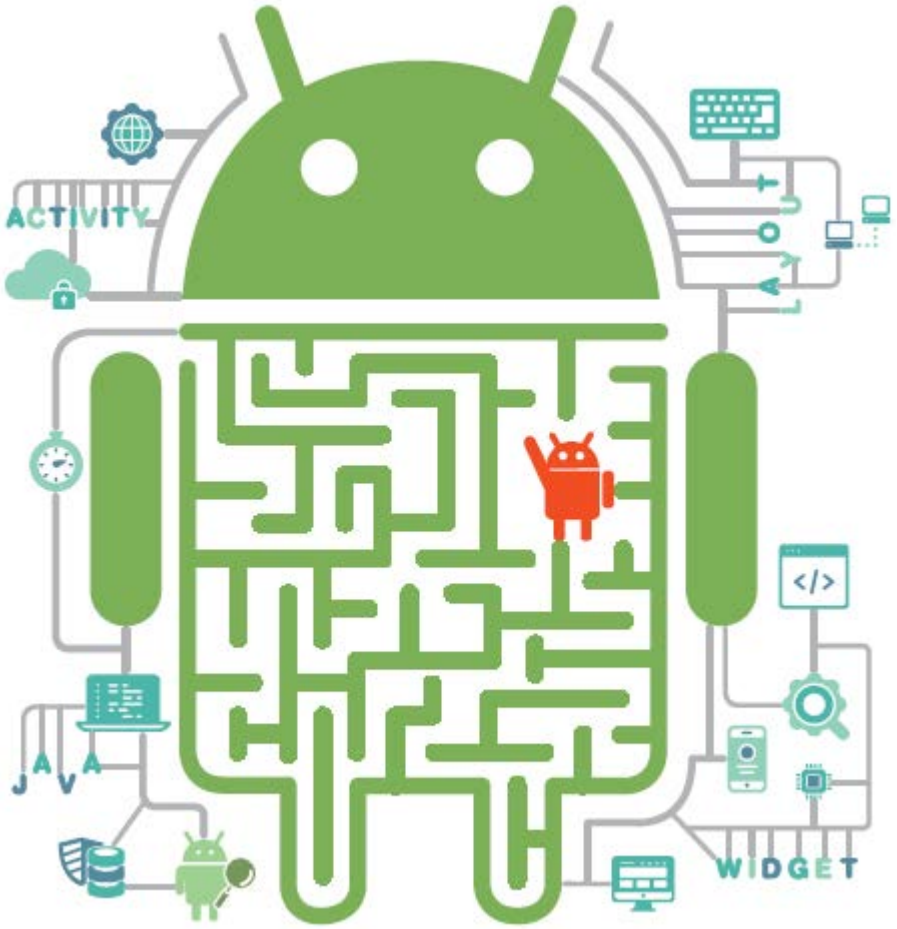
원리를 알면 IT가 맛있다
IT COOKBOOK

단계별로 배우는 안드로이드 프로그래밍

[강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미㈜에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 최고 5년 이하의 징역 또는 5천만원 이하의 벌금에 처할 수 있고 이를 병과(倂科)할 수도 있습니다.



단계별로 배우는

안드로이드 프로그래밍

Chapter 05. 그래픽 출력과 UI 이벤트 처리

한빛아카데미 Hanbit Academy, Inc.

목차

학습목표

- 그래픽 출력의 핵심 요소를 이해한다.
- 도형, 텍스트, 비트맵을 처리할 수 있다.
- UI 이벤트 처리 방식의 특징을 이해한다.
- 키와 터치 입력을 처리할 수 있다.

01 그래픽 출력 ▶ 캔버스와 페인트

- **안드로이드 앱의 그래픽 출력에 필요: 비트맵, 캔버스, 페인트**

- **비트맵(Bitmap)**
  - 그래픽 출력의 최종 결과인 픽셀(화소) 정보를 담는 메모리 영역
- **캔버스(Canvas)**
  - 비트맵에 그래픽 출력을 할 때 캔버스 객체를 사용. 캔버스 클래스는 다양한 그리기 API를 제공하므로 그릴 대상에 따라 적절한 메서드를 호출
- **페인트(Paint)**
  - 그래픽 출력에 사용할 색상과 각종 스타일(선 두께, 폰트 크기 등) 정보를 담고 있음. 똑같은 그리기 API를 호출하더라도 페인트 객체가 다르면 출력 결과물도 달라짐.

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-1 GraphicBasics

MainActivity 클래스 수정

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(R.layout.activity_main);
6           setContentView(new MyView(this));
7       }
8
9       private static class MyView extends View {
10          private Paint mPaint;
11
12          public MyView(Context context) {
13              super(context);
14              mPaint = new Paint();
15              mPaint.setAntiAlias(true);
16              mPaint.setColor(Color.RED);
17          }
18
19          @Override
20          protected void onDraw(Canvas canvas) {
21              //super.onDraw(canvas);
22              canvas.drawColor(Color.YELLOW);
23              canvas.drawCircle(100, 100, 80, mPaint);
24          }
25      }
26  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-1 GraphicBasics

MainActivity 클래스 수정

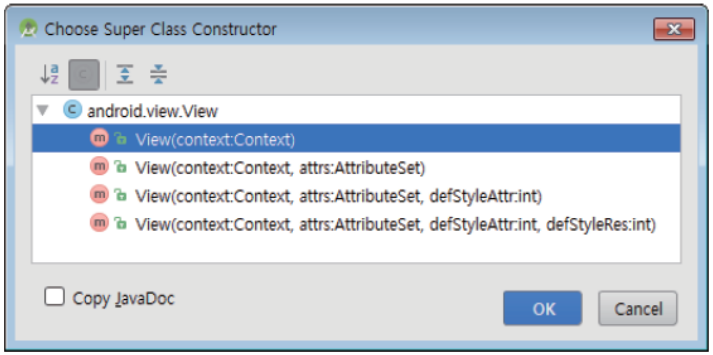
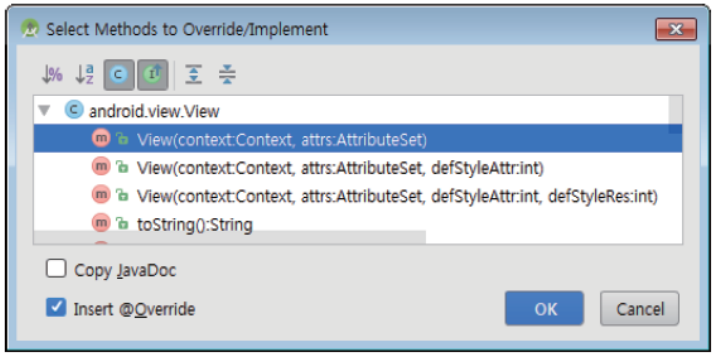
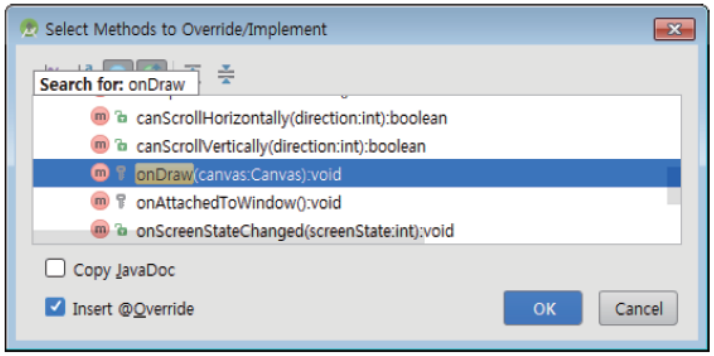


그림 5-1 수퍼클래스의 생성자를 토대로 서브클래스의 생성자 정의



(a) [Code] – [Override Methods…] 메뉴 선택



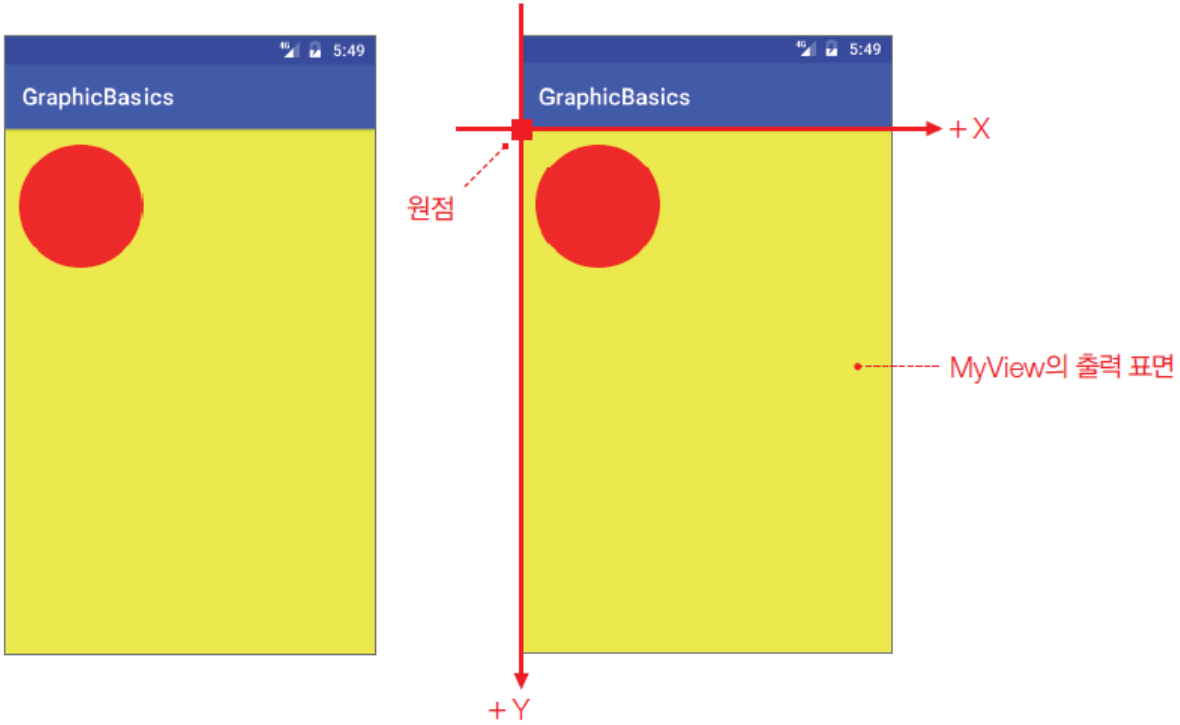
(b) onDraw를 입력하여 메서드가 선택되면 [OK] 클릭

그림 5-2 수퍼클래스의 OnDraw() 메서드 재정의

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-1 GraphicBasics

- (a) 실행 화면
- (b) 원점의 위치와 x, y 좌표축의 방향, MyView의 출력 표면



(a) 초기 화면 (b) 화면 분석

그림 5-3 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

페인트 객체

- 그래픽 출력에 사용될 여러 속성 정보를 제공함으로써 어떻게 출력할지를 결정

Paint 클래스에서 자주 사용되는 메서드

표 5-1 Paint 클래스의 메서드

구분	메서드	설명
초기화	void reset()	페인트 객체의 속성을 기본값으로 되돌린다.
앤티앨리어싱	void setAntiAlias(boolean aa)	그래픽 출력 시 가장자리의 계단 현상을 제거할지를 지정한다.
스타일	void setStyle(Paint.Style style)	스타일을 지정한다. STROKE(테두리만 그리기), FILL(내부만 채우기), FILL_AND_STROKE(테두리와 내부 모두 그리기)를 줄 수 있다. 도형과 텍스트 출력에 적용되며 비트맵 출력에는 영향을 주지 않는다.
색상	void setColor(int color) void setARGB(int a, int r, int g,int b)	출력에 사용할 색상을 지정한다. 스타일 속성에 따라 테두리 색상이 될 수도 있고 내부를 채우는 색상이 될 수도 있다.
선 두께	void setStrokeWidth(float width)	선 두께를 픽셀 단위로 지정한다.
텍스트 크기	void setTextSize(float textSize)	텍스트 크기를 픽셀 단위로 지정한다.
텍스트 정렬	void setTextAlign(Paint.Align align)	텍스트 정렬 방식을 지정한다. LEFT(왼쪽), CENTER(가운데), RIGHT(오른쪽)를 줄 수 있다.

원리를 알면 IT가 맛있다
IT COOKBOOK

Canvas 클래스의 메서드 중 기본적인 도형과 텍스트 출력 메서드

표 5-2 Canvas 클래스의 메서드

구분	메서드	설명

색으로 채우기	`drawColor(int color)`
	`drawRGB(int r, int g, int b)`
	`drawARGB(int a, int r, int g, int b)`
		캔버스 객체와 연결된 비트맵 전체를 주어진 색으로 채운다.

점 찍기	`drawPoint(float x, float y, Paint paint)`	(x, y) 좌표에 점을 찍는다.
직선 그리기	`drawLine(float startX, float startY, float stopX, float stopY, Paint paint)`	(startX, startY) – (stopX, stopY) 좌표를 잇는 직선을 그린다.

직사각형 그리기	`drawRect(float left, float top, float right, float bottom, Paint paint)`
	`drawRect(Rect r, Paint paint)`
	`drawRect(RectF rect, Paint paint)`
		직사각형을 그린다. 좌상단(left, top)과 우하단(right, bottom) 좌표를 직접 주거나, Rect 또는 RectF 객체로 직사각형 좌표를 전달할 수 있다.

원 그리기	`drawCircle(float cx, float cy, float radius, Paint paint)`	(cx, cy) 좌표를 중심으로 반지름이 radius인 원을 그린다.
타원 그리기	`drawOval(RectF oval, Paint paint)`	RectF가 나타내는 직사각형에 내접하는 타원을 그린다.
경로 그리기	`drawPath(Path path, Paint paint)`	Path 객체에 저장된 경로를 그린다.
텍스트 출력	`drawText(String text, float x, float y, Paint paint)`	텍스트를 (x, y) 좌표에 출력한다.

원리를 알면 IT가 맛있다
IT COOKBOOK

경로(Path)

- 직선과 곡선을 조합하여 복잡한 형태를 정의한 것

전형적인 사용 절차

```
/* ① 경로 객체 생성하기 */
Path path = new Path();
/* ② 경로 정의하기 */
path.moveTo(x1, y1); // (x1, y1) - (x2, y2) - (x3, y3)를 잇는 직선 그리기
path.lineTo(x2, y2);
path.lineTo(x3, y3);
path.addCircle(…); // 원 그리기
path.addRect(…); // 직사각형 그리기
path.quadTo(…); // 곡선 그리기
/* ③ 경로 사용하기 */
canvas.drawPath(path, mPaint); // 경로 자체를 그리기
canvas.drawTextOnPath(…); // 경로 위에 텍스트 출력하기
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-2 GraphicPrimitives

MainActivity 클래스 수정

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(new MyView(this));
7       }
8
9       private static class MyView extends View {
10          private Paint mPaint;
11
12          public MyView(Context context) {
13              super(context);
14              mPaint = new Paint();
15          }
16
17          @Override
18          protected void onDraw(Canvas canvas) {
19              // 페인트 객체 초기화
20              mPaint.reset();
21              mPaint.setAntiAlias(true);
22              // (1) 점 찍기
23              mPaint.setStrokeWidth(20);
24              canvas.drawPoint(30, 30, mPaint);
25              // (2) 선 그리기
26              mPaint.setStrokeWidth(8);
27              canvas.drawLine(50, 50, 200, 100, mPaint);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-2 GraphicPrimitives

MainActivity 클래스 수정

```
28              // (3) 도형 그리기: 직사각형, 원, 타원
29              mPaint.setStyle(Paint.Style.STROKE);
30              mPaint.setColor(Color.RED);
31              canvas.drawRect(250, 50, 450, 100, mPaint);
32              mPaint.setColor(Color.GREEN);
33              canvas.drawCircle(100, 200, 50, mPaint);
34              mPaint.setColor(Color.BLUE);
35              canvas.drawOval(new RectF(200, 150, 400, 250), mPaint);
36              // (4) 경로 그리기
37              mPaint.setColor(Color.CYAN);
38              Path path = new Path();
39              path.moveTo(50, 300);
40              path.lineTo(90, 310);
41              path.lineTo(110, 290);
42              path.lineTo(130, 330);
43              path.lineTo(160, 310);
44              canvas.drawPath(path, mPaint);
45              // (5) 텍스트 출력
46              mPaint.setStyle(Paint.Style.FILL);
47              mPaint.setColor(Color.MAGENTA);
48              mPaint.setTextSize(50);
49              mPaint.setTextAlign(Paint.Align.CENTER);
50              canvas.drawText("Hello", 100, 400, mPaint);
51          }
52      }
53  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-2 GraphicPrimitives

실행 화면

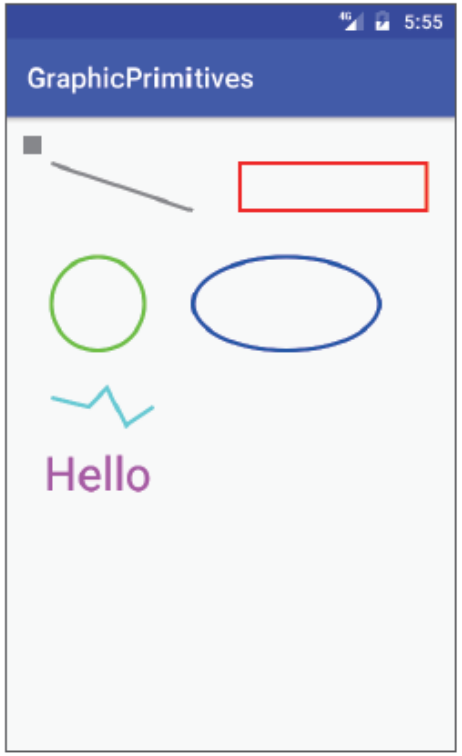


그림 5-4 실행 화면

01 그래픽 출력 ▶ 비트맵 출력

비트맵(Bitmap)

- 색상을 가진 점(Pixel)의 집합
- 일상 생활에서 디지털 카메라 등을 이용해 손쉽게 데이터를 생성

자바 코드로 비트맵 직접 만들기

```
static Bitmap createBitmap(int width, int height, Bitmap.Config config)
        // 수정 가능 Bitmap 객체 리턴
Bitmap copy(Bitmap.Config config, boolean isMutable)
        // 읽기 전용 또는 수정 가능 Bitmap 객체 리턴
```

원리를 알면 IT가 맛있다
IT COOKBOOK

그래픽 이미지를 읽어서 비트맵 만들기

- 그래픽 이미지를 읽어서 비트맵을 만드려면 .png, .jpg, .gif, .bmp, .webp 형식의 이미지 파일을 준비

```
static Bitmap decodeResource(Resources res, int id)
static Bitmap decodeResource(Resources res, int id, BitmapFactory.Options opts)
```

- 프로젝트의 res/drawable 폴더에 복사해두면 자바 코드에서 R.drawable.확장자 제외한 파일명 형태로 참조할 수 있음

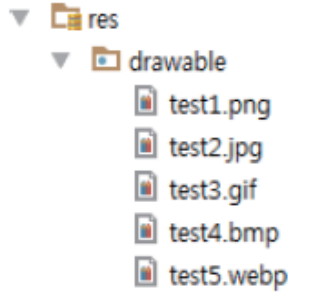


그림 5-5 이미지 파일

```
// 그래픽 이미지를 읽어서 비트맵 만들기
// → 이렇게 만들어진 비트맵은 읽기 전용이다!
Bitmap bitmap;
bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.test1);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

비트맵 속성을 얻거나 변경하기

- 생성된 비트맵 객체로부터 비트맵의 속성을 얻거나 변경 가능

자주 사용되는 Bitmap 클래스의 속성 관련 메서드

표 5-3 Bitmap 클래스의 메서드

구분	메서드	설명
색상 구성	`Bitmap.Config getConfig()` `void setConfig(Bitmap.Config config)`	비트맵의 색상 구성을 얻거나 변경한다. 색상 구성으로 가장 많이 사용되는 값은 Bitmap.Config.ARGB_8888이다.

폭과 높이	`int getWidth()`
	`int getHeight()`
	`void setWidth(int width)`
	`void setHeight(int height)`
		비트맵의 폭과 높이를 얻거나 변경한다.

픽셀값	`int getPixel(int x, int y)` `void setPixel(int x, int y, int color)`	(x, y) 위치의 픽셀값을 얻거나 변경한다.

원리를 알면 IT가 맛있다
IT COOKBOOK

비트맵 표면에 출력하기

- 생성된 비트맵 표면에 출력할 때는 캔버스 객체를 사용

전형적인 코드 형태

```
Bitmap bitmap = …; // ① 비트맵 객체를 생성한다.
Canvas canvas = new Canvas(mBitmap); // ② 캔버스 객체의 생성자로 비트맵 객체를 전달한다.
canvas.drawRect(…); // ③ Canvas 클래스의 다양한 메서드를 호출하여 비트맵 표면에 출력한다.
```

01 그래픽 출력 ▶ 비트맵 출력

- **비트맵 자체를 출력**
  - Canvas 클래스의 drawBitmap( ) 메서드를 사용

- **이름이 같으면서 전달 인자가 다른 여러 형태가 제공되는데 활용도가 높은 형태**

```
void drawBitmap(Bitmap bitmap, float left, float top, Paint paint)
void drawBitmap(Bitmap bitmap, Rect src, Rect dst, Paint paint)
```

01 그래픽 출력 ▶ 비트맵 출력

실습 5-3 BitmapBasics

- [그림 5-5]와 똑같은 이름의 이미지 파일을 종류별로 준비해서 res/drawable 폴더에 복사해두기

- MainActivity 클래스를 수정

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-3 BitmapBasics

MainActivity 클래스를 수정

```
6           setContentView(new MyView(this));
7       }
8
9       private static class MyView extends View {
10          private Paint mPaint;
11          private Bitmap mBitmap1;
12          private Canvas mCanvas1;
13          private Bitmap mBitmap2;
14          private Canvas mCanvas2;
15
16          public MyView(Context context) {
17              super(context);
18              mPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
19              mPaint.setStyle(Paint.Style.STROKE);
20              mPaint.setStrokeWidth(1);
21              // 자바 코드로 비트맵을 직접 생성하고 그 위에 도형 그리기
22              mBitmap1 = Bitmap.createBitmap(400, 300, Bitmap.Config.ARGB_8888);
23              mCanvas1 = new Canvas(mBitmap1);
24              mCanvas1.drawRect(0, 0, 400, 300, mPaint);
25              mCanvas1.drawCircle(150, 100, 50, mPaint);
26              // 그래픽 이미지를 읽어서 비트맵을 생성하고 그 위에 도형 그리기
27              mBitmap2 = BitmapFactory.decodeResource(getResources(), R.drawable.test1);
28              mBitmap2 = mBitmap2.copy(mBitmap2.getConfig(), true);
29              mCanvas2 = new Canvas(mBitmap2);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-3 BitmapBasics

MainActivity 클래스를 수정

```
30              mPaint.setColor(Color.RED);
31              mPaint.setStrokeWidth(10);
32              mCanvas2.drawLine(0, 0, mBitmap2.getWidth(), mBitmap2.getHeight(), mPaint);
33          }
34
35          @Override
36          protected void onDraw(Canvas canvas) {
37              canvas.drawColor(Color.YELLOW);
38              // (10, 10) 지점에 mBitmap1을 원래 크기(400x300)로 출력
39              canvas.translate(10, 10);
40              canvas.drawBitmap(mBitmap1, 0, 0, null);
41              // y 방향으로 10픽셀 이동 후 mBitmap2를 400x300 크기로 리사이징하여 출력
42              canvas.translate(0, mBitmap1.getHeight() + 10);
43              canvas.drawBitmap(mBitmap2, null, new Rect(0, 0, 400, 300), null);
44          }
45      }
46  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실행 화면

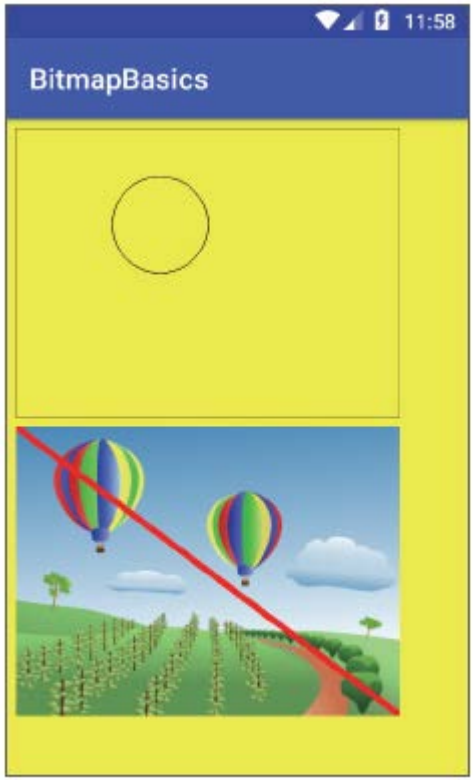


그림 5-6 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

전송 모드(transfer mode)

- 그래픽 출력 시 목적지 픽셀과 원본 픽셀이 어떻게 결합되는지 결정

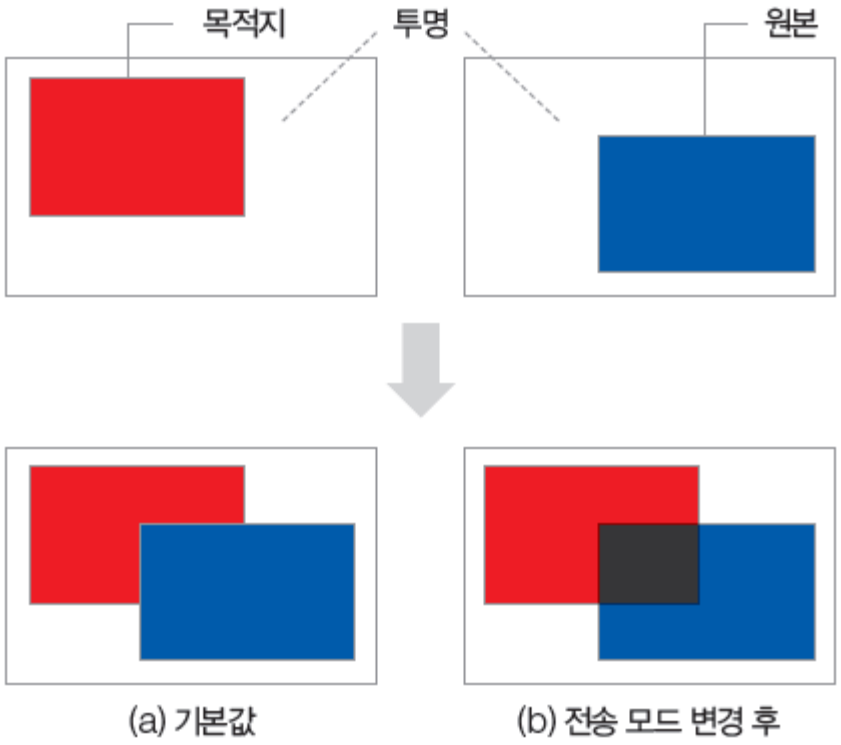


그림 5-7 전송 모드에 따른 출력 결과

01 그래픽 출력 ▶ 전송 모드

목적지에 원본을 전송하여 출력할 때 네 가지 영역

① 목적지 픽셀만 있는 영역
② 원본 픽셀만 있는 영역
③ 목적지와 원본 픽셀이 겹치는 영역
④ 나머지 영역([그림 5-7]에서 투명 영역)

Paint 클래스의 setXfermode( ) 메서드로 전송 모드 설정

```
Xfermode setXfermode(Xfermode xfermode)
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-4 Xfermode

- res/drawable 폴더에 그림 파일을 복사

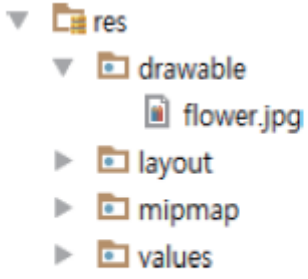


그림 5-8 그림 파일

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-4 Xfermode

MainActivity 클래스 수정

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(new MyView(this));
7       }
8
9       private static class MyView extends View {
10          private static final int WIDTH = 240;
11          private static final int HEIGHT = 240;
12          private Paint mPaint;
13          private Bitmap mBitmapDst;
14          private Bitmap mBitmapSrc;
15
16          public MyView(Context context) {
17              super(context);
18              setBackgroundResource(R.drawable.flower);
19              mPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
20              // destination 비트맵
21              mBitmapDst = Bitmap.createBitmap(WIDTH, HEIGHT, Bitmap.Config.ARGB_8888);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-4 Xfermode

MainActivity 클래스 수정

```
22              Canvas canvasDst = new Canvas(mBitmapDst);
23              mPaint.setColor(Color.argb(255, 255, 0, 0));
24              canvasDst.drawOval(new RectF(0, 0, WIDTH * 3 / 4, HEIGHT * 3 / 4), mPaint);
25              // source 비트맵
26              mBitmapSrc = Bitmap.createBitmap(WIDTH, HEIGHT, Bitmap.Config.ARGB_8888);
27              Canvas canvasSrc = new Canvas(mBitmapSrc);
28              mPaint.setColor(Color.argb(255, 0, 255, 0));
29              canvasSrc.drawRect(WIDTH / 3, HEIGHT / 3, WIDTH, HEIGHT, mPaint);
30          }
31
32          @Override
33          protected void onDraw(Canvas canvas) {
34              // 전송 모드를 원하는 값으로 설정
35              mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.ADD));
36              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.CLEAR));
37              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.DST));
38              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.DST_ATOP));
39              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.DST_IN));
40              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.DST_OUT));
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-4 Xfermode

MainActivity 클래스 수정

```
41              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.DST_OVER));
42              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.SRC));
43              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.SRC_ATOP));
44              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.SRC_IN));
45              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.SRC_OUT));
46              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.SRC_OVER));
47              //mPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.XOR));
48
49              // destination 비트맵에 source 비트맵 출력
50              Canvas canvasDst = new Canvas(mBitmapDst);
51              canvasDst.drawBitmap(mBitmapSrc, 0, 0, mPaint);
52
53              // 전송 모드를 기본값으로 재설정하고 destination 비트맵 출력
54              mPaint.setXfermode(null);
55              canvas.drawBitmap(mBitmapDst, 0, 0, mPaint);
56          }
57      }
58  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-4 Xfermode

실행 화면

(a) PorterDuff.Mode.ADD 적용

(b) PorterDuff.Mode.XOR 적용

그림 5-9 실행 화면

02 UI 이벤트 처리

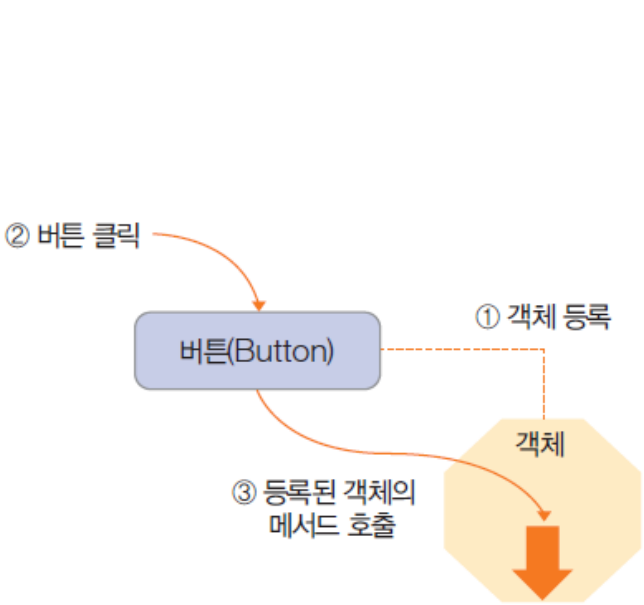
UI 이벤트(User Interface event)

- 사용자의 행위로 응용 프로그램에 전달되는 입력(예: 키 누름, 화면 터치)
- 입력 이벤트(Input event)라고도 부름

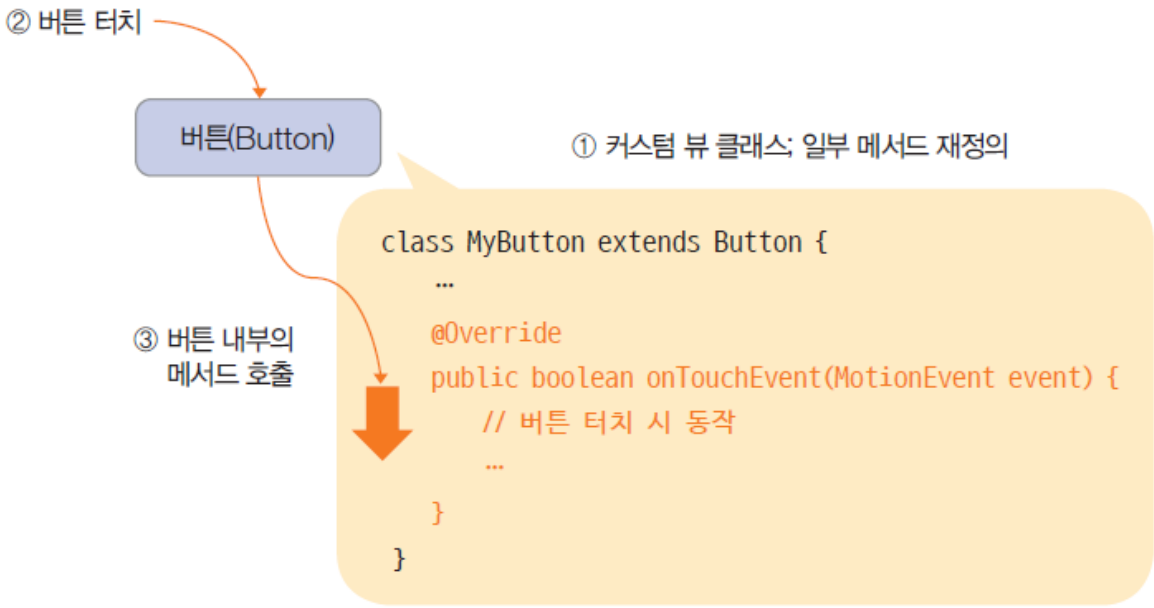
원리를 알면 IT가 맛있다
IT COOKBOOK

UI 이벤트를 처리하는 방식

- 이벤트 리스너(Event listener): 이벤트를 처리하는 객체를 만들어서 뷰에 등록해두면 이벤트 발생 시 등록된 객체의 메서드가 호출되어 처리
- 이벤트 핸들러(Event handler): 이벤트 발생 시 뷰 내부의 미리 정해진 메서드가 호출되어 처리



(a) 이벤트 리스너



(b) 이벤트 핸들러

그림 5-10 UI 이벤트 처리 방식 비교

원리를 알면 IT가 맛있다
IT COOKBOOK

이벤트 리스너

표 5-4 View 클래스의 UI 이벤트 리스너

UI 이벤트	UI 이벤트 처리 객체를 등록하는 메서드	UI 이벤트 발생 시 호출되는 메서드
키 입력	void setOnKeyListener(View.OnKeyListener l)	boolean onKey(View v, int keyCode, KeyEvent event)
화면 터치	void setOnTouchListener(View.OnTouchListener l)	boolean onTouch(View v, MotionEvent event)
클릭	void setOnClickListener(View.OnClickListener l)	void onClick(View v)
길게 클릭	void setOnLongClickListener(View.OnLongClickListener l)	boolean onLongClick(View v)

원리를 알면 IT가 맛있다
IT COOKBOOK

버튼 클릭의 처리: 첫 번째 방법

버튼 클릭 처리 – 첫 번째 방법

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(R.layout.activity_main);
7           findViewById(R.id.button).setOnClickListener( new MyClickListener() );
8       }
9   }
10
11  class MyClickListener implements View.OnClickListener {
12      @Override
13      public void onClick(View v) {
14          Toast.makeText(v.getContext(), "버튼 클릭!", Toast.LENGTH_SHORT).show();
15      }
16  }
17
```

원리를 알면 IT가 맛있다
IT COOKBOOK

버튼 클릭의 처리: 첫 번째 방법

버튼 클릭 처리 – 두 번째 방법

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(R.layout.activity_main);
7           findViewById(R.id.button).setOnClickListener( new View.OnClickListener() {
8               @Override
9               public void onClick(View v) {
10                  Toast.makeText(v.getContext(), "버튼 클릭!", Toast.LENGTH_SHORT).show();
11              }
12          } );
13      }
14  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

이벤트 핸들러

표 5-5 View 클래스의 UI 이벤트 핸들러

UI 이벤트	UI 이벤트 발생 시 호출되는 핸들러
키 누름	boolean **onKeyDown**(int keyCode, KeyEvent event)
키 뗌	boolean **onKeyUp**(int keyCode, KeyEvent event)
화면 터치	boolean **onTouchEvent**(MotionEvent event)

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-5 ListenerHandler

- 리니어 레이아웃에 버튼 두 개를 배치
- 첫 번째 버튼은 이벤트 리스너, 두 번째 버튼은 이벤트 핸들러 방식

activity_main.xml

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       android:layout_width="match_parent"
5       android:layout_height="match_parent"
6       android:orientation="vertical">
7       <Button
8           android:id="@+id/button1"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:text="버튼1 (이벤트 리스너)"/>
12      <Button
13          android:id="@+id/button2"
14          android:layout_width="match_parent"
15          android:layout_height="wrap_content"
16          android:text="버튼2 (이벤트 핸들러)"/>
17  </LinearLayout>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-5 ListenerHandler

MainActivity 클래스 수정

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(R.layout.activity_main);
7
8           findViewById(R.id.button1).setOnClickListener(new View.OnClickListener() {
9               @Override
10              public void onClick(View v) {
11                  Toast.makeText(v.getContext(), "버튼1 클릭!", Toast.LENGTH_SHORT).show();
12              }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-5 ListenerHandler

MainActivity 클래스 수정

```
13          });
14      }
15  }
16
17  class MyButton extends Button {
18
19      public MyButton(Context context) {
20          super(context);
21          init();
22      }
23
24      public MyButton(Context context, AttributeSet attrs) {
25          super(context, attrs);
26          init();
27      }
28
29      public MyButton(Context context, AttributeSet attrs, int defStyleAttr) {
30          super(context, attrs, defStyleAttr);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-5 ListenerHandler

MainActivity 클래스 수정

```
31          init();
32      }
33
34      private void init() {
35          // 초기화 코드를 넣는다.
36          setBackgroundColor(Color.YELLOW);
37          setTextColor(Color.RED);
38      }
39
40      @Override
41      public boolean onTouchEvent(MotionEvent event) {
42          if (event.getAction() == MotionEvent.ACTION_DOWN) {
43              Toast.makeText(getContext(), "버튼2 클릭!", Toast.LENGTH_SHORT).show();
44          }
45          return super.onTouchEvent(event);
46      }
47  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-5 ListenerHandler

MainActivity 클래스 수정

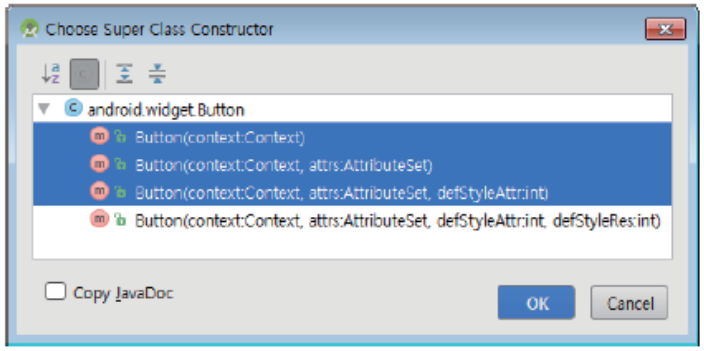
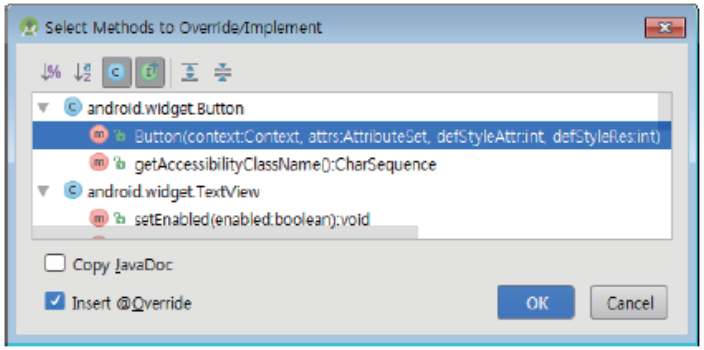
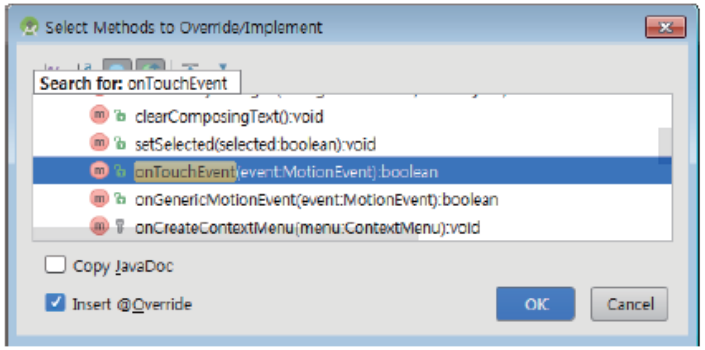


그림 5-11 수퍼클래스의 생성자를 토대로 서브클래스의 생성자 정의



(a) [Code]-[Override Methods…] 메뉴 선택



(b) onTouch 입력하여 메서드가 선택되면 [OK] 클릭

그림 5-12 수퍼클래스의 onTouchEvent() 메서드 재정의

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-5 ListenerHandler

- **res/layout/activity_main.xml의 12행 수정**

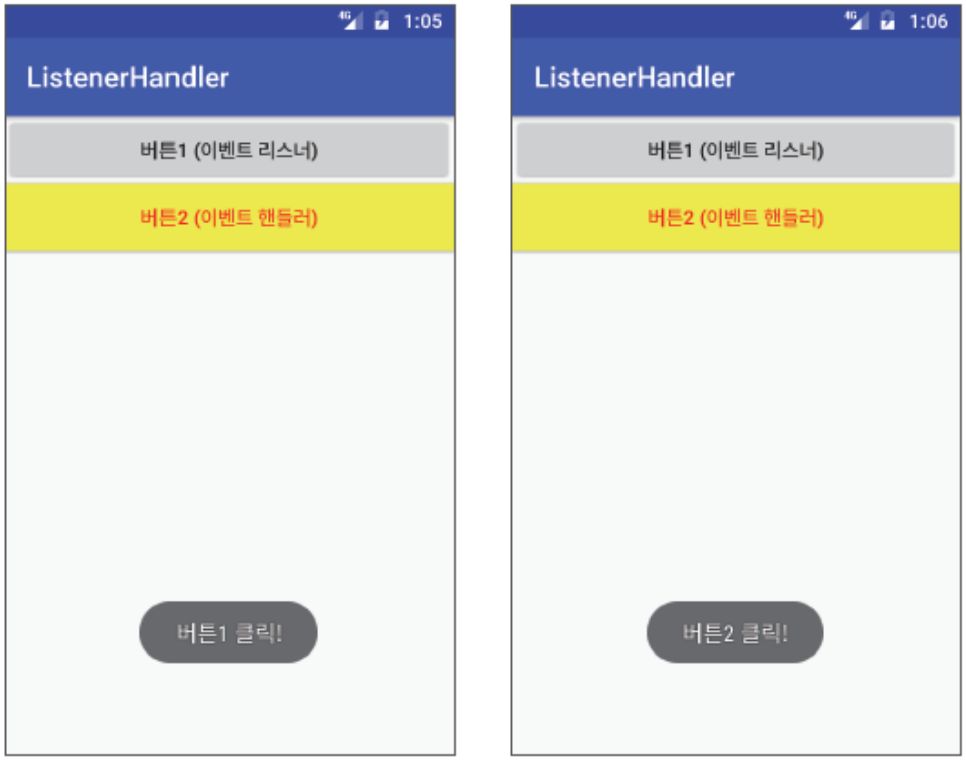
activity_main.xml

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       android:layout_width="match_parent"
5       android:layout_height="match_parent"
6       android:orientation="vertical">
7       <Button
8           android:id="@+id/button1"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:text="버튼1 (이벤트 리스너)"/>
12      <andbook.example.listenerhandler.MyButton
13          android:id="@+id/button2"
14          android:layout_width="match_parent"
15          android:layout_height="wrap_content"
16          android:text="버튼2 (이벤트 핸들러)"/>
17  </LinearLayout>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-5 ListenerHandler

- 실행 화면: (a) 이벤트 리스너, (b) 이벤트 헨들러



(a) 버튼1 클릭 (b) 버튼2 클릭

그림 5-13 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

키 입력 처리 메서드

이벤트 리스너	boolean **onKey**(View v, int keyCode, KeyEvent event)
이벤트 핸들러	boolean **onKeyDown**(int keyCode, KeyEvent event) boolean **onKeyUp**(int keyCode, KeyEvent event)

키 코드(Key Code): 키에 할당된 고유한 숫자

표 5-6 키 코드(일부)

키 이름	키 코드	키 이름	키 코드
백(Back)	KEYCODE_BACK	메뉴	KEYCODE_MENU
카메라	KEYCODE_CAMERA	볼륨 줄이기	KEYCODE_VOLUME_DOWN
엔터(Enter)	KEYCODE_ENTER	볼륨 키우기	KEYCODE_VOLUME_UP
홈	KEYCODE_HOME	전원	KEYCODE_POWER
숫자	KEYCODE_0 ~ KEYCODE_12	알파벳	KEYCODE_A ~ KEYCODE_Z

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-6 KeyTest

res/layout/activity_main.xml 수정

activity_main.xml

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       android:layout_width="match_parent"
5       android:layout_height="match_parent"
6       android:orientation="vertical">
7       <EditText
8           android:id="@+id/editText"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:inputType="text"
12          android:text=""/>
13  </LinearLayout>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-6 KeyTest

MainActivity 클래스 수정

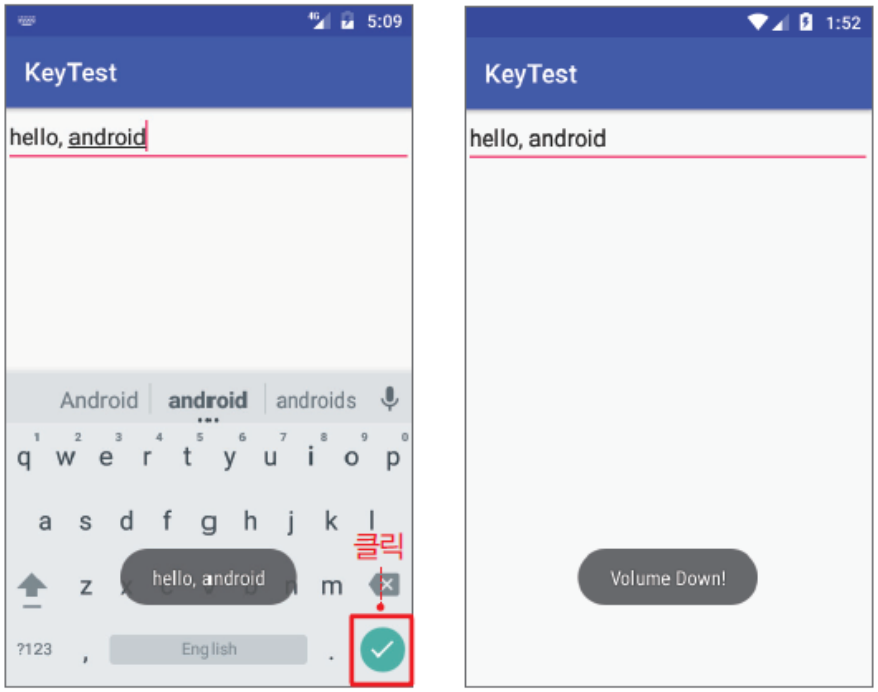
MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(R.layout.activity_main);
7
8           findViewById(R.id.editText).setOnKeyListener(new View.OnKeyListener() {
9               @Override
10              public boolean onKey(View v, int keyCode, KeyEvent event) {
11                  if (event.getAction() == KeyEvent.ACTION_DOWN) {
12                      switch (keyCode) {
13                      case KeyEvent.KEYCODE_ENTER:
14                          Toast.makeText(v.getContext(), ((EditText) v).getText().toString(),
15                                  Toast.LENGTH_SHORT).show();
16                          return true;
17                      case KeyEvent.KEYCODE_VOLUME_DOWN:
18                          Toast.makeText(v.getContext(), "Volume Down!",
19                                  Toast.LENGTH_SHORT).show();
20                          return true;
21                      }
22                  }
23                   return false;
24              }
25          });
26      }
27  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-6 KeyTest

실행 화면



(a) 텍스트 입력 후 엔터 키 누름 (b) 볼륨 줄이기 버튼 누름

그림 5-14 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

터치 입력을 처리하는 메서드

이벤트 리스너	boolean onTouch(View v, MotionEvent event)
이벤트 핸들러	boolean onTouchEvent(MotionEvent event)

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-7 TouchTest

터치 입력을 처리하는 메서드

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       @Override
4       protected void onCreate(Bundle savedInstanceState) {
5           super.onCreate(savedInstanceState);
6           setContentView(new MyView(this));
7       }
8
9       private static class MyView extends View {
10          private static final int R = 10;
11          private ArrayList<Float> mPoints;
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-7 TouchTest

터치 입력을 처리하는 메서드

```
12          private Paint mPaint;
13
14          public MyView(Context context) {
15              super(context);
16              mPoints = new ArrayList<>();
17              mPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
18              mPaint.setColor(Color.MAGENTA);
19          }
20
21          @Override
22          protected void onDraw(Canvas canvas) {
23              for (int i = 0; i < mPoints.size(); i += 2) {
24                  float x = mPoints.get(i);
25                  float y = mPoints.get(i + 1);
26                  canvas.drawRect(x - R, y - R, x + R, y + R, mPaint);
27              }
28          }
29
30          @Override
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-7 TouchTest

터치 입력을 처리하는 메서드

```
31          public boolean onTouchEvent(MotionEvent event) {
32              switch (event.getAction()) {
33              case MotionEvent.ACTION_DOWN:
34                  mPoints.add(event.getX());
35                  mPoints.add(event.getY());
36                  invalidate();
37                  return true;
38              case MotionEvent.ACTION_MOVE:
39                  mPoints.add(event.getX());
40                  mPoints.add(event.getY());
41                  invalidate();
42                  return true;
43              case MotionEvent.ACTION_UP:
44                  return true;
45              }
46              return super.onTouchEvent(event);
47          }
48      }
49  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 5-7 TouchTest

실행 화면

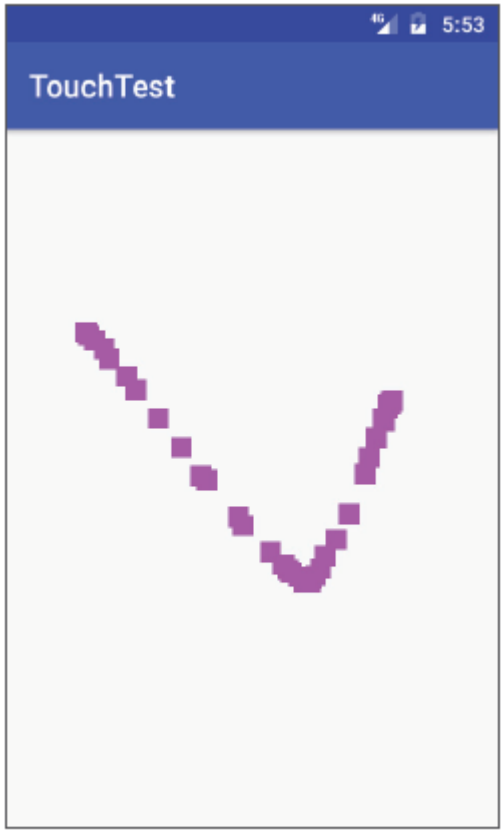
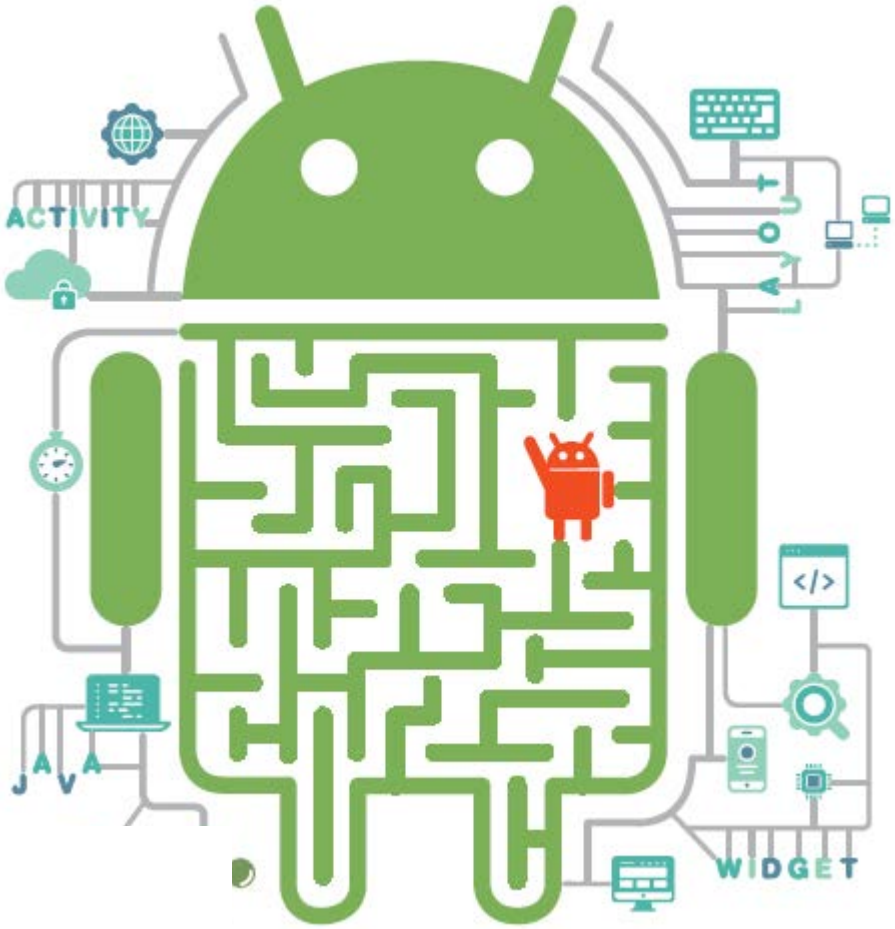


그림 5-15 실행 화면



단계별로 배우는

안드로이드 프로그래밍