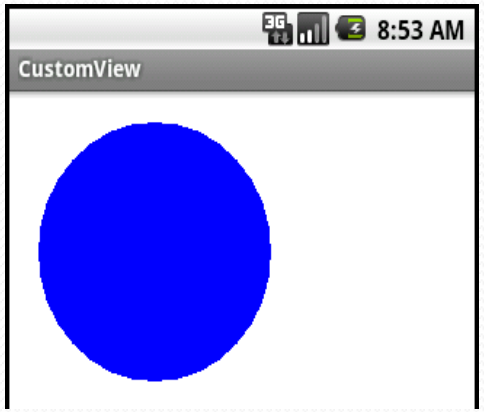
2023 안드로이드
Android Application 01

대전대 온디바이스 AI 응용시스템 개발자
하성호

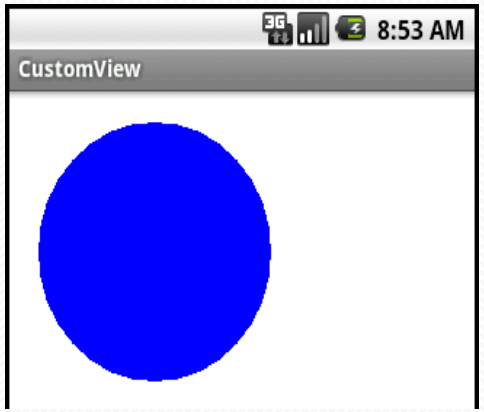
[실습]원 그리기

```
public class LinearLayoutTest extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MyView myview = new MyView(this);
        setContentView(myview);
    }
}
```



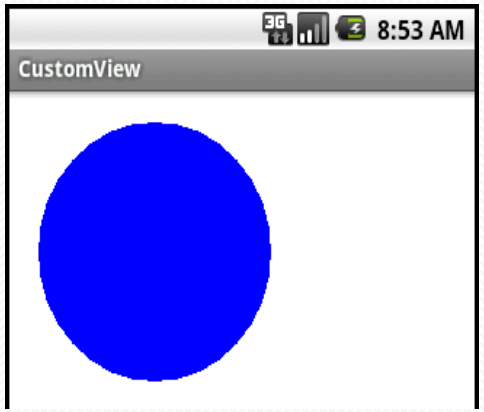
[실습]원 그리기

```
public class LinearLayoutTest extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MyView myview = new MyView(this);
        setContentView(myview);
    }
    protected class MyView extends View{
        /*생성자*/
        public MyView(Context context) {
            super(context);
        }
}
}
```



[실습]원 그리기

```
public class LinearLayoutTest extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MyView myview = new MyView(this);
        setContentView(myview);
    }
    protected class MyView extends View{
        /*생성자*/
        public MyView(Context context) {
            super(context);
        }
        @Override
        protected void onDraw(Canvas canvas) {
            // TODO Auto-generated method stub
            Paint paint = new Paint();
            paint.setColor(Color.BLUE);
            canvas.drawColor(Color.WHITE);
            canvas.drawCircle(100, 100, 80, paint);
            super.onDraw(canvas);
        }
    }
}
```



Canvas(캔버스)

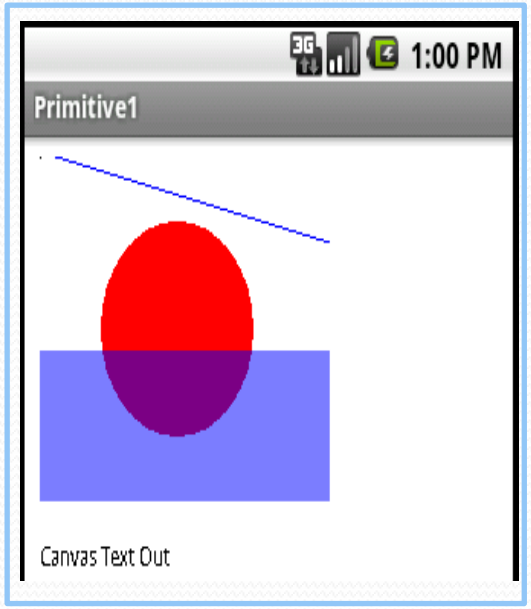
- 그리기 표면이며 **onDraw**의 인수로 전달
  - 이 메서드를 재정의하려면 **View**를 상속받아야 함
- 캔버스에는 다양한 그리기 메서드을 제공

```
void drawPoint (float x, float y, Paint paint)
void drawLine (float startX, float startY, float stopX, float stopY, Paint paint)
void drawCircle (float cx, float cy, float radius, Paint paint)
void drawRect (float left, float top, float right, float bottom, Paint paint)
void drawText (String text, float x, float y, Paint paint)
```

- 모든 그리기 메서드의 마지막 인수는 항상 Paint 객체이다.
- **Paint**는 색상, 글꼴, 스타일, 그리기 모드 등의 정보를 지정하며 그려진다.
- 색상에는 알파를 지정하여 반투명 출력이 가능하다.

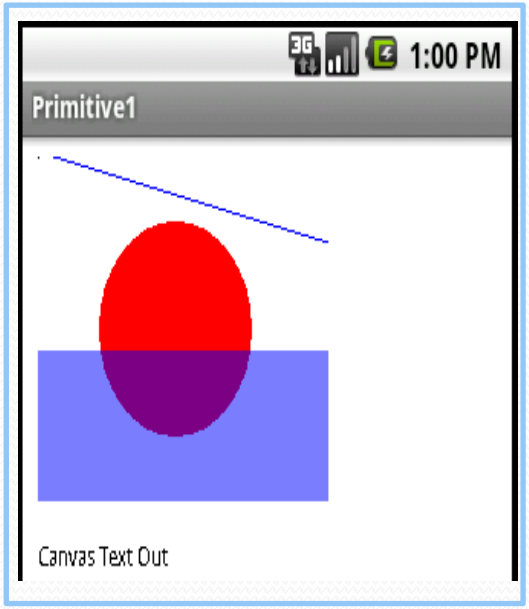
선그리기

```
import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.View;

public class ShapeTest extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ShapeView sv = new ShapeView(this);
        setContentView(sv);
    }
    protected class ShapeView extends View{
        public ShapeView(Context context){
            super(context);
            // TODO Auto-generated constructor stub
        }
        @Override
        protected void onDraw(Canvas canvas) {
            // TODO Auto-generated method stub
            Paint paint = new Paint();
            super.onDraw(canvas);
        }
    }
}
```



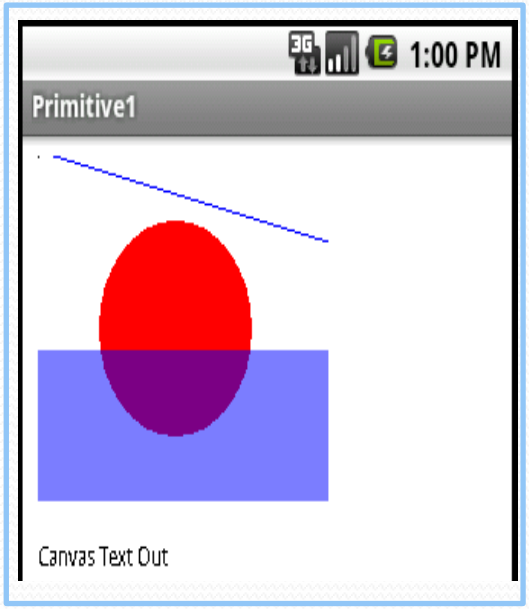
선그리기

```
@Override
        protected void onDraw(Canvas canvas) {
            // TODO Auto-generated method stub
            Paint paint = new Paint();
            canvas.drawColor(Color.WHITE); //배경
            canvas.drawPoint(10, 10, paint); //점
            paint.setColor(Color.RED);
            canvas.drawLine(20, 10, 200, 50, paint);//라인
        }
```



선그리기

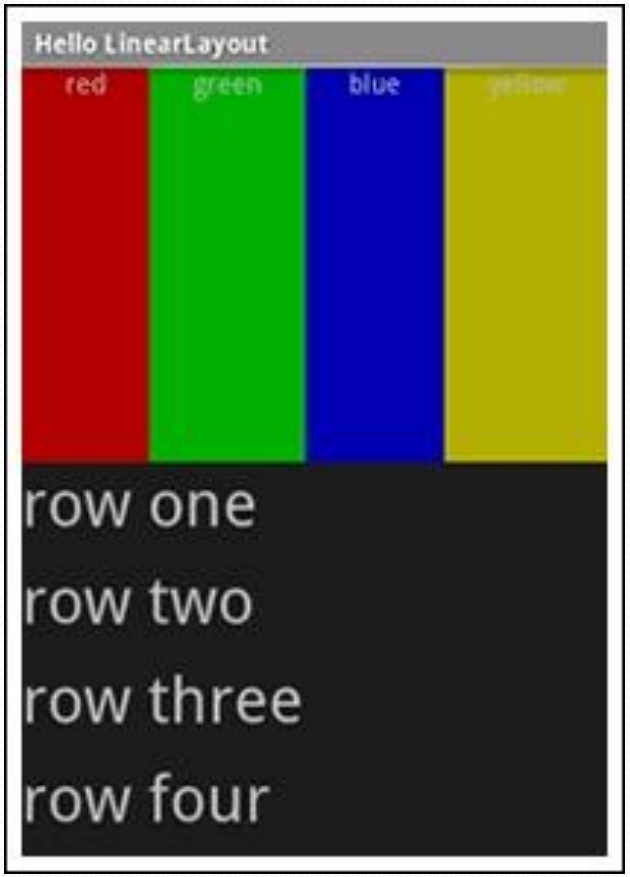
```
@Override
    protected void onDraw(Canvas canvas) {
        // TODO Auto-generated method stub
        Paint paint = new Paint();
        canvas.drawColor(Color.WHITE); //배경
        canvas.drawPoint(10, 10, paint); //점
        paint.setColor(Color.BLUE);
        canvas.drawLine(20, 10, 200, 50, paint);//라인
        paint.setColor(Color.RED);
        canvas.drawCircle(100, 90, 50, paint);
        paint.setColor(0x800000ff);
        canvas.drawRect(10, 100, 200, 170, paint);
        paint.setColor(Color.BLACK);
        canvas.drawText("Canvas text out", 10,
            200, paint);
    }
```



레이아웃

레이아웃

- LinearLayout



LinearLayout(1)

```
<?xml version="1.0" encoding="utf-8"?>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
       android:orientation="vertical"
       android:layout_width="fill_parent”    android:layout_height="fill_parent">

      <LinearLayout
         android:orientation="horizontal“
         android:layout_width="fill_parent“      android:layout_height="fill_parent"
         android:layout_weight="1">
         <TextView
            android:text="red"
            android:gravity="center_horizontal"
            android:background="#aa0000"
            android:layout_width="wrap_content"  android:layout_height="fill_parent"
            android:layout_weight="1"/>
```

LinearLayout(2)

```
<TextView
        android:text="green“   android:gravity="center_horizontal"
        android:background="#00aa00“  android:layout_width="wrap_content"
        android:layout_height="fill_parent“   android:layout_weight="1"/>
    <TextView
        android:text="blue“   android:gravity="center_horizontal"
        android:background=“#0000aa“  android:layout_width="wrap_content"
        android:layout_height="fill_parent“   android:layout_weight="1"/>
    <TextView
        android:text="yellow"  android:gravity="center_horizontal"
        android:background="#aaaa00"  android:layout_width="wrap_content"
        android:layout_height="fill_parent"  android:layout_weight="1"/>
</LinearLayout>
```

LinearLayout(3)

```
<LinearLayout
    android:orientation="vertical“  android:layout_width="fill_parent“
    android:layout_height="fill_parent“   android:layout_weight="1">
    <TextView   android:text="row one“   android:textSize="15pt"
      android:layout_width="fill_parent“    android:layout_height="wrap_content"
      android:layout_weight="1"/>
    <TextView   android:text="row two”  android:textSize="15pt"
      android:layout_width="fill_parent“   android:layout_height="wrap_content"
      android:layout_weight="1"/>
    <TextView   android:text="row three“   android:textSize="15pt"
      android:layout_width="fill_parent“   android:layout_height="wrap_content"
      android:layout_weight="1"/>
    <TextView  android:text="row four“   android:textSize="15pt"
      android:layout_width="fill_parent“  android:layout_height="wrap_content"
      android:layout_weight="1"/>
  </LinearLayout> </LinearLayout>
```

레이아웃

- Relative Layout

Relative Layout(1)

```
<?xml version="1.0" encoding="utf-8"?>
    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TextView
            android:id="@+id/label"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Type here:"/>
        <EditText
            android:id="@+id/entry"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:background="@android:drawable/editbox_background"
            android:layout_below="@id/label"/>
```

Relative Layout(2)

```
<Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/entry"
        android:layout_alignParentRight="true"
        android:layout_marginLeft="10dip"
        android:text="OK" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toLeftOf="@id/ok"
        android:layout_alignTop="@id/ok"
        android:text="Cancel" />
</RelativeLayout>
```

레이아웃

- Table layout

Table layout(2)

```
<?xml version="1.0" encoding="utf-8"?>
    <TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:stretchColumns="1">

        <TableRow>
            <TextView
                android:layout_column="1"
                android:text="Open..."
                android:padding="3dip" />
            <TextView
                android:text="Ctrl-O"
                android:gravity="right"
                android:padding="3dip" />
        </TableRow>
```

Table layout(1)

```
<TableRow>
        <TextView
           android:layout_column="1”   android:text="Save..."
           android:padding="3dip" />
        <TextView
           android:text="Ctrl-S“          android:gravity="right"
           android:padding="3dip" />
     </TableRow>
     <TableRow>
        <TextView
           android:layout_column="1"
           android:text="Save As... "    android:padding="3dip" />
        <TextView
           android:text="Ctrl-Shift-S"
           android:gravity="right”          android:padding="3dip" />
     </TableRow>
     <View
        android:layout_height="2dip"   android:background="#FF909090" />
```

Table layout(3)

```
<TableRow>
        <TextView   android:text="X"   android:padding="3dip" />
        <TextView
           android:text="Import... "  android:padding="3dip" />
    </TableRow>
    <TableRow>
        <TextView    android:text="X“   android:padding="3dip" />
        <TextView    android:text="Export... "  android:padding="3dip" />
        <TextView    android:text="Ctrl-E“   android:gravity="right"
           android:padding="3dip" />
    </TableRow>
    <View  android:layout_height="2dip“  android:background="#FF909090" />
    <TableRow>
        <TextView   android:layout_column="1“  android:text="Quit"
           android:padding="3dip" />
    </TableRow>
</TableLayout>
```

CustomView

안드로이드 프로젝트 생성

- 다음 내용을 입력
- [완료] 선택

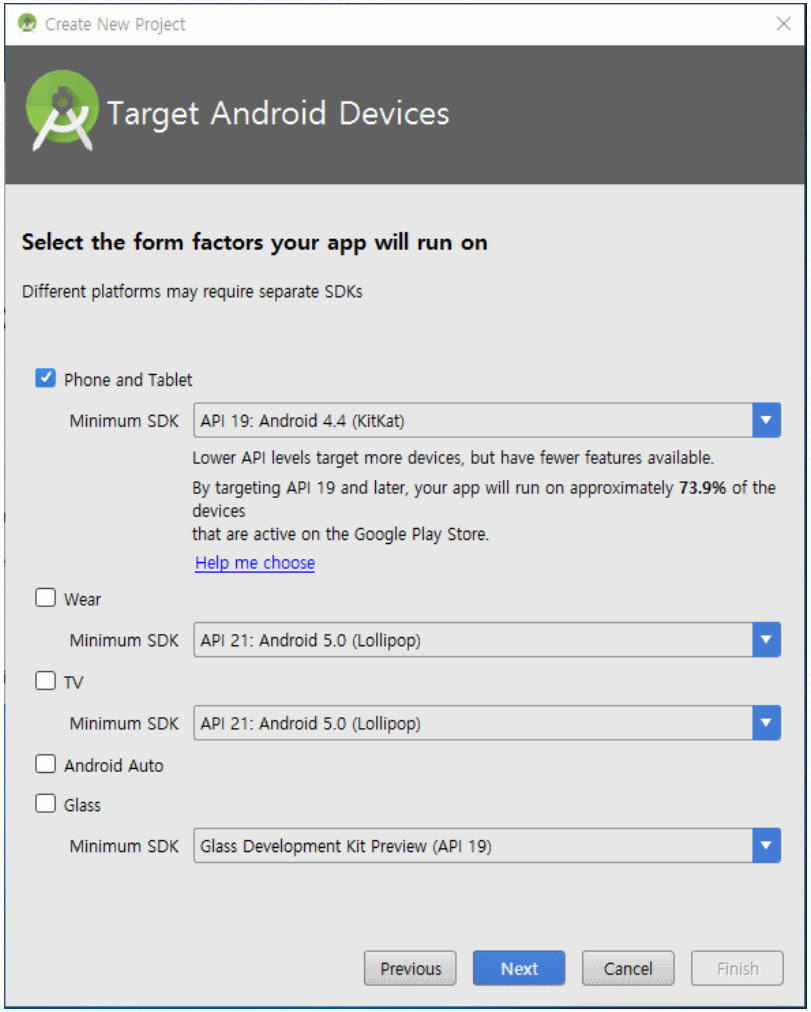
[Project Name]
TestCustomView

[Minimum SDK]
API 19 : Android 4.4 (KitKat)

[Application Name]
TestCustomView

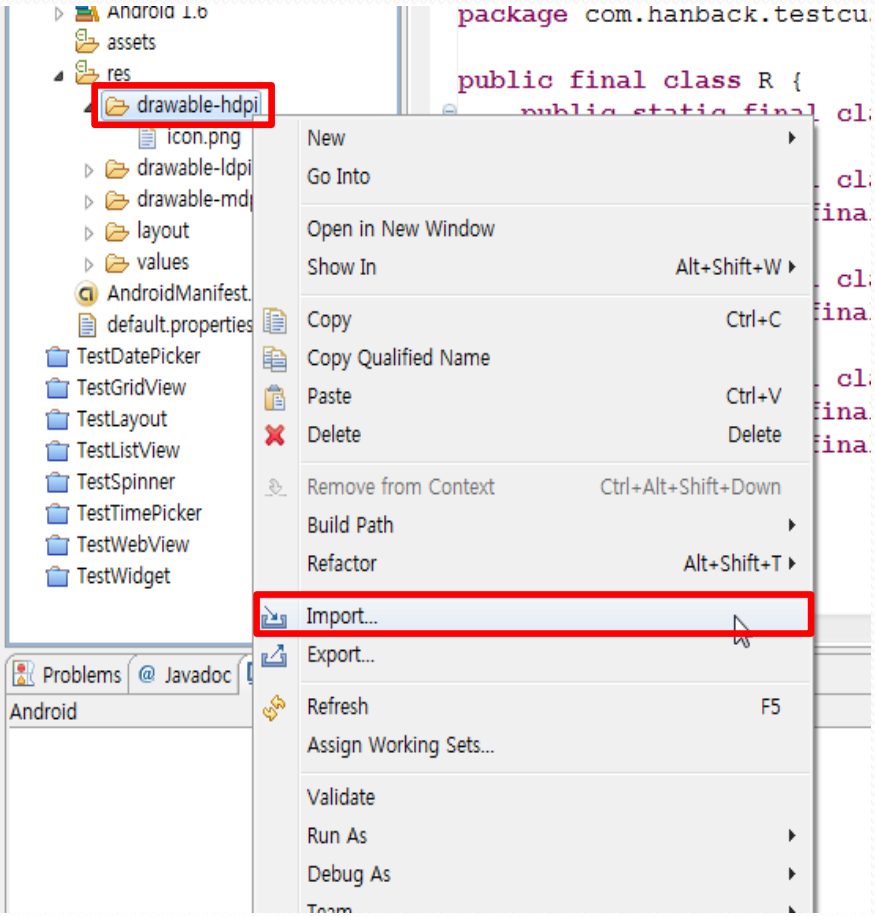
[Company Domain]
Usnschool.com

[Activity Name]
MainActivity



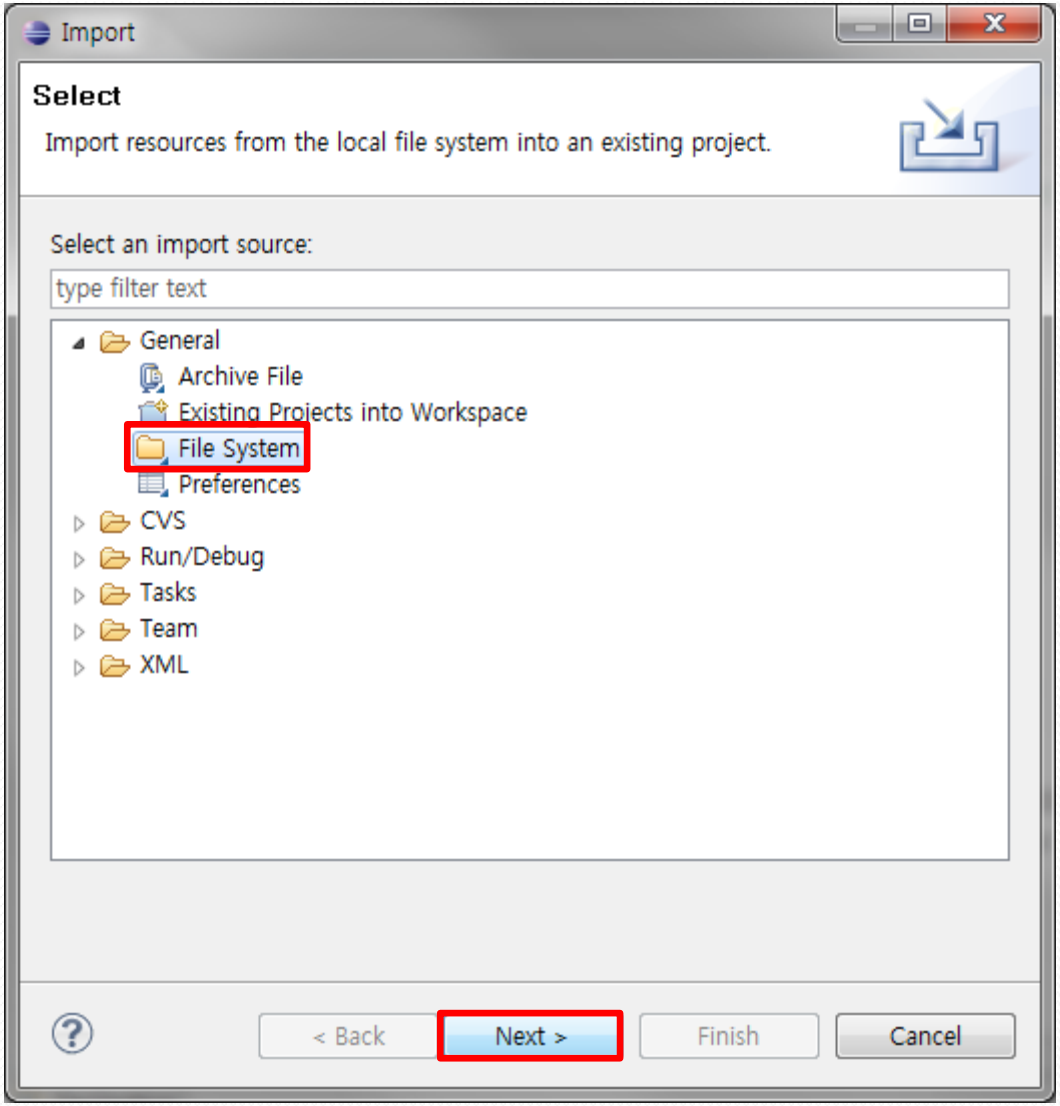
출력할 이미지 가져오기

- [res] > [drawable-hdpi] 오른쪽 버튼 클릭 – Import 선택



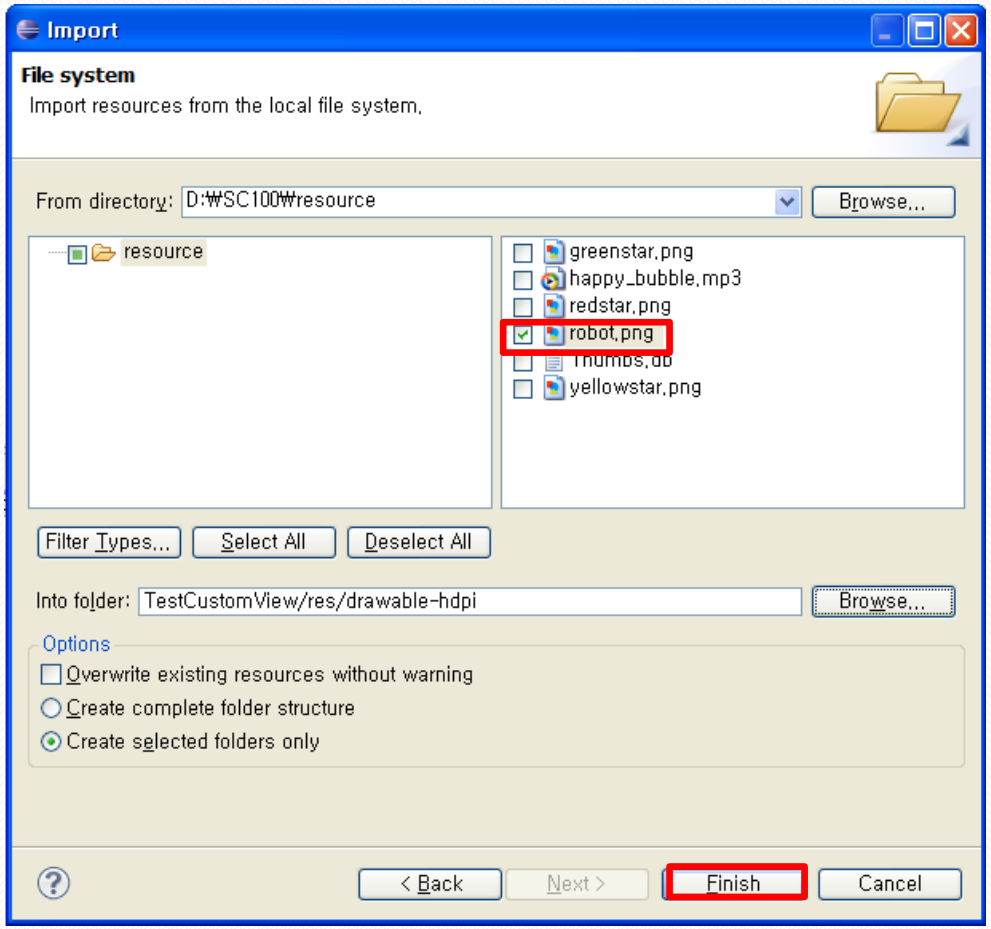
출력할 이미지 가져오기

- [General] – [File System] 선택 & [Next]



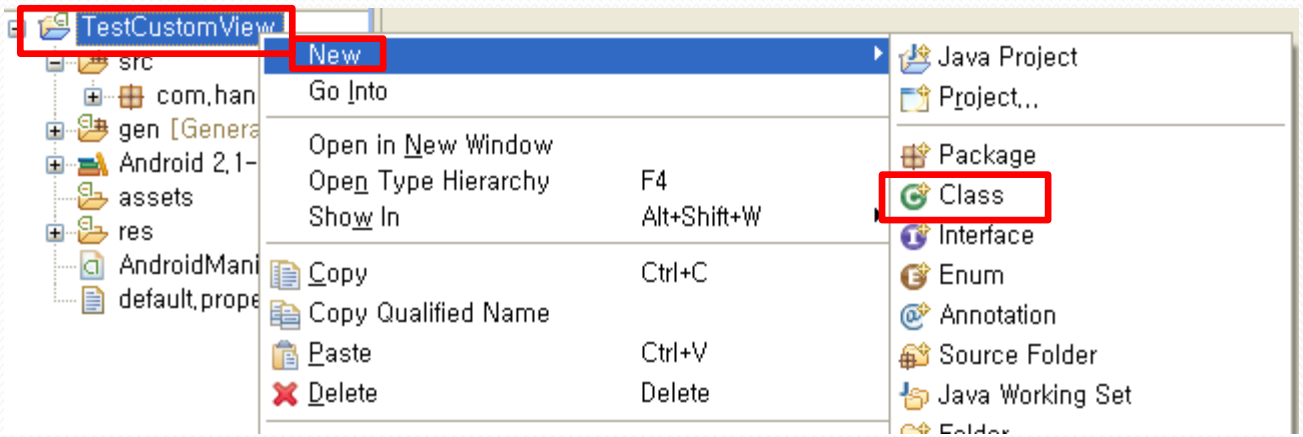
출력할 이미지 가져오기

- [Android2.1]/resource/robot.png 파일 추가
- [Finish] 선택



사용자 정의 View를 위한 클래스 추가

- src/com.hn.testcustomview 오른쪽 버튼
- [New] – [Class] 선택

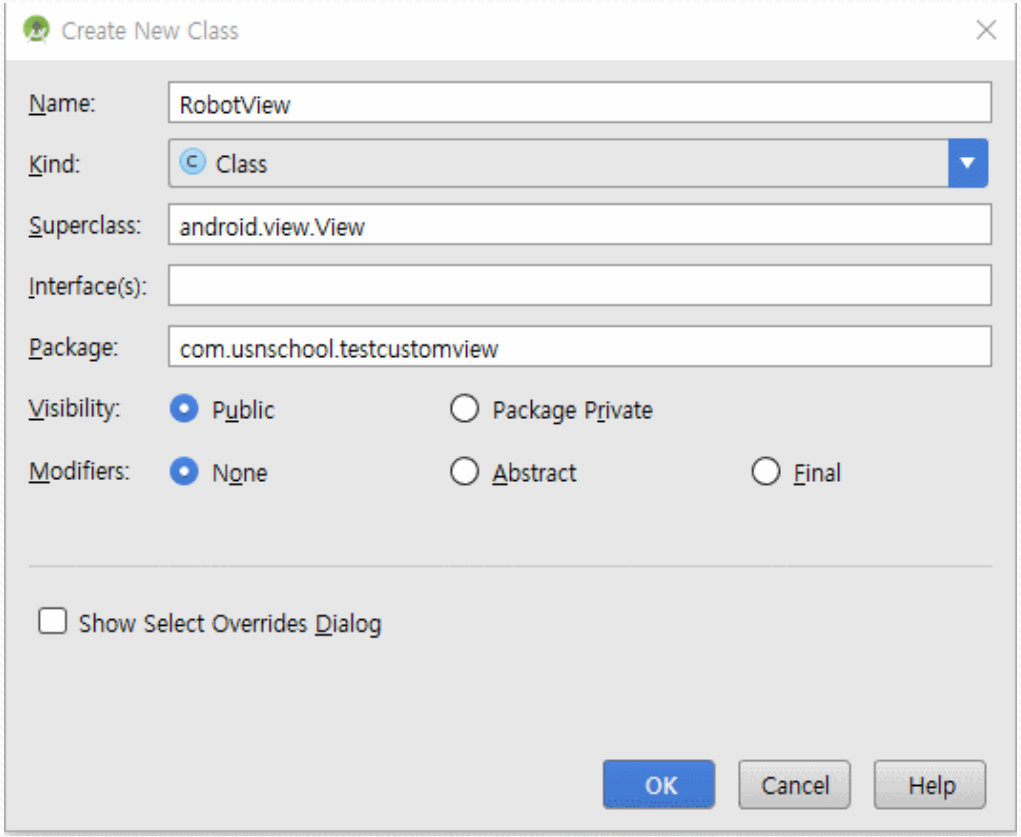


사용자 정의 View를 위한 클래스 추가

패키지 :
com.usnschool.testcustomview

이름 :
RobotView

수퍼클래스 :
android.view.View



RobotView클래스 생성자 추가

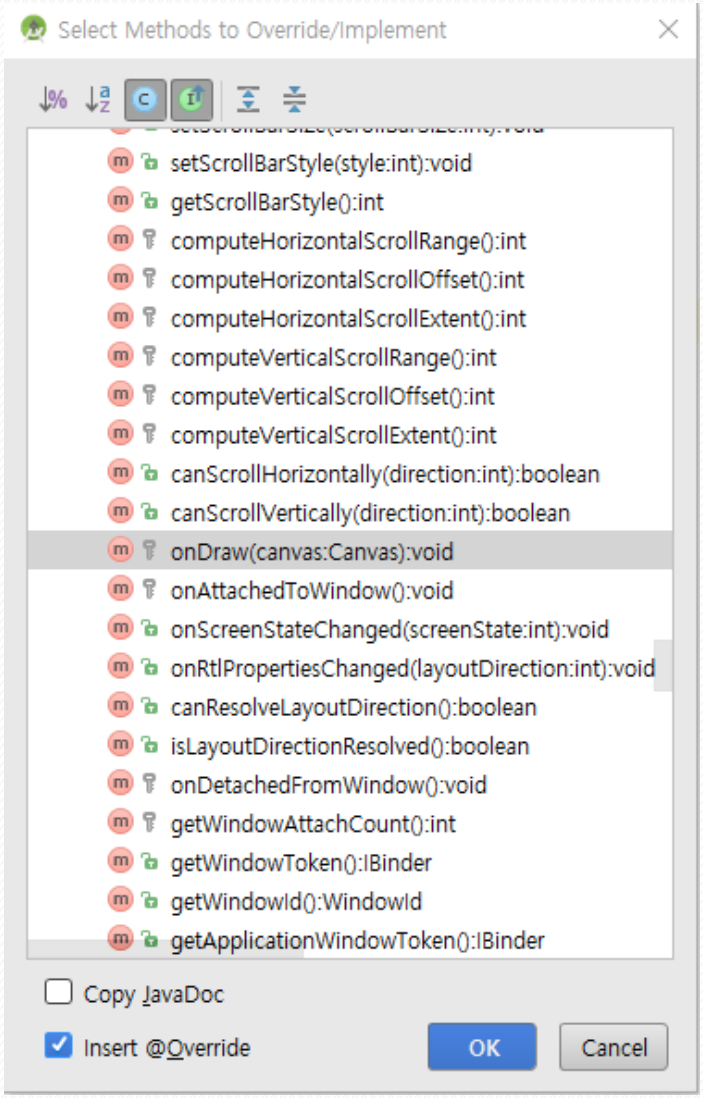
- RobotView(Context, AttributeSet) 생성자 추가

: package com.usnschol.testcustomview;

```
import android.content.Context;
import android.graphics.Canvas;
import android.util.AttributeSet;
import android.view.View;

public class RobotView extends View {


        public RobotView(Context context, AttributeSet attrs) {
                super(context, attrs);

        }
```

RobotView 클래스 멤버함수 추가

- onDraw(Canvas) 선택 – [OK] 선택



생성된 코드 결과

```
package com.usnschool.testcustomview;

import android.content.Context;
import android.graphics.Canvas;
import android.util.AttributeSet;
import android.view.View;


public class RobotView extends View {


    public RobotView(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
    }
}
```

코드 추가

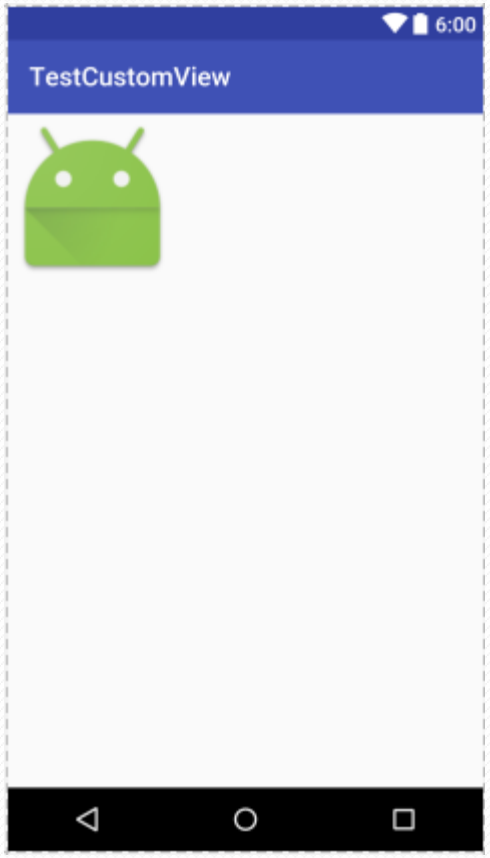
```
package com.usnschool.testcustomview;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.drawable.Drawable;
import android.support.v4.content.res.ResourcesCompat;
import android.util.AttributeSet;
import android.view.View;


public class RobotView extends View {

    private Drawable image;

    public RobotView(Context context, AttributeSet attrs) {
        super(context, attrs);

       // image = getResources().getDrawable(R.drawable.ic_launcher);
        image = ResourcesCompat.getDrawable(getResources(),
R.drawable.ic_launcher, null);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
    }
}
```

코드 추가

```
019:        @Override
020:        protected void onDraw(Canvas canvas) {
021:
022:                image.setBounds(0, 0, 128, 128);
023:                image.draw(canvas);
024:
025:                super.onDraw(canvas);
026:        }
027:
028: }
029:
```

main.xml 파일 수정

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="match_parent"
              android:layout_height="match_parent">

  <com.usnschool.testcustomview.RobotView

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    />

</LinearLayout>
```



실행 결과 화면

소스코드 수정

- 이미지를 화면 가운데 위치로 이동
  - RobotView.java

```
018:
019:        @Override
020:        protected void onDraw(Canvas canvas) {
021:                    int viewWidth = this.getWidth();
022:                    int viewHeight = this.getHeight();
023:
024:                    int imageWidth = image.getIntrinsicWidth();
025:                    int imageHeight = image.getIntrinsicHeight();
026:
027:                    int x = viewWidth / 2 - imageWidth / 2;
028:                    int y = viewHeight / 2 - imageHeight / 2;
029:
030:                    image.setBounds(x, y, x + imageWidth, y + imageHeight);
031:                    image.draw(canvas);
032:
033:                    super.onDraw(canvas);
034:        }
035:
```

실행 결과 화면

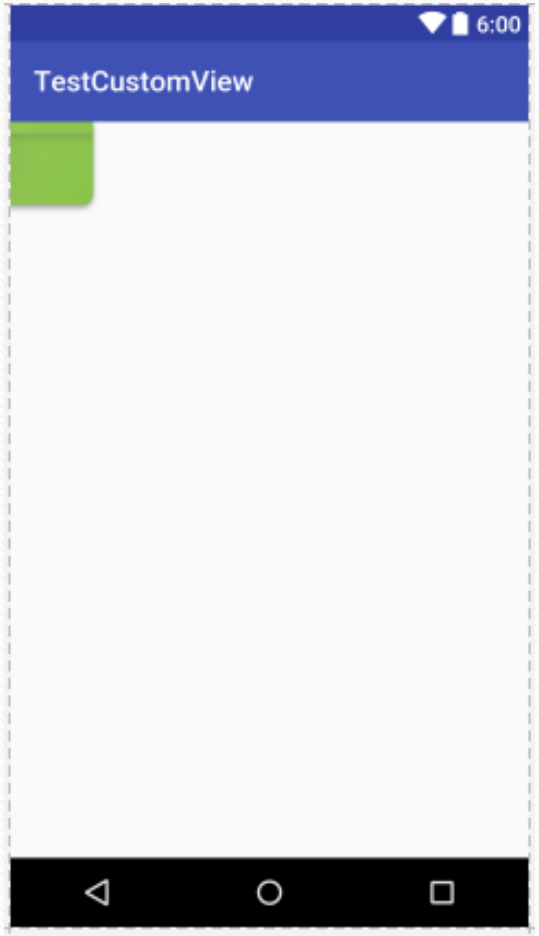
시작 시 가운데 출력을 위한 초기화

- onDraw()에 있는 초기 코드를 생성자로 옮김
- 생성자 RobotView()와 onDraw()에서 사용하는 변수; 멤버 변수로!

```
010: public class RobotView extends View {
011:
012:        private Drawable image;
013:        private int viewWidth, viewHeight;
014:        private int imageWidth, imageHeight;
015:        private int x, y;
016:
017:        public RobotView(Context context, AttributeSet attrs) {
018:                super(context, attrs);
019:
020:                image = this.getResources().getDrawable(R.drawable.robot);
021:
022:                viewWidth = this.getWidth();
023:                viewHeight = this.getHeight();
024:
025:                imageWidth = image.getIntrinsicWidth();
026:                imageHeight = image.getIntrinsicHeight();
027:
028:                x = viewWidth / 2 - imageWidth / 2;
029:                y = viewHeight / 2 - imageHeight / 2;
030:        }
```

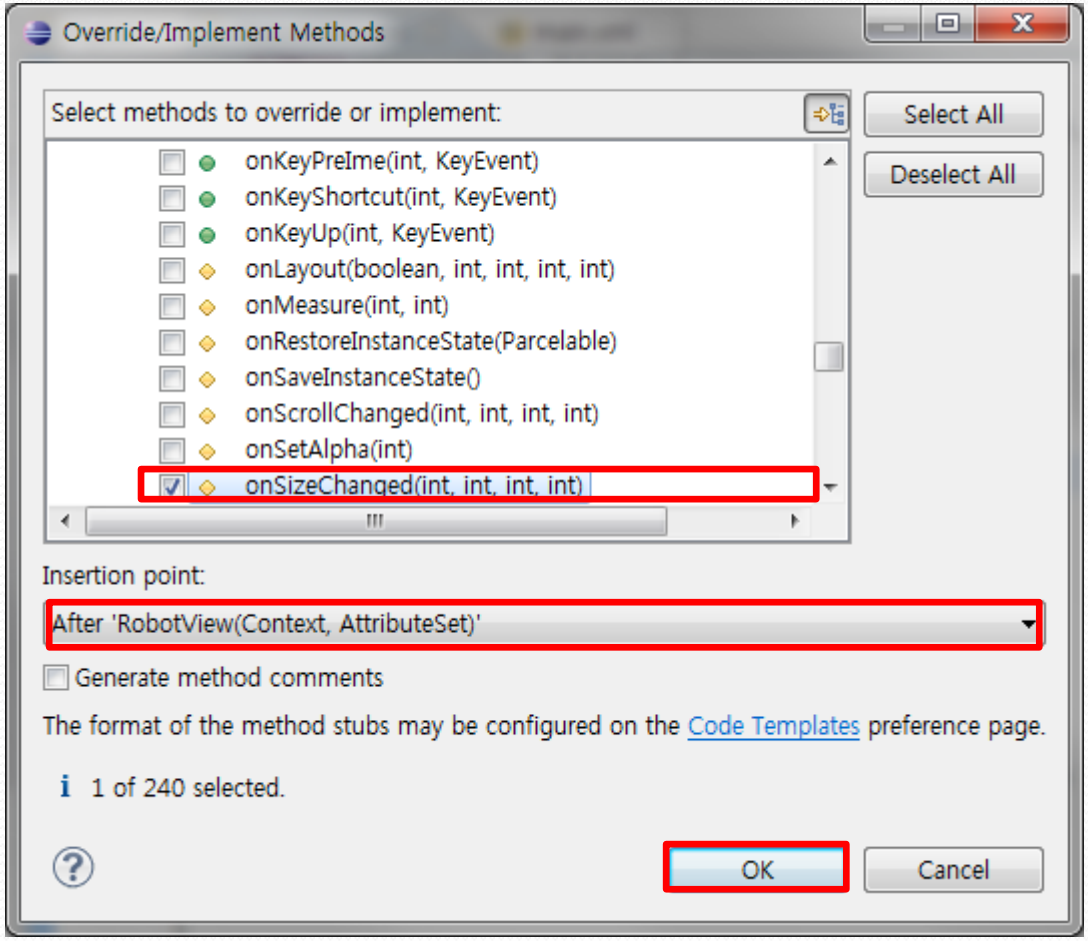
실행 결과 화면

- 문제발생
- this.getWidth(), this.getHeight() 문제!
  - 함수 결과값 : 0



[해법]PG 초기화시 화면 가운데 이미지 위치

- onSizeChanged() 메소드 생성자 밑에 추가



코드 수정

- 생성자에 있는 초기화 코드를 onSizeChanged()로.

```
016:
017:        public RobotView(Context context, AttributeSet attrs) {
018:                    super(context, attrs);
019:
020:                    image = this.getResources().getDrawable(R.drawable.robot);
021:        }
022:
023:        @Override
024:        protected void onSizeChanged(int w, int h, int oldw, int oldh) {
025:
026:                    viewWidth = this.getWidth();
027:                    viewHeight = this.getHeight();
028:
029:                    imageWidth = image.getIntrinsicWidth();
030:                    imageHeight = image.getIntrinsicHeight();
031:
032:                    x = viewWidth / 2 - imageWidth / 2;
033:                    y = viewHeight / 2 - imageHeight / 2;
034:
035:                    super.onSizeChanged(w, h, oldw, oldh);
036:        }
037:
```

onDraw()

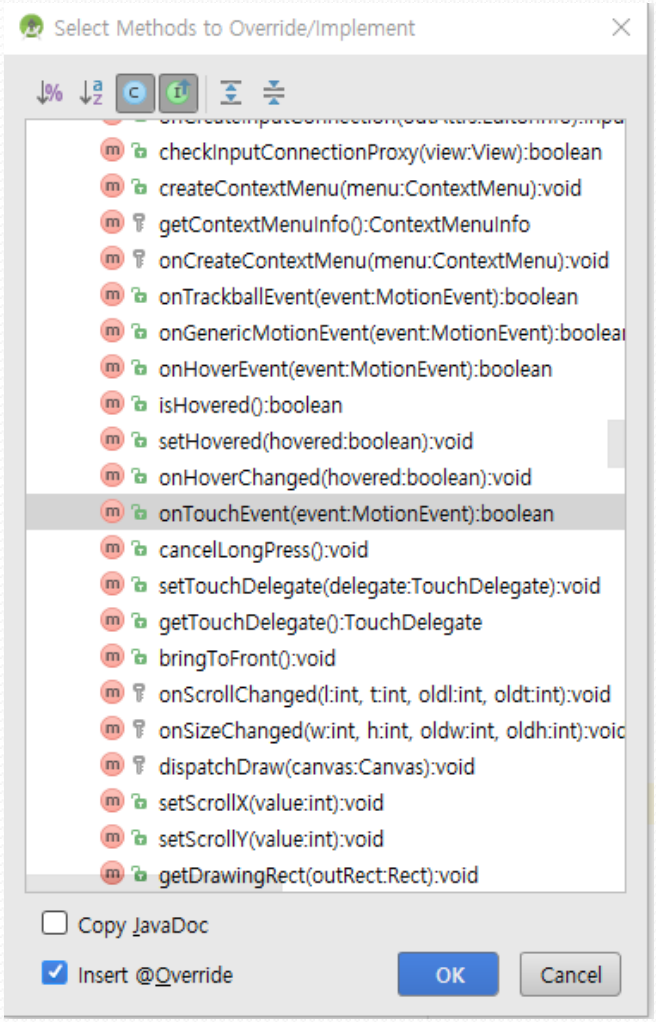
- 이미지를 화면 가운데 위치로 이동
  - RobotView.java

```
018:
019:        @Override
020:        protected void onDraw(Canvas canvas) {
021:
030:                    image.setBounds(x, y, x + imageWidth, y + imageHeight);
031:                    image.draw(canvas);
032:
033:                    super.onDraw(canvas);
034:        }
035:
```

실행 결과 화면

터치 이벤트 처리하기

- onTouchEvent(MotionEvent) 추가



터치 이벤트 처리를 위한 코드 수정

- onTouchEvent() 함수 코드 작성

```
042:        @Override
043:        public boolean onTouchEvent(MotionEvent event) {
044:                    x = (int)event.getX();
045:                    y = (int)event.getY();
046:                    this.invalidate();
047:
048:                    return true;
049:        }
```

터치 이벤트 처리를 위한 코드 수정

```
018:
019:        @Override
020:        protected void onDraw(Canvas canvas) {
021:                    viewWidth = this.getWidth();
022:                    viewHeight = this.getHeight();
023:
024:                    imageWidth = image.getIntrinsicWidth();
025:                    imageHeight = image.getIntrinsicHeight();
026:
027:                    //int x = viewWidth / 2 - imageWidth / 2;
028:                    //int y = viewHeight / 2 - imageHeight / 2;
029:
030:                    image.setBounds(x, y, x + imageWidth, y + imageHeight);
031:                    image.draw(canvas);
032:
033:                    super.onDraw(canvas);
034:        }
035:
```

떨어지는 로봇 (스레드 사용)

```
011: public class RobotView extends View implements Runnable {
012:         private final static int STEP = 10;
013:
014:         private Drawable image;
015:         private int viewWidth, viewHeight;
016:         private int imageWidth, imageHeight;
017:         private int x, y;
018:
019:         public RobotView(Context context, AttributeSet attrs) {
020:                 super(context, attrs);
021:
022:                 image = this.getResources().getDrawable(R.drawable.robot);
023:
024:                 Thread thread = new Thread(this);
025:                 thread.start();
026:         }
027:
```

코드 수정(스레드 사용)

```
080:
081:        @Override
082:        public void run() {
083:            for (;;) {
084:                    try {
085:                        Thread.sleep(500);
086:                        y = Math.min(viewHeight - imageHeight, y + STEP);
087:                                this.postInvalidate();
088:                    } catch (InterruptedException e) {
089:                        // TODO Auto-generated catch block
090:                        e.printStackTrace();
091:                    }
092:            }
093:        }
```