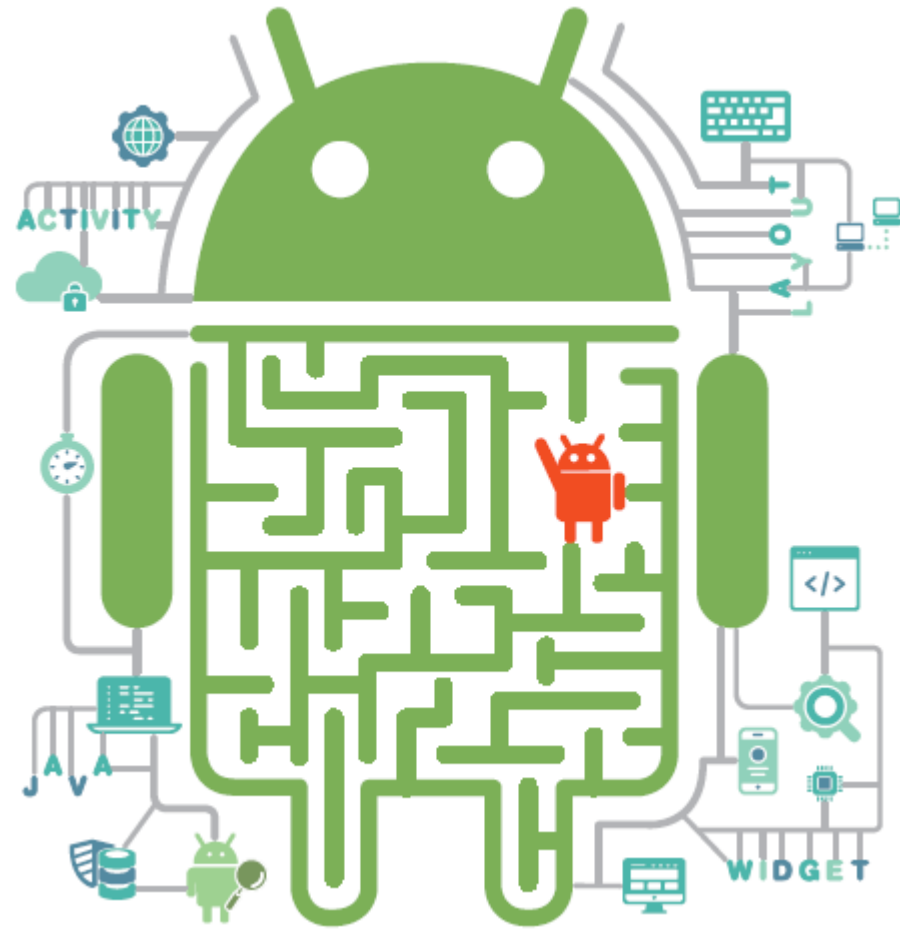


## 단계 별로 배우는 안드로이드 프로그래밍

### [강의교안 이용 안내]

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



단계별로 배우는

# 안드로이드 프로그래밍

## Chapter 08. 액티비티(II)

# 목차

01 인텐트 개념

02 인텐트 활용

# 학습목표

- 앱 구성 요소와 인텐트의 관계를 파악한다.
- 명시적 인텐트를 정의하고 사용할 수 있다.
- 암시적 인텐트를 정의하고 사용할 수 있다.
- 인텐트를 이용한 데이터 교환 구조를 이해한다.
- 알림 메시지를 생성하고 처리할 수 있다.

## ■ 인텐트(Intent)

- 안드로이드 앱의 구성 요소 중, 콘텐츠 제공자를 제외한 액티비티, 서비스, 브로드캐스트 수신기는 인텐트라 불리는 자바 객체를 통해 시작(실행)됨

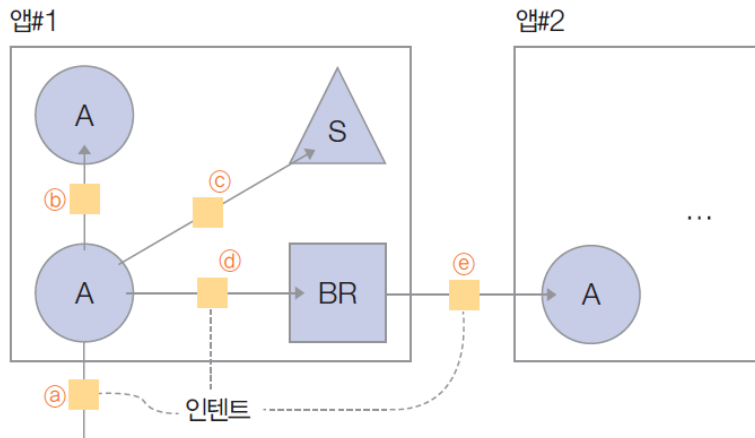


그림 8-1 앱 구성 요소와 인텐트의 관계

- ① 외부에서 인텐트로 액티비티를 시작하여 앱 실행
- ② 같은 앱에 포함된 다른 액티비티 시작
- ③ 같은 앱에 포함된 서비스 시작
- ④ 같은 앱에 포함된 브로드캐스트 수신기 시작
- ⑤ 다른 앱에 포함된 액티비티 시작

## ■ 인텐트

- 안드로이드 시스템에 요청할 작업을 담고 있는 자바 객체
- 인텐트를 생성하고 초기화하여 앱의 구성 요소를 실행하는 코드

```
Intent intent = new Intent(); // 인텐트 객체를 생성한다.  
intent.setAction(...);      // Intent 클래스의 각종 메서드를  
...                          // 호출하여 초기화한다.  
startActivity(intent);       // 액티비티를 시작하도록 시스템에 요청한다.  
startService(intent);        // 서비스를 시작하도록 시스템에 요청한다.  
sendBroadcast(intent);       // (브로드캐스트를 보내서) 브로드캐스트 수신기를  
                              // 시작하도록 시스템에 요청한다.
```

# 01 인텐트 개념 ▶ 액티비티와 명시적 인텐트

## ■ 인텐트의 종류

- 명시적 인텐트: 인텐트가 가리키는 대상 명확. 주로 내부 구성 요소를 실행할 때 사용.
- 암시적 인텐트: 인텐트가 가리키는 대상 모호. 주로 외부 구성 요소를 실행할 때 사용.

## ■ 명시적 인텐트와 암시적 인텐트의 차이점

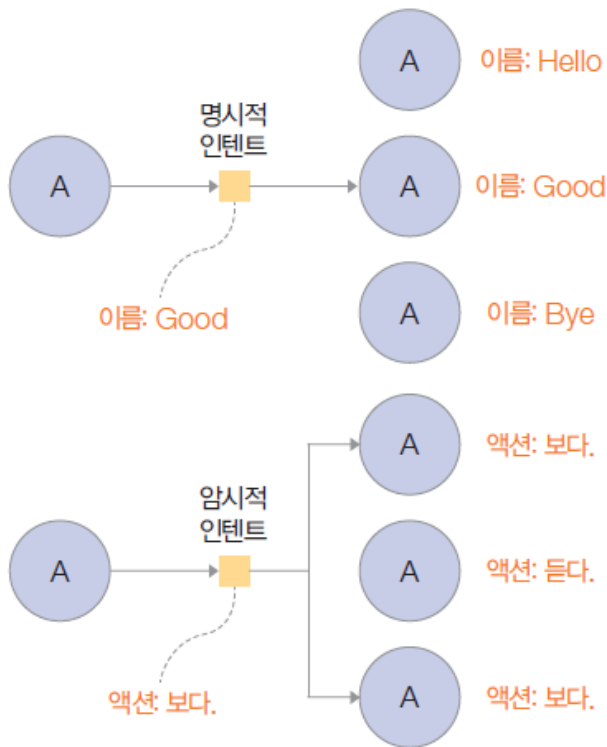


그림 8-2 명시적 인텐트와 암시적 인텐트

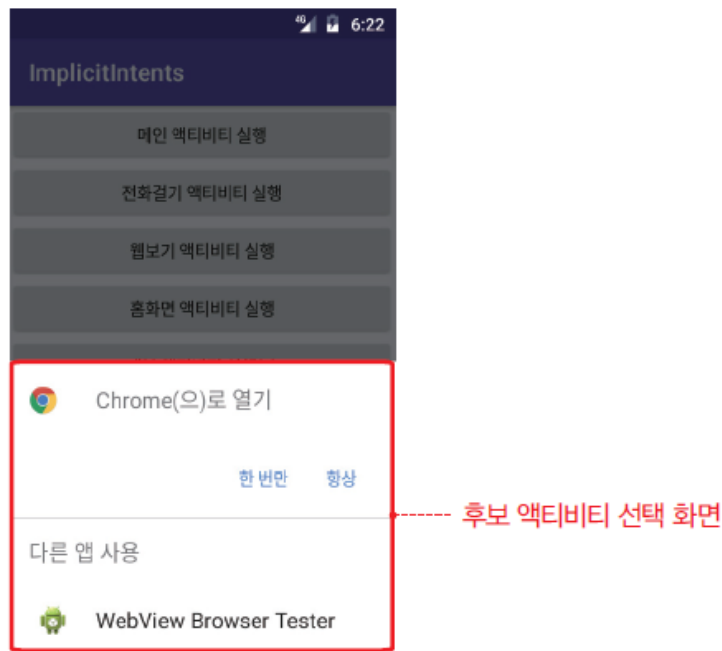


그림 8-3 암시적 인텐트의 후보가 둘 이상인 경우

# 01 인텐트 개념 ▶ 액티비티와 명시적 인텐트



## 실습 8-1

## ExplicitIntents

### ■ 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

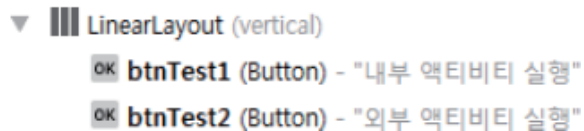


그림 8-4 컴포넌트 트리

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/btnTest1"
9          android:layout_width="wrap_content"
10         android:layout_height="wrap_content"
11         android:onClick="mOnClick"
12         android:text="내부 액티비티 실행"/>
13     <Button
14         android:id="@+id/btnTest2"
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         android:onClick="mOnClick"
18         android:text="외부 액티비티 실행"/>
19 </LinearLayout>
```



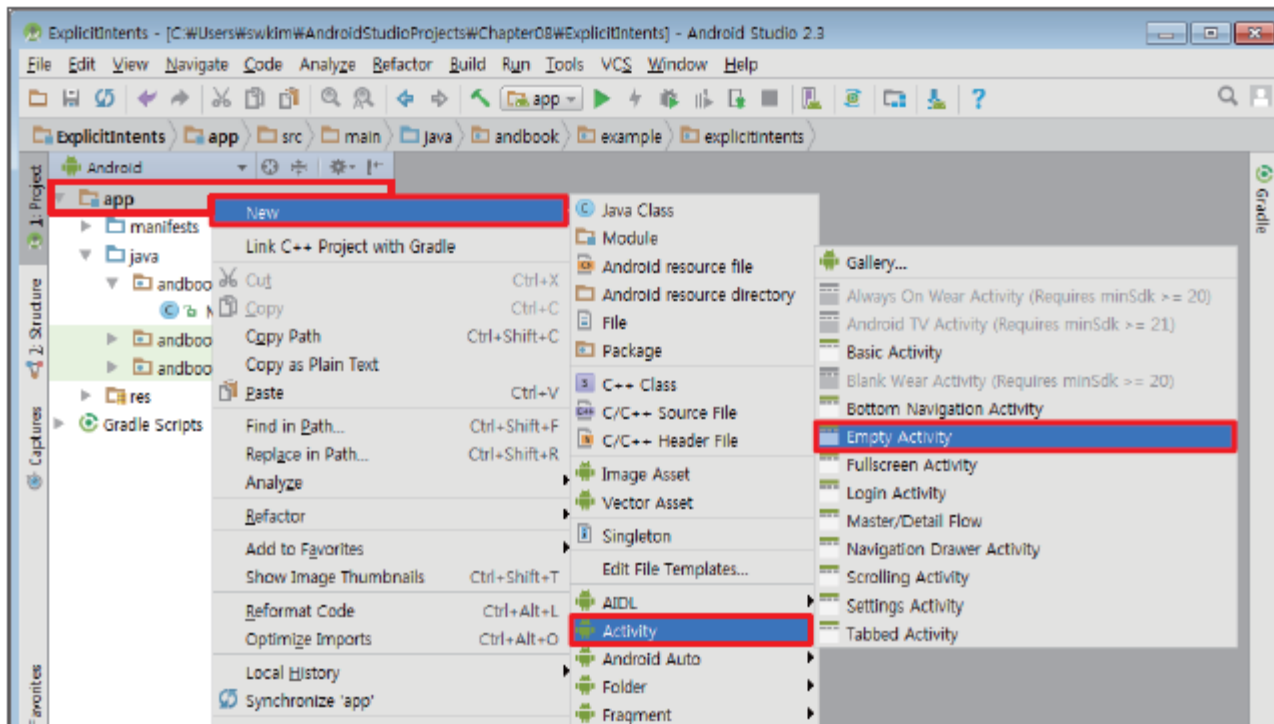
# 01 인텐트 개념 ▶ 액티비티와 명시적 인텐트



## 실습 8-1

## ExplicitIntents

### ■ 내부 액티비티 생성



(a) [New]–[Activity]–[Empty Activity] 메뉴 선택

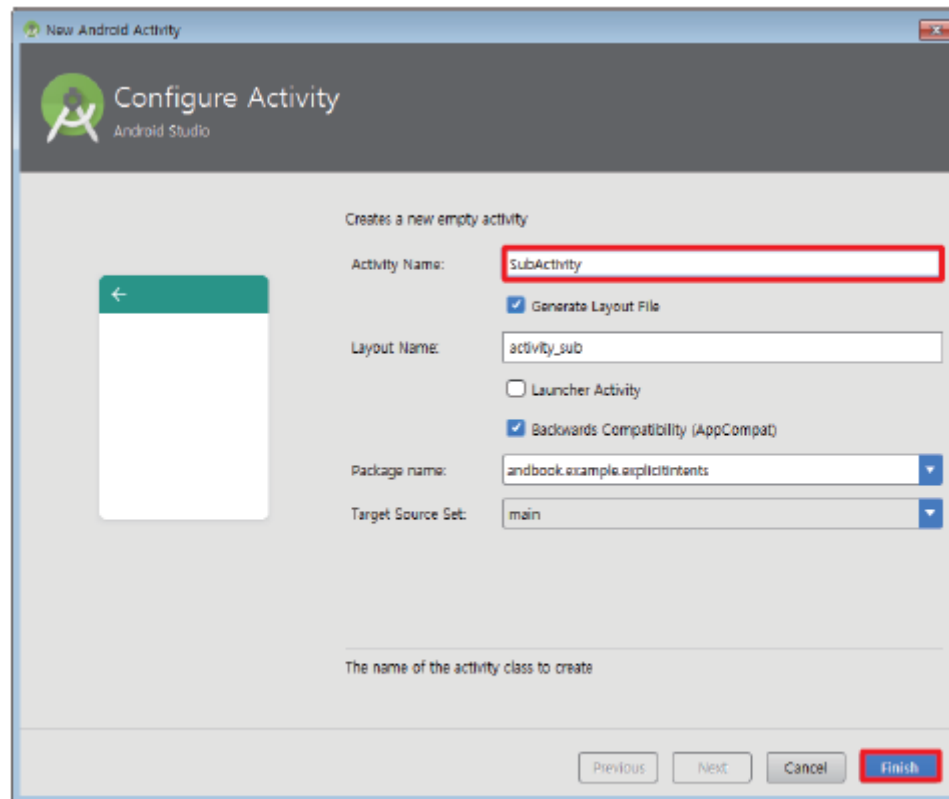
# 01 인텐트 개념 ▶ 액티비티와 명시적 인텐트



실습 8-1

ExplicitIntents

## ■ 내부 액티비티 생성



(b) 액티비티 이름 변경



## 실습 8-1

## ExplicitIntents

### ■ res/layout/activity\_sub.xml 파일 수정

activity\_sub.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <AnalogClock
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:background="#ffff77">
7  </AnalogClock>
```

(c) activity\_sub.xml 수정

그림 8-5 내부 액티비티 생성

# 01 인텐트 개념 ▶ 액티비티와 명시적 인텐트



## 실습 8-1

## ExplicitIntents

### ■ AndroidManifest.xml에 새로운 activity 엘리먼트 추가

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="andbook.example.explicitintents">
    <application ...>
        <activity android:name=".MainActivity">
            ...
        </activity>
        <activity android:name=".SubActivity">
        </activity>
    </application>
</manifest>
```

# 01 인텐트 개념 ▶ 액티비티와 명시적 인텐트



## 실습 8-1

## ExplicitIntents

### ■ MainActivity 클래스에 코드 작성

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
9      public void mOnClick(View v) {
10         Intent intent;
11         switch (v.getId()) {
12             case R.id.btnTest1: // 내부 액티비티 실행
13                 intent = new Intent(this, SubActivity.class);
14                 startActivity(intent);
15                 break;
16             case R.id.btnTest2: // 외부 액티비티 실행
17                 intent = new Intent();
18                 intent.setClassName("com.android.settings", "com.android.settings.Settings");
19                 startActivity(intent);
20                 break;
21         }
22     }
23 }
```

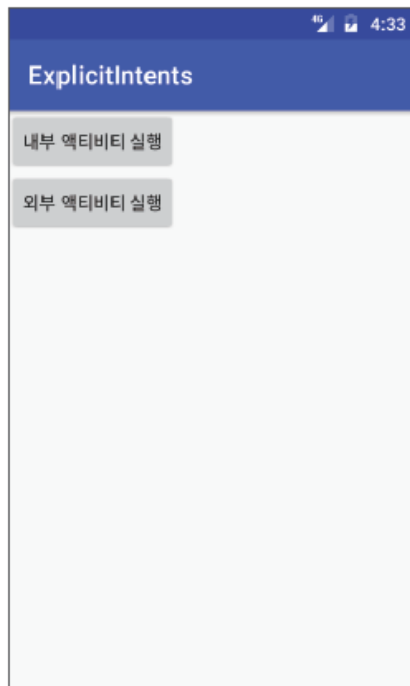
# 01 인텐트 개념 ▶ 액티비티와 명시적 인텐트



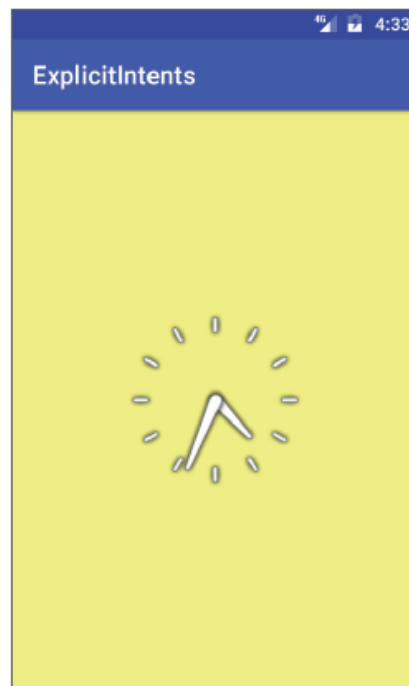
실습 8-1

ExplicitIntents

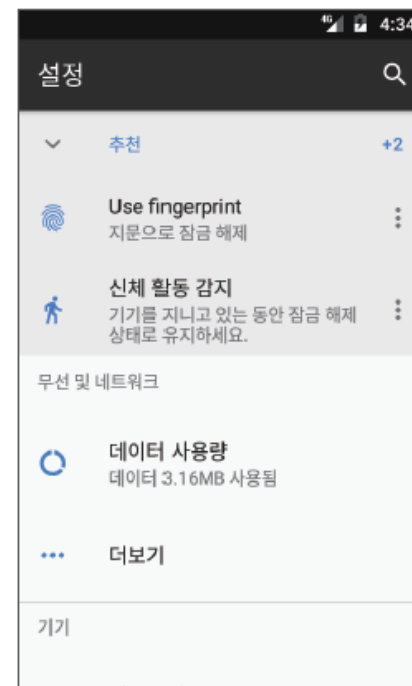
## ■ 실행 화면



(a) 초기 화면



(b) 내부 액티비티 실행



(c) 외부 액티비티 실행

그림 8-6 실행 화면

# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터

- 수신 대상의 이름을 포함하지 않는 인텐트는 암시적 인텐트로 간주
- 암시적 인텐트는 액션, 데이터, 카테고리로 구성

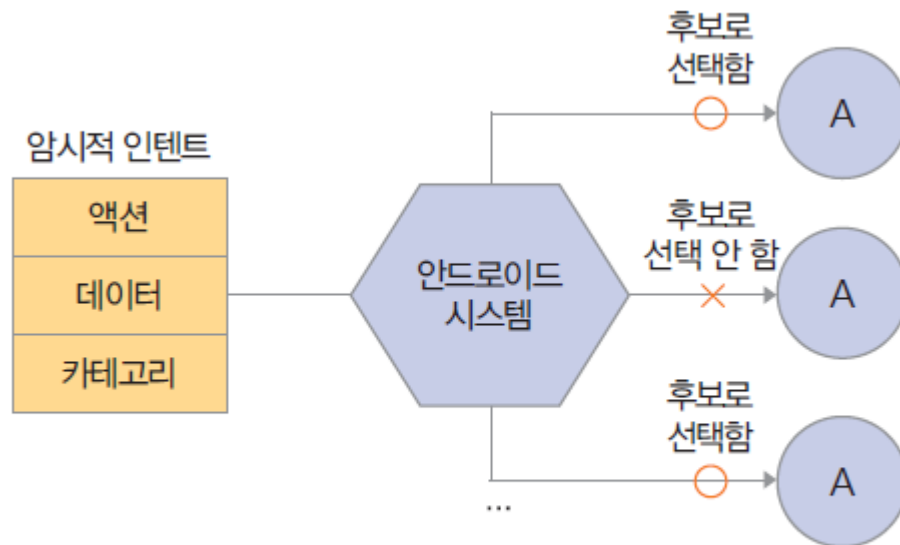


그림 8-7 안드로이드 시스템의 암시적 인텐트 처리 (1)

## ■ 암시적 인텐트로 액티비티를 실행하는 전형적인 코드

```
1  Intent intent = new Intent(); // 기본 인텐트 생성
2  intent.setAction(Intent.ACTION_VIEW); // 액션 지정
3  intent.setDataAndType(Uri.parse("https://www.google.co.kr/images/branding/googlelogo/1x/
   googlelogo_color_272x92dp.png"), "image/*"); // 위치(URI)와 타입(MIME 타입) 지정
4  intent.addCategory(Intent.CATEGORY_DEFAULT); // 카테고리 지정
5  startActivity(intent); // 액티비티 실행
```



# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터

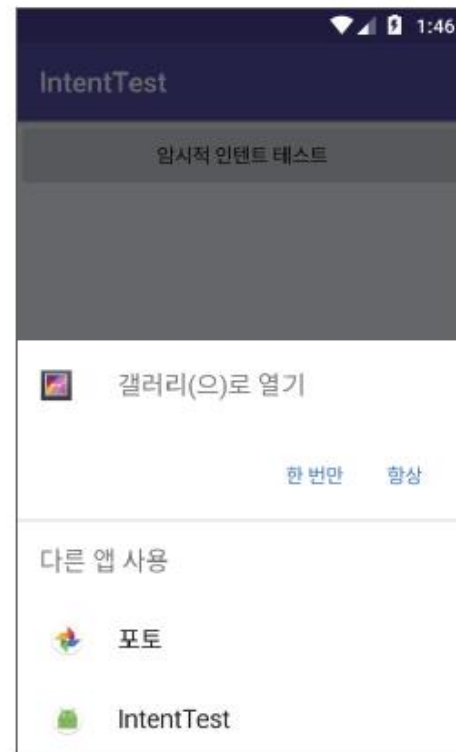
## ■ 실행 화면



(a) 후보 액티비티 선택 (1)



(b) 갤러리 액티비티 실행



(c) 후보 액티비티 선택 (2)

그림 8-8 실행 화면

# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터

## ■ 인텐트 필터

- 어떤 앱 구성 요소의 암시적 인텐트 후보 결정
- 구성 요소가 처리할 수 있는 액션과 데이터, 카테고리로 구성

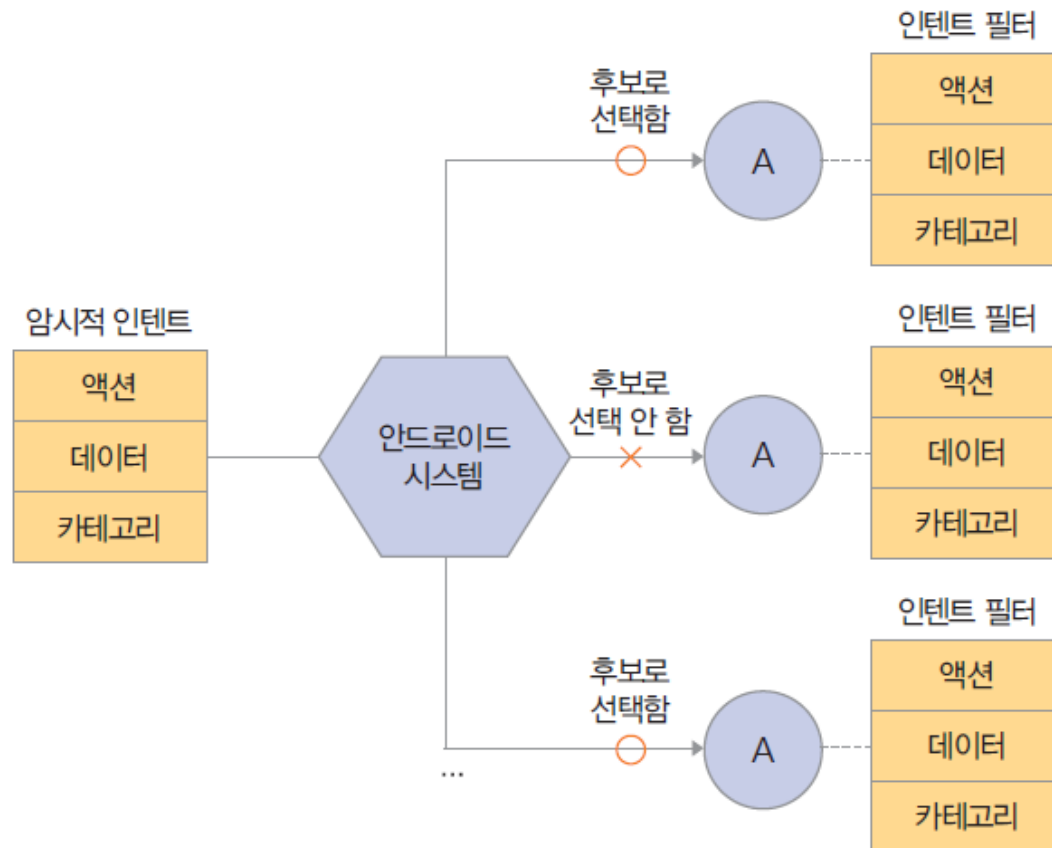


그림 8-9 안드로이드 시스템의 암시적 인텐트 처리 (2)

## ■ TestActivity라는 액티비티에 인텐트 필터 추가

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3    package="andbook.example.hello">
4    <application ...>
5      <activity android:name=".TestActivity">
6        <intent-filter>
7          <action android:name="android.intent.action.VIEW"/>
8          <data android:scheme="https" android:mimeType="image/*"/>
9          <category android:name="android.intent.category.DEFAULT"/>
10       </intent-filter>
11       <intent-filter>
12         ...
13       </intent-filter>
14       <intent-filter>
15         ...
16       </intent-filter>
17     </activity>
18     ...
19   </application>
20 </manifest>
```

# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터



## 실습 8-2

## ImplicitIntents

### ■ 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

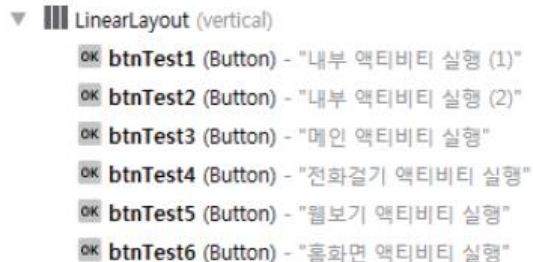


그림 8-10 컴포넌트 트리

#### 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/btnTest1"
9          android:layout_width="match_parent"
```

# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터



## 실습 8-2

## ImplicitIntents

### ■ 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

```
10         android:layout_height="wrap_content"
11         android:onClick="mOnClick"
12         android:text="내부 액티비티 실행 (1)"/>
13     <Button
14         android:id="@+id/btnTest2"
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:onClick="mOnClick"
18         android:text="내부 액티비티 실행 (2)"/>
19     <Button
20         android:id="@+id/btnTest3"
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content"
23         android:onClick="mOnClick"
24         android:text="메인 액티비티 실행"/>
25     <Button
26         android:id="@+id/btnTest4"
27         android:layout_width="match_parent"
```



## 실습 8-2

## ImplicitIntents

- 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

```
29         android:onClick="mOnClick"
30         android:text="전화걸기 액티비티 실행"/>
31     <Button
32         android:id="@+id/btnTest5"
33         android:layout_width="match_parent"
34         android:layout_height="wrap_content"
35         android:onClick="mOnClick"
36         android:text="웹보기 액티비티 실행"/>
37     <Button
38         android:id="@+id/btnTest6"
39         android:layout_width="match_parent"
40         android:layout_height="wrap_content"
41         android:onClick="mOnClick"
42         android:text="홈화면 액티비티 실행"/>
43 </LinearLayout>
```

# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터



실습 8-2

ImplicitIntents

- SubActivity라는 이름의 내부 액티비티 생성
- AndroidManifest.xml에 SubActivity의 하위 엘리먼트로 인텐트 필터 두 개 추가

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="andbook.example.implicitintents">
4      <application
5          android:allowBackup="true"
6          android:icon="@mipmap/ic_launcher"
7          android:label="@string/app_name"
8          android:roundIcon="@mipmap/ic_launcher_round"
9          android:supportsRtl="true"
10         android:theme="@style/AppTheme">
11         <activity android:name=".MainActivity">
12             <intent-filter>
13                 <action android:name="android.intent.action.MAIN"/>
14                 <category android:name="android.intent.category.LAUNCHER"/>
15             </intent-filter>
```



## 실습 8-2

## ImplicitIntents

- AndroidManifest.xml에 SubActivity의 하위 엘리먼트로 인텐트 필터 두 개 추가

```
16         </activity>
17         <activity android:name=".SubActivity">
18             <intent-filter>
19                 <action android:name="andbook.example.implicitintents.TEST1"/>
20                 <category android:name="android.intent.category.DEFAULT"/>
21             </intent-filter>
22             <intent-filter>
23                 <action android:name="andbook.example.implicitintents.TEST2"/>
24                 <data android:mimeType="image/*"/>
25                 <category android:name="android.intent.category.DEFAULT"/>
26             </intent-filter>
27         </activity>
28     </application>
29 </manifest>
```



# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터



실습 8-2

ImplicitIntents

## ■ MainActivity 클래스에 코드 작성

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
9      public void mOnClick(View v) {
10         Intent intent = new Intent(); // 기본 인텐트 생성
11         switch (v.getId()) {
12             case R.id.btnTest1: // 내부 액티비티 실행 (1)
13                 intent.setAction("andbook.example.implicitintents.TEST1");
14                 break;
15             case R.id.btnTest2: // 내부 액티비티 실행 (2)
16                 intent.setAction("andbook.example.implicitintents.TEST2");
17                 intent.setType("image/png");
```

# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터



## 실습 8-2

## ImplicitIntents

### ■ MainActivity 클래스에 코드 작성

```
18         break;
19     case R.id.btnTest3: // 메인 액티비티 실행
20         intent.setAction(Intent.ACTION_MAIN);
21         break;
22     case R.id.btnTest4: // '전화 걸기' 액티비티 실행
23         intent.setAction(Intent.ACTION_DIAL);
24         intent.setData(Uri.parse("tel:010-2222-3333"));
25         break;
26     case R.id.btnTest5: // '웹 보기' 액티비티 실행
27         intent.setAction(Intent.ACTION_VIEW);
28         intent.setData(Uri.parse("http://www.android.com"));
29         break;
30     case R.id.btnTest6: // '홈 화면' 액티비티 실행
31         intent.setAction(Intent.ACTION_MAIN);
32         intent.addCategory(Intent.CATEGORY_HOME);
33         break;
34     }
35     startActivity(intent);
36 }
37 }
```

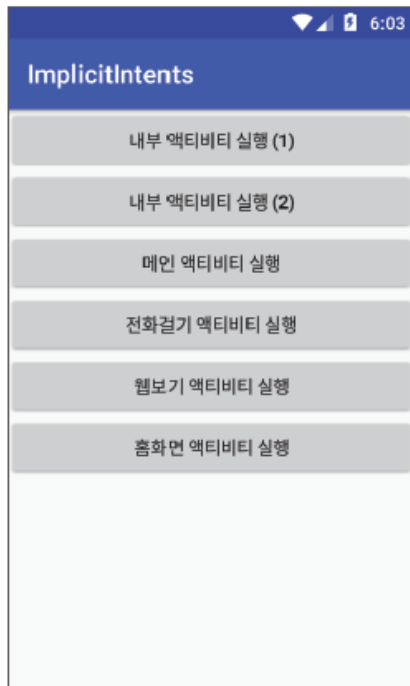
# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터



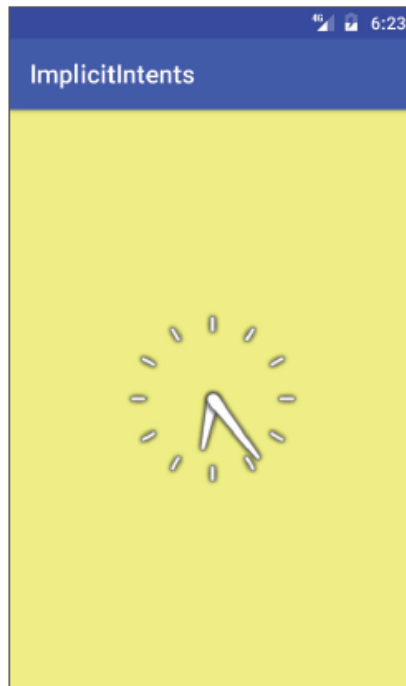
## 실습 8-2

## ImplicitIntents

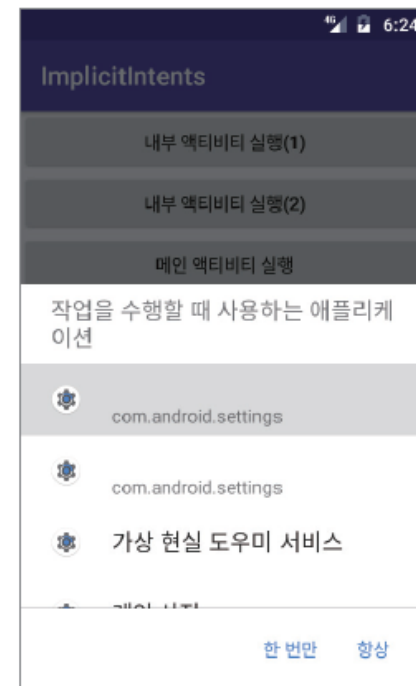
### ■ 실행 화면



(a) 초기 화면



(b) 내부 액티비티 실행 후



(c) 메인 액티비티 실행 전

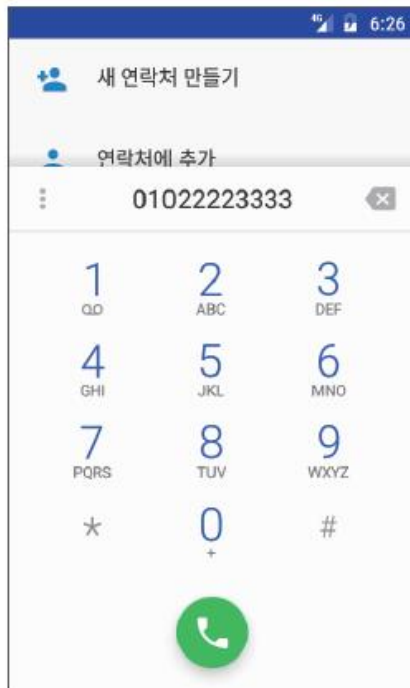
# 01 인텐트 개념 ▶ 암시적 인텐트와 인텐트 필터



## 실습 8-2

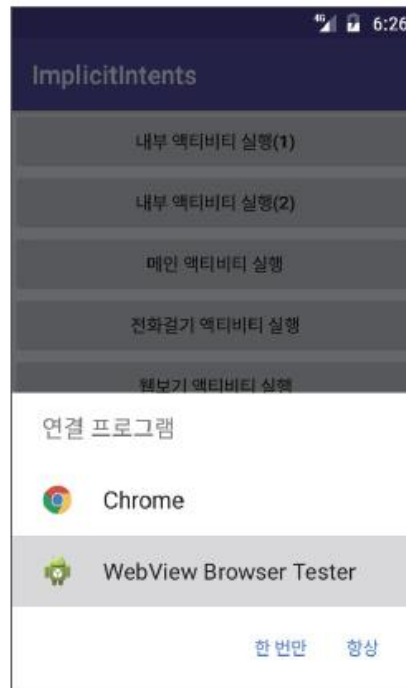
## ImplicitIntents

### ■ 실행 화면

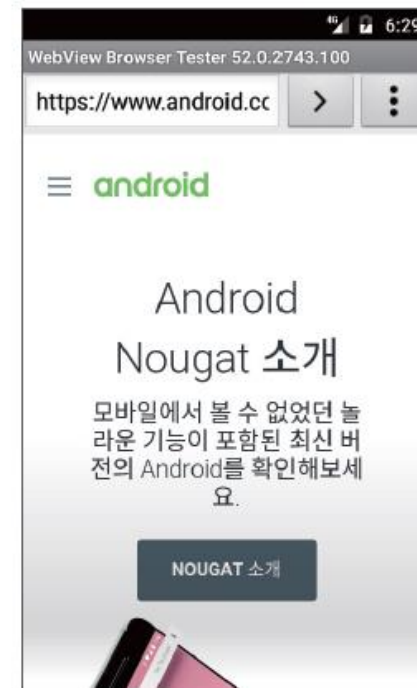


(d) '전화 걸기' 액티비티 실행 후

그림 8-11 실행 화면



(e) '웹 보기' 액티비티 실행 전



(f) '웹 보기' 액티비티 실행 후

# 01 인텐트 개념 ▶ 인텐트 구성 요약

## ■ 인텐트에 이름, 액션/데이터/카테고리 외에 다른 정보 추가 가능

인텐트		
명시적 인텐트	이름	시작할 대상의 클래스 이름
	액션	수행할 작업 또는 발생한 브로드캐스트
암시적 인텐트	데이터	데이터의 위치와 타입
	카테고리	시작할 대상의 종류
	엑스트라	부가적인 데이터
	플래그	기타 제어 정보

그림 8-12 인텐트의 내부 구성

## ■ Intent 클래스가 제공하는 엑스트라 관련 메서드 중 일부

표 8-1 Intent 클래스의 엑스트라 관련 메서드 (일부)

작업	메서드
데이터 넣기	Intent.putExtra(String name, boolean value) // 자바 기본형
	Intent.putExtra(String name, boolean[] value) // 자바 기본형의 배열
	Intent.putExtra(String name, byte value)
	Intent.putExtra(String name, byte[] value)
	Intent.putExtra(String name, char value)
	Intent.putExtra(String name, char[] value)
	Intent.putExtra(String name, short value)
	Intent.putExtra(String name, short[] value)
	Intent.putExtra(String name, int value) // 자바 기본형
	Intent.putExtra(String name, int[] value) // 자바 기본형의 배열
	Intent.putExtra(String name, long value)
	Intent.putExtra(String name, long[] value)
	Intent.putExtra(String name, float value)
	Intent.putExtra(String name, float[] value)
	Intent.putExtra(String name, double value)
	Intent.putExtra(String name, double[] value)
	Intent.putExtra(String name, String value) // 자바 문자열
	Intent.putExtra(String name, String[] value) // 자바 문자열의 배열

## ■ Intent 클래스가 제공하는 엑스트라 관련 메서드 중 일부

데이터 꺼내기	<code>boolean getBooleanExtra(String name, boolean defaultValue) // 자바 기본형; 기본값 지정</code>
	<code>boolean[] getBooleanArrayExtra(String name) // 자바 기본형의 배열</code>
	<code>...</code>
	<code>int getIntExtra(String name, int defaultValue) // 자바 기본형; 기본값 지정</code>
	<code>int[] getIntArrayExtra(String name) // 자바 기본형의 배열</code>
	<code>...</code>
	<code>String getStringExtra(String name) // 자바 문자열</code>
	<code>String[] getStringArrayExtra(String name) // 자바 문자열의 배열</code>

## ■ Intent 클래스가 제공하는 플래그 관련 메서드

- Intent 클래스에 FLAG\_로 시작하는 30개 이상의 플래그가 정의

표 8-2 Intent 클래스의 플래그 관련 메서드

메서드	기능
Intent setFlags(int flags) Intent addFlags(int flags)	setFlags()는 인텐트에 플래그를 새로 넣고, addFlags()는 인텐트에 플래그를 더 추가한다.
int getFlags()	인텐트에 설정된 플래그를 리턴한다.

## ■ 인텐트에 엑스트라와 플래그를 설정하는 코드의 예

```
1 Intent intent = new Intent("andbook.example.TESTACTION");
2 intent.putExtra("count", 10);
3 intent.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
4 startActivity(intent);
```



### ■ 액티비티 A, B의 데이터 교환



그림 8-13 액티비티 A, B의 데이터 교환

### ■ 액티비티 A

① 명시적 혹은 암시적 인텐트를 생성한다.

– 이때 실어 보내고자 하는 데이터는 [표 8-1]을 참조하여 엑스트라로 넣는다.

② startActivityForResult(인텐트, requestCode) 메서드를 호출하여 액티비티 B를 시작한다.

– requestCode는 0보다 크거나 같은 정숫값으로, ⑦에서 결과를 구분할 때 사용된다.

⑦ ⑥단계에서 액티비티(B)가 종료하면 현재 액티비티(A)로 돌아온다.

이때 onActivityResult(requestCode, resultCode, 인텐트) 메서드가 자동 호출된다.

– requestCode는 ②에서 전달했던 값과 일치한다.

– resultCode가 RESULT\_OK이면 성공을, 그렇지 않으면 실패를 뜻한다.

– 인텐트의 엑스트라로 전달된 결과(데이터)는 [표 8-1]을 참조하여 꺼낼 수 있다.

### ■ 액티비티 B

- ③ 액티비티 B가 실행되면서 onCreate()가 호출된다.
  - getIntent() 메서드를 호출하면 ②에서 전달한 인텐트가 그대로 리턴된다.
  - 인텐트의 엑스트라로 전달된 데이터는 [표 8-1]을 참조하여 꺼낼 수 있다.
- ④ 액티비티 B가 액티비티 A에게 결과(데이터)를 리턴하려면 새로운 인텐트를 생성한다.
  - 이때 전달하고자 하는 결과(데이터)는 [표 8-1]을 참조하여 엑스트라로 넣는다.
- ⑤ setResult(resultCode, 인텐트) 메서드를 호출하면 현재 액티비티(B)가 자신을 호출한 액티비티(A)에게 전달할 데이터가 정해진다.
  - resultCode로는 성공이면 RESULT\_OK, 실패이면 RESULT\_CANCELED를 넣는다.
- ⑥ finish()를 호출하여 현재 액티비티(B)를 종료하고 이전 액티비티(A)로 돌아간다.



### 실습 8-3

### DataExchange

## ■ 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

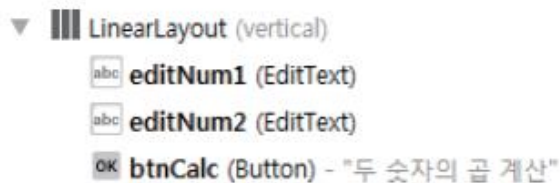


그림 8-14 컴포넌트 트리

#### 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">
```



### 실습 8-3

### DataExchange

#### ■ 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

```
7      <EditText
8          android:id="@+id/editNum1"
9          android:layout_width="match_parent"
10         android:layout_height="wrap_content"
11         android:hint="숫자1"
12         android:inputType="number"/>
13     <EditText
14         android:id="@+id/editNum2"
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:hint="숫자2"
18         android:inputType="number"/>
19     <Button
20         android:id="@+id/btnCalc"
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content"
23         android:onClick="mOnClick"
24         android:text="두 숫자의 곱 계산"/>
25 </LinearLayout>
```



### 실습 8-3

### DataExchange

- SubActivity라는 이름의 내부 액티비티 생성
- res/layout/activity\_sub.xml 내용은 컴포넌트 트리를 참고해 수정

▼ LinearLayout (vertical)  
Ab textNum1 (TextView)  
Ab textNum2 (TextView)  
OK btnQuit (Button) - "화면 닫기"

그림 8-15 컴포넌트 트리

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      <TextView
```



### 실습 8-3

### DataExchange

- res/layout/activity\_sub.xml 내용은 컴포넌트 트리를 참고해 수정

```
8         android:id="@+id/textNum1"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:hint="숫자1"
12        android:textSize="16dp"/>
13    <TextView
14        android:id="@+id/textNum2"
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:hint="숫자2"
18        android:textSize="16dp"/>
19    <Button
20        android:id="@+id/btnQuit"
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
23        android:onClick="mOnClick"
24        android:text="화면 닫기"/>
25 </LinearLayout>
```

## 02 인텐트 활용 ▶ 데이터 교환



### 실습 8-3

### DataExchange

#### ■ MainActivity 클래스에 다음 코드 작성

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {  
2  
3      private EditText mEditNum1;  
4      private EditText mEditNum2;  
5  
6      @Override  
7      protected void onCreate(Bundle savedInstanceState) {  
8          super.onCreate(savedInstanceState);  
9          setContentView(R.layout.activity_main);  
10  
11         mEditNum1 = (EditText) findViewById(R.id.editNum1);  
12         mEditNum2 = (EditText) findViewById(R.id.editNum2);  
13     }  
14  
15     public void mOnClick(View v) {  
16         Intent intent = new Intent(this, SubActivity.class);  
17         intent.putExtra("num1", Integer.parseInt(mEditNum1.getText().toString()));  
18         intent.putExtra("num2", Integer.parseInt(mEditNum2.getText().toString()));
```





### 실습 8-3

### DataExchange

#### ■ MainActivity 클래스에 다음 코드 작성

```
19         startActivityForResult(intent, 0);
20     }
21
22     @Override
23     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
24         if (requestCode == 0 && resultCode == RESULT_OK) {
25             int result = data.getIntExtra("result", 0);
26             Toast.makeText(this, "두 숫자의 곱: " + result, Toast.LENGTH_SHORT).show();
27         }
28         super.onActivityResult(requestCode, resultCode, data);
29     }
30 }
```



### 실습 8-3

### DataExchange

#### ■ SubActivity.java에 다음 코드 작성

SubActivity.java

```
1  public class SubActivity extends AppCompatActivity {  
2  
3      private int num1;  
4      private int num2;  
5  
6      @Override  
7      protected void onCreate(Bundle savedInstanceState) {  
8          super.onCreate(savedInstanceState);  
9          setContentView(R.layout.activity_sub);  
10  
11         TextView textNum1 = (TextView) findViewById(R.id.textNum1);  
12         TextView textNum2 = (TextView) findViewById(R.id.textNum2);  
13  
14         Intent intent = getIntent();
```



### 실습 8-3

### DataExchange

#### ■ SubActivity.java에 다음 코드 작성

```
15         num1 = intent.getIntExtra("num1", -1);
16         num2 = intent.getIntExtra("num2", -1);
17         if (num1 != -1 && num2 != -1) {
18             textNum1.setText(String.format(Locale.KOREAN, "숫자1: %d", num1));
19             textNum2.setText(String.format(Locale.KOREAN, "숫자2: %d", num2));
20         }
21     }
22
23     public void mOnClick(View v) {
24         Intent data = new Intent();
25         data.putExtra("result", num1 * num2);
26         setResult(RESULT_OK, data);
27         finish();
28     }
29 }
```

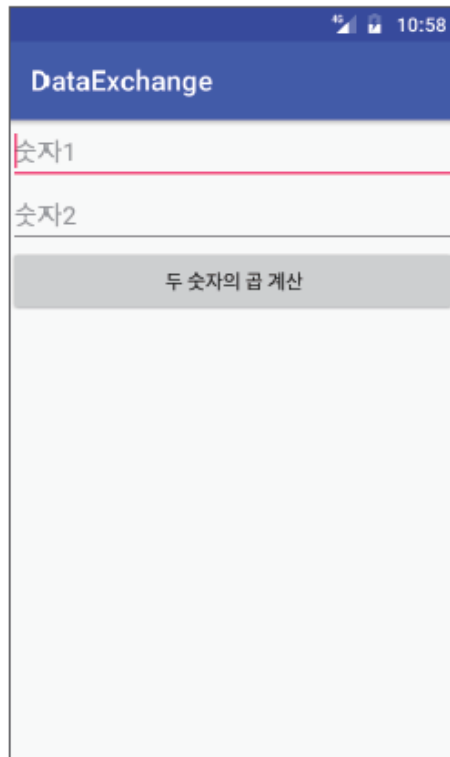
## 02 인텐트 활용 ▶ 데이터 교환



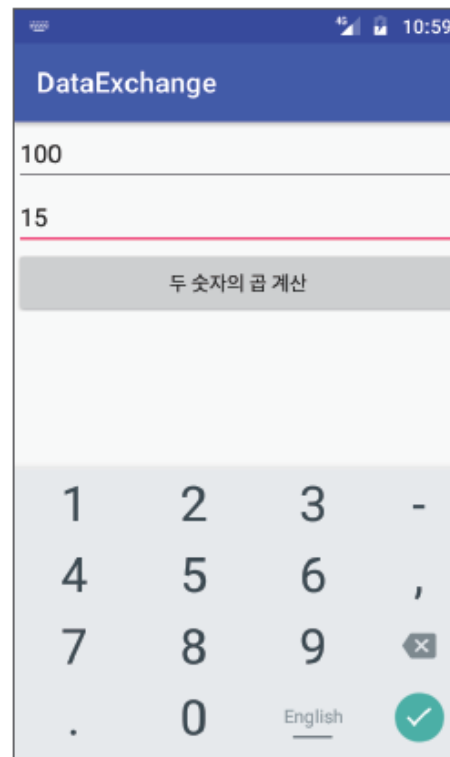
실습 8-3

DataExchange

### ■ 실행 화면



(a) MainActivity 화면; 초기 상태



(b) MainActivity 화면; 두 숫자 입력 후

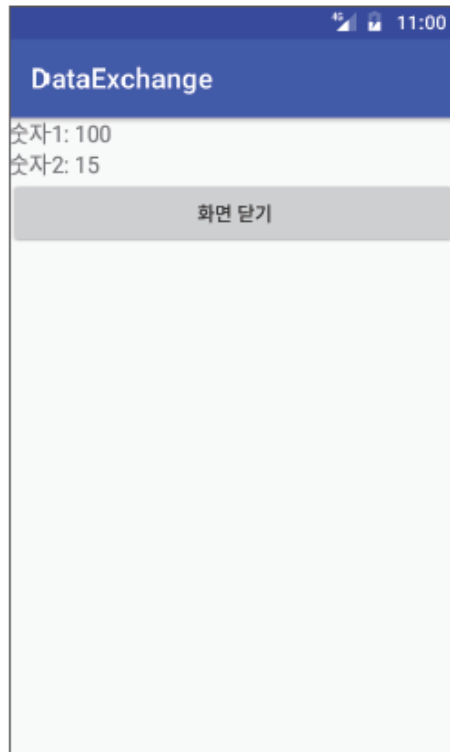
## 02 인텐트 활용 ▶ 데이터 교환



실습 8-3

DataExchange

### ■ 실행 화면



(c) SubActivity 화면; 두 숫자 출력



(d) MainActivity 화면; SubActivity 종료 후

그림 8-16 실행 화면

## 02 인텐트 활용 ▶ 알림 처리

### ■ 알림(Notification)

- 안드로이드 시스템의 대표적인 특징
- 앱 UI의 외부에서 사용자에게 표시하는 메시지



(a) 알림 영역에 표시된 알림

(b) 알림 창에 표시된 알림

그림 8-17 알림 영역과 알림 창에 표시된 알림

### ■ Notification 객체

- 안드로이드 앱에서 알림을 표시하고 싶을 때 만들어 초기화
- Notification 객체를 알림 관리자에 넘기면, 안드로이드 시스템의 알림 서비스에 전달되어 알림 영역에 아이콘이 표시

안드로이드 시스템

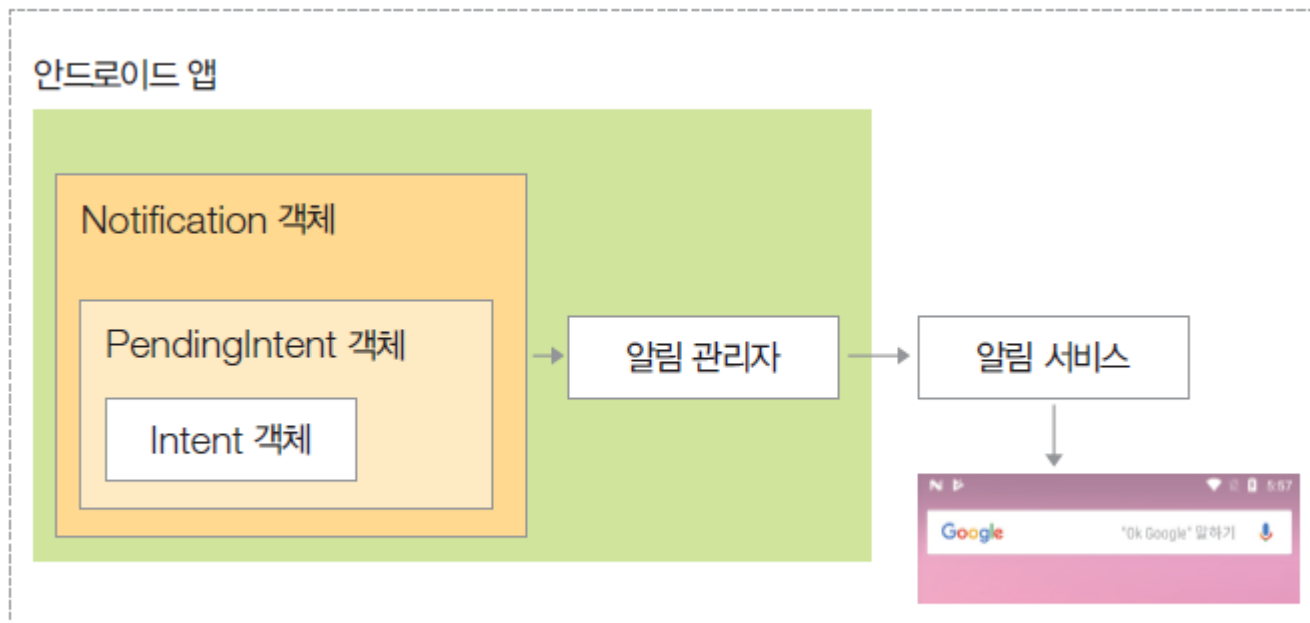


그림 8-18 알림 처리 과정

### ■ 예시 코드

- 알림을 터치하면 등록해둔 액티비티를 시작하면서 알림 제거

```
1  NotificationManager notificationManager =
2      (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
3
4  Intent resultIntent = new Intent(this, SubActivity.class);
5  PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
6      resultIntent, PendingIntent.FLAG_UPDATE_CURRENT);
7
8  NotificationCompat.Builder builder = new NotificationCompat.Builder(this);
9  builder.setSmallIcon(R.mipmap.ic_launcher);
10 builder.setContentTitle("알림 제목");
11 builder.setContentText("알림 텍스트");
12 builder.setContentIntent(contentIntent);
13 builder.setAutoCancel(true);
14
15 notificationManager.notify(0, builder.build());
```



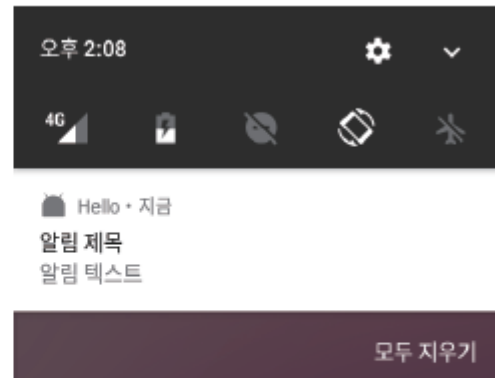
## 02 인텐트 활용 ▶ 알림 처리

### ■ 실행 화면



(a) 알림 영역에 표시된 알림

그림 8-19 실행 화면



(b) 알림 창에 표시된 알림



### 실습 8-4

### Notifications

## ■ 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

- ▼ LinearLayout (vertical)
  - OK btnTest1 (Button) - "첫 번째 알림 테스트"
  - OK btnTest2 (Button) - "두 번째 알림 테스트"

그림 8-20 컴포넌트 트리

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/btnTest1"
9          android:layout_width="match_parent"
10         android:layout_height="wrap_content"
11         android:onClick="mOnClick"
12         android:text="첫 번째 알림 테스트"/>
13     <Button
14         android:id="@+id/btnTest2"
```



### 실습 8-4

### Notifications

- 컴포넌트 트리를 참고하여 res/layout/activity\_main.xml 수정

```
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:onClick="mOnClick"
18         android:text="두 번째 알림 테스트"/>
19     </LinearLayout>
```



### 실습 8-4

### Notifications

- SubActivity라는 이름의 내부 액티비티 생성
- MainActivity 클래스에 서로 다른 알림을 표시하는 코드 작성

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  
9      public void mOnClick(View v) {  
10         NotificationManager notificationManager =  
11             (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);  
12  
13         switch (v.getId()) {  
14             case R.id.btnTest1:
```



### 실습 8-4

### Notifications

#### ■ MainActivity 클래스에 서로 다른 알림을 표시하는 코드 작성

```
15      Intent resultIntent1 = new Intent(this, SubActivity.class);
16      PendingIntent contentIntent1 = PendingIntent.getActivity(this, 0,
17          resultIntent1, PendingIntent.FLAG_UPDATE_CURRENT);
18
19      NotificationCompat.Builder builder1 = new NotificationCompat.Builder(this);
20      builder1.setSmallIcon(R.mipmap.ic_launcher);
21      builder1.setContentTitle("첫 번째 알림 제목");
22      builder1.setContentText("첫 번째 알림 텍스트");
23      builder1.setContentIntent(contentIntent1);
24      builder1.setAutoCancel(true);
25
26      notificationManager.notify(0, builder1.build());
```



### 실습 8-4

### Notifications

#### ■ MainActivity 클래스에 서로 다른 알림을 표시하는 코드 작성

```
27         break;
28     case R.id.btnTest2:
29         Intent resultIntent2 = new Intent(this, SubActivity.class);
30         PendingIntent contentIntent2 = PendingIntent.getActivity(this, 0,
31             resultIntent2, PendingIntent.FLAG_UPDATE_CURRENT);
32
33         NotificationCompat.Builder builder2 = new NotificationCompat.Builder(this);
34         builder2.setSmallIcon(R.mipmap.ic_launcher);
35         builder2.setContentTitle("두 번째 알림 제목");
36         builder2.setContentText("두 번째 알림 텍스트");
37         builder2.setContentIntent(contentIntent2);
38         builder2.setAutoCancel(true);
39
40         notificationManager.notify(1, builder2.build());
41         break;
42     }
43 }
44 }
```

## 02 인텐트 활용 ▶ 알림 처리



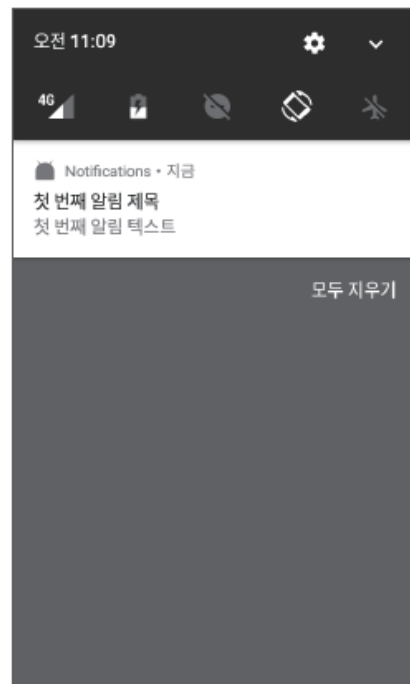
실습 8-4

Notifications

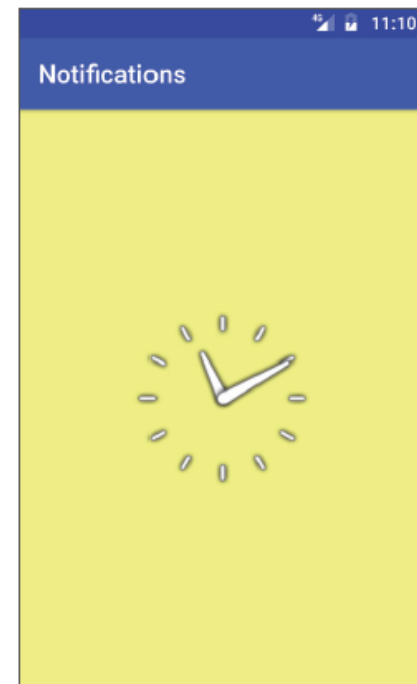
### ■ 실행 화면



(a) 초기 화면



(b) 첫 번째 알림



(c) 첫 번째 알림 터치 후

## 02 인텐트 활용 ▶ 알림 처리



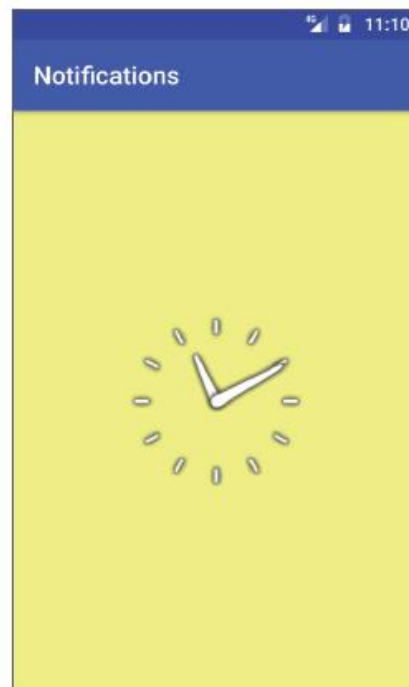
실습 8-4

Notifications

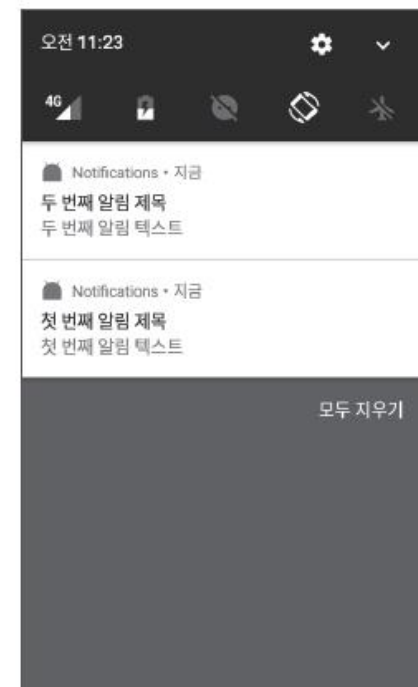
### ■ 실행 화면



(d) 두 번째 알림



(e) 두 번째 알림 터치 후



(f) 첫 번째와 두 번째 알림

그림 8-21 실행 화면





단계별로 배우는

# 안드로이드 프로그래밍