재생 구현 (24)

❖ Button Event Handler

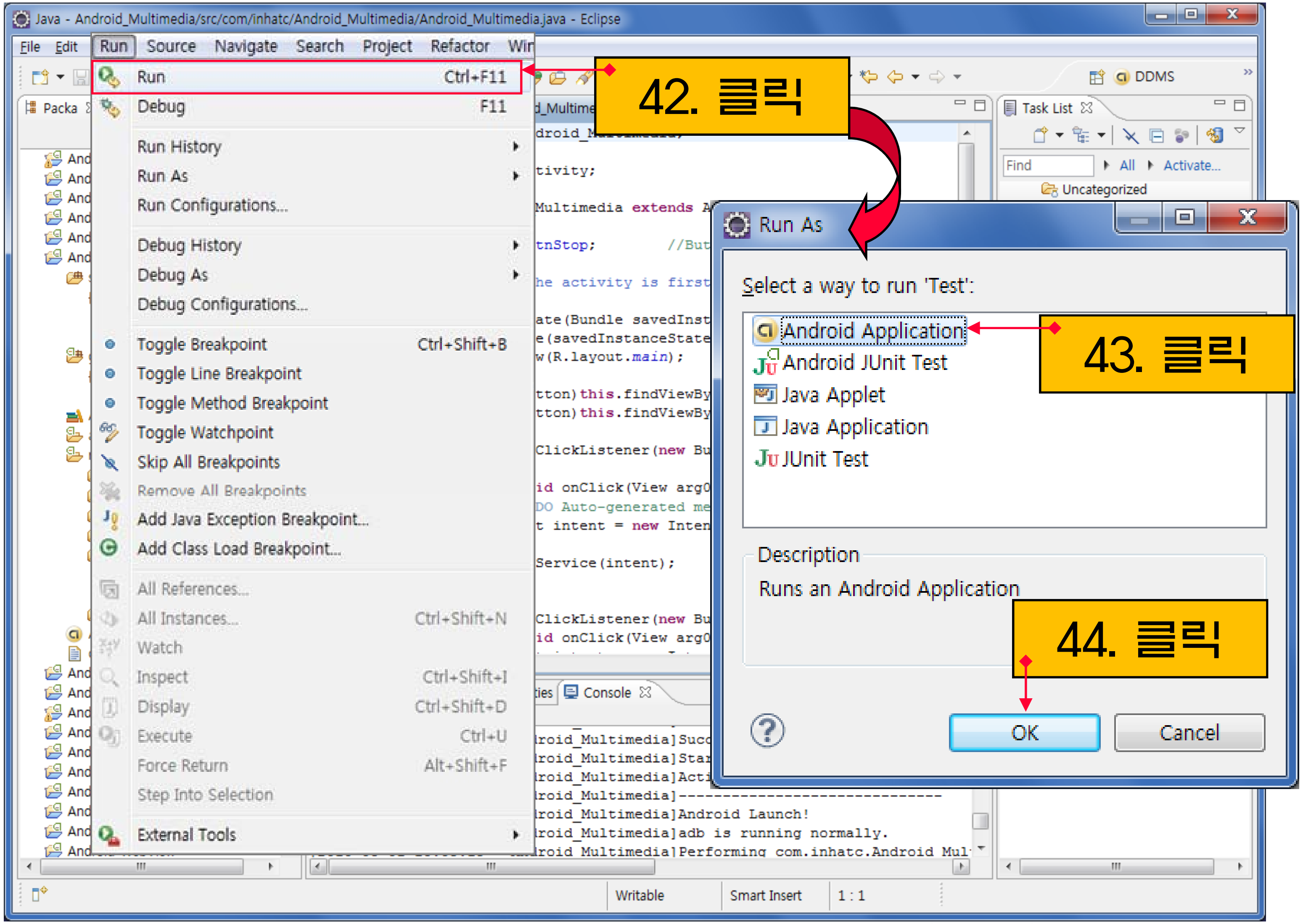


```java

        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



Select a way to run 'Test':

Description

Runs an Android Application

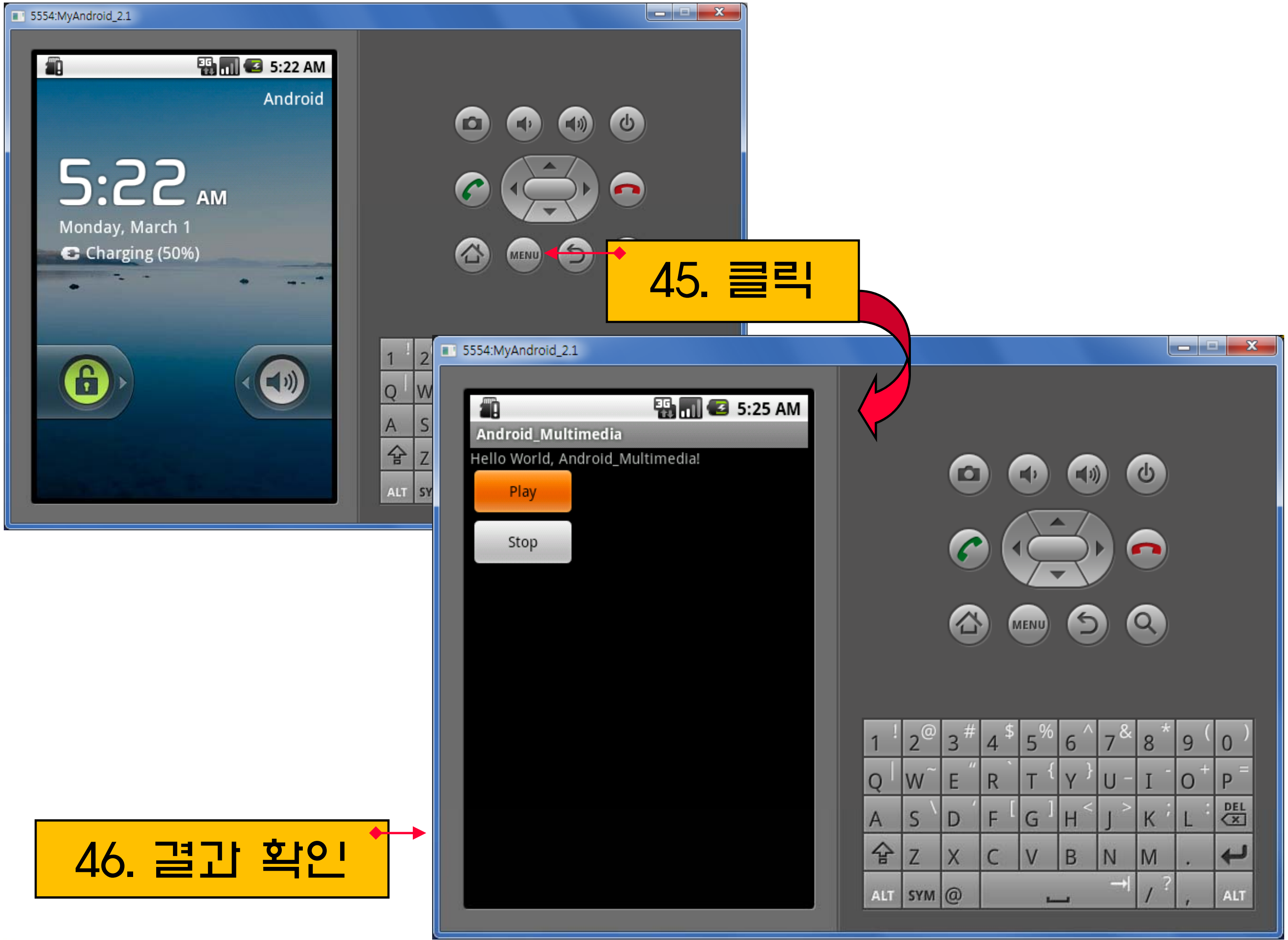
42. 클릭

43. 클릭

44. 클릭

Android Audio 재생 구현 (26)

❖ 실행 결과



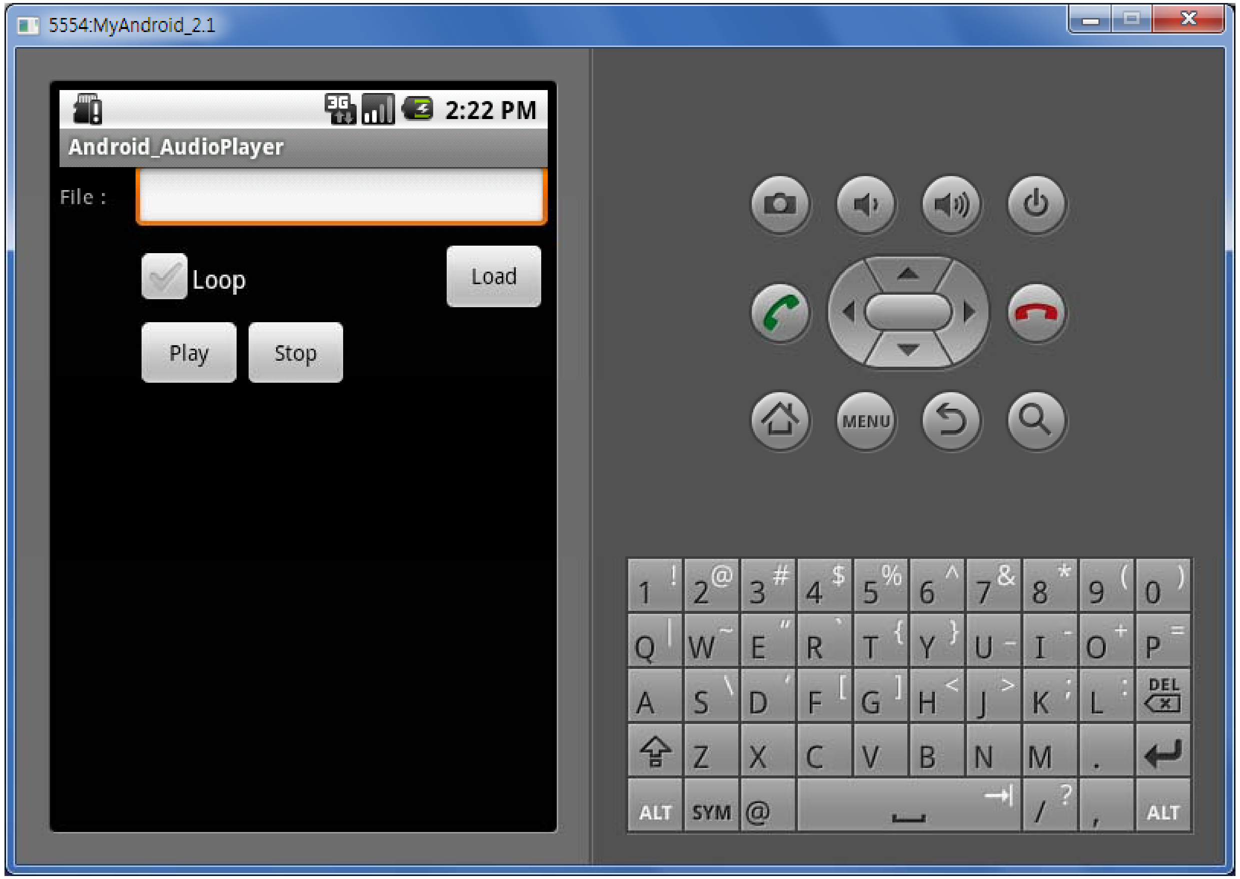
Hello World, Android_Multimedia!

45. 클릭

46. 결과 확인

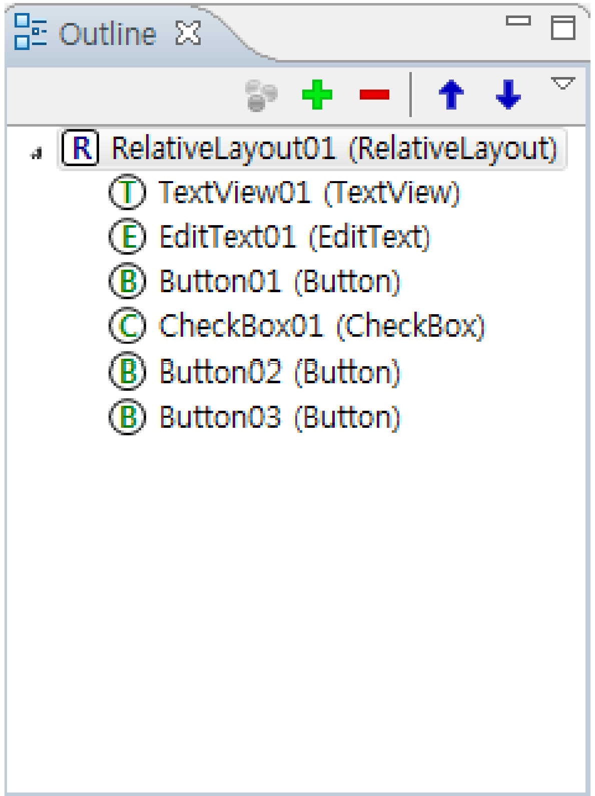
실습 I : Audio Player 구현

- Android_AudioPlayer (실습 시간 : 30분)
  - 아래 그림과 같이 AudioPlayer를 구현하시오.
    - Project Name : Android_AudioPlayer

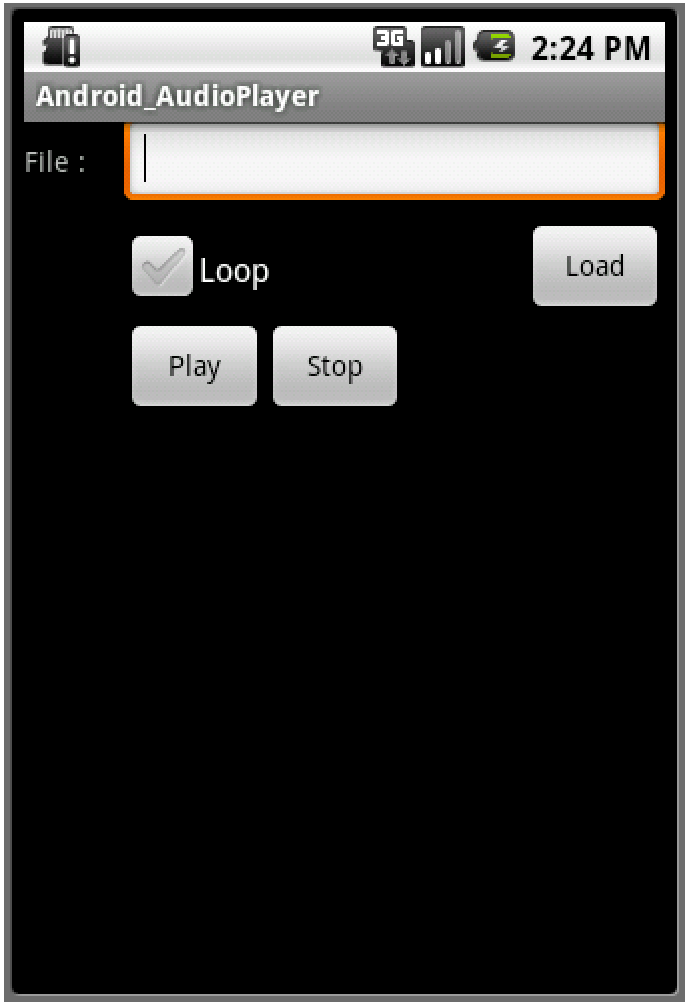


실습 I : Audio Player 구현 (1)

UI 설계



RelativeLayout01 (RelativeLayout)
- TextView01 (TextView)
- EditText01 (EditText)
- Button01 (Button)
- CheckBox01 (CheckBox)
- Button02 (Button)
- Button03 (Button)



1. UI 설계 및 속성 지정

실습 I : Audio Player 구현 (2)

❖ TextView01, EditText01, Button 01 속성 지정

컨트롤	속성 지정
TextView01	• android:id="@+id/txtFLocation" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:text="File : " • android:layout_marginTop="10dip"
EditText01	• android:id="@+id/edtSrcFile" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:layout_alignBaseline="@id/txtFLocation" • android:layout_marginLeft=“50dip" • android:minWidth=“270dip"
Button01	• android:id="@+id/btnLoad" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:text="Load" • android:minWidth="70dip" • android:layout_marginLeft=“250dip"

실습 I : Audio Player 구현 (2)

❖ CheckBox01, Button 02 속성 지정

컨트롤	속성 지정
CheckBox01	• android:id="@+id/chkLoop" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:layout_marginTop="50dip" • android:layout_marginLeft="50dip" • android:text="Loop" • android:checked="false"
Button02	• android:id="@+id/btnPlay" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:layout_marginTop="100dip" • android:layout_marginLeft="50dip" • android:text="Play" • android:minWidth="70dip"

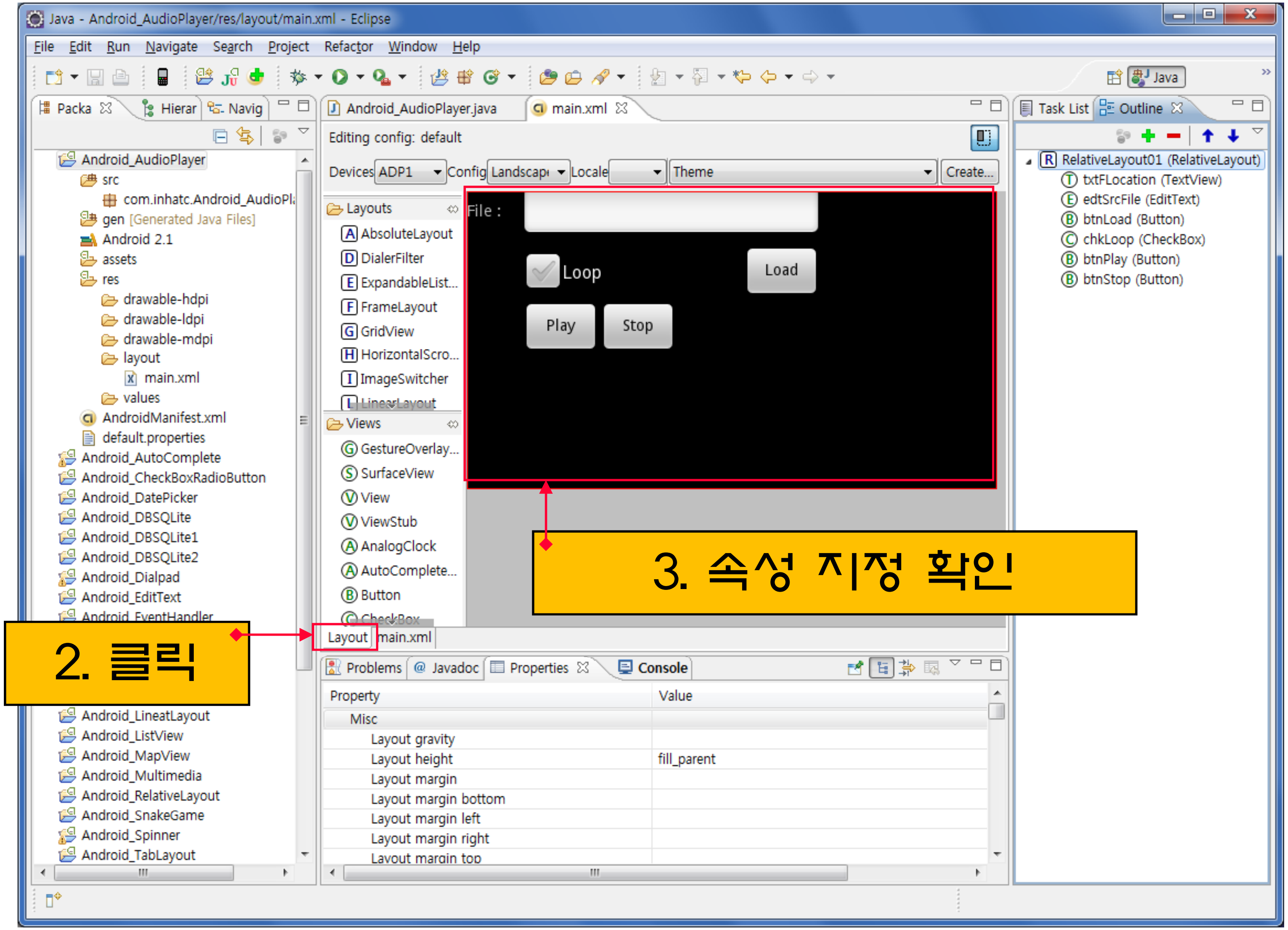
실습 I : Audio Player 구현 (3)

❖ Button 03 속성 지정

컨트롤	속성 지정
Button03	• android:id="@+id/btnStop" • android:layout_width="wrap_content" • android:layout_height="wrap_content" • android:layout_alignBaseline="@id/btnPlay" • android:layout_marginTop="100dip" • android:text="Stop" • android:minWidth="70dip" • android:layout_marginLeft="130dip"

실습 I : Audio Player 구현 (4)

❖ Main.xml 수정 확인!



Editing config: default

Devices ADP1 · Config Landscape · Locale · Theme

RelativeLayout01 (RelativeLayout)
- txtFLocation (TextView)
- edtSrcFile (EditText)
- btnLoad (Button)
- chkLoop (CheckBox)
- btnPlay (Button)
- btnStop (Button)

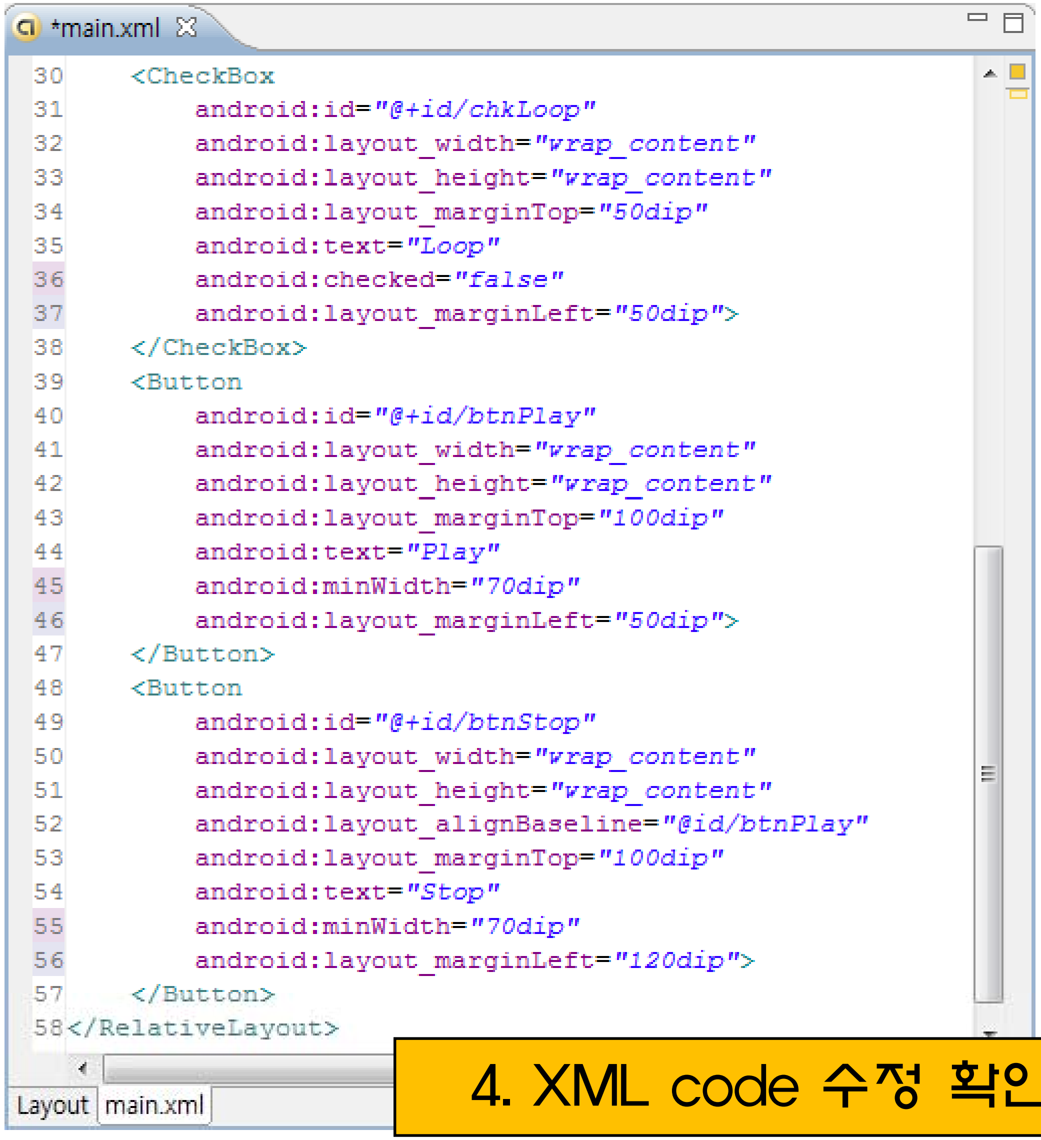
Property	Value
Misc	
Layout gravity	
Layout height	fill_parent
Layout margin	
Layout margin bottom	
Layout margin left	
Layout margin right	
Layout margin top	

2. 클릭

3. 속성 지정 확인!

실습 I : Audio Player 구현 (5)

- Main.xml 수정

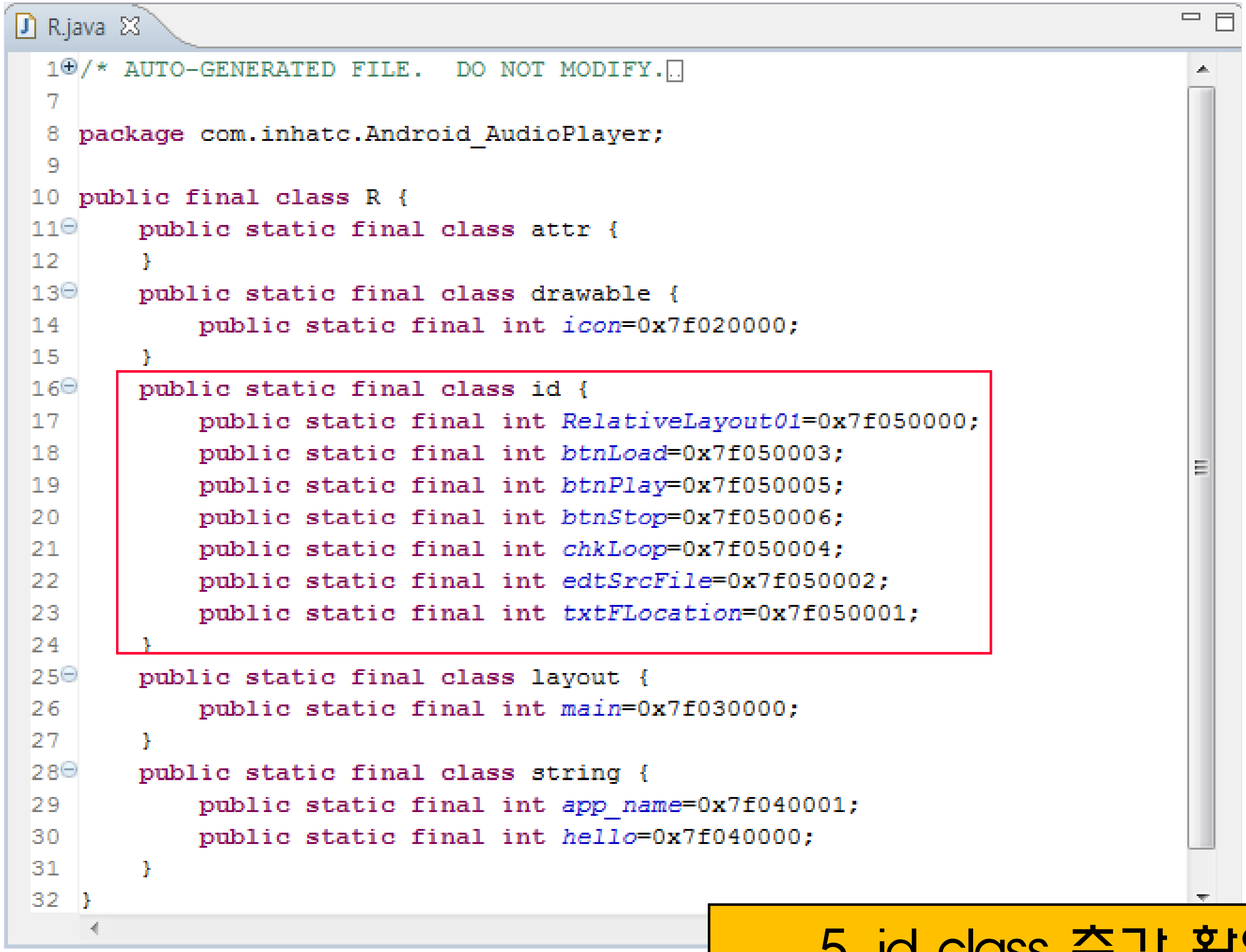


```
    <CheckBox
        android:id="@+id/chkLoop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dip"
        android:text="Loop"
        android:checked="false"
        android:layout_marginLeft="50dip">
    </CheckBox>
    <Button
        android:id="@+id/btnPlay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dip"
        android:text="Play"
        android:minWidth="70dip"
        android:layout_marginLeft="50dip">
    </Button>
    <Button
        android:id="@+id/btnStop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@id/btnPlay"
        android:layout_marginTop="100dip"
        android:text="Stop"
        android:minWidth="70dip"
        android:layout_marginLeft="120dip">
    </Button>
</RelativeLayout>
```

4. XML code 수정 확인!

실습 I : Audio Player 구현 (6)

- R.java

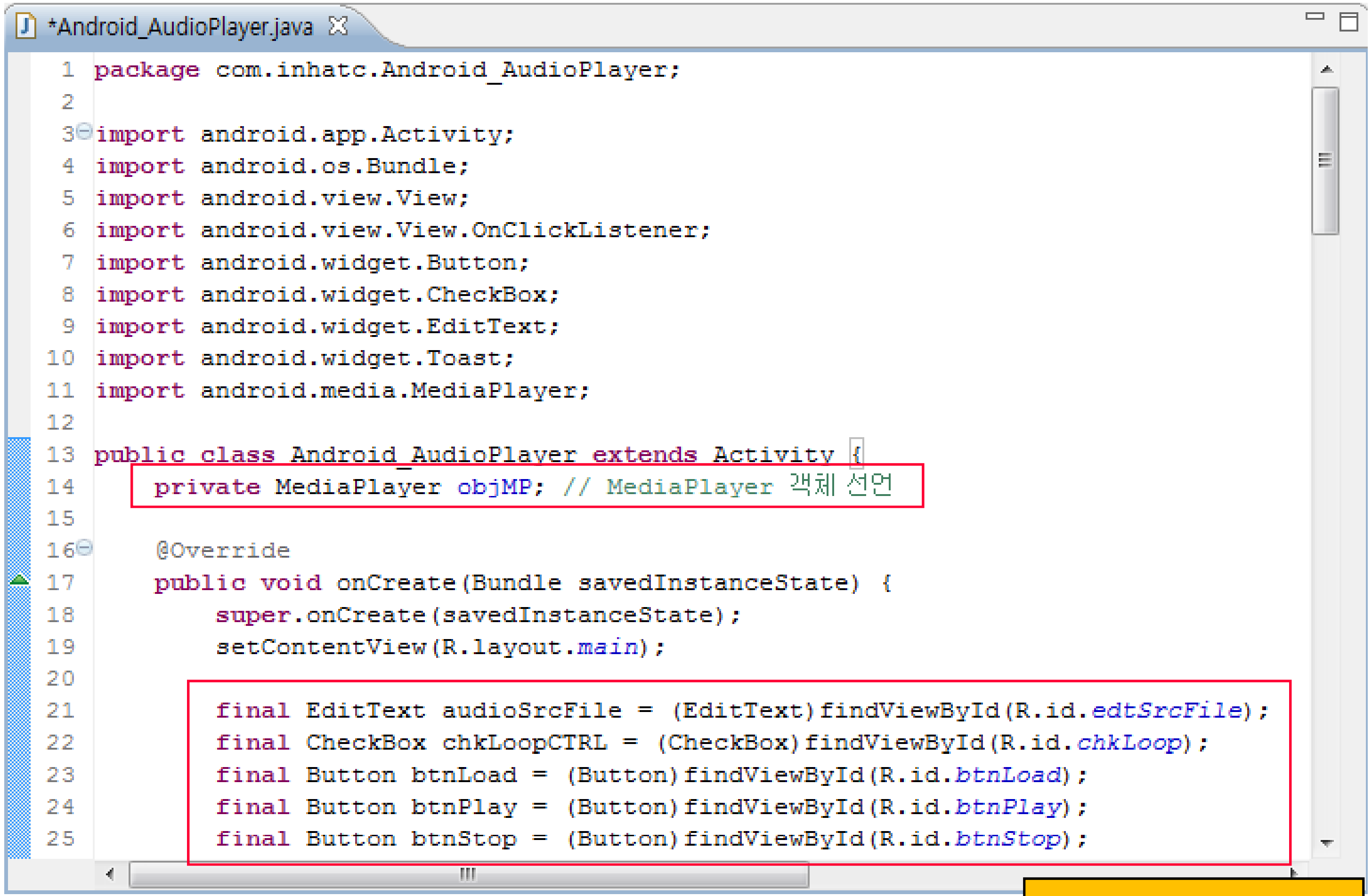


```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.

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 I : Audio Player 구현 (7)

Android_AudioPlayer.java

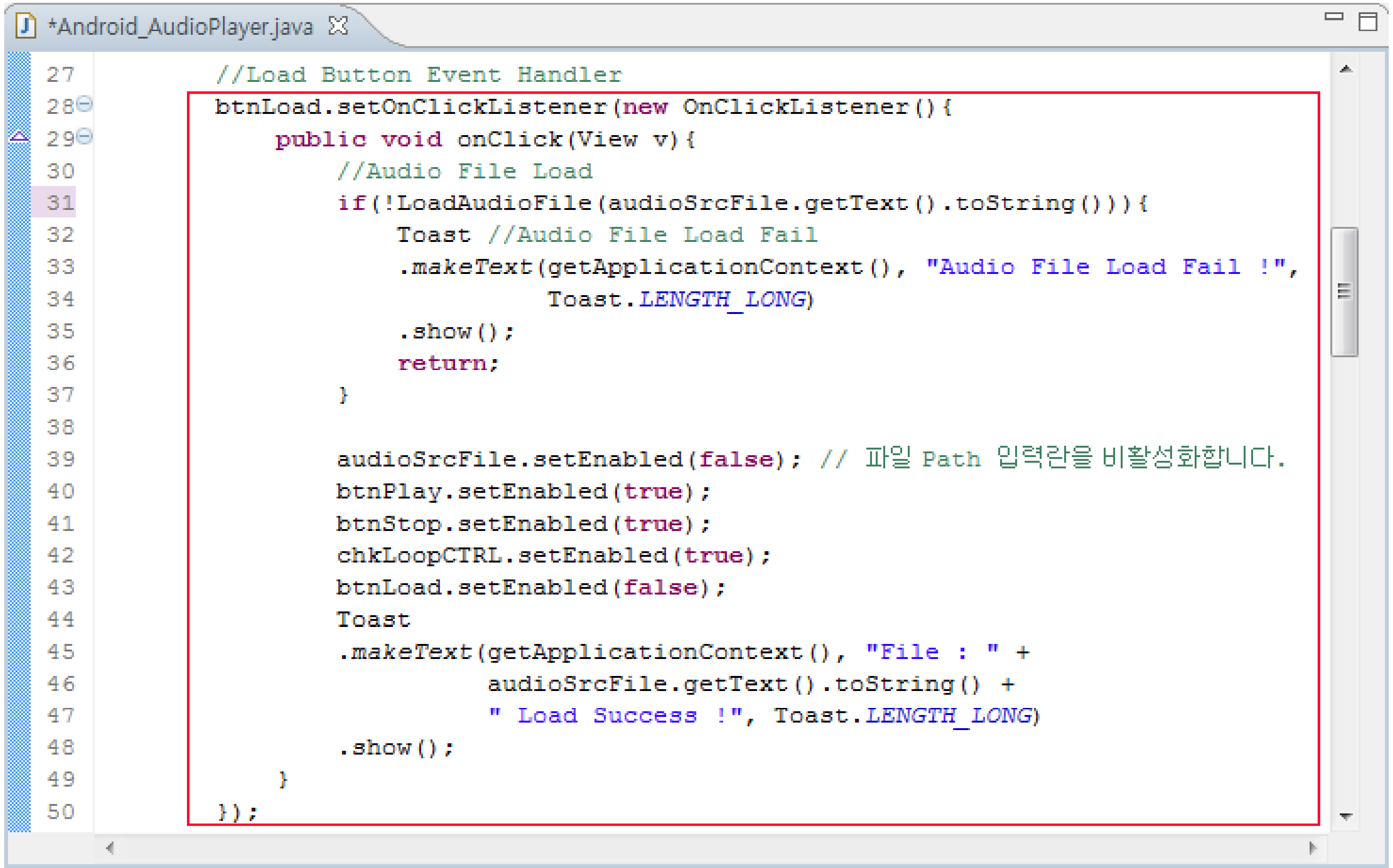


```java
package com.inhatc.Android_AudioPlayer;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;
import android.media.MediaPlayer;

public class Android_AudioPlayer extends Activity {
    private MediaPlayer objMP; // MediaPlayer 객체 선언

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        final EditText audioSrcFile = (EditText)findViewById(R.id.edtSrcFile);
        final CheckBox chkLoopCTRL = (CheckBox)findViewById(R.id.chkLoop);
        final Button btnLoad = (Button)findViewById(R.id.btnLoad);
        final Button btnPlay = (Button)findViewById(R.id.btnPlay);
        final Button btnStop = (Button)findViewById(R.id.btnStop);
```

6. Coding

실습 I : Audio Player 구현 (8)

❖ Load Button Event Handler 구현

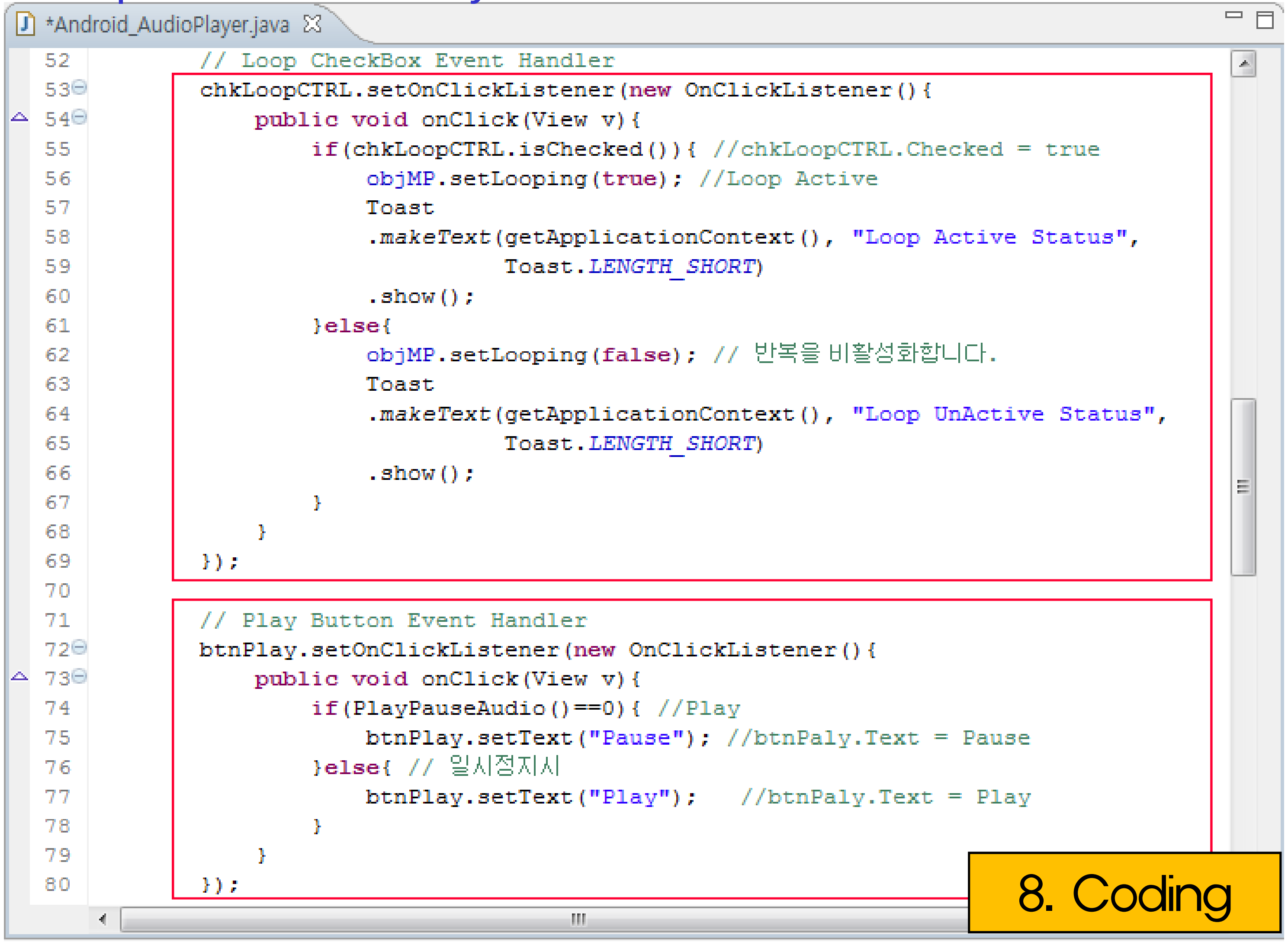


```
//Load Button Event Handler
btnLoad.setOnClickListener(new OnClickListener(){
    public void onClick(View v){
        //Audio File Load
        if(!LoadAudioFile(audioSrcFile.getText().toString())){
            Toast //Audio File Load Fail
            .makeText(getApplicationContext(), "Audio File Load Fail !",
                      Toast.LENGTH_LONG)
            .show();
            return;
        }

        audioSrcFile.setEnabled(false); // 파일 Path 입력란을 비활성화합니다.
        btnPlay.setEnabled(true);
        btnStop.setEnabled(true);
        chkLoopCTRL.setEnabled(true);
        btnLoad.setEnabled(false);
        Toast
        .makeText(getApplicationContext(), "File : " +
                  audioSrcFile.getText().toString() +
                  " Load Success !", Toast.LENGTH_LONG)
        .show();
    }
});
```

7. Coding

실습 I : Audio Player 구현 (9)

❖ Loop CheckBox / Play Button Event Handler 구현

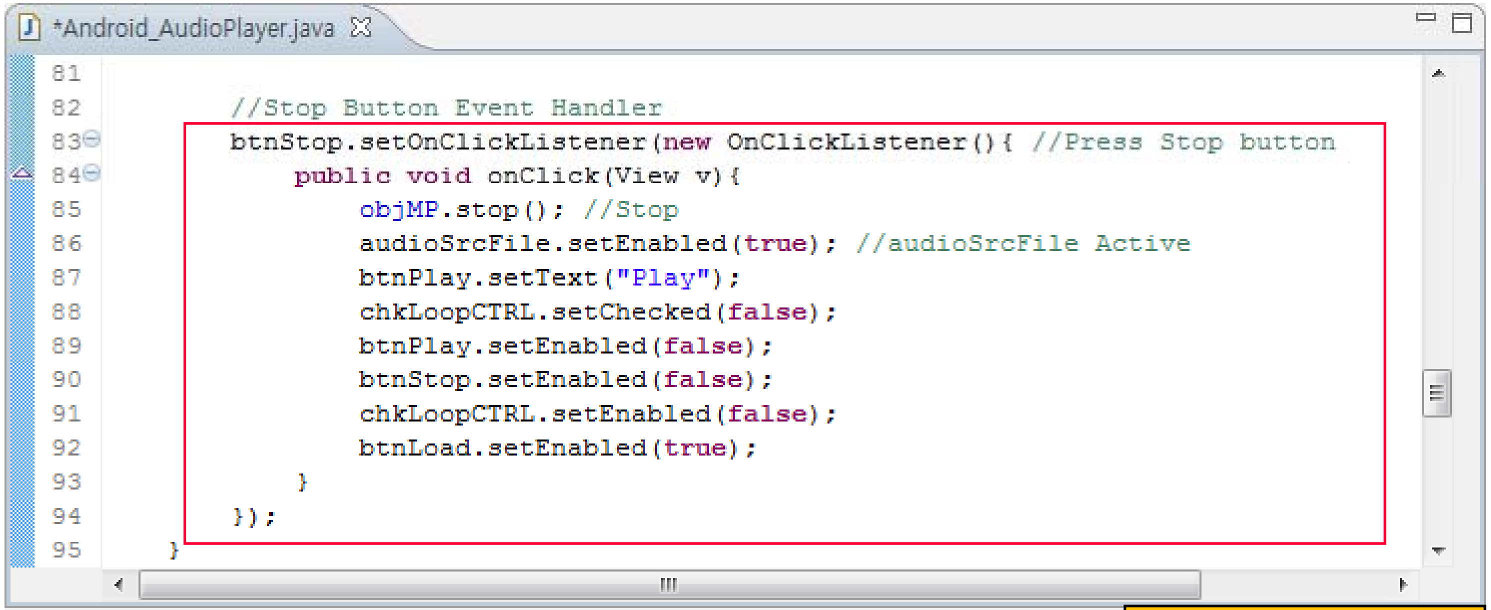


```
// Loop CheckBox Event Handler
chkLoopCTRL.setOnClickListener(new OnClickListener(){
    public void onClick(View v){
        if(chkLoopCTRL.isChecked()){ //chkLoopCTRL.Checked = true
            objMP.setLooping(true); //Loop Active
            Toast
            .makeText(getApplicationContext(), "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{ // 일시정지시
            btnPlay.setText("Play");   //btnPaly.Text = Play
        }
    }
});
```

8. Coding

실습 I : Audio Player 구현 (10)

❖ Stop Button Event Handler 구현

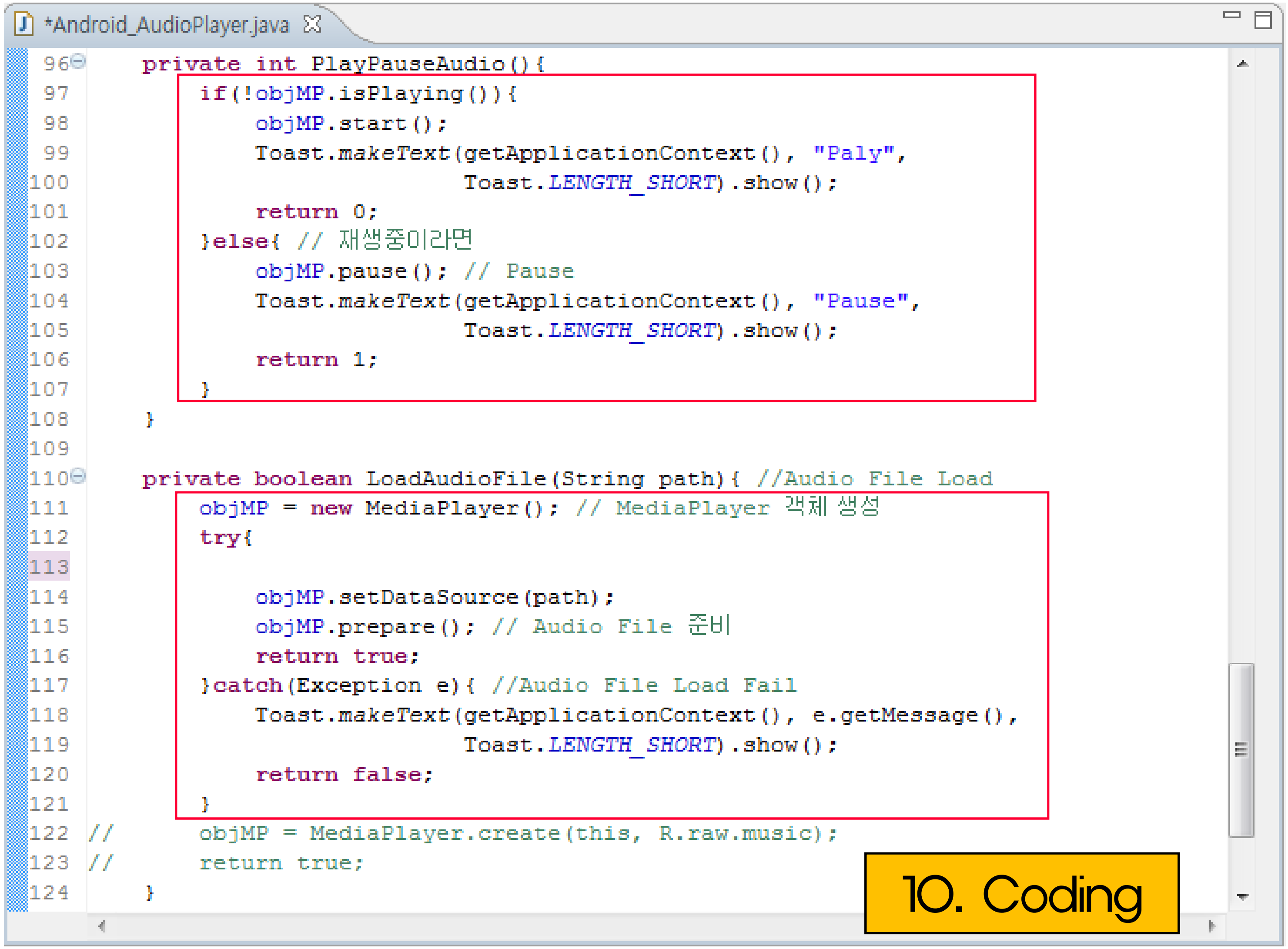


```
        //Stop Button Event Handler
        btnStop.setOnClickListener(new OnClickListener(){ //Press Stop button
            public void onClick(View v){
                objMP.stop(); //Stop
                audioSrcFile.setEnabled(true); //audioSrcFile Active
                btnPlay.setText("Play");
                chkLoopCTRL.setChecked(false);
                btnPlay.setEnabled(false);
                btnStop.setEnabled(false);
                chkLoopCTRL.setEnabled(false);
                btnLoad.setEnabled(true);
            }
        });
    }
```

9. Coding

실습 I : Audio Player 구현 (11)

❖ PlayPauseAudio() / LoadAudioFile() Method 구현

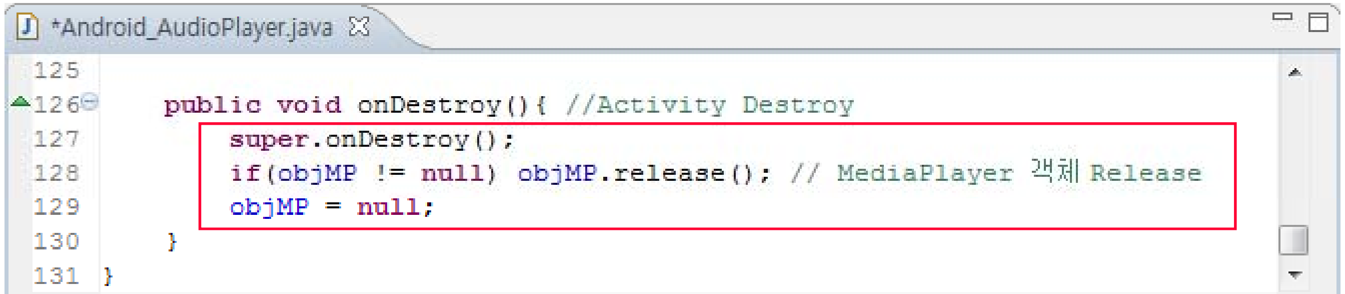


```java
    private int PlayPauseAudio(){
        if(!objMP.isPlaying()){
            objMP.start();
            Toast.makeText(getApplicationContext(), "Paly",
                           Toast.LENGTH_SHORT).show();
            return 0;
        }else{ // 재생중이라면
            objMP.pause(); // Pause
            Toast.makeText(getApplicationContext(), "Pause",
                           Toast.LENGTH_SHORT).show();
            return 1;
        }
    }

    private boolean LoadAudioFile(String path){ //Audio File Load
        objMP = new MediaPlayer(); // MediaPlayer 객체 생성
        try{

            objMP.setDataSource(path);
            objMP.prepare(); // Audio File 준비
            return true;
        }catch(Exception e){ //Audio File Load Fail
            Toast.makeText(getApplicationContext(), e.getMessage(),
                           Toast.LENGTH_SHORT).show();
            return false;
        }
//      objMP = MediaPlayer.create(this, R.raw.music);
//      return true;
    }
```

10. Coding

실습 I : Audio Player 구현 (12)

- onDestroy() Method 구현

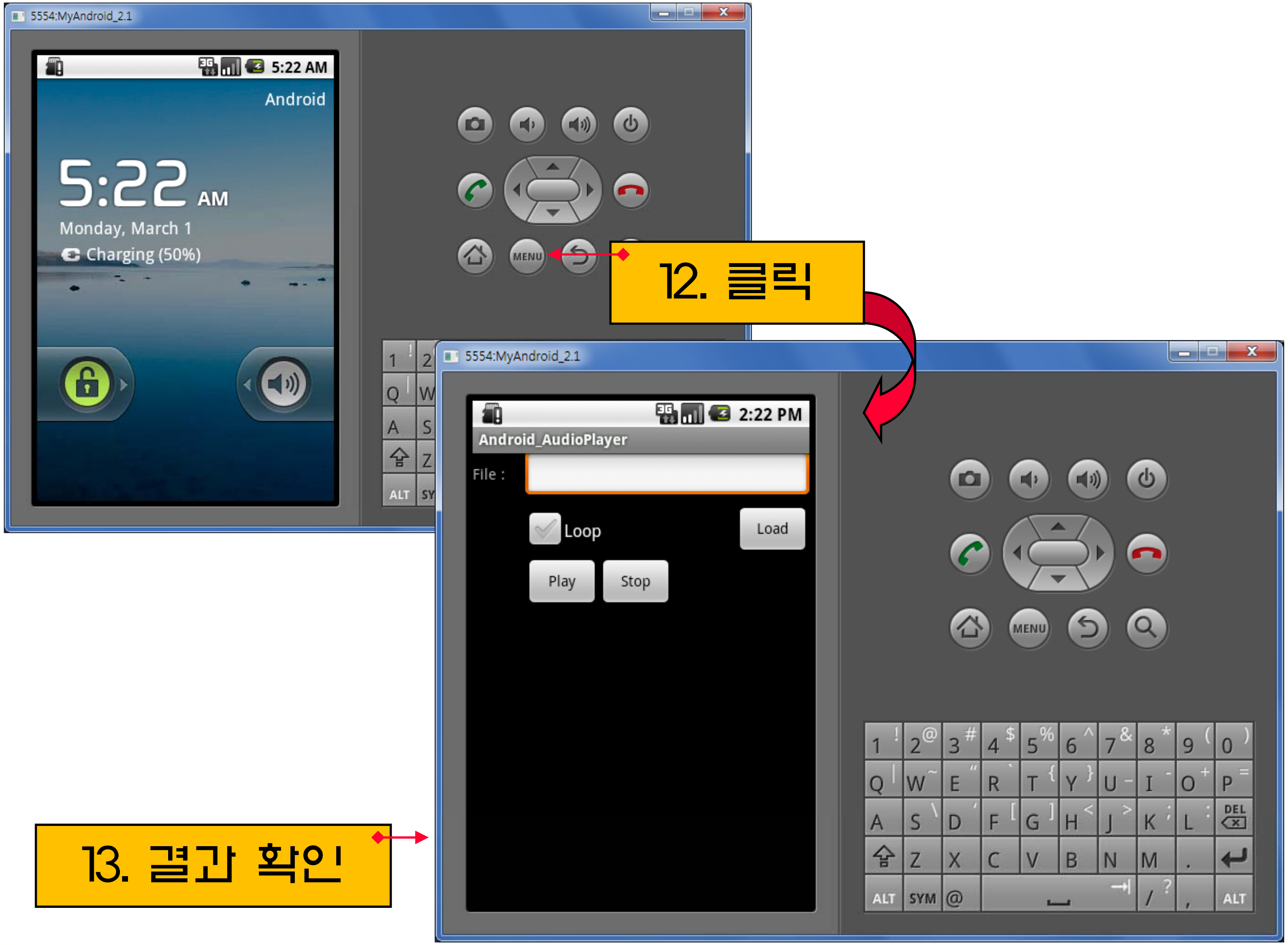


```java

    public void onDestroy(){ //Activity Destroy
        super.onDestroy();
        if(objMP != null) objMP.release(); // MediaPlayer 객체 Release
        objMP = null;
    }
}
```

11. Coding

실습 I : Audio Player 구현 (8)

❖ 실행 결과



12. 클릭

13. 결과 확인

학습 요약

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