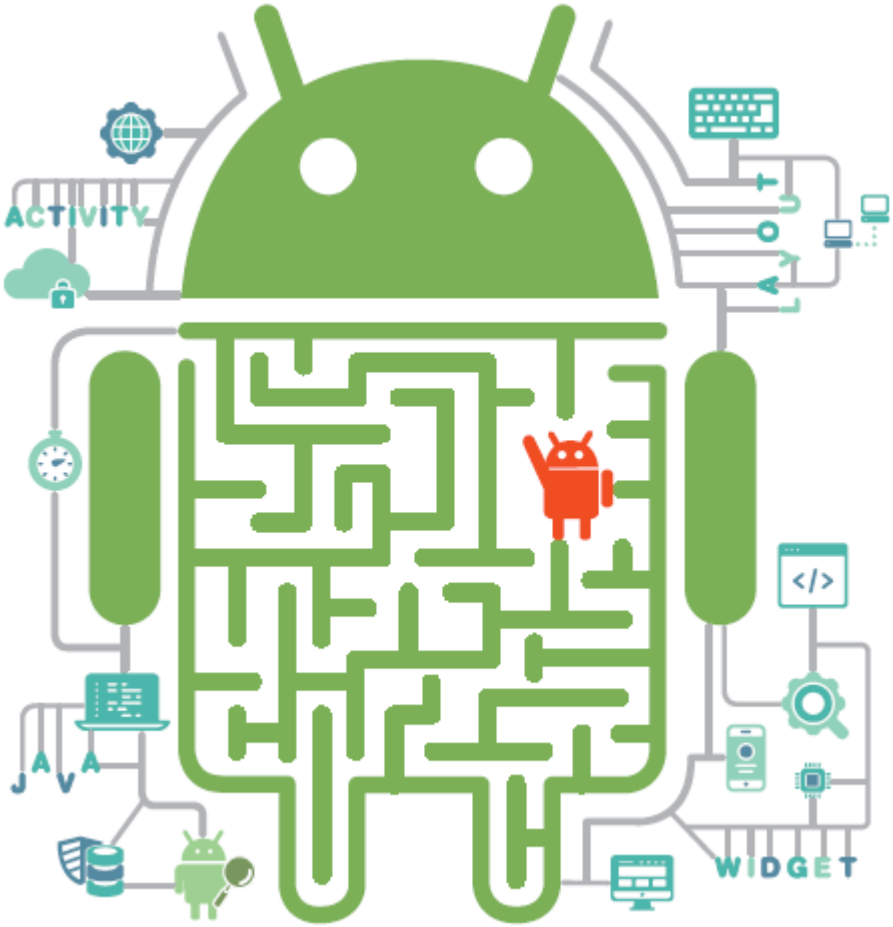
원리를 알면 IT가 맛있다
IT COOKBOOK

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 10. 서비스와 브로드캐스트 수신기

한빛아카데미 Hanbit Academy, Inc.

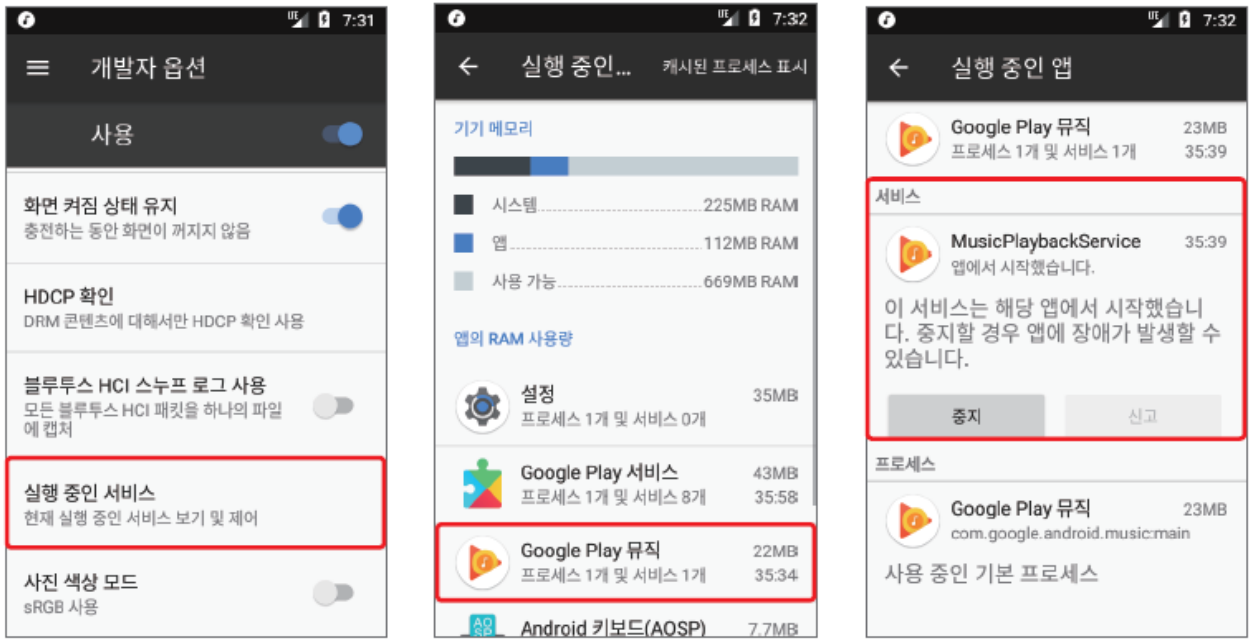
목차

학습목표

- 서비스의 생명주기를 이해한다.
- 시작된 서비스를 작성할 수 있다.
- 바인드된 서비스를 작성할 수 있다.
- 루퍼와 핸들러 구조를 파악한다.
- 브로드캐스트 수신기의 생명주기를 이해한다.
- 브로드캐스트 수신기를 작성할 수 있다.

서비스(Service)

- 시간이 오래 걸리는 작업을 백그라운드에서 수행
- 액티비티와 달리 UI를 가지지 않는 앱 구성 요소
- 서비스의 사용 예: 음악 재생이나 파일 다운로드



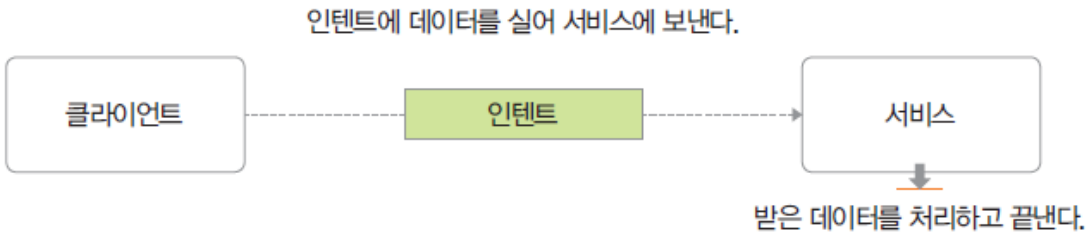
(a) 개발자 옵션 항목 선택 (b) 앱 선택 (c) 서비스 보기

그림 10-1 실행 중인 앱의 서비스 보기

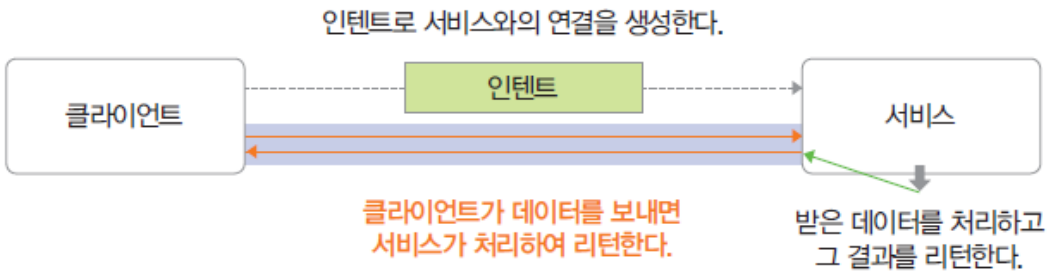
원리를 알면 IT가 맛있다
IT COOKBOOK

서비스의 작동

- 클라이언트: 서비스를 사용하는 앱 구성 요소. 대개는 액티비티
- 시작된 서비스: 클라이언트가 인텐트에 실어 보낸 데이터를 받아 처리하고 끝냄
- 바인드된 서비스: 클라이언트와 서비스 사이에 생성된 연결 통로. 데이터를 받아 처리한 후 결과 리턴



(a) 시작된 서비스



(b) 바인드된 서비스

그림 10-2 서비스의 작동 방식

원리를 알면 IT가 맛있다
IT COOKBOOK

서비스 작동 방식에 따른 생명주기

(a) 시작된 서비스 (b) 바인드된 서비스

그림 10-3 서비스의 생명주기

01 서비스 ▶ 기본 구조와 생명주기

서비스는 Service 클래스를 상속받고 일부 메서드를 재정의해 만듦

```
1   public class MyService extends Service {
2
3       @Override
4       public void onCreate() {
5           // 서비스 생성 시 수행할 코드를 여기에 작성한다.
6       }
7
8       @Override
9       public int onStartCommand(Intent intent, int flags, int startId) {
10          // 시작된(started) 서비스가 수행할 코드를 작성하되,
11          // 시간이 오래 걸리는 작업은 스레드를 생성하여 처리한다.
12          return START_STICKY;
13      }
14
15      @Override
16      public IBinder onBind(Intent intent) {
17          // 바인드된(bound) 서비스가 클라이언트와의 통신을 위한
18          // IBinder 타입 객체를 리턴하는 코드를 작성한다.
19          return null;
20      }
21
22      @Override
23      public void onDestroy() {
24          // 서비스 종료 시 수행할 코드를 작성한다.
25      }
26  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

시작된 서비스 작성 및 사용 방법

표 10-1 시작된 서비스 작성과 사용

시작된 서비스 작성	시작된 서비스 사용
① [필수] Service 클래스를 상속받아 **서비스 클래스를 생성**한다. ② [필수] **onStartCommand() 메서드를 재정의하고 수신된 인텐트를 처리**하는 코드를 작성한다. – 간단한 작업은 곧바로 처리하면 되지만, 시간이 오래 걸릴 작업이면 스레드를 생성하여 처리한다. ③ [선택] 작업 처리가 끝나면 **stopSelf()를 호출하여 서비스 스스로 중지(종료)**한다.	① [필수] 명시적 인텐트를 준비하고 **startService()를 호출하여 서비스를 시작**한다. – 동일 서비스에게 서로 다른 작업을 지시하고 싶다면 명시적 인텐트에 포함되는 부가 정보를 다르게 하여 매번 startService()를 호출하면 된다. ② [선택] 명시적 인텐트를 준비하고 **stopService()를 호출하여 서비스를 중지(종료)**시킨다.

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

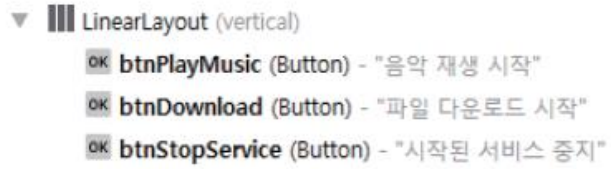


그림 10-4 컴포넌트 트리

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/btnPlayMusic"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
```

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

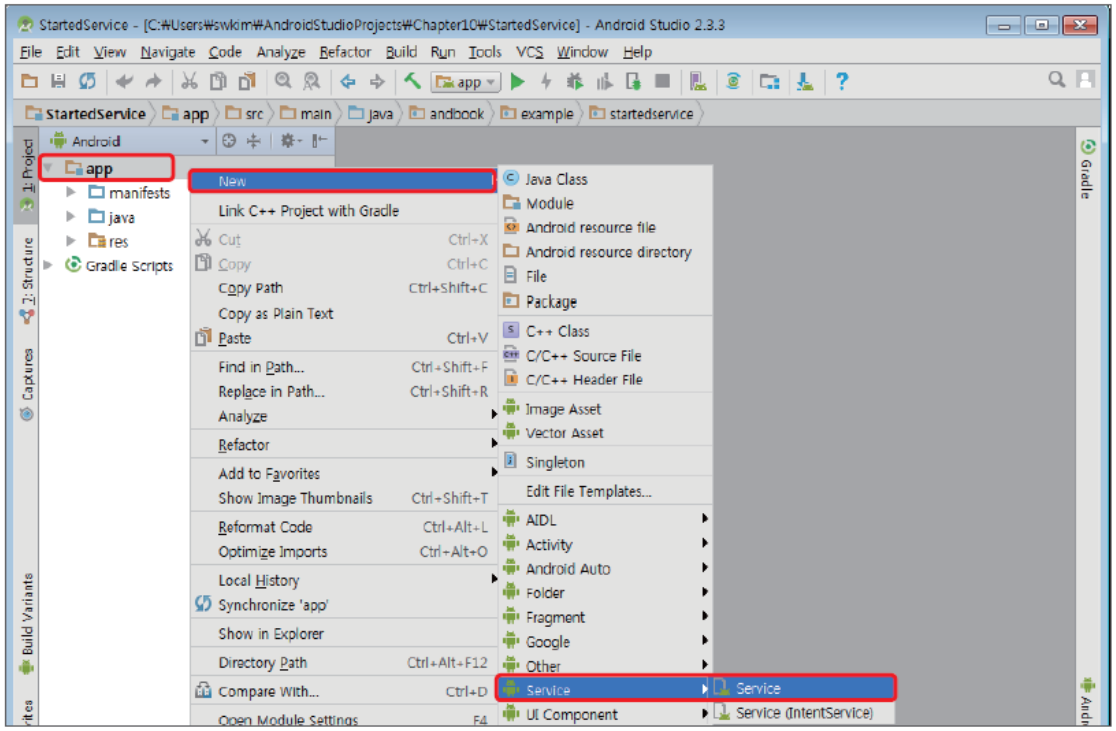
- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
11          android:onClick="mOnClick"
12          android:text="음악 재생 시작"/>
13      <Button
14          android:id="@+id/btnDownload"
15          android:layout_width="match_parent"
16          android:layout_height="wrap_content"
17          android:onClick="mOnClick"
18          android:text="파일 다운로드 시작"/>
19      <Button
20          android:id="@+id/btnStopService"
21          android:layout_width="match_parent"
22          android:layout_height="wrap_content"
23          android:onClick="mOnClick"
24          android:text="시작된 서비스 중지"/>
25  </LinearLayout>
```

01 서비스 ▶ 시작된 서비스 작성

실습 10-1 StartedService

- 좌측 프로젝트 창의 app 폴더에서 우클릭
- [New]-[Service]-[Service] 선택

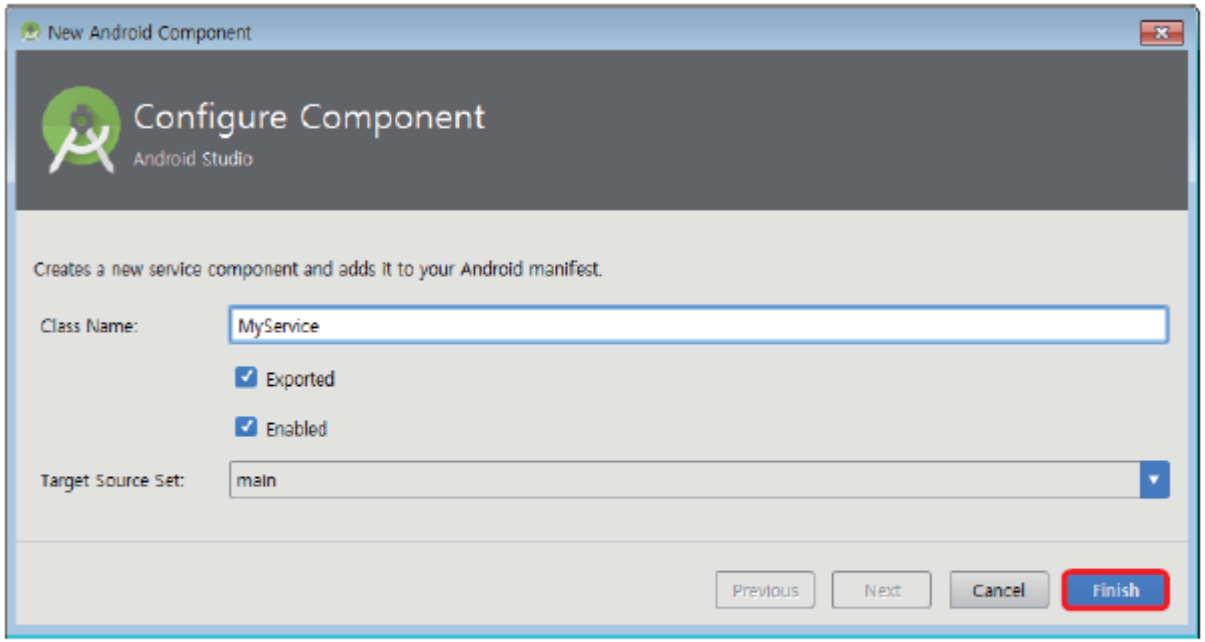


(a) 서비스 클래스 생성 (1)

01 서비스 ▶ 시작된 서비스 작성

실습 10-1 StartedService

- [Finish] 버튼을 클릭하여 기본값으로 서비스 클래스를 생성



(b) 서비스 클래스 생성 (2)

그림 10-5 서비스 클래스(MyService) 생성

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

- onCreate ( ), onStartCommand ( ) , onDestroy ( ) 메서드 재정의
- onStartCommand( ) 메서드는 빨간 줄 부분을 삭제하고 진행

```
@Override
public int onStartCommand(Intent intent, @IntDef(value = { Service.START_FLAG_REDELIVERY, Service.START_FLAG_RETRY }, flag = true) int flags, int startId) {
    return super.onStartCommand(intent, flags, startId);
}
```

그림 10-6 onStartCommand() 메서드의 초기 상태

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

- onCreate ( ), onStartCommand ( ) , onDestroy ( ) 메서드 재정의

MyService.java

```
1   public class MyService extends Service { // ① [필수] 서비스 클래스 생성
2
3       public MyService() {
4       }
5
6       @Override
7       public IBinder onBind(Intent intent) {
8           return null;
9       }
10
11      @Override
12      public void onCreate() {
13          super.onCreate();
14          Log.d("test", "onCreate() 호출!");
```

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

- onCreate ( ), onStartCommand ( ) , onDestroy ( ) 메서드 재정의

```
15      }
16
17      @Override
18      public int onStartCommand(Intent intent, int flags, int startId) {
19          // ② [필수] onStartCommand() 메서드 재정의; 수신된 인텐트 처리
20          if (intent.getAction().equals("andbook.example.PLAYMUSIC")) {
21              new MusicThread().start();
22          } else if (intent.getAction().equals("andbook.example.DOWNLOAD")) {
23              new DownloadThread().start();
24          }
25          return super.onStartCommand(intent, flags, startId);
26      }
27
28      @Override
29      public void onDestroy() {
```

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

- onCreate ( ), onStartCommand ( ) , onDestroy ( ) 메서드 재정의

```
30          super.onDestroy();
31          Log.d("test", "onDestroy() 호출!");
32      }
33  }
34
35  class MusicThread extends Thread {
36      @Override
37      public void run() {
38          Log.d("test", "음악 재생 시작!");
39          try {
40              Thread.sleep(2000);
41          } catch (InterruptedException e) {
42              e.printStackTrace();
43          }
44          Log.d("test", "음악 재생 종료!");
45      }
46  }
```

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

- onCreate ( ), onStartCommand ( ) , onDestroy ( ) 메서드 재정의

```
47
48  class DownloadThread extends Thread {
49      @Override
50      public void run() {
51          Log.d("test", "파일 다운로드 시작!");
52          try {
53              Thread.sleep(1000);
54          } catch (InterruptedException e) {
55              e.printStackTrace();
56          }
57          Log.d("test", "파일 다운로드 종료!");
58      }
59  }
```

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

- MainActivity 클래스에 서비스 시작 혹은 중지 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(this, MyService.class);
11          switch (v.getId()) {
12          case R.id.btnPlayMusic:
```

01 서비스 ▶ 시작된 서비스 작성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

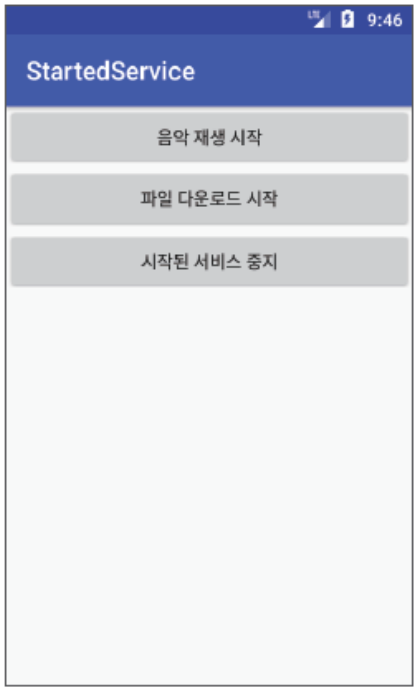
- MainActivity 클래스에 서비스 시작 혹은 중지 코드 작성

```
13              intent.setAction("andbook.example.PLAYMUSIC");
14              startService(intent); // ① [필수] 서비스 시작
15              break;
16          case R.id.btnDownload:
17              intent.setAction("andbook.example.DOWNLOAD");
18              startService(intent); // ① [필수] 서비스 시작
19              break;
20          case R.id.btnStopService:
21              stopService(intent); // ② [선택] 서비스 중지(=종료)
22              break;
23          }
24      }
25  }
```

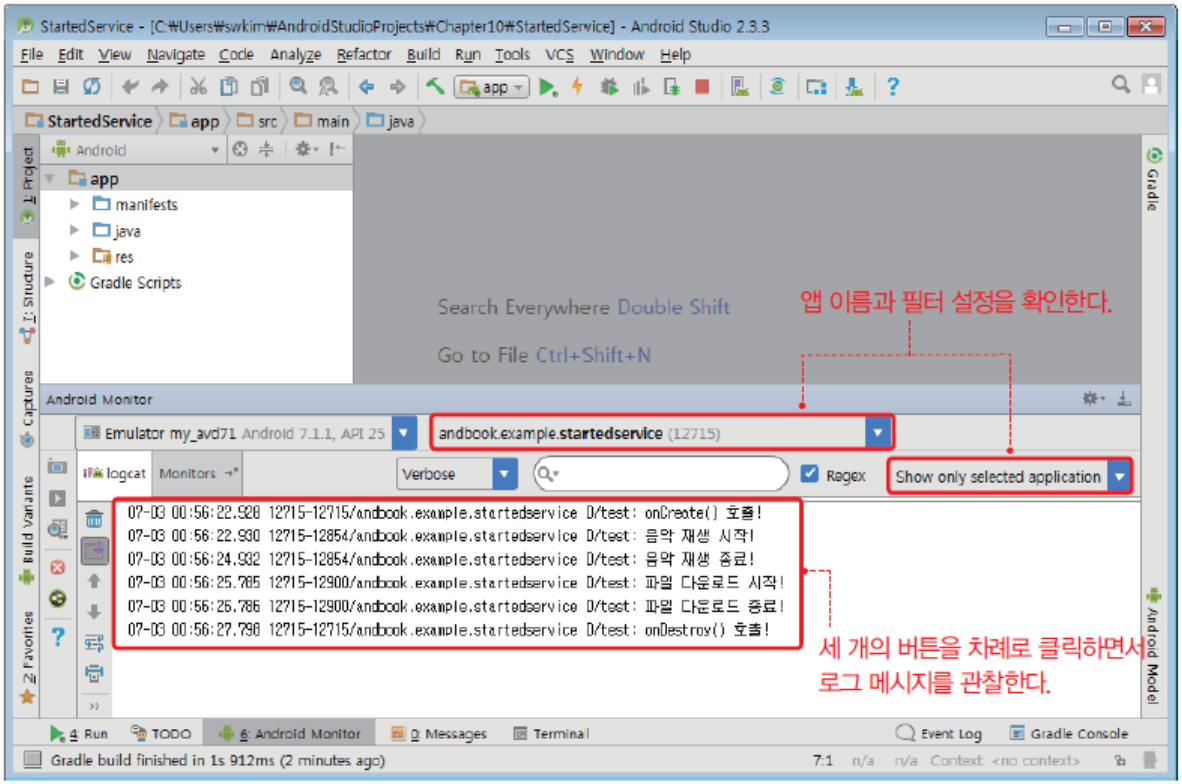
원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-1 StartedService

실행 화면



(a) 초기 화면



(b) Android Monitor 화면

그림 10-7 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

바인드된 서비스를 작성하고 사용하는 방법

표 10-2 바인드된 서비스 작성과 사용

바인드된 서비스 작성	바인드된 서비스 사용
① [필수] Service 클래스를 상속받아 **서비스 클래스를 생성**한다. ② [필수] Binder 클래스를 상속받아 **바인더 클래스를 정의**하고, 임의의 메서드(예: getService())를 하나 추가하여 **현재의 서비스 객체를 리턴**하는 코드를 작성한다. ③ [필수] **onBind() 메서드를 재정의하고 바인더 객체를 생성하여 리턴**하는 코드를 작성한다. – 리턴된 바인더 객체는 클라이언트 측에서 bindService()를 호출하여 얻을 수 있다. ④ [필수] 클라이언트를 위해 작업을 처리할 **서비스 메서드를 추가**한다. – 서비스 메서드의 형태와 개수에 제약은 없다.	① [필수] **ServiceConnection 인터페이스를 구현하는 객체를 생성**한다. – ServiceConnection 객체는 서비스와의 연결을 감시하는 역할을 한다. ② [필수] 명시적 인텐트를 준비하고 **bindService()를 호출하여 서비스와 연결**한다. – 이때 ①의 ServiceConnection 객체를 bindService()에 넘겨주어 서비스와 연결이 성공하거나 연결이 강제로 끊어지는 상황을 감지한다. ③ [필수] 서비스와 연결이 성공하면 ServiceConnection 객체의 onServiceConnected() 메서드가 호출되어 바인더 객체가 리턴된다. – **바인더 객체의 메서드(예: getService())를 호출**하면 서비스 객체를 얻을 수 있다. ④ [필수] ③에서 얻은 **서비스 객체를 통해 서비스 메서드를 호출**하여 서비스가 제공하는 기능을 자유롭게 사용한다. ⑤ [필수] 명시적 인텐트를 준비하고 **unbindService()를 호출하여 서비스와 연결을 해제**한다.

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-2 BoundService

- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

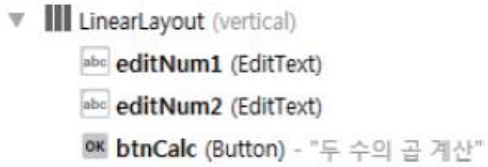


그림 10-8 컴포넌트 트리

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       <EditText
8           android:id="@+id/editNum1"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:inputType="number"/>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-2 BoundService

- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
12      <EditText
13          android:id="@+id/editNum2"
14          android:layout_width="match_parent"
15          android:layout_height="wrap_content"
16          android:inputType="number"/>
17      <Button
18          android:id="@+id/btnCalc"
19          android:layout_width="match_parent"
20          android:layout_height="wrap_content"
21          android:onClick="mOnClick"
22          android:text="두 수의 곱 계산"/>
23  </LinearLayout>
```

01 서비스 ▶ 시작된 서비스 작성

실습 10-2 BoundService

- [실습 10-1] 3단계를 따라 MyService 클래스 생성
- 생성된 MyService 클래스의 기존 코드 일부 변경과 새코드 추가

MainActivity.java

```
1    public class MyService extends Service { // ① [필수] 서비스 클래스 생성
2
3        public MyService() {
4        }
5
6        @Override
7        public IBinder onBind(Intent intent) {
8            // ③ [필수] onBind() 메서드 재정의; 바인더 객체 생성 & 리턴
```

01 서비스 ▶ 시작된 서비스 작성

실습 10-2 BoundService

- 생성된 MyService 클래스의 기존 코드 일부 변경과 새코드 추가

```
9           return new LocalBinder();
10      }
11
12      public class LocalBinder extends Binder {
13          // ② [필수] 바인더 클래스 정의; 현재의 서비스 객체 리턴
14          MyService getService() {
15              return MyService.this;
16          }
17      }
18
19      public int CalcNum(int m, int n) { // ④ [필수] 서비스 메서드 추가
20          return m * n;
21      }
22  }
```

01 서비스 ▶ 시작된 서비스 작성

실습 10-2 BoundService

- MainActivity 클래스에 버튼 클릭 시 계산 기능을 사용하는 코드 작성

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       private MyService mService;
4       private boolean mBound = false;
5
6       @Override
7       protected void onCreate(Bundle savedInstanceState) {
8           super.onCreate(savedInstanceState);
9           setContentView(R.layout.activity_main);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-2 BoundService

- **MainActivity 클래스에 버튼 클릭 시 계산 기능을 사용하는 코드 작성**

```
10      }
11
12      ServiceConnection mConn = new ServiceConnection() { // ① ServiceConnection 객체 생성
13
14          @Override
15          public void onServiceConnected(ComponentName name, IBinder service) {
16              // ③ 바인더 객체의 메서드(예: getService()) 호출; 서비스 객체 얻기
17              mService = ((MyService.LocalBinder) service).getService();
18              mBound = true;
19          }
20
21          @Override
22          public void onServiceDisconnected(ComponentName name) {
23              mBound = false;
24          }
25      };
26
27      @Override
28      protected void onStart() {
29          super.onStart();
30          Intent intent = new Intent(this, MyService.class);
31          bindService(intent, mConn, Context.BIND_AUTO_CREATE); // ② 서비스와 연결하기
32      }
33
34      @Override
```

01 서비스 ▶ 시작된 서비스 작성

실습 10-2 BoundService

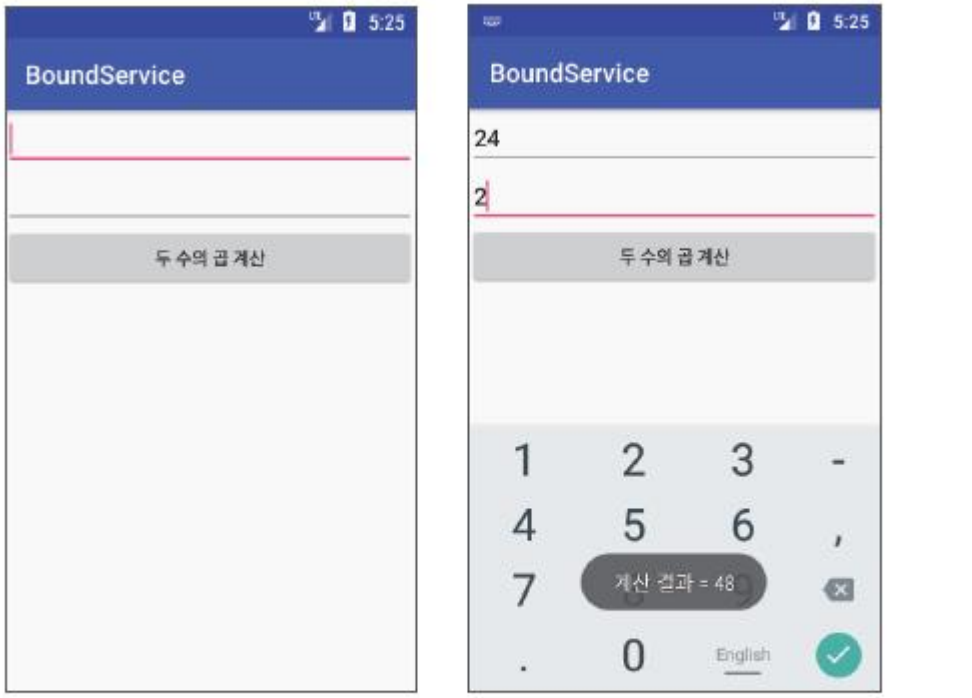
- MainActivity 클래스에 버튼 클릭 시 계산 기능을 사용하는 코드 작성

```
35      protected void onStop() {
36          super.onStop();
37          if (mBound) {
38              unbindService(mConn); // ⑤ 서비스와 연결 해제
39              mBound = false;
40          }
41      }
42
43      public void mOnClick(View v) {
44          switch (v.getId()) {
45          case R.id.btnCalc:
46              if (mBound) {
47                  EditText editNum1 = (EditText) findViewById(R.id.editNum1);
48                  EditText editNum2 = (EditText) findViewById(R.id.editNum2);
49                      if (editNum1.length() > 0 && editNum2.length() > 0) {
50                      int num1 = Integer.parseInt(editNum1.getText().toString());
51                      int num2 = Integer.parseInt(editNum2.getText().toString());
52                      // ④ 서비스 객체를 통해 서비스 메서드 호출
53                      int result = mService.CalcNum(num1, num2);
54                      Toast.makeText(this, "계산 결과 = " + result, Toast.LENGTH_SHORT).show();
55                  }
56              }
57              break;
58          }
59      }
60  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-2 BoundService

실행 화면



(a) 초기 화면 (b) 숫자 입력 후 [두 수의 곱 계산] 버튼 클릭

그림 10-9 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

예시

- StartedService 예제의 MyService 클래스 코드를 약간 수정한 것
- "andbook.example.PLAYMUSIC" 액션을 담고 있는 인텐트를 수신했을 때 별도의 스레드를 생성하지 않고 곧바로 처리하되 시간이 오래 걸리는 상황

```
public class MyService extends Service { // ① [필수] 서비스 클래스 생성
    …
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // ② [필수] onStartCommand() 메서드 재정의; 수신된 인텐트 처리
        if (intent.getAction().equals("andbook.example.PLAYMUSIC")) {
            try {
                Thread.sleep(10000); // 클라이언트의 요청을 처리하는 데 10초가 걸림!
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        } else if (intent.getAction().equals("andbook.example.DOWNLOAD")) {
            new DownloadThread().start();
        }
        return super.onStartCommand(intent, flags, startId);
    }
    …
}
```

원리를 알면 IT가 맛있다
IT COOKBOOK

ANR (Application Not Responding) 발생

- 메인 스레드가 작업을 오래 처리하느라 UI 이벤트(예: 화면 터치)를 처리하지 못할 때 주로 발생

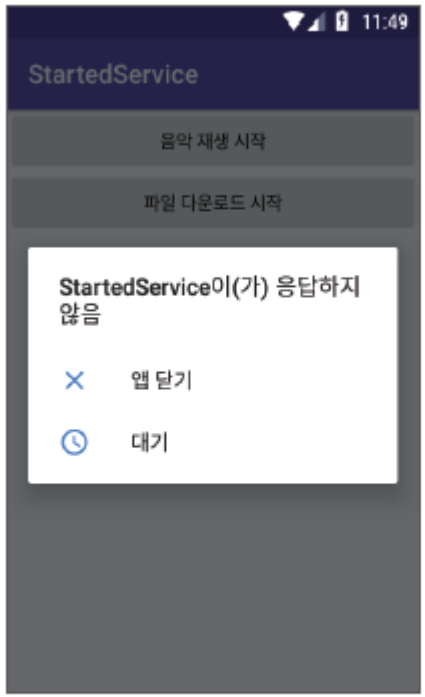


그림 10-10 ANR 발생

원리를 알면 IT가 맛있다
IT COOKBOOK

메시지 큐 (Message Queue)

- 하나의 스레드에서 작업을 처리하되 당장 처리하지 못하는 것은 자료 구조에 넣어 두었다가 차례로 꺼내서 처리하는 방식. 기본으로 제공

한 스레드가 여러 작업을 순차적으로 처리하는 구조

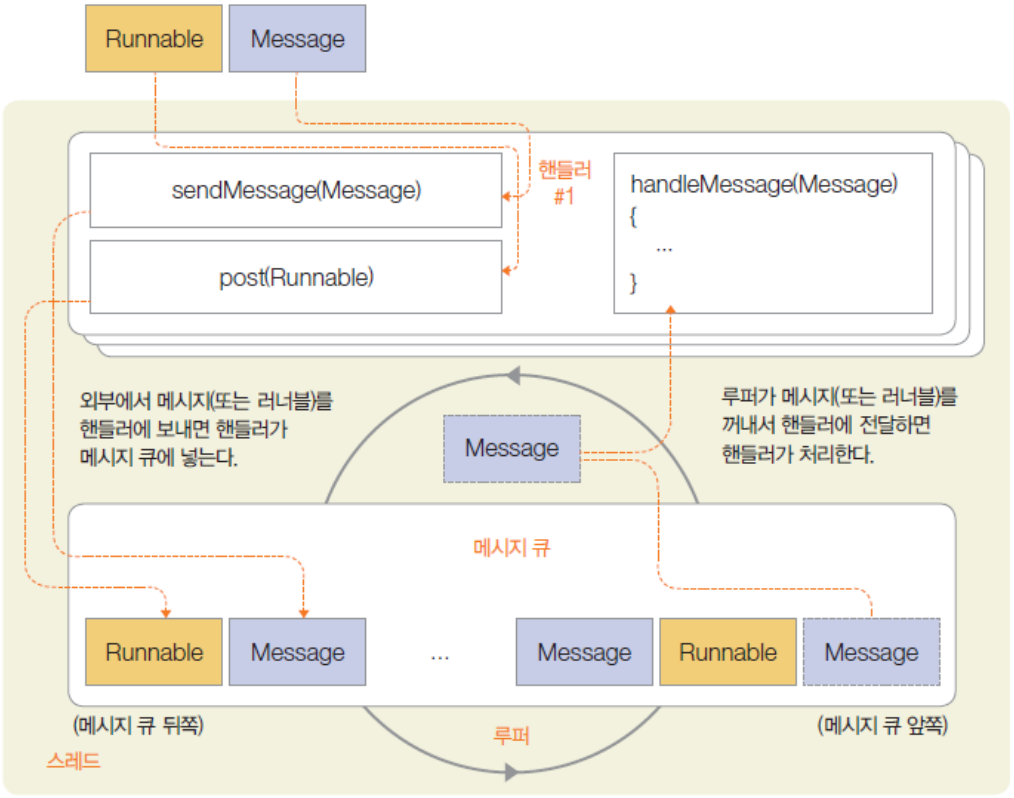


그림 10-11 스레드, 메시지 큐, 루퍼, 핸들러의 관계

원리를 알면 IT가 맛있다
IT COOKBOOK

각 개체의 이름과 기능

표 10-3 [그림 10-11]을 구성하는 개체 이름과 기능

이름	기능
메시지Message **러너블**Runnable	메시지와 러너블 모두 스레드가 처리할 대상이다. 메시지는 처리할 데이터를 담고 있고, 러너블은 실행할 코드를 담고 있다는 차이가 있다.
메시지 큐 Message Queue	큐Queue는 FIFOFirst In First Out(먼저 들어온 것이 먼저 나감) 방식으로 동작하는 대표적인 자료 구조이다. 메시지 큐는 스레드당 한 개만 존재하며, 메시지와 러너블이 들어온 순으로 저장되어 있다.
루퍼Looper	루퍼는 스레드당 한 개만 존재하며, 메시지 큐의 맨 앞에서 메시지나 러너블을 꺼내서 핸들러에 전달한다.
핸들러Handler	핸들러는 메시지 큐에 메시지나 러너블을 넣는 기능도 하고, 루퍼가 전달해준 메시지나 러너블을 처리하는 기능도 한다. 메시지 큐나 루퍼와 달리 핸들러는 스레드당 여러 개가 존재할 수 있다. 특정 핸들러를 통해 넣은 메시지나 러너블은 반드시 같은 핸들러에 전달되어 처리된다.

원리를 알면 IT가 맛있다
IT COOKBOOK

메시지를 보내고 처리하는 전형적인 코드

메시지 보내기

```
Message msg = Message.obtain(); // 메시지 객체를 얻는다.
msg.what = 0; // 메시지를 구분하는 숫자다.
msg.arg1 = 10; // arg1에는 임의 정숫값을 넣을 수 있다.
msg.arg2 = 20; // arg2에도 임의 정숫값을 넣을 수 있다.
msg.obj = "hello"; // obj에는 임의 객체를 넣을 수 있다.
mHandler.sendMessage(msg); // 핸들러를 통해 메시지를 보낸다.
```

메시지 처리하기

```
mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        if (msg.what == 0) { // 메시지를 구분하는 숫자를 체크한다.
            Log.d("test", "다음 데이터를 처리합니다.");
            int n1 = msg.arg1; // 정숫값을 꺼낸다.
            int n2 = msg.arg2; // 정숫값을 꺼낸다.
            String str = (String) msg.obj; // 객체를 꺼낸다.
        }
    }
};
```

01 서비스 ▶ 루퍼와 핸들러

러너블을 보내고 처리하는 전형적인 코드

러너블 보내기

```
mHandler.post( new Runnable() {
    @Override
    public void run() {
        Log.d("test", "다음 코드를 실행합니다.");
    }
} );
```

러너블 처리하기

```
mHandler = new Handler(); // Runnable 객체가 전달되면 run() 메서드를 호출하여 실행한다.
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-3 LooperHandler1

이 예제의 제어 흐름

액티비티에서 버튼 클릭 → 1씩 증가되는 숫자를 메시지에 담아 핸들러로 전송 → 스레드의 메시지 큐에 저장 → 루퍼가 메시지를 꺼내서 핸들러에 전달 → 핸들러가 메시지를 받아 계산하고 그 결과를 로그 메시지로 남김

컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

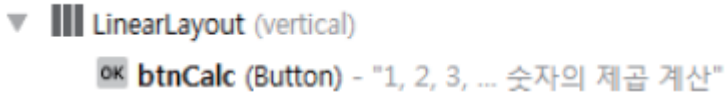


그림 10-12 컴포넌트 트리

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-3 LooperHandler1

- 컴포넌트 트리를 참고하여 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       <Button
8           android:id="@+id/btnCalc"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:onClick="mOnClick"
12          android:text="1, 2, 3, … 숫자의 제곱 계산"/>
13  </LinearLayout>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-3 LooperHandler1

- MainActivity 클래스에 메시지 보내고 핸들러에서 처리하는 코드 작성

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       private MyThread mThread;
4       private int mNumber;
5
6       @Override
7       protected void onCreate(Bundle savedInstanceState) {
8           super.onCreate(savedInstanceState);
9           setContentView(R.layout.activity_main);
10      }
11
12      @Override
13      protected void onStart() {
14          super.onStart();
15          mThread = new MyThread();
16          mThread.start(); // 스레드 시작
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-3 LooperHandler1

- MainActivity 클래스에 메시지 보내고 핸들러에서 처리하는 코드 작성

```
17      }
18
19      @Override
20      protected void onStop() {
21          super.onStop();
22          Message msg = Message.obtain();
23          msg.what = -1;
24          mThread.mHandler.sendMessage(msg); // 메시지 보내기; 루퍼 & 스레드 종료
25      }
26
27      public void mOnClick(View v) {
28          Message msg = Message.obtain();
29          msg.what = 0;
30          msg.arg1 = ++mNumber;
31          mThread.mHandler.sendMessage(msg); // 메시지 보내기; 처리할 숫자 데이터
32      }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-3 LooperHandler1

- MainActivity 클래스에 메시지 보내고 핸들러에서 처리하는 코드 작성

```
33
34      private class MyThread extends Thread {
35
36          public Handler mHandler;
37
38          @Override
39          public void run() {
40              Looper.prepare(); // 루퍼 초기화
41
42              mHandler = new Handler() { // 핸들러 생성
43                  @Override
44                  public void handleMessage(Message msg) {
45                      if (msg.what == 0) {
46                          // 계산 지연 시간
47                          try {
48                              Thread.sleep(3000);
49                          } catch (InterruptedException e) {
50                              e.printStackTrace();
51                          }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-3 LooperHandler1

- MainActivity 클래스에 메시지 보내고 핸들러에서 처리하는 코드 작성

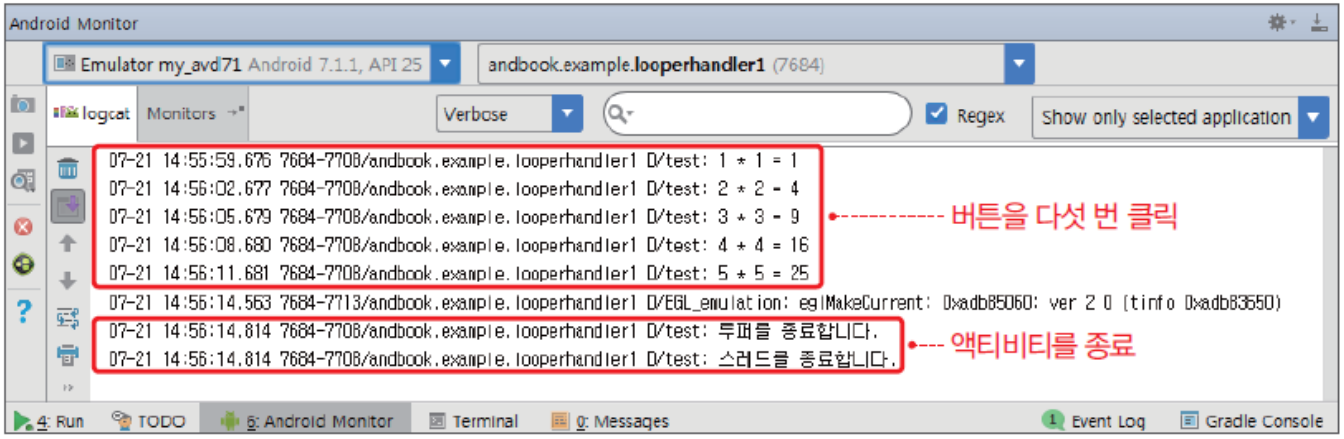
```
52                      // 계산 결과 출력
53                      int result = msg.arg1 * msg.arg1;
54                      Log.d("test", msg.arg1 + " * " + msg.arg1 + " = " + result);
55                  } else if (msg.what == -1) {
56                      Looper.myLooper().quit(); // 루퍼 종료 → 스레드 종료
57                      Log.d("test", "루퍼를 중료합니다.");
58                  }
59              }
60          };
61
62          Looper.loop(); // 루퍼 실행
63          Log.d("test", "스레드를 중료합니다.");
64      }
65  }
66  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-3 LooperHandler1

실행 화면

(a) 초기 화면



(b) Android Monitor 화면

그림 10-13 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-4 LooperHandler2

이 예제의 제어 흐름

액티비티에서 버튼 클릭 → 1씩 증가되는 숫자를 메시지에 담아 핸들러로 전송 → 스레드의 메시지 큐에 저장 → 루퍼가 메시지를 꺼내서 핸들러에 전달 → 핸들러가 메시지를 받아 계산하고 결과를 메시지에 담아 메인 핸들러로 전송 → 메인 스레드의 메시지 큐에 저장 → 메인 루퍼가 메시지를 꺼내서 메인 핸들러에 전달 → 메인 핸들러가 메시지를 받아 결과를 액티비티 화면에 출력

컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

그림 10-14 컴포넌트 트리

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-4 LooperHandler2

- 컴포넌트 트리를 참고하여 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       <Button
8           android:id="@+id/btnCalc"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:onClick="mOnClick"
12          android:text="1, 2, 3, … 숫자의 제곱 계산"/>
13      <TextView
14          android:id="@+id/textResult"
15          android:layout_width="match_parent"
16          android:layout_height="wrap_content"
17          android:gravity="center_horizontal"
18          android:textSize="24dp"
19          android:textStyle="bold|italic"/>
20  </LinearLayout>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-4 LooperHandler2

- MainActivity.java는 LooperHandler1 예제의 MainActivity.java 내용 재사용
- 계산한 결과를 액티비티 화면에 표시하는 데 필요한 코드 추가

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       private MyThread mThread;
4       private int mNumber;
5       private TextView mTextResult;
6
7       @Override
8       protected void onCreate(Bundle savedInstanceState) {
9           super.onCreate(savedInstanceState);
10          setContentView(R.layout.activity_main);
11          mTextResult = (TextView) findViewById(R.id.textResult);
12      }
13
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-4 LooperHandler2

- 계산한 결과를 액티비티 화면에 표시하는 데 필요한 코드 추가

```
14      @Override
15      protected void onStart() {
16          super.onStart();
17          mThread = new MyThread();
18          mThread.start(); // 스레드 시작
19      }
20
21      @Override
22      protected void onStop() {
23          super.onStop();
24          Message msg = Message.obtain();
25          msg.what = -1;
26          mThread.mHandler.sendMessage(msg); // 메시지 보내기; 루퍼 & 스레드 종료
27      }
28
29      public void mOnClick(View v) {
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-4 LooperHandler2

- 계산한 결과를 액티비티 화면에 표시하는 데 필요한 코드 추가

```
30          Message msg = Message.obtain();
31          msg.what = 0;
32          msg.arg1 = ++mNumber;
33          mThread.mHandler.sendMessage(msg); // 메시지 보내기; 처리할 숫자 데이터
34      }
35
36      private Handler mMainHandler = new Handler() {
37          @Override
38          public void handleMessage(Message msg) {
39              // 계산 결과 출력
40              if (msg.what == 0) {
41                  mTextResult.setText(msg.arg1 + " * " + msg.arg1 + " = " + msg.arg2);
42              }
43          }
44      };
45
46      private class MyThread extends Thread {
47
48          public Handler mHandler;
49
50          @Override
51          public void run() {
52              Looper.prepare(); // 루퍼 초기화
53
54              mHandler = new Handler() { // 핸들러 생성
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-4 LooperHandler2

- 계산한 결과를 액티비티 화면에 표시하는 데 필요한 코드 추가

```
55                @Override
56                public void handleMessage(Message msg) {
57                    if (msg.what == 0) {
58                        // 계산 지연 시간
59                        try {
60                            Thread.sleep(3000);
61                        } catch (InterruptedException e) {
62                            e.printStackTrace();
63                        }
64                        // 계산 결과 전송
65                        int result = msg.arg1 * msg.arg1;
66                        //mTextResult.setText(msg.arg1 + " * " + msg.arg1 + " = " + msg.arg2);
67                        Message msgResult = Message.obtain();
68                        msgResult.what = 0;
69                        msgResult.arg1 = msg.arg1;
70                        msgResult.arg2 = result;
71                        mMainHandler.sendMessage(msgResult);
72                    } else if (msg.what == -1) {
73                        Looper.myLooper().quit(); // 루퍼 종료 → 스레드 종료
74                        Log.d("test", "루퍼를 종료합니다.");
75                    }
76                }
77            };
78
79            Looper.loop(); // 루퍼 실행
80            Log.d("test", "스레드를 종료합니다.");
81        }
82    }
83 }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-4 LooperHandler2

실행 화면

(a) 메인 스레드에서 화면 출력→ 성공

(b) 사용자 스레드에서 화면 출력 → 실패

그림 10-15 실행 화면

02 브로드캐스트 수신기

브로드캐스트 수신기(Broadcast Receiver)

- 안드로이드 시스템, 다른 앱, 자신의 앱에서 발생시킨 이벤트를 수신하여 처리하는 앱 구성 요소
- 브로드캐스트 이벤트의 활용 예로는 배터리 상태를 표시하는 기능

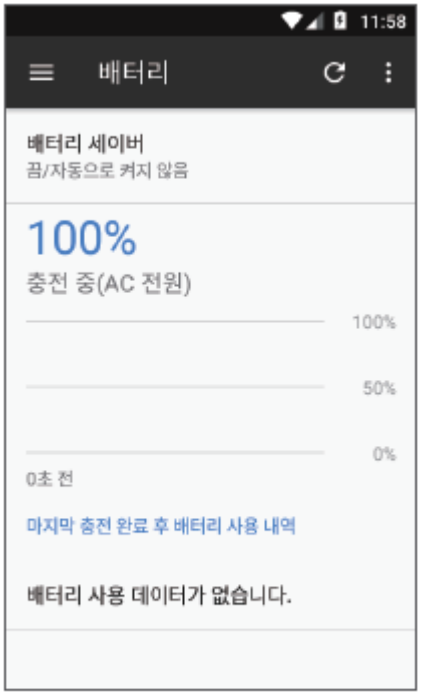


그림 10-16 브로드캐스트 수신기 활용 (예: 배터리 상태 얻기)

원리를 알면 IT가 맛있다
IT COOKBOOK

브로드캐스트 수신기

- 안드로이드 앱의 4대 구성 요소 중 구조가 가장 간단
- BroadcastReceiver 클래스를 상속받고 onReceive( ) 메서드를 재정의하여 만듦

```
1   public class MyReceiver extends BroadcastReceiver {
2
3       @Override
4       public void onReceive(Context context, Intent intent) {
5           /* 액션 = 브로드캐스트 이벤트를 의미한다. */
6           if (intent.getAction().equals("android.intent.action.BATTERY_CHANGED")) {
7               // 인텐트를 분석하여 BATTERY_CHANGED 이벤트를 처리한다.
8           } else if (intent.getAction().equals("android.intent.action.BATTERY_LOW")) {
9               // 인텐트를 분석하여 BATTERY_LOW 이벤트를 처리한다.
10          }
11      }
12  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

브로드캐스트 수신기 이벤트 처리

- 이벤트를 처리하기 위해 안드로이드 시스템에 등록할 때, 두 가지 방법

ⓐ 매니페스트에 정적 등록한다.

– 이벤트 발생 시 항상 응답한다.

ⓑ 앱 실행 중에 자바 코드를 이용하여 필요하면 동적 등록하고 필요 없으면 동적 해제한다.

– 등록되어 있는 동안에만 이벤트 발생 시 응답한다.

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

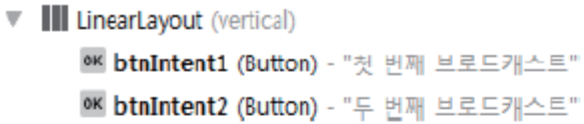


그림 10-17 컴포넌트 트리

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">
```

02 브로드캐스트 수신기 ▶ 기본 구조와 생명주기

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

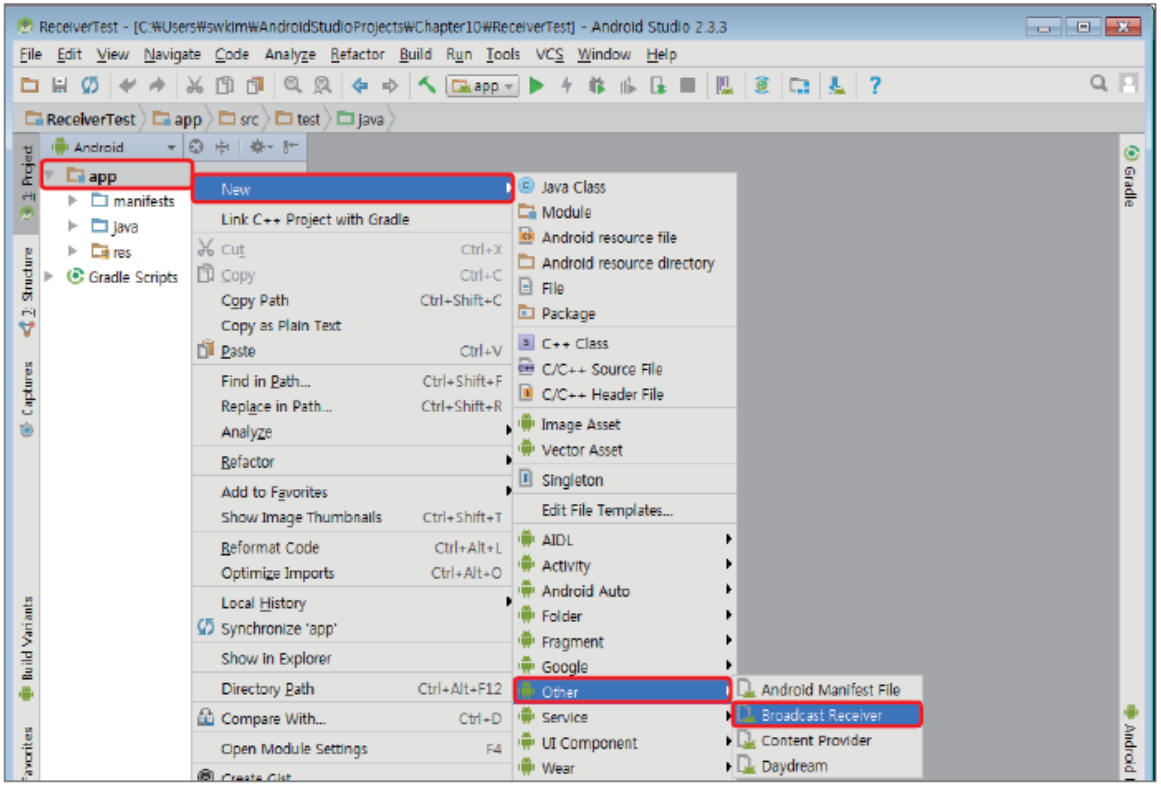
- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
7       <Button
8           android:id="@+id/btnIntent1"
9           android:layout_width="match_parent"
10          android:layout_height="wrap_content"
11          android:onClick="mOnClick"
12          android:text="첫 번째 브로드캐스트"/>
13      <Button
14          android:id="@+id/btnIntent2"
15          android:layout_width="match_parent"
16          android:layout_height="wrap_content"
17          android:onClick="mOnClick"
18          android:text="두 번째 브로드캐스트"/>
19  </LinearLayout>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

- 좌측 프로젝트 창의 app 폴더에서 우클릭
- [New]-[Other]-[Broadcast Receiver] 클릭

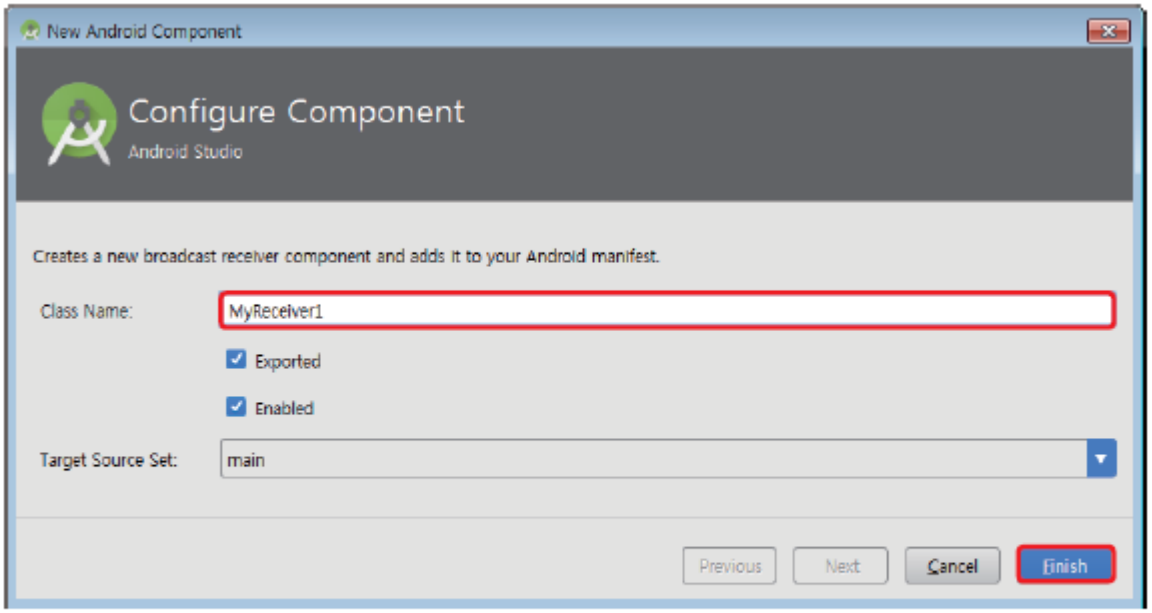


(a) 브로드캐스트 수신기 클래스 생성 (1)

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

- 대화상자가 열리면 클래스 이름을 “MyReceiver1”로 변경
- [Finish] 클릭



(b) 브로드캐스트 수신기 클래스 생성 (2)

그림 10-18 브로드캐스트 수신기 클래스(MyReceiver1, MyReceiver2) 생성

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

- 전달받은 인텐트에서 "mydata"라는 정수형 엑스트라를 꺼내서 출력하는 코드 작성

MyReceiver1.java

```
1   public class MyReceiver1 extends BroadcastReceiver {
2
3       @Override
4       public void onReceive(Context context, Intent intent) {
5           int data = intent.getIntExtra("mydata", -1);
6           Toast.makeText(context, "MyReceiver1: mydata = " + data,
7                   Toast.LENGTH_SHORT).show();
8       }
9   }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

- 전달받은 인텐트에서 "mydata"라는 정수형 엑스트라를 꺼내서 출력하는 코드 작성

MyReceiver2.java

```
1   public class MyReceiver2 extends BroadcastReceiver {
2
3       @Override
4       public void onReceive(Context context, Intent intent) {
5           int data = intent.getIntExtra("mydata", -1);
6           Toast.makeText(context, "MyReceiver2: mydata = " + data,
7                   Toast.LENGTH_SHORT).show();
8       }
9   }
```

02 브로드캐스트 수신기 ▶ 기본 구조와 생명주기

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

- AndroidManifest.xml 열기
- MyReceiver1 엘리먼트에 인텐트 필터 추가
- MyReceiver2 엘리먼트 삭제

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.receivertest">
…       …
17          <receiver
18              android:name=".MyReceiver1"
19              android:enabled="true"
20              android:exported="true">
21              <intent-filter>
22                  <action android:name="andbook.example.TESTEVENT1"/>
23              </intent-filter>
24          </receiver>
25          <receiver
26              android:name=".MyReceiver2"
27              android:enabled="true"
28              android:exported="true">
29          </receiver>
30      </application>
31  </manifest>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

코드 작성

- MyReceiver2를 동적 등록 및 해제하는 코드와, 버튼 클릭에 따라 MyReceiver1과 MyReceiver2를 각각 활성화하는 코드 작성

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       private BroadcastReceiver mReceiver;
4       private IntentFilter mFilter;
5
6       @Override
7       protected void onCreate(Bundle savedInstanceState) {
8           super.onCreate(savedInstanceState);
9           setContentView(R.layout.activity_main);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

코드 작성

- MyReceiver2를 동적 등록 및 해제하는 코드와, 버튼 클릭에 따라 MyReceiver1과 MyReceiver2를 각각 활성화하는 코드 작성

```
10
11          mReceiver = new MyReceiver2();
12          mFilter = new IntentFilter("andbook.example.TESTEVENT2");
13      }
14
15      @Override
16      protected void onStart() {
17          super.onStart();
18          registerReceiver(mReceiver, mFilter);
19      }
20
21      @Override
22      protected void onStop() {
23          super.onStop();
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

코드 작성

- MyReceiver2를 동적 등록 및 해제하는 코드와, 버튼 클릭에 따라 MyReceiver1과 MyReceiver2를 각각 활성화하는 코드 작성

```
24            unregisterReceiver(mReceiver);
25        }
26
27        public void mOnClick(View v) {
28            Intent intent = new Intent();
29            switch (v.getId()) {
30            case R.id.btnIntent1:
31                intent.setAction("andbook.example.TESTEVENT1");
32                intent.putExtra("mydata", 100);
33                sendBroadcast(intent);
34                break;
35            case R.id.btnIntent2:
36                intent.setAction("andbook.example.TESTEVENT2");
37                intent.putExtra("mydata", 200);
38                sendBroadcast(intent);
39                break;
40            }
41        }
42    }
```

02 브로드캐스트 수신기 ▶ 기본 구조와 생명주기

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

16~19행

```
▶ Intent registerReceiver(BroadcastReceiver receiver, IntentFilter filter)
```

22~25행

```
▶ void unregisterReceiver(BroadcastReceiver receiver)
```

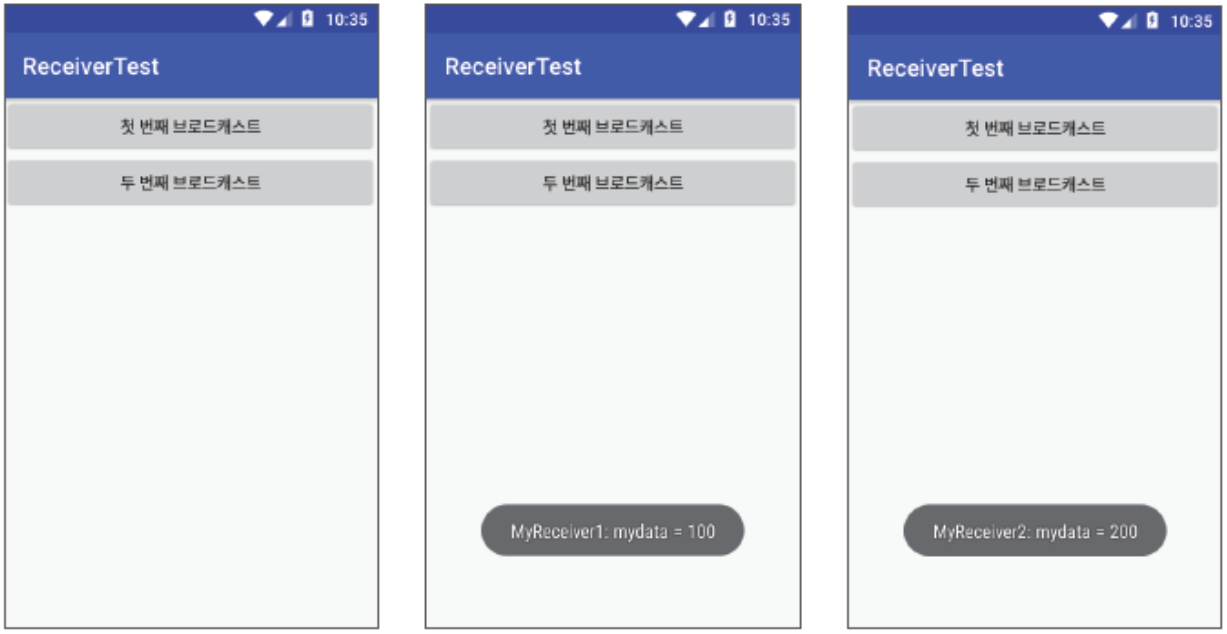
31~33행, 36~38행

```
▶ void sendBroadcast(Intent intent)
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-5 ReceiverTest

실행 화면



(a) 초기 화면 (b) MyReceiver1 활성화 (c) MyReceiver2 활성화

그림 10-19 실행 화면

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-6 DetectSystemEvent

ACTION_BOOT_COMPLETED 이벤트 처리

- 매니페스트에 다음 한 줄을 추가

```
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
```

res/layout/activity_main.xml에 앱 동작의 설명 텍스트 추가

activity_main.xml

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <TextView
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       android:layout_width="match_parent"
5       android:layout_height="match_parent"
6       android:gravity="center"
7       android:text="부트 완료(ACTION_BOOT_COMPLETED),
8         \n사용자 존재(ACTION_USER_PRESENT),
9         \n헤드셋 플러그(ACTION_HEADSET_PLUG)
10        \n이벤트를 감지합니다."
11      android:textSize="16dp"
12      android:textStyle="bold|italic">
13  </TextView>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-6 DetectSystemEvent

- [실습 10-5] 3단계를 따라 MyReceiver1과 MyReceiver2 클래스 생성
- 생성된 MyReceiver1 클래스 수정

MyReceiver1.java

```
1   public class MyReceiver1 extends BroadcastReceiver {
2
3       @Override
4       public void onReceive(Context context, Intent intent) {
5           if (intent.getAction().equals(Intent.ACTION_BOOT_COMPLETED)) {
6               Toast.makeText(context, "ACTION_BOOT_COMPLETED",
7                       Toast.LENGTH_SHORT).show();
8           } else if (intent.getAction().equals(Intent. ACTION_TIMEZONE_CHANGED)) {
9               Toast.makeText(context, " ACTION_TIMEZONE_CHANGED ",
10                      Toast.LENGTH_SHORT).show();
11          }
12      }
13  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-6 DetectSystemEvent

생성된 MyReceiver2 클래스 수정

MyReceiver2.java

```
1   public class MyReceiver2 extends BroadcastReceiver {
2
3       @Override
4       public void onReceive(Context context, Intent intent) {
5           if (intent.getAction().equals(Intent.ACTION_HEADSET_PLUG)) {
6               int state = intent.getIntExtra("state", -1);
7               Toast.makeText(context, "ACTION_HEADSET_PLUG: state = " + state,
8                       Toast.LENGTH_SHORT).show();
9           }
10      }
11  }
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-6 DetectSystemEvent

AndroidManifest.xml 수정

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.detectsystemevent">
4       <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
5       <application
6           android:allowBackup="true"
…           …
18          <receiver
19              android:name=".MyReceiver1"
20              android:enabled="true"
21              android:exported="true">
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-6 DetectSystemEvent

AndroidManifest.xml 수정

```
22              <intent-filter>
23                  <action android:name="android.intent.action.BOOT_COMPLETED"/>
24              </intent-filter>
25              <intent-filter>
26                  <action android:name="android.intent.action. TIMEZONE_CHANGED"/>
27              </intent-filter>
28          </receiver>
29          <receiver
30              android:name=".MyReceiver2"
31              android:enabled="true"
32              android:exported="true">
33          </receiver>
34      </application>
35  </manifest>
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-6 DetectSystemEvent

- **MainActivity 클래스에 MyReceiver2 동적 등록 및 해제 코드 작성**

MainActivity.java

```
1   public class MainActivity extends AppCompatActivity {
2
3       private BroadcastReceiver mReceiver;
4       private IntentFilter mFilter;
5
6       @Override
7       protected void onCreate(Bundle savedInstanceState) {
8           super.onCreate(savedInstanceState);
9           setContentView(R.layout.activity_main);
10
11          mReceiver = new MyReceiver2();
12          mFilter = new IntentFilter(Intent.ACTION_HEADSET_PLUG);
```

원리를 알면 IT가 맛있다
IT COOKBOOK

실습 10-6 DetectSystemEvent

- MainActivity 클래스에 MyReceiver2 동적 등록 및 해제 코드 작성

```
13      }
14
15      @Override
16      protected void onStart() {
17          super.onStart();
18          registerReceiver(mReceiver, mFilter);
19      }
20
21      @Override
22      protected void onStop() {
23          super.onStop();
24          unregisterReceiver(mReceiver);
25      }
26  }
```

실습 10-6 DetectSystemEvent

실행 화면

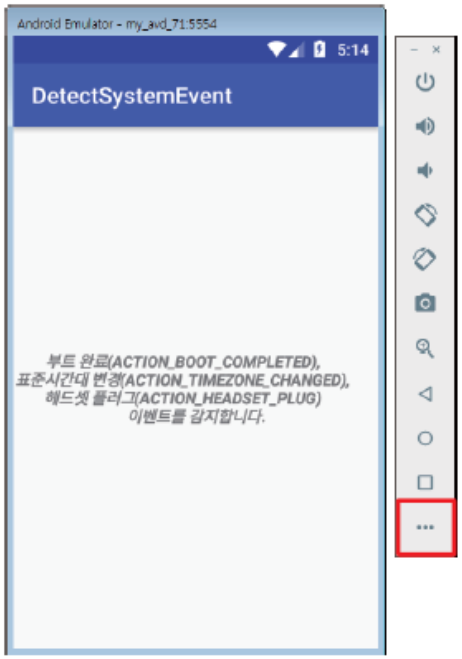
(a) 초기 화면

(b) 부팅 직후

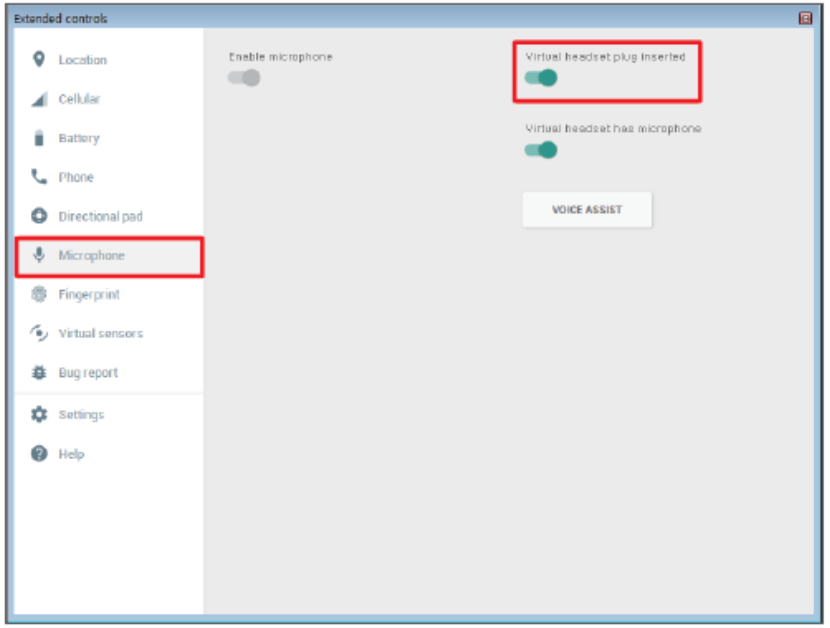
(c) 표준시간대 변경

실습 10-6 DetectSystemEvent

실행 화면



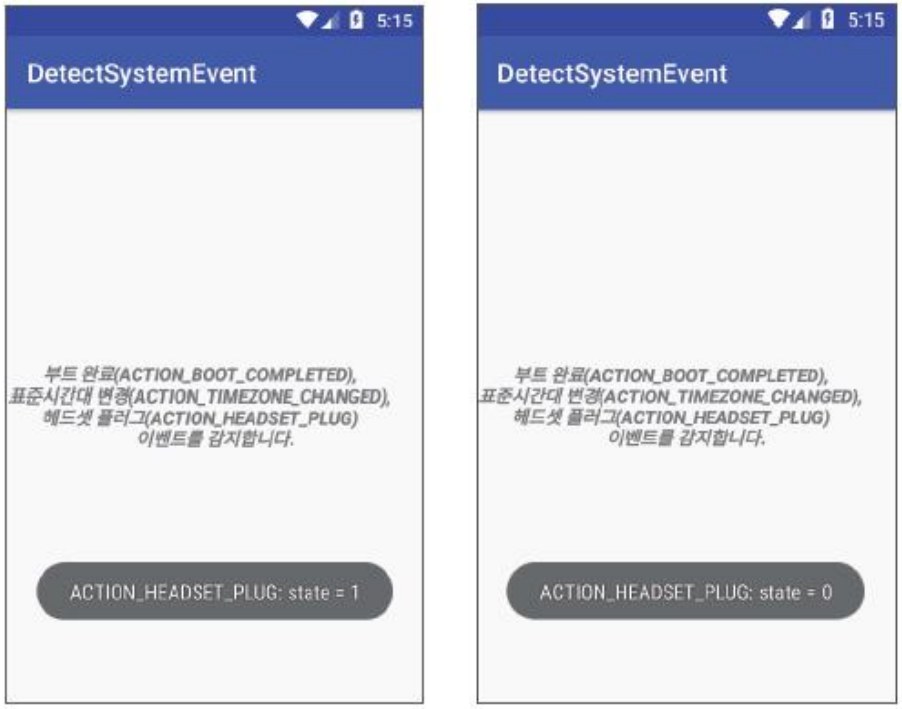
(d) 에뮬레이터 제어 대화상자 열기



(e) 헤드셋 꽂거나 빼기

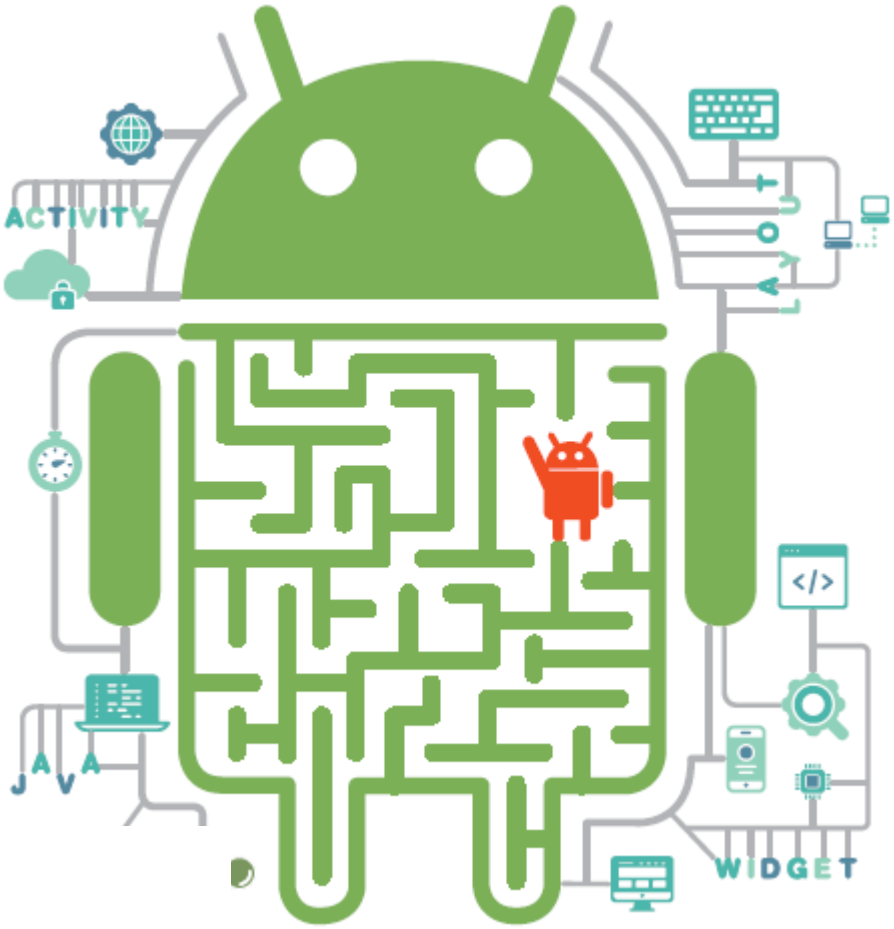
실습 10-6 DetectSystemEvent

실행 화면



(f) 헤드셋을 꽂았을 때 (g) 헤드셋을 뺐을 때

그림 10-20 실행 화면



단계별로 배우는

안드로이드 프로그래밍