



Graphic 2D





학습 목표

교육 목표

- ❖ CustomView 구현
- ❖ Keyboard 입력 처리
- ❖ Touch Screen 처리
- ❖ 실습 I
 - ◆ Android Image 화면 중앙 출력 구현
- ❖ 실습 II
 - ◆ Android KeyEvent 처리 구현
- ❖ 실습 III
 - ◆ Android Screen Touch Event 처리 구현



Navigation voice



Traffic view



Street View





Android Image

■ Android 지원 Image 형식

❖ JPEG

◆ Joint Photographic Experts Group

❖ PNG

◆ Portable Network Graphics

❖ GIF

◆ Graphics Interchange Format

❖ BMP

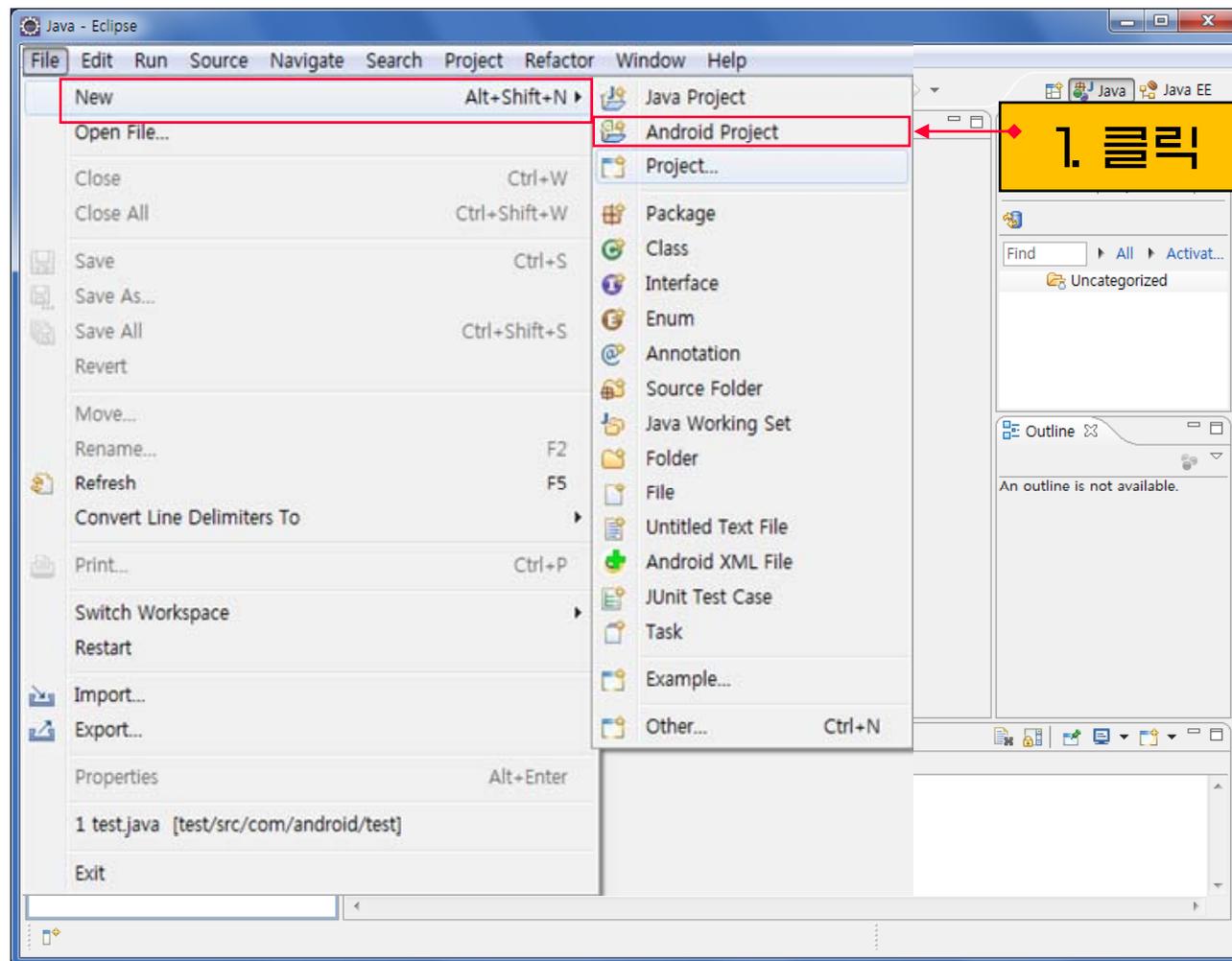




Android Custom View 구현 (1)

■ Android 프로젝트 생성

❖ 프로젝트 명 : Android_Image





Android Custom View 구현 (2)

New Android Project

Creates a new Android Project resource.

Project name:

Contents

- Create new project in workspace
- Create project from existing source
- Use default location

Location:

- Create project from existing sample

Samples:

Build Target

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:

Package name:

- Create Activity:

Min SDK Version:

2. Android_Image 입력

3. 클릭

4. Android_Image 입력

5. com.inhatc.android_Image 입력

6. Android_Image 입력

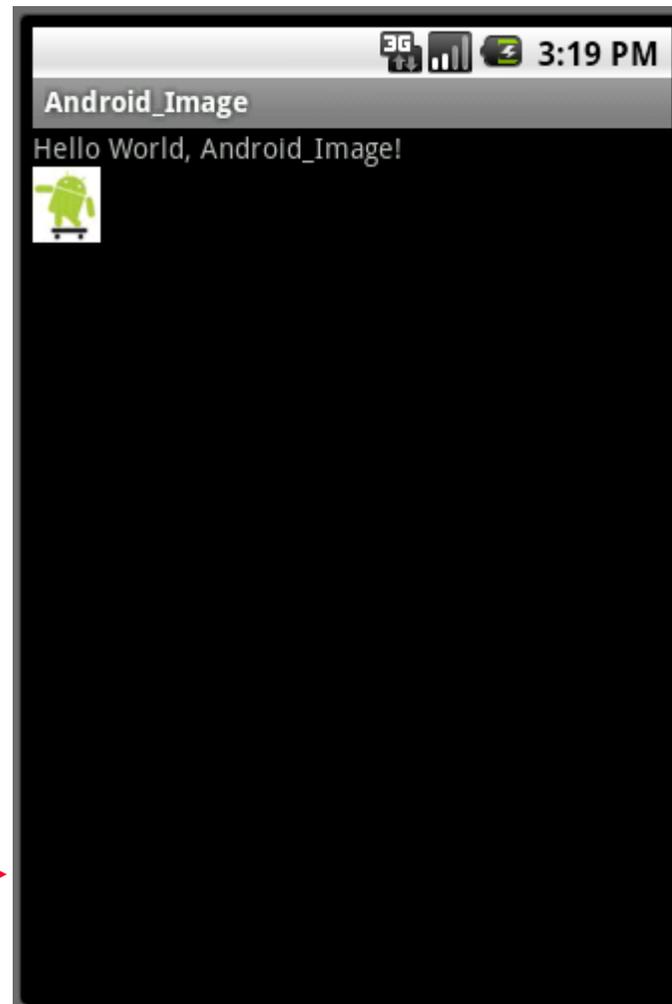
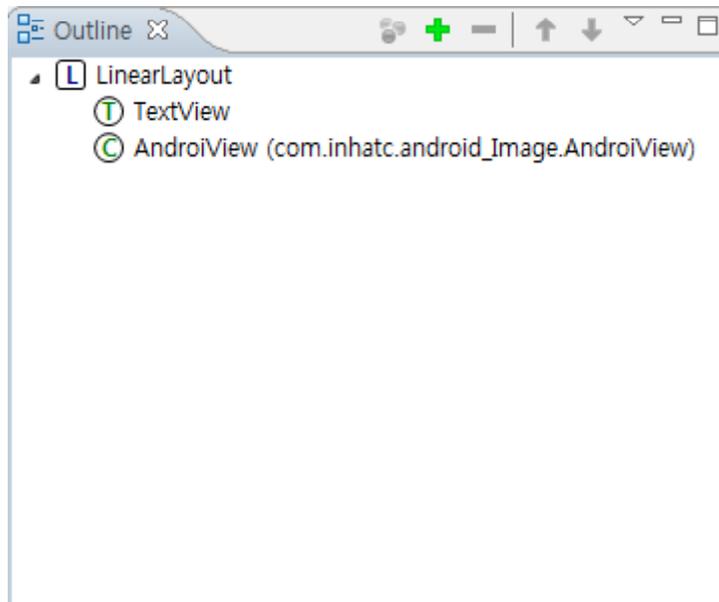
7. 클릭





Android Custom View 구현 (3)

■ UI 설계



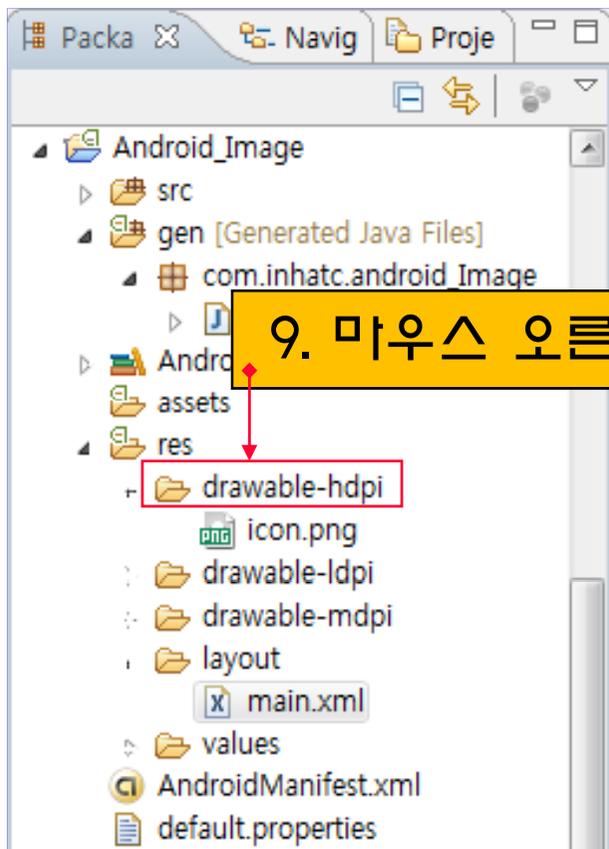
8. UI 설계 및 속성 지정



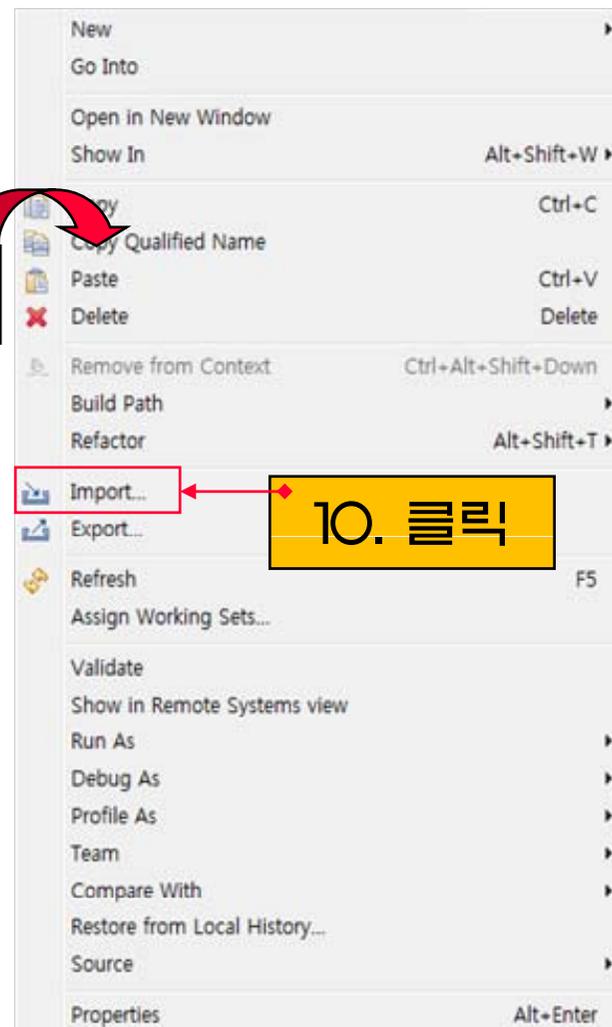


Android Custom View 구현 (4)

Image import



9. 마우스 오른쪽 버튼 클릭

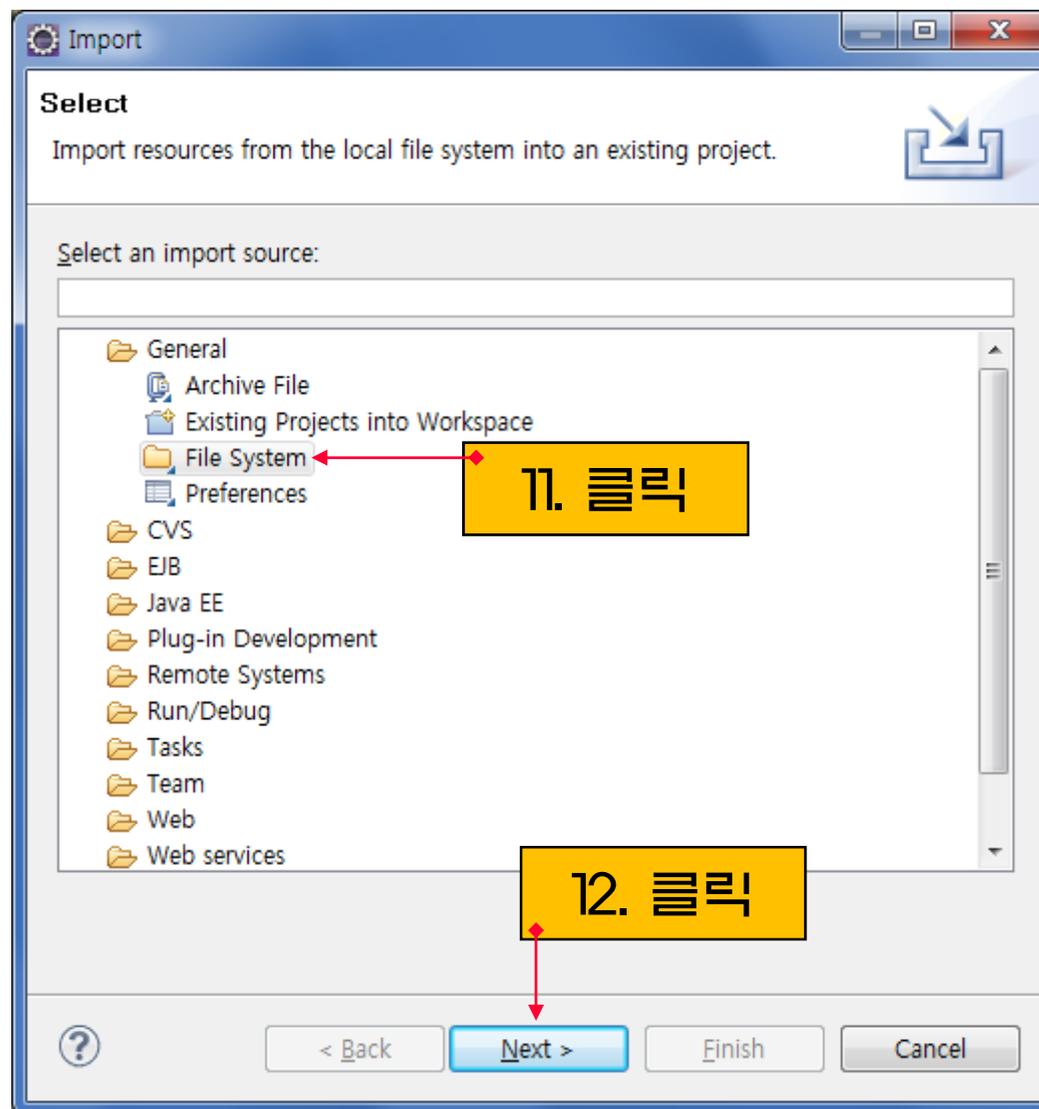


10. 클릭



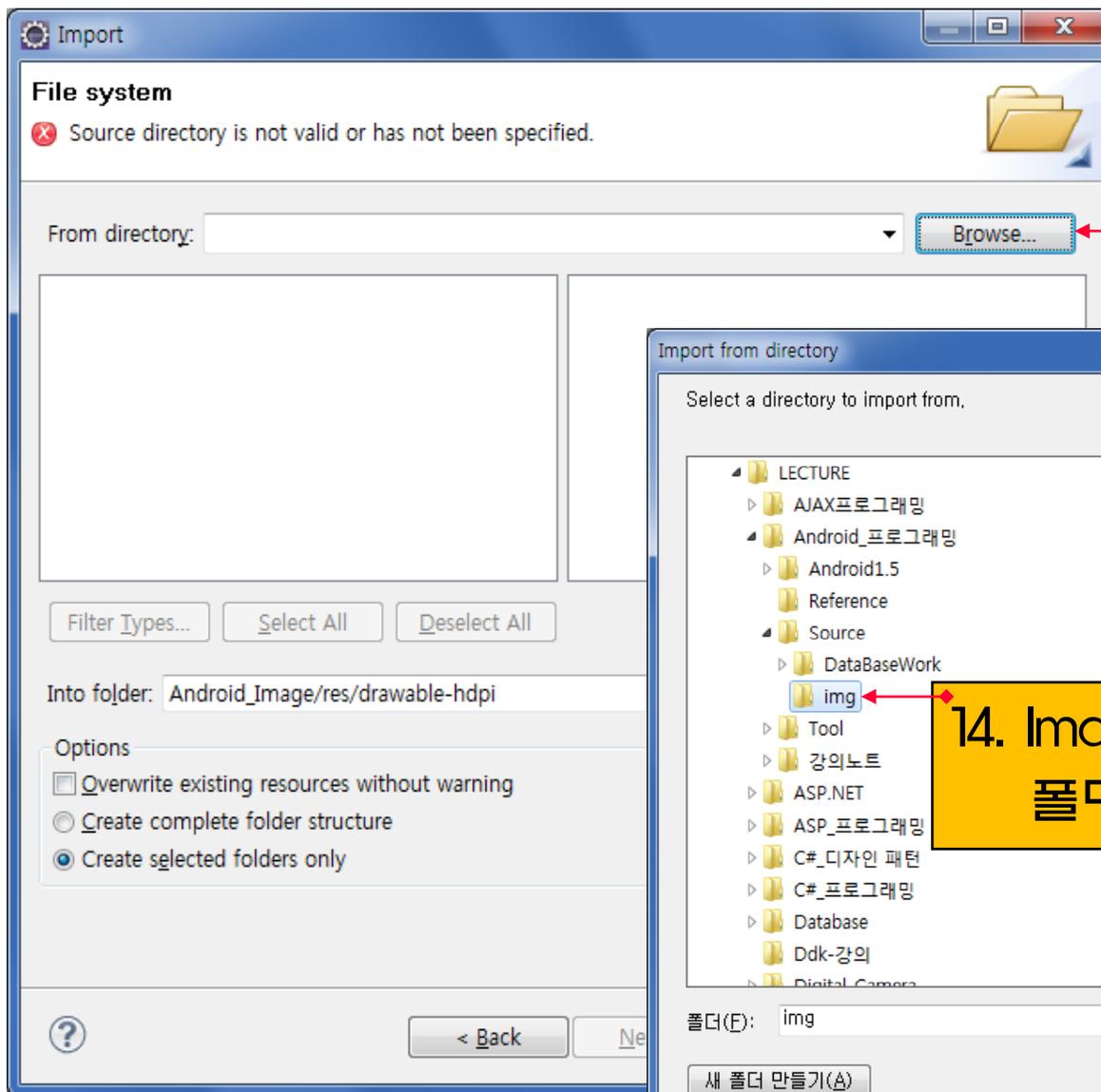


Android Custom View 구현 (5)

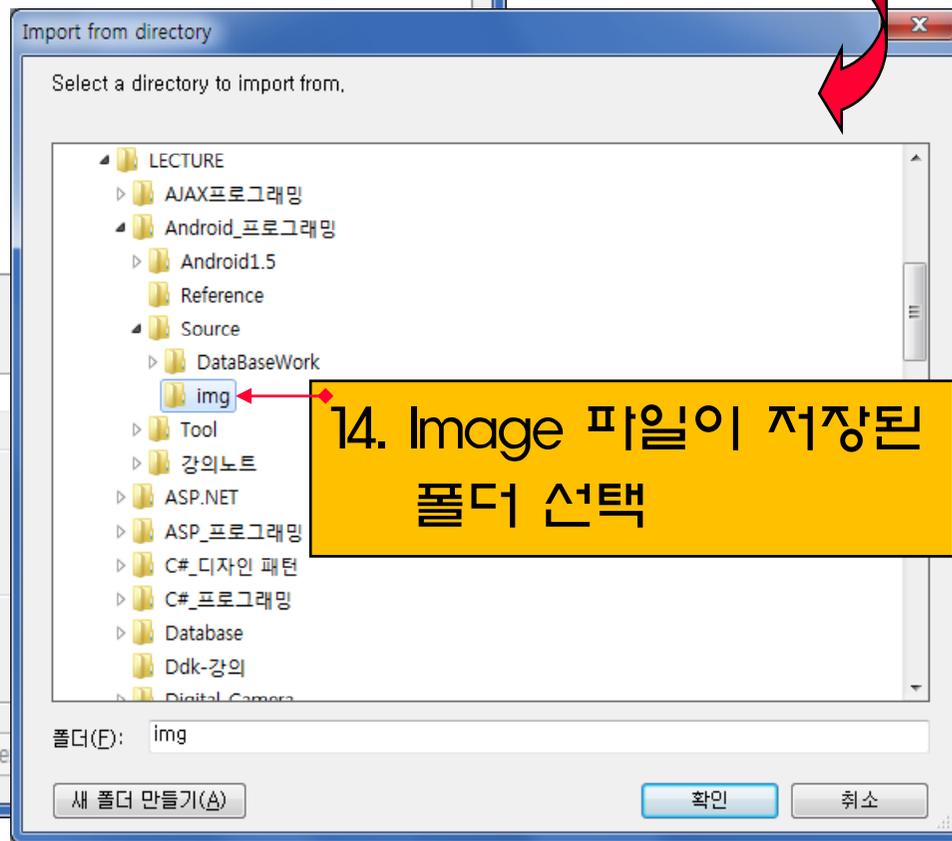




Android Custom View 구현 (6)



13. 클릭



14. Image 파일이 저장된
폴더 선택

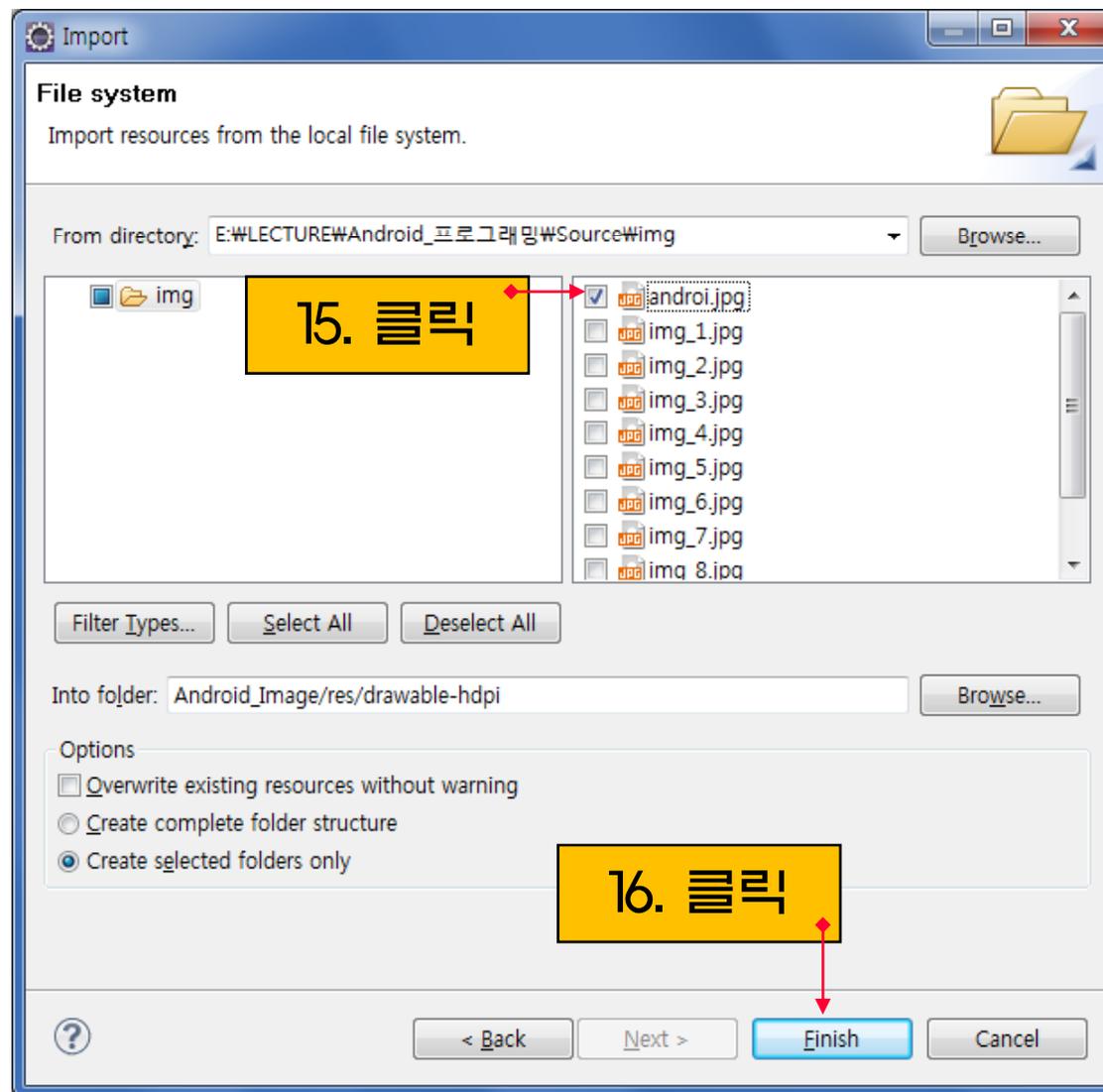




Android Custom View 구현 (7)

❖ Image 파일 선택

◆ android.jpg

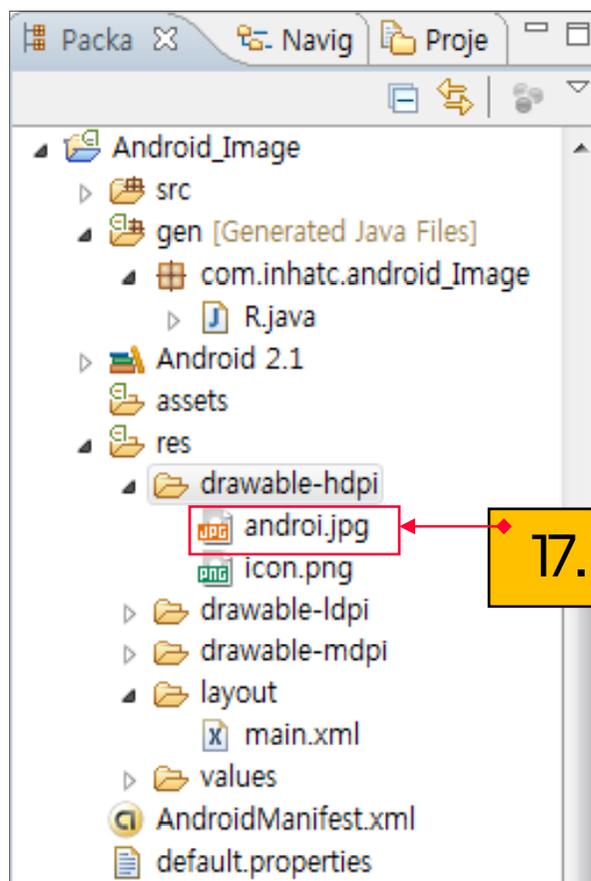




Android Custom View 구현 (8)

❖ Image 파일 import 확인

◆ android.jpg



17. Image 파일 import 확인





Android Custom View 구현 (9)

■ Android_Image.java 수정

❖ TitleBar 제거

```
*Android_Image.java X
1 package com.inhatec.android_Image;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Window;
6
7 public class Android_Image extends Activity {
8     /** Called when the activity is first created. */
9     @Override
10    public void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12
13        requestWindowFeature(Window.FEATURE_NO_TITLE); //TitleBar 제거
14
15        setContentView(R.layout.main);
16    }
17 }
```

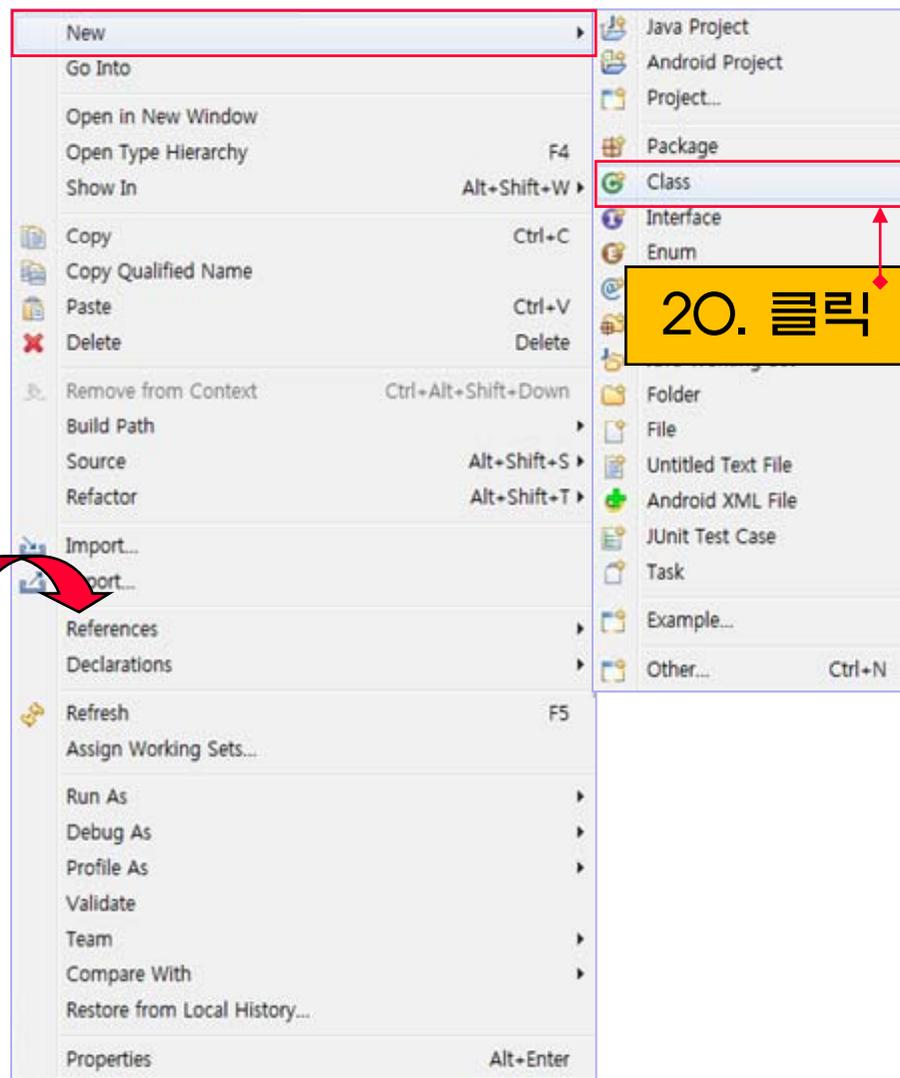
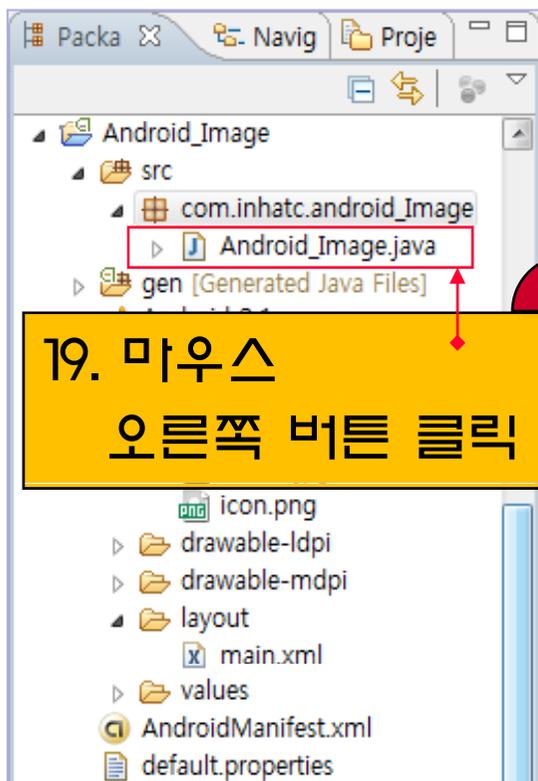
18. Coding





Android Custom View 구현 (10)

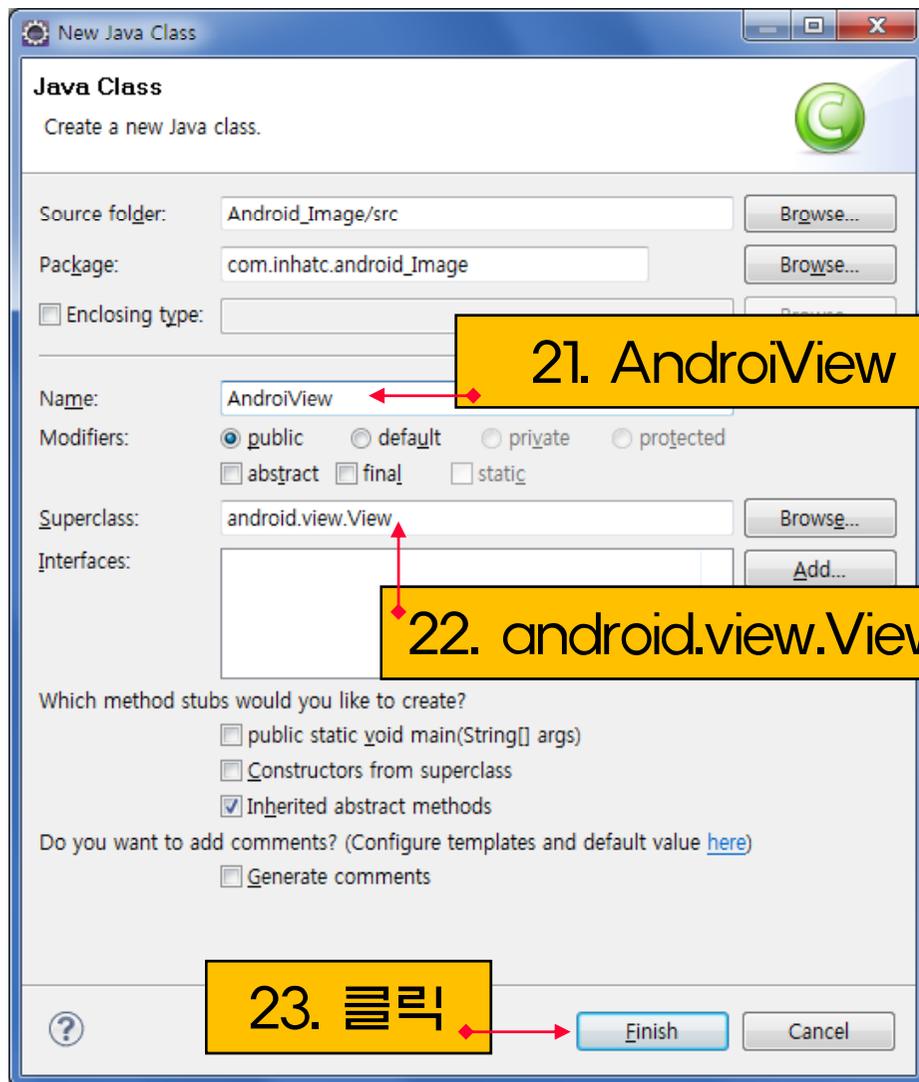
■ New Class 추가





Android Custom View 구현 (11)

❖ AndroiView Class



21. AndroiView 입력

22. android.view.View 입력

23. 클릭





Android Custom View 구현 (12)

❖ AndroidView 생성자 추가

24. 클릭

25. 더블 클릭

```
1 package com.inhatec.android_image;
2
3 import android.view.View;
4
5 public class AndroiView extends View {
6
7 }
```

```
import android.content.Context;
import android.util.AttributeSet;
import android.view.View;
public class AndroiView extends View {

public AndroiView(Context context, AttributeSet attrs) {
super(context, attrs);
// TODO Auto-generated constructor stub
}
}
```

editable	true
last modified	2010년 4월 4일 오후 10:43:59
linked	false
location	C:\Android\Workspace\Android_Image\src\com\inhatec\...
name	AndroiView.java
path	/Android_Image/src/com/inhatec/android_image/AndroiView.j...
size	111 bytes

Implicit super constructor View() is undef...uctor. Must define an explicit constructor Writable Smart Insert 5 : 24





Android Custom View 구현 (13)

❖ AndroidView 생성자 추가 확인

```
*AndroiView.java ✕
1 package com.inhatc.android_image;
2
3 import android.content.Context;
4 import android.util.AttributeSet;
5 import android.view.View;
6
7 public class AndroiView extends View {
8
9     public AndroiView(Context context, AttributeSet attrs) {
10         super(context, attrs);
11         // TODO Auto-generated constructor stub
12     }
13
14 }
```

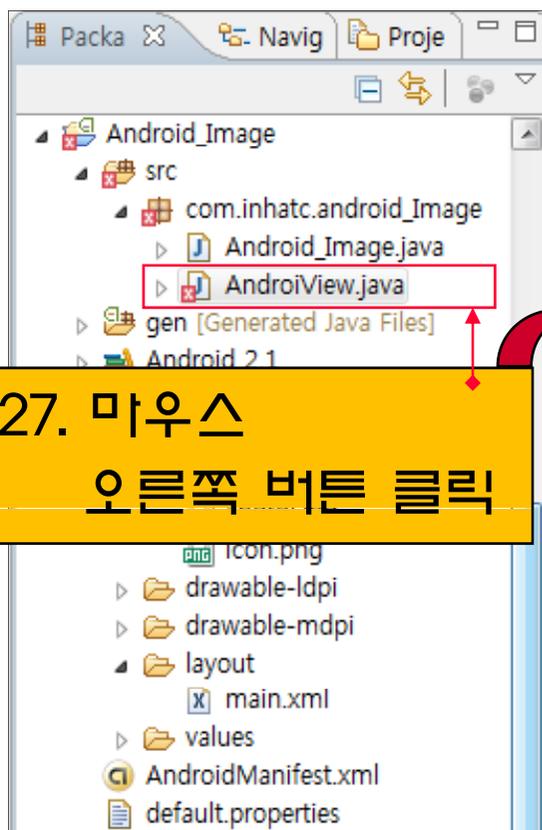
26. AndroidView 생성자 추가 확인



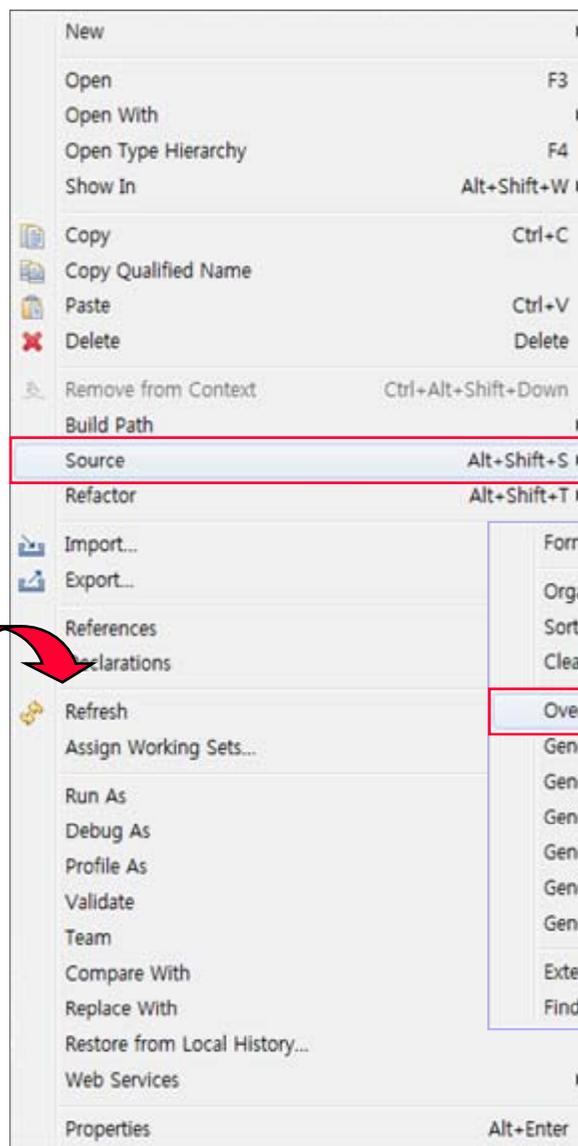


Android Custom View 구현 (14)

Method 추가



27. 마우스
오른쪽 버튼 클릭



28. 클릭

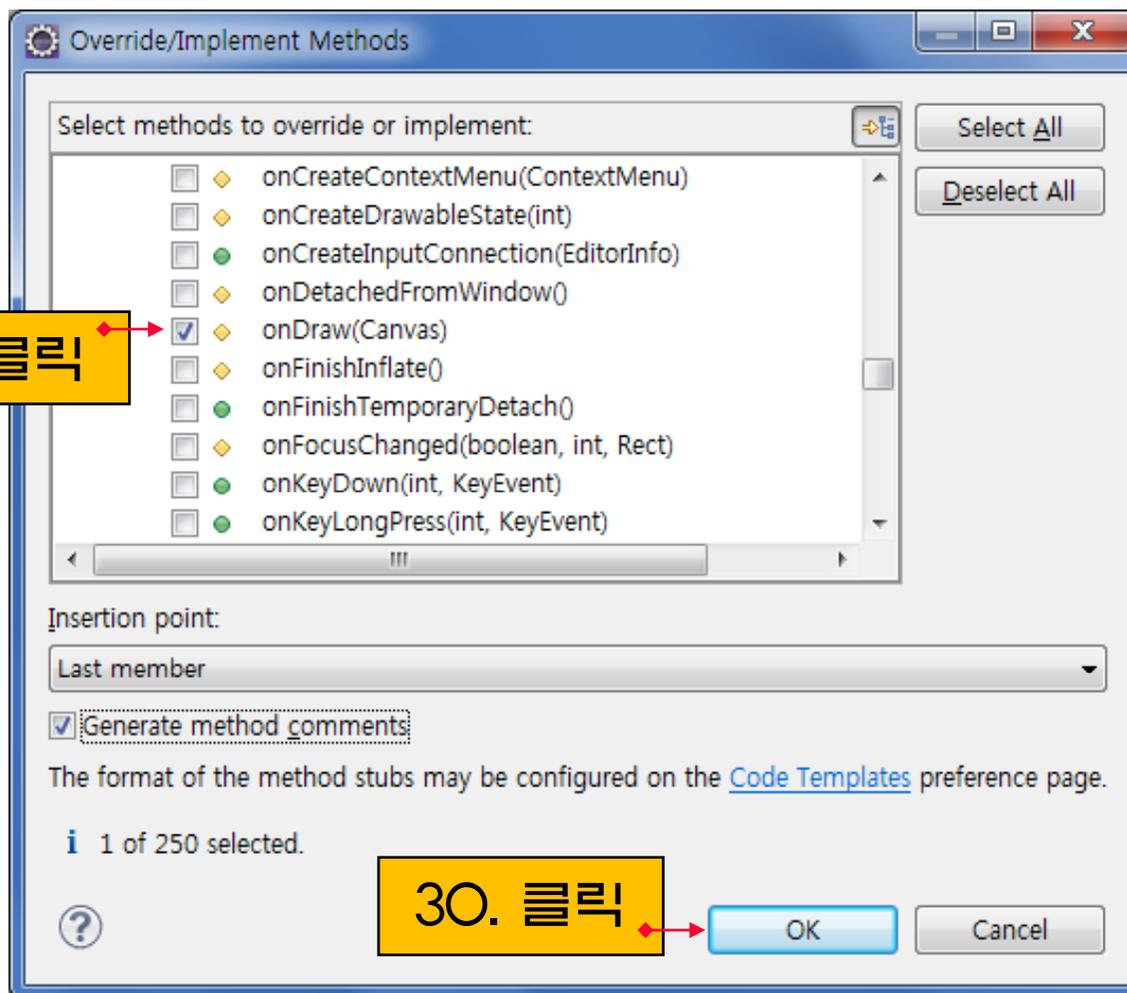




Android Custom View 구현 (15)

Method 추가

onDraw(Canvas) 선택



29. 클릭

30. 클릭





Android Custom View 구현 (16)

```
*AndroiView.java
1 package com.inhatc.android_Image;
2
3 import android.content.Context;
4 import android.graphics.Canvas;
5 import android.util.AttributeSet;
6 import android.view.View;
7
8 public class AndroiView extends View {
9
10     public AndroiView(Context context, AttributeSet attrs) {
11         super(context, attrs);
12         // TODO Auto-generated constructor stub
13     }
14
15     /* (non-Javadoc)
16      * @see android.view.View#onDraw(android.graphics.Canvas)
17      */
18     @Override
19     protected void onDraw(Canvas canvas) {
20         // TODO Auto-generated method stub
21         super.onDraw(canvas);
22     }
23
24 }
```

31. onDraw(Canvas) method 추가 확인





Android Custom View 구현 (17)

■ Main.xml 수정

❖ AndroidView 추가

```
main.xml
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    <com.inhatc.android_image.AndroiView
13        android:id="@+id/AndroiView"
14        android:layout_width="fill_parent"
15        android:layout_height="fill_parent"
16        android:focusable="true">
17    </com.inhatc.android_image.AndroiView>
18</LinearLayout>
```

32. 추가





Android Custom View 구현 (18)

❖ main.xml 저장

33. 클릭

34. 클릭

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

Property	Value
Layout gravity	center
Layout height	wrap_content
Layout margin	
Layout margin bottom	
Layout margin left	





Android Custom View 구현 (19)

❖ R.java

```
R.java X
1+/* AUTO-GENERATED FILE. DO NOT MODIFY.
7
8 package com.inhatc.android_Image;
9
10 public final class R {
11     public static final class attr {
12     }
13     public static final class drawable {
14         public static final int androi=0x7f020000;
15         public static final int icon=0x7f020001;
16     }
17     public static final class id {
18         public static final int AndroiView=0x7f050000;
19     }
20     public static final class layout {
21         public static final int main=0x7f030000;
22     }
23     public static final class string {
24         public static final int app_name=0x7f040001;
25         public static final int hello=0x7f040000;
26     }
27 }
```

35. drawable 클래스 변수 추가 확인





Android Custom View 구현 (20)

■ AndroidView.java 수정

```
AndroidView.java ✕
1 package com.inhatec.android_Image;
2
3 import android.content.Context;
4 import android.graphics.Canvas;
5 import android.graphics.drawable.Drawable;
6 import android.util.AttributeSet;
7 import android.view.View;
8
9 public class AndroiView extends View {
10     Drawable imgAndroi; //image 객체 선언
11     int ix, iy; //Width, Height 저장
12     int imgWidth, imgHeight; //Image Width, Height 저장
13
14     public AndroiView(Context context, AttributeSet attrs) {
15         super(context, attrs);
16         // TODO Auto-generated constructor stub
17
18         imgAndroi=this.getResources().getDrawable(R.drawable.androi);
19     }
20
21     @Override
22     protected void onDraw(Canvas canvas) {
23         // TODO Auto-generated method stub
24         imgWidth = imgAndroi.getIntrinsicWidth();
25         imgHeight = imgAndroi.getIntrinsicHeight();
26         ix = 0;
27         iy = 0;
28
29         imgAndroi.setBounds(ix, iy, ix+imgWidth, iy+imgHeight);
30         imgAndroi.draw(canvas);
31
32         super.onDraw(canvas);
33     }
34 }
```

36. Coding





Android Custom View 구현 (21)

❖ Android 프로젝트 실행

The screenshot shows the Eclipse IDE interface. The **Run** menu is open, and the **Run As** dialog box is displayed. The following steps are highlighted with red arrows and yellow callouts:

- 37. 클릭**: Click on the **Run** menu item (Ctrl+F11).
- 38. 클릭**: Click on **Android Application** in the **Run As** dialog box.
- 39. 클릭**: Click on the **OK** button in the **Run As** dialog box.

The **Run As** dialog box shows the following options:

- Android Application
- Android JUnit Test
- Java Applet
- Java Application
- JUnit Test

The description for **Android Application** is: **Runs an Android Application**.





Android Custom View 구현 (22)

실행 결과

The image displays two sequential screenshots of an Android emulator window titled "5554:MyAndroid_2.1_SDcard".

The top screenshot shows the lock screen with the time "3:15 PM", date "Sunday, April 4", and "Charging (50%)". A yellow callout box with the text "40. 클릭" (Click) has a red arrow pointing to the MENU button on the virtual device's keypad.

The bottom screenshot shows the same emulator with a custom view titled "Android_Image" displayed. The view contains the text "Hello World, Android_Image!" and a small Android logo. A yellow callout box with the text "41. 결과 확인" (Check result) has a red arrow pointing to the custom view area.

A red curved arrow indicates the transition from the lock screen to the custom view.



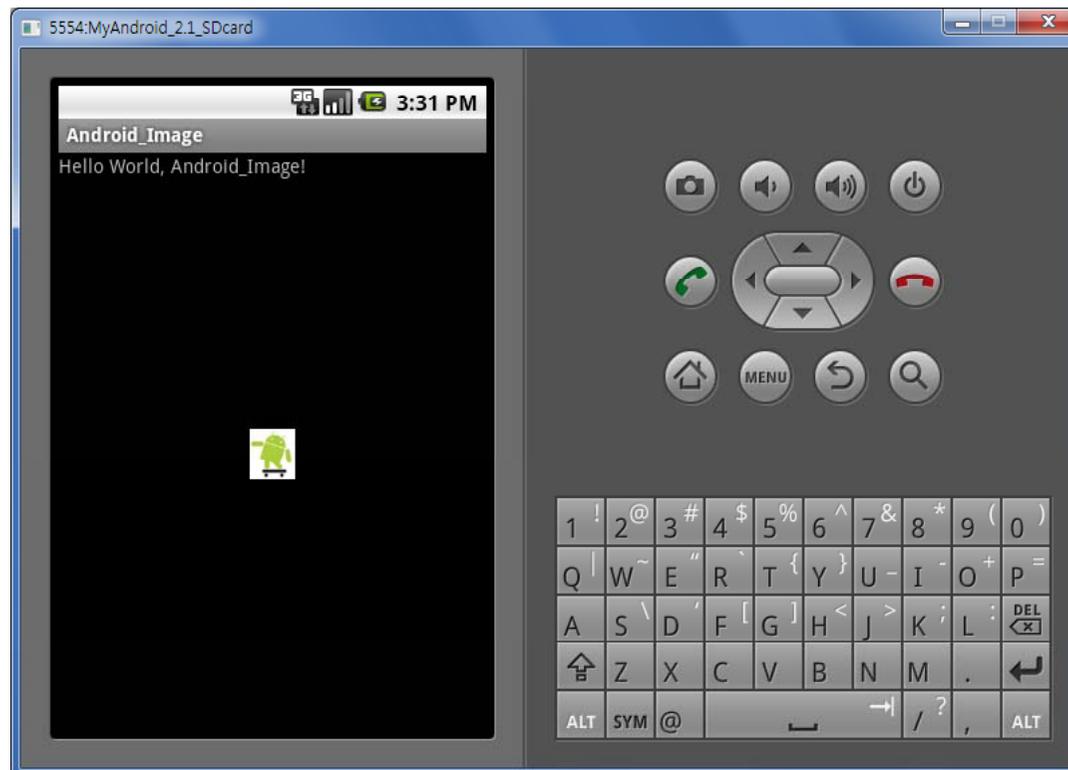


실습 I : Android Image 화면 중앙 출력 구현

■ Android_Image (실습 시간 : 20분)

❖ 아래 그림과 같이 Androi(안드로이드) 이미지가 화면 중앙에 나타나도록 Coding 하시오.

◆ AndroiView.java 의 onDraw(Canvas canvas) 수정





Android Image 화면 중앙 출력 구현

```
AndroiView.java
1 package com.inhatec.android_Image;
2
3 import android.content.Context;
4 import android.graphics.Canvas;
5 import android.graphics.drawable.Drawable;
6 import android.util.AttributeSet;
7 import android.view.View;
8
9 public class AndroiView extends View {
10     Drawable imgAndroi;        //image 객체 선언
11     int ix, iy;                //Width, Height 저장
12     int imgWidth, imgHeight;   //Image Width, Height 저장
13
14     public AndroiView(Context context, AttributeSet attrs) {
15         super(context, attrs);
16         // TODO Auto-generated constructor stub
17
18         imgAndroi=this.getResources().getDrawable(R.drawable.androi);
19     }
20
21     @Override
22     protected void onDraw(Canvas canvas) {
23         // TODO Auto-generated method stub
24         imgWidth = imgAndroi.getIntrinsicWidth();
25         imgHeight = imgAndroi.getIntrinsicHeight();
26         ix = (this.getWidth()-imgWidth)/2;
27         iy = (this.getHeight()-imgHeight)/2;
28
29         imgAndroi.setBounds(ix, iy, ix+imgWidth, iy+imgHeight);
30         imgAndroi.draw(canvas);
31
32         super.onDraw(canvas);
33     }
34 }
```

Coding





실습 II : Android KeyEvent 처리 구현

■ Android_Image (실습 시간 : 20분)

❖ 아래 그림과 같이     Key에 따라 android() 이미지의 위치를 이동하도록 Coding 하시오.

◆ `AndroiView.java` 의 `onKeyDown(int, KeyEvent)` method 추가





Android KeyEvent 처리 구현 (1)

```
*AndroiView.java X
1 package com.inhatc.android_Image;
2
3 import android.content.Context;
4 import android.graphics.Canvas;
5 import android.graphics.Color;
6 import android.graphics.drawable.Drawable;
7 import android.util.AttributeSet;
8 import android.view.KeyEvent;
9 import android.view.View;
10
11 public class AndroiView extends View {
12     Drawable imgAndroi;           //image 객체 선언
13     int ix, iy;                   //Width, Height 저장
14     int imgWidth, imgHeight;     //Image Width, Height 저장
15
16     public AndroiView(Context context, AttributeSet attrs) {
17         super(context, attrs);
18         // TODO Auto-generated constructor stub
19         imgAndroi=this.getResources().getDrawable(R.drawable.androi);
20         imgWidth = imgAndroi.getIntrinsicWidth();
21         imgHeight = imgAndroi.getIntrinsicHeight();
22     }
23
24     @Override
25     protected void onDraw(Canvas canvas) {
26
27         canvas.drawColor(Color.parseColor("#FFFFFF")); //Canvas 배경색 (white)
28         imgAndroi.setBounds(ix, iy, ix+imgWidth, iy+imgHeight);
29         imgAndroi.draw(canvas);
30
31         super.onDraw(canvas);
32     }
}
```

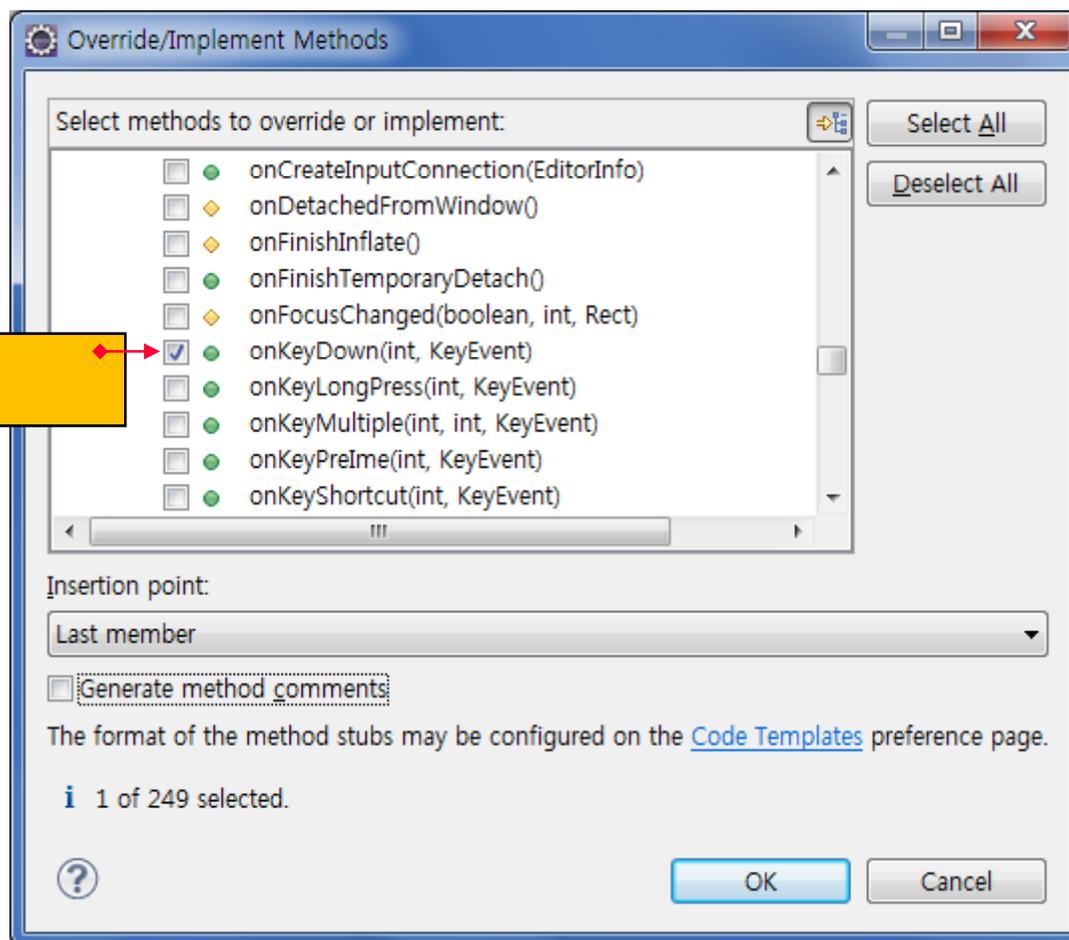
Coding





Android KeyEvent 처리 구현 (2)

■ onKeyDown(int, KeyEvent) method 추가



선택





Android KeyEvent 처리 구현 (3)

```
AndroiView.java X
34 @Override
35 public boolean onKeyDown(int keyCode, KeyEvent event) {
36     // TODO Auto-generated method stub
37     switch(keyCode)
38     {
39         case KeyEvent.KEYCODE_DPAD_LEFT:
40             ix -= 15;
41             break;
42         case KeyEvent.KEYCODE_DPAD_RIGHT:
43             ix += 15;
44             break;
45         case KeyEvent.KEYCODE_DPAD_UP:
46             iy -= 15;
47             break;
48         case KeyEvent.KEYCODE_DPAD_DOWN:
49             iy += 15;
50             break;
51     }
52     this.invalidate();
53     return super.onKeyDown(keyCode, event);
54 }
55 @Override
56 protected void onSizeChanged(int w, int h, int oldw, int oldh) {
57     // TODO Auto-generated method stub
58     ix = (this.getWidth()-imgWidth)/2;
59     iy = (this.getHeight()-imgHeight)/2;
60
61     super.onSizeChanged(w, h, oldw, oldh);
62 }
```

Coding





실습 III : Android Screen Touch Event 처리 구현

■ Android_Image (실습 시간 : 20분)

❖ 아래 그림과 같이 screen을 touch한 위치로 android(🤖) 이미지를 이동하도록 Coding 하시오.

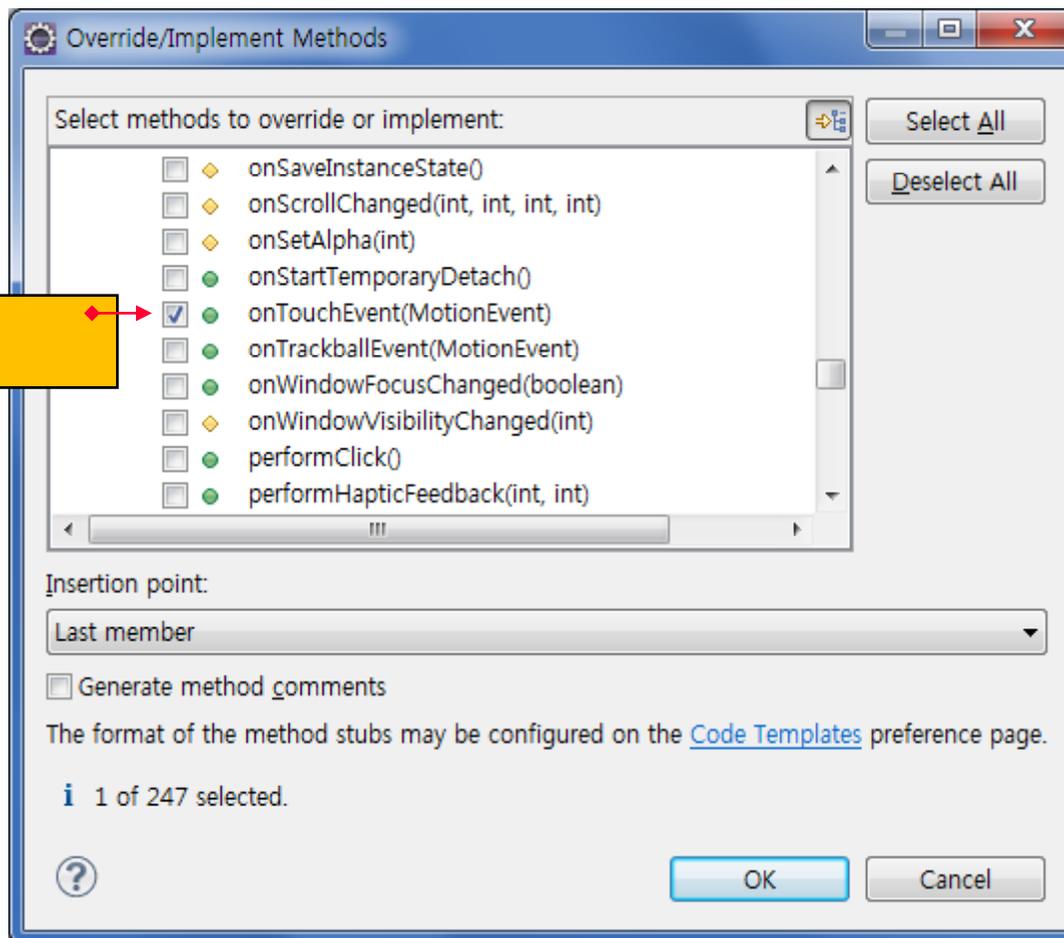
◆ AndroidView.java 의 onTouchEvent(MotionEvent, event) method 추가





Android Screen Touch Event 처리 구현 (1)

■ onKeyDown(int, KeyEvent) method 추가



선택





Android Screen Touch Event 처리 구현 (2)

❖ onKeyDown(int, KeyEvent) method 추가

```
*AndroidView.java
62
63 @Override
64 public boolean onTouchEvent(MotionEvent event) {
65     // TODO Auto-generated method stub
66     ix = (int)event.getX();    //Touch 한 Screen 위치의 X 좌표
67     iy = (int)event.getY();    //Touch 한 Screen 위치의 Y 좌표
68     this.invalidate();
69
70     return super.onTouchEvent(event);
71 }
72 }
```

Coding





학습 요약

- CustomView 구현
- Keyboard 입력 처리
- Touch Screen 처리
- 실습 I
 - ❖ Android Image 화면 중앙 출력 구현
- 실습 II
 - ❖ Android KeyEvent 처리 구현
- 실습 III
 - ❖ Android Screen Touch Event 처리 구현



open handset alliance

