

## 4장. Android 응용 위젯

### 1. Spinner 사용하기

#### 예제1. Spinner 사용하기

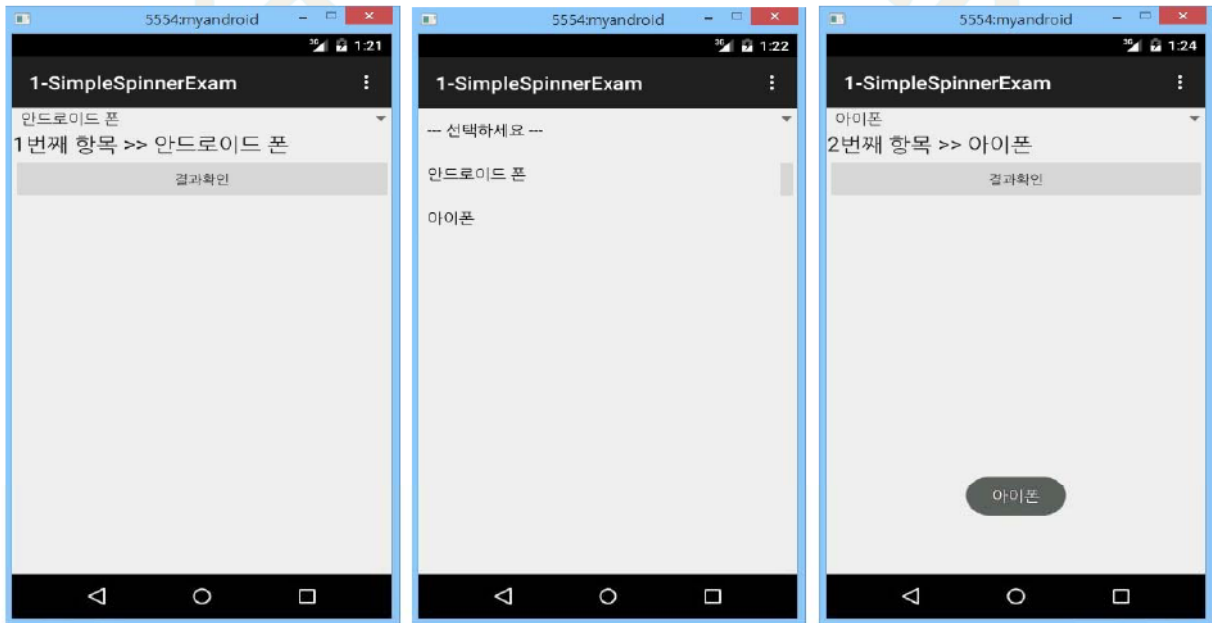
Application Name : 1-SimpleSpinnerExam

Company Domain : .example.com

Package Name : com.example.simplespinnerexam

Project location : D:\Android\workspace\ch04\1-SimpleSpinnerExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- arrays.xml 파일 추가하고, 내용 작성
  - => app/res/values/arrays.xml
  - => /res/values/오\_버/new/XML/Values XML file 선택
  - => Values File Name : arrays
  - => Finish
- Design에서 Spinner를 드래그하여 1개 추가한다.
  - => android:entries="@array/spinner\_data" 추가
- Large Text를 추가한다.
- Button을 드래그하여 1개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.

#### ① 리소스 파일

1) app/res/values/arrays.xml

① app/res/layout/activity\_main.xml

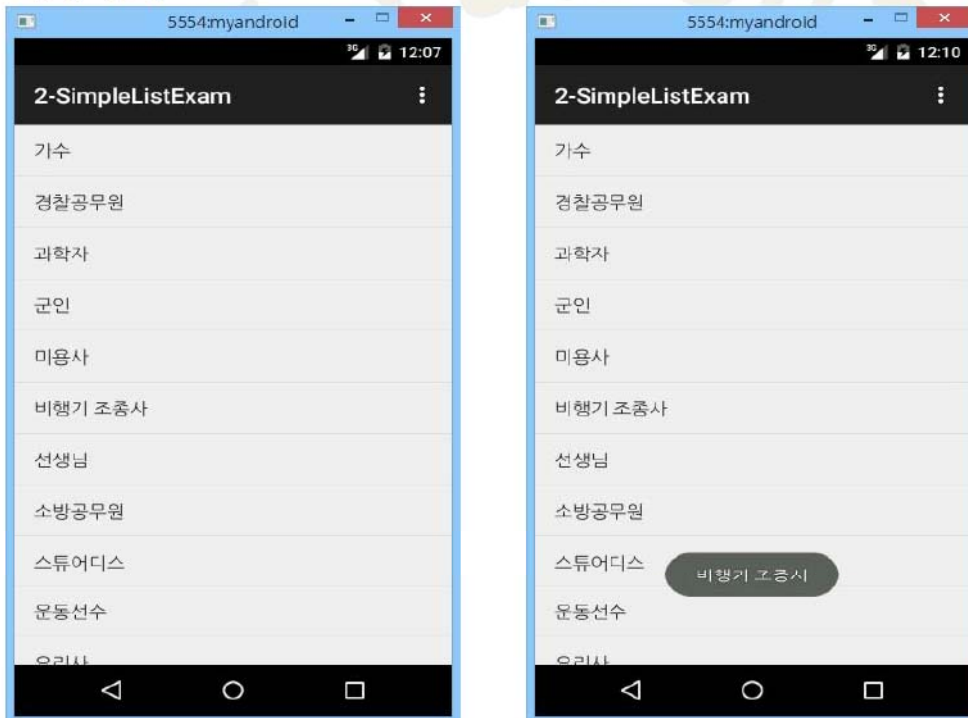
② app/java/com.example.simplespinnerexam/MainActivity.java

## 2. ListView 사용하기

### 예제2. 기본적인 ListView 사용하기

Application Name : 2-SimpleListExam  
Company Domain : .example.com  
Package Name : com.example.simplelistexam  
Project location : D:\Android\workspace\ch04\2-SimpleListExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- arrays.xml 파일 추가하고, 내용 작성
  - => app/res/values/arrays.xml
  - => /res/values/오\_버/new/XML/Values XML file 선택
  - => Values File Name : arrays
  - => Finish
  - => 직업들을 아이템에 추가한다.
- Design에서 ListView를 드래그하여 1개 추가한다.
  - => android:entries="@array/list\_data" 추가
- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.

#### ① 리소스 파일

1) app/res/values/arrays.xml

① app/res/layout/activity\_main.xml

② app/java/com.example.simplelistexam/MainActivity.java

### 3. ListView의 커스터마이징

#### 예제3. ListView의 커스터마이징

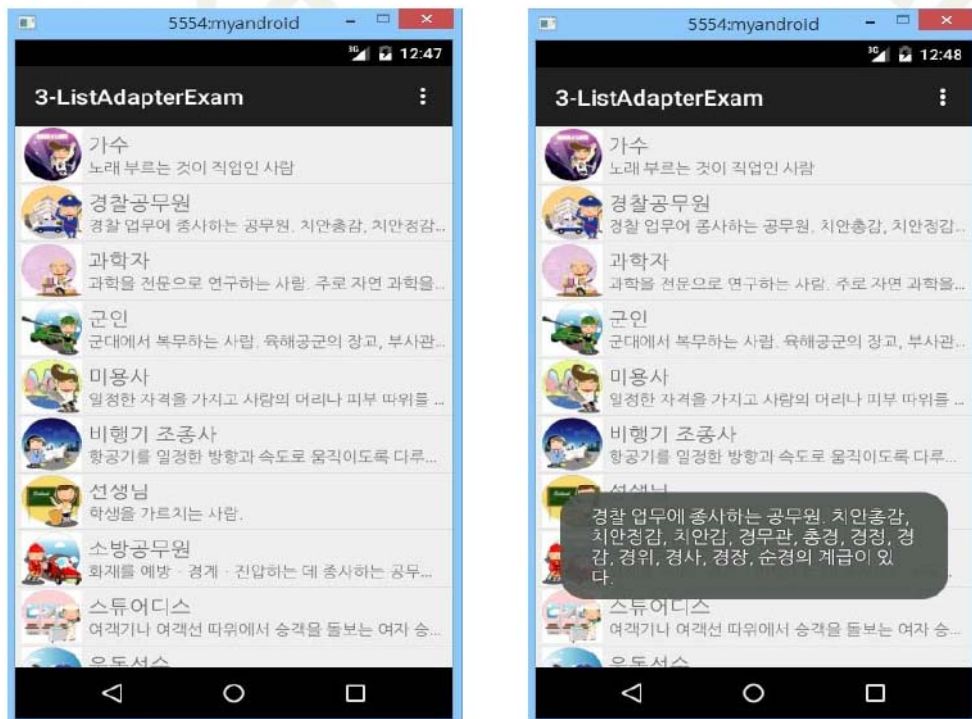
Application Name : 3-ListAdapterExam

Company Domain : .example.com

Package Name : com.example.listadapterexam

Project location : D:\Android\workspace\ch04\3-ListAdapterExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 ListView를 드래그하여 1개 추가한다.
- /res/drawable/ 폴더에 그림을 복사해 놓는다.
  - > job01.png~job14.png
- list\_item.xml 파일을 추가한다.
  - => 아래의 <Layout 작업순서>를 참조.
- Java Beans 만들기
  - => 패키지 만들기 : com.example.listadapterexam/model
  - => Job 클래스 만들기
  - => 멤버 변수 : title, description, image

- ListView에 사용할 Adapter 만들기

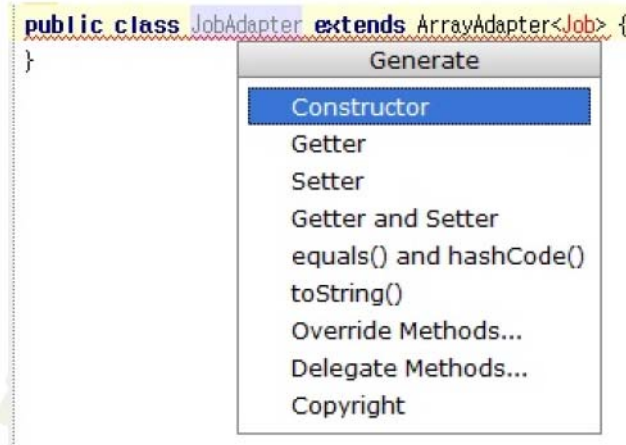
=> 패키지 만들기 : com.example.listadapterexam/adapter

=> JobAdapter 클래스 만들기

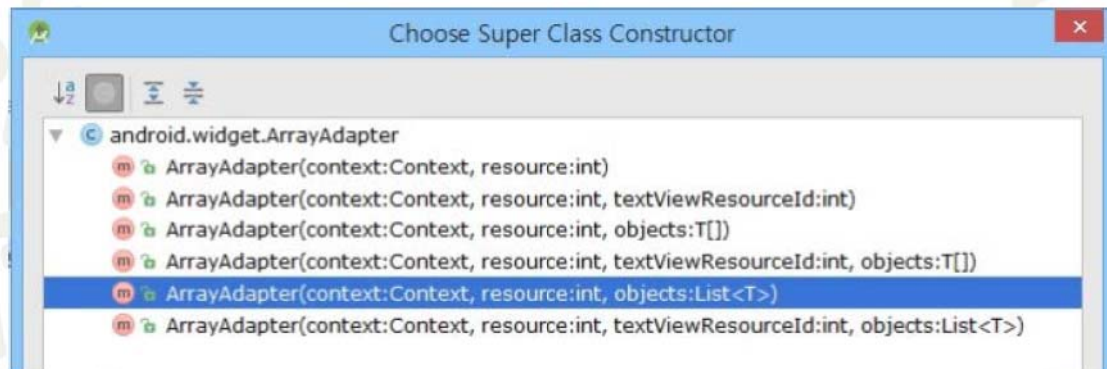
=> 상속받기 : ArrayAdapter<Job>

=> 생성자 추가하기

① 메뉴 -> Code -> Generate 클릭 (단축키 : Alt + Insert) -> Constructor 선택



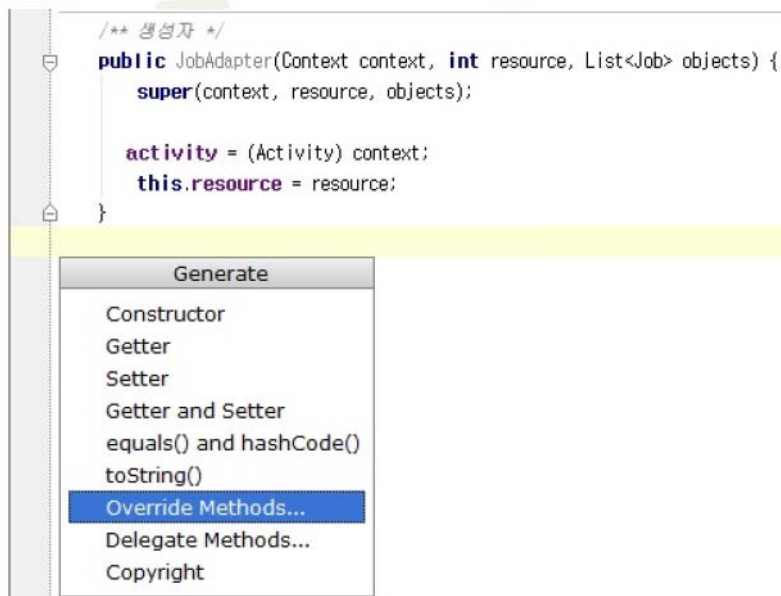
② 아래 그림의 생성자를 선택



=> 멤버 변수 : activity, resource 추가

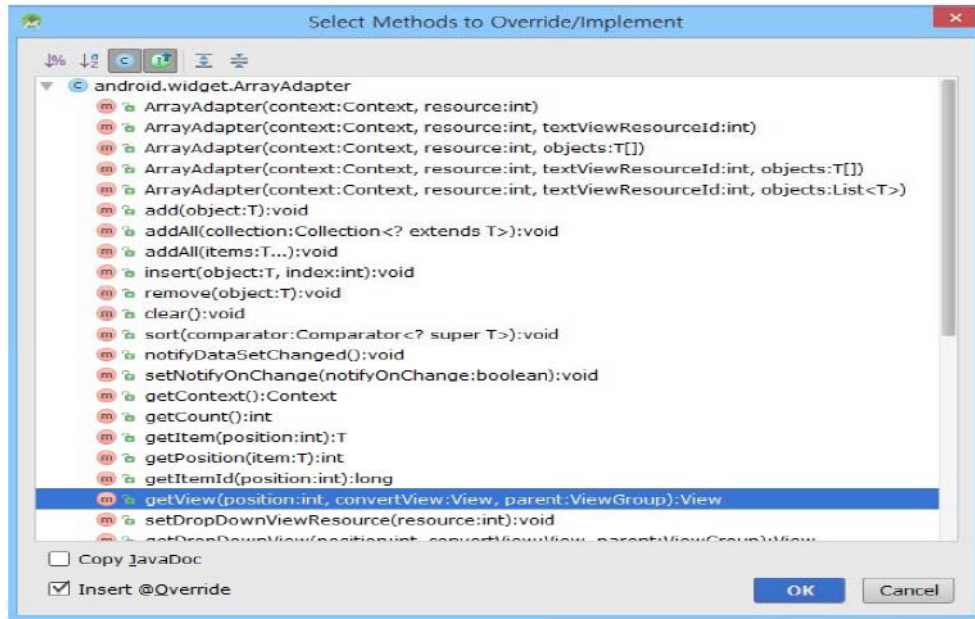
=> getView함수 추가

① 메뉴 -> Code -> Generate 클릭 (단축키 : Alt + Insert) -> Override Methods... 선택





## ② getView 메소드를 선택

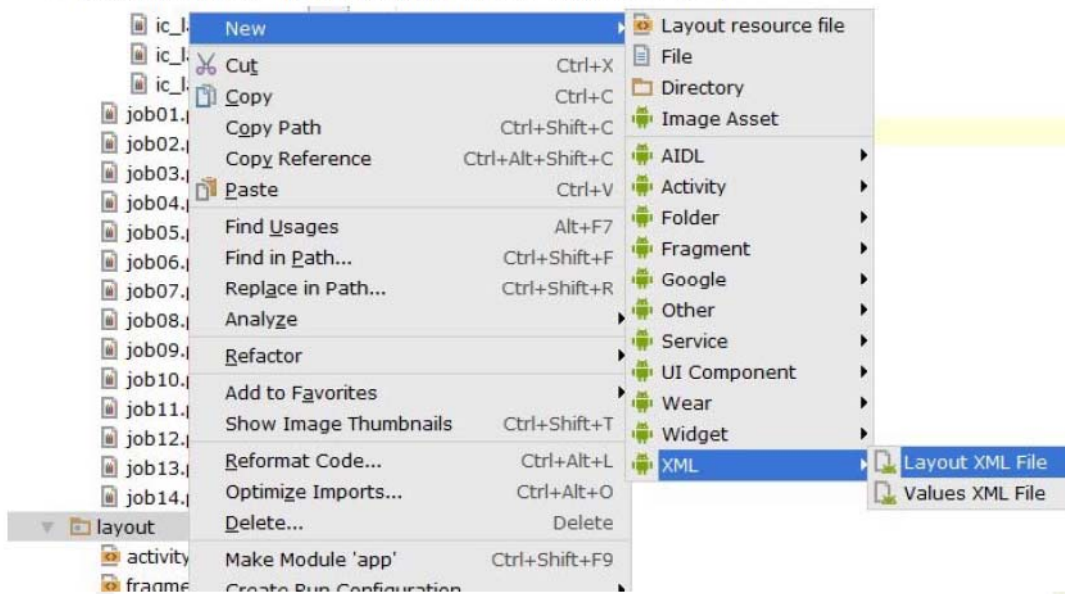


- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트를 처리를 한다.

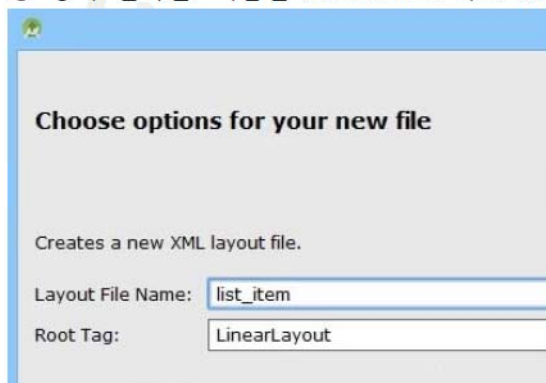
## <Layout 작업 순서>

① /res/layout/ 폴더에 list\_item.xml 파일을 추가한다.

=> res/layout/오\_버/new/XML/Layout XML File 선택



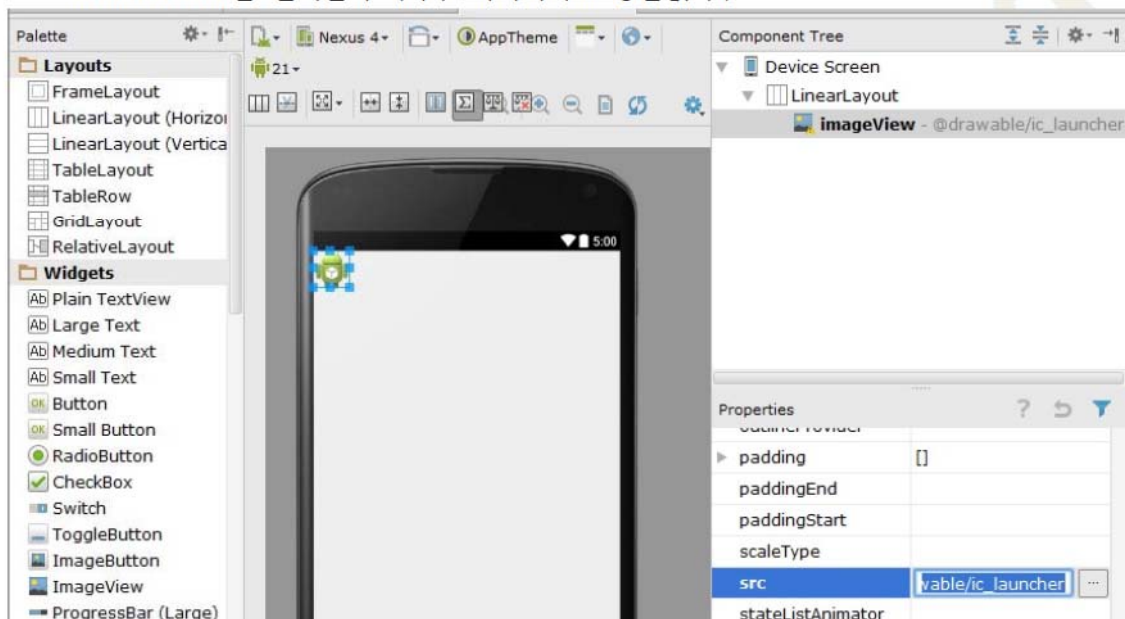
② 창이 열리면, 이름만 list\_item로 하고 finish한다.



③ Design에서 ImageView를 드래그하여 1개 추가한다.

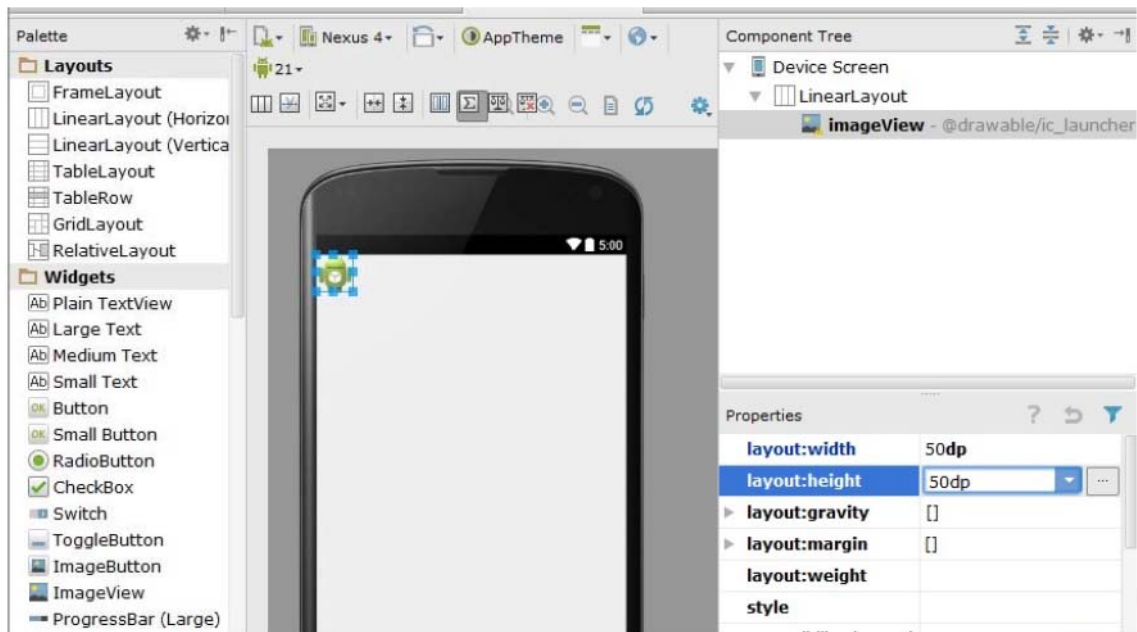
=> ImageView의 src에 이미지를 추가한다.

=> ic\_launcher를 선택한다. (아무 이미지라도 상관없다.)



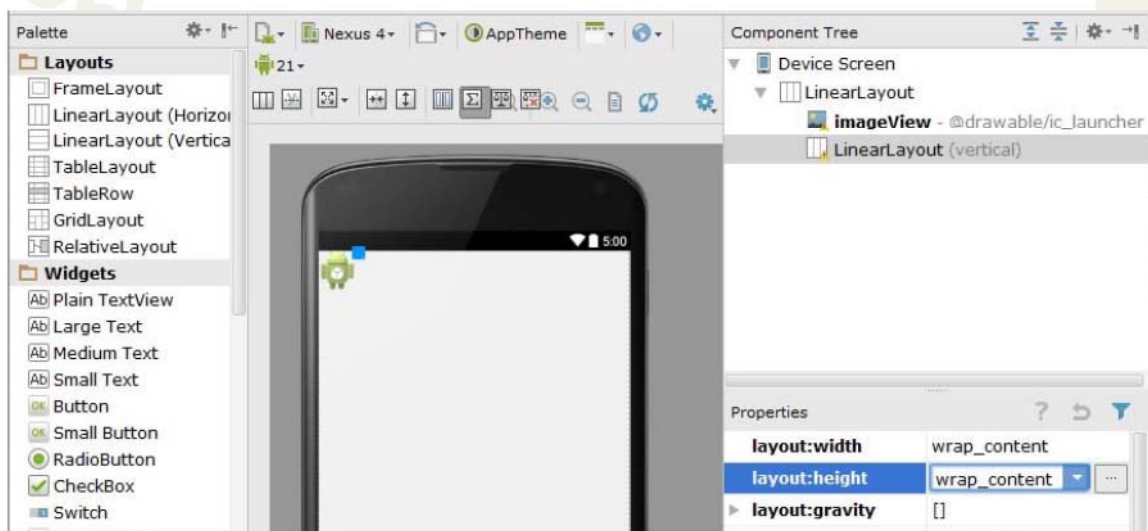
④ ImageView의 크기를 고정시킨다.

=> layout:width : 50dp, layout:height : 50dp 로 수정한다.



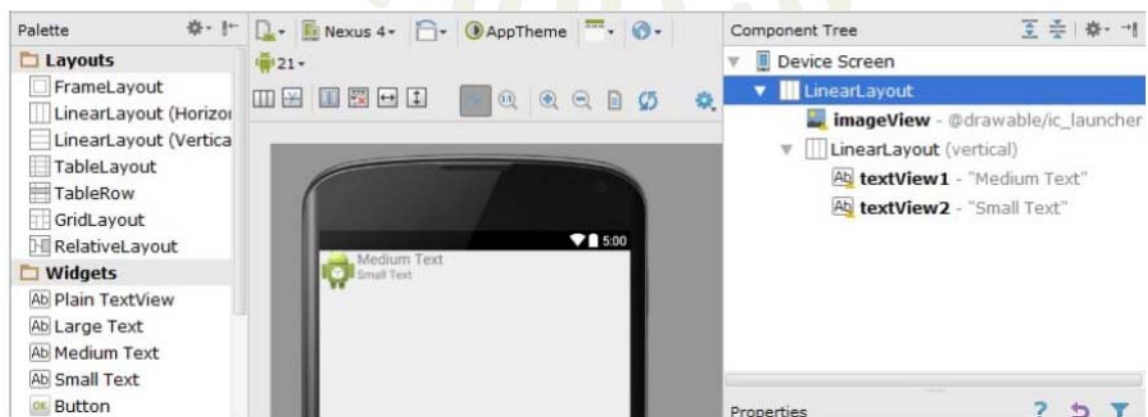
⑤ Design에서 LinearLayout(Vertical)을 드래그하여 1개 추가한다.

=> layout:height : wrap\_content 로 수정한다.



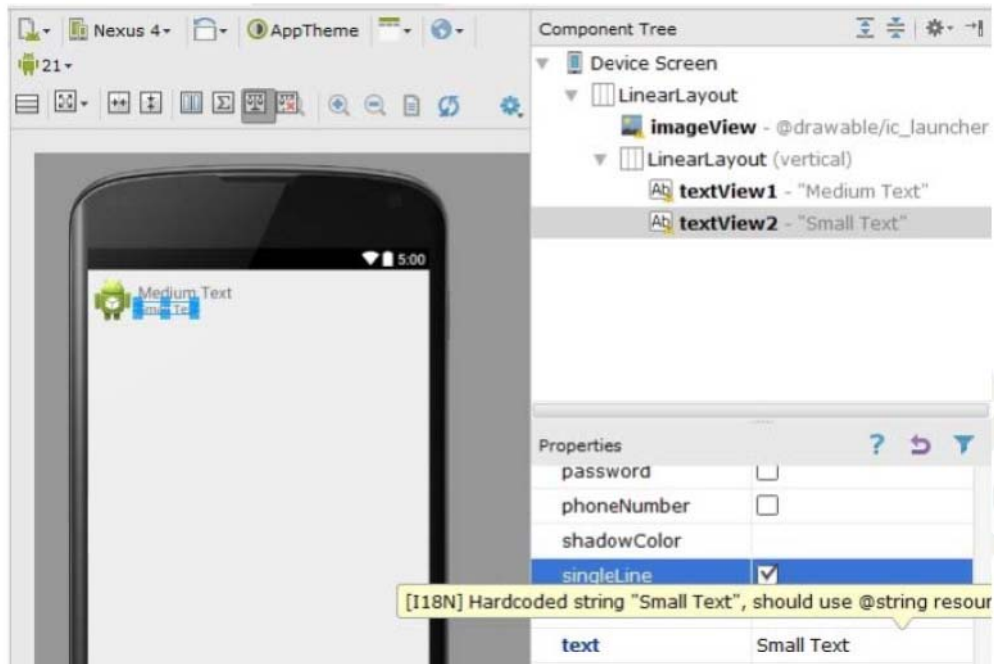
⑥ Design에서 Medium Text와 Small Text를 드래그하여 각각 1개씩 추가한다.

=> SingleLine : true 로 체크한다.



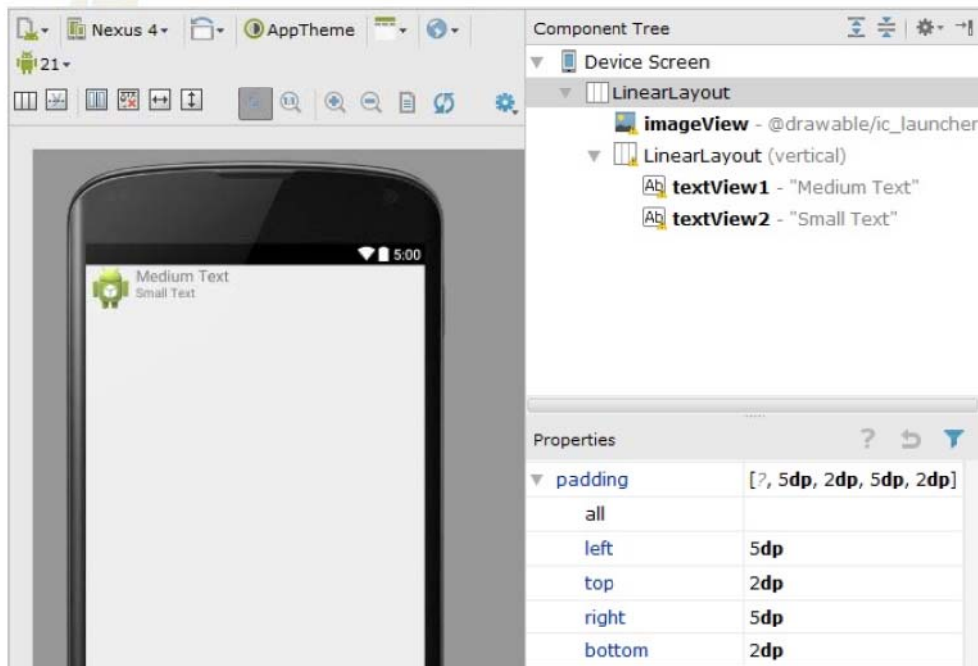
⑦ Design에서 Medium Text와 Small Text 수정

=> SingleLine : true 로 체크한다.



⑧ 제일 상위의 LinearLayout의 padding을 조정한다.

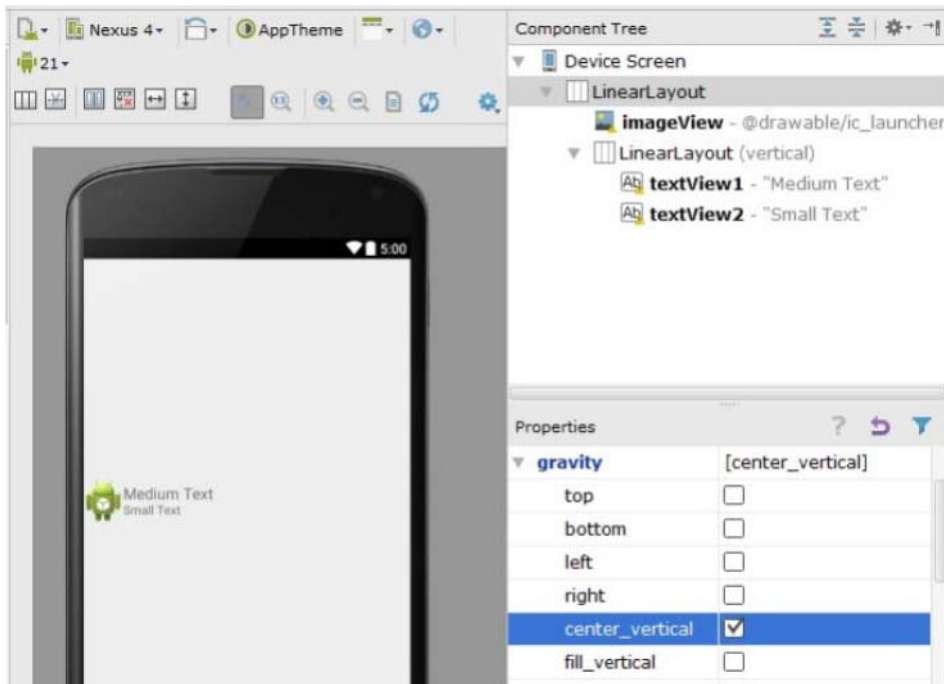
=> left : 5dp, top : 2dp, right : 5dp, bottom : 2dp 로 수정한다.





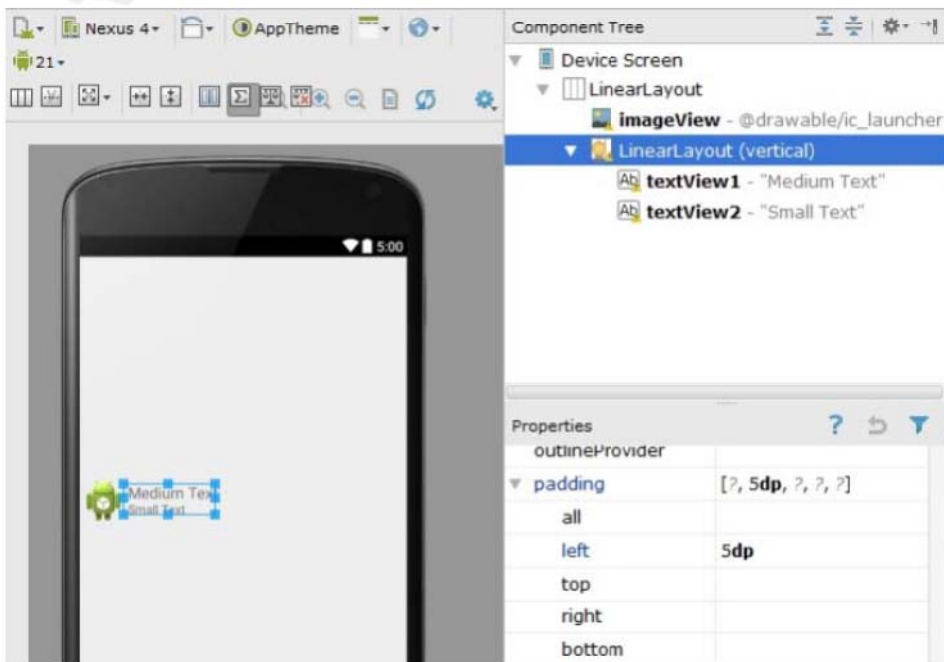
⑨ 제일 상위의 LinearLayout의 gravity를 수정한다.

=> center\_vertical 을 체크한다.



⑩ 두번째 LinearLayout의 padding을 조정한다.

=> left : 5dp 만 수정한다.



⑪ 리소스 파일

1) app/res/drawable/

-> job01.jpg ~ job14.jpg

① app/res/layout/activity\_main.xml

② app/res/layout/list\_item.xml

③ app/java/com.example.listadapterexam/model/Job.java

④ app/java/com.example.listadapterexam/adapter/JobAdapter.java

⑤ app/java/com.example.listadapterexam/MainActivity.java

## 문제1. ListView의 커스텀마이징

Application Name : BookList

Company Domain : .example.com

Package Name : com.example.booklist

Project location : D:\Android\workspace\ch04\BookList

### <실행 결과>



### <작업 순서>

- 예제3과 동일
- list\_item.xml 파일



- Book 클래스 : Java Beans에 포함할 변수

```
public class Book {
    private String subject;
    private String writer;
    private String publisher;
    private int image;
}
```

⑥ 리소스 파일

1) app/res/drawable/

-> book01.jpeg, book02.jpeg, book03.jpeg, book04.jpeg, book05.jpeg, book06.jpeg,  
book07.jpeg, book08.jpeg, book09.jpeg, book10.jpeg

① app/res/layout/activity\_main.xml

② app/res/layout/list\_item.xml

③ app/java/com.example.booklist/model/Book.java

④ app/java/com.example.booklist/adapters/BookAdapter.java

⑤ app/java/com.example.booklist/MainActivity.java

## 4. ListView 내부에 포함된 컴포넌트의 이벤트 처리

### 예제4. ListView 내부에 포함된 컴포넌트의 이벤트 처리

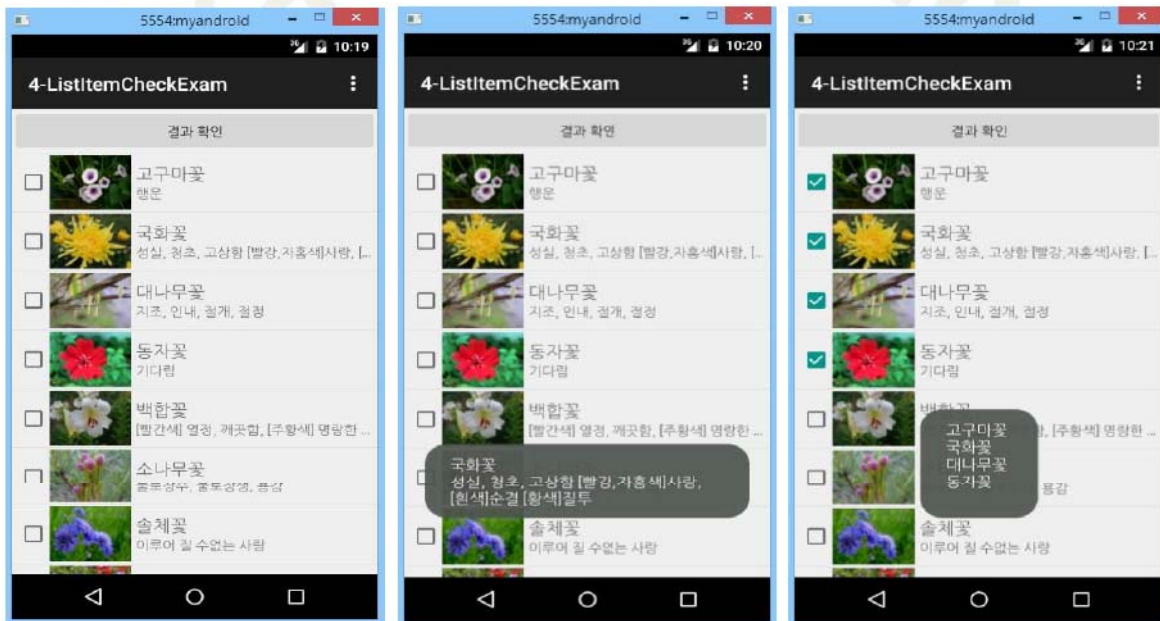
Application Name : 4-ListItemCheckExam

Company Domain : .example.com

Package Name : com.example.listitemcheckexam

Project location : D:\Android\workspace\ch04\4-ListItemCheckExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 ListView를 드래그하여 1개 추가한다.
- /res/drawable/ 폴더에 그림을 복사해 놓는다.  
=> flower01.png~flower13.png
- /res/layout/ 폴더에 list\_item.xml 파일을 추가한다.  
=> 컴포넌트를 추가한다.  
=> CheckBox, ImageView, Medium Text, Small Text  
=> CheckBox에 아래 내용을 추가한다.  
android:focusable="false"  
android:focusableInTouchMode="false"





- Java Beans 만들기
  - => 패키지 만들기 : com.example.listitemcheckexam/**model**
  - => Flower 클래스 만들기
  - => 멤버 변수 : title, description, image, check
- ListView에 사용할 Adapter 만들기
  - => 패키지 만들기 : com.example.listadapterexam/**adapter**
  - => FlowerAdapter 클래스 만들기
  - => 상속받기 : ArrayAdapter<Flower>
  - => 생성자 추가하기
    - ① 메뉴 -> Code -> Generate 클릭 (단축키 : Alt + Insert) -> Constructor 선택
    - ② 생성자를 선택
  - => 멤버 변수 : activity, resource 추가
  - => getView함수 추가
    - ① 메뉴 -> Code -> Generate 클릭 (단축키 : Alt + Insert) -> Override Methods... 선택
    - ② getView 메소드를 선택
- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.

#### ⑥ 리소스 파일

- 1) app/res/drawable/
  - > flower01.png ~ flower13.png
- ① app/res/layout/activity\_main.xml
- ② app/res/layout/list\_item.xml
- ③ app/java/com.example.listitemcheckexam/model/Flower.java
- ④ app/java/com.example.listitemcheckexam/adapter/FlowerAdapter.java
- ⑤ app/java/com.example.listitemcheckexam/MainActivity.java

## 5. ListFooter 사용하기

### 예제5. ListFooter 사용하기

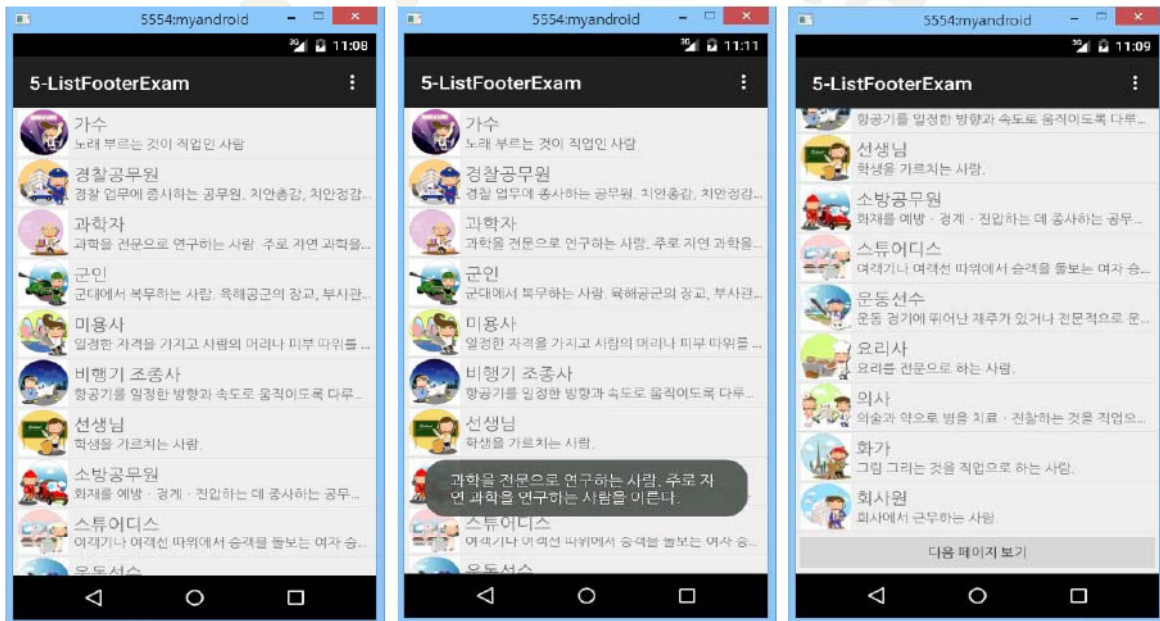
Application Name : 5-ListFooterExam

Company Domain : .example.com

Package Name : com.example.listfooterexam

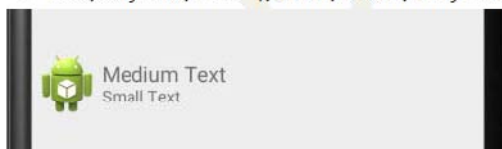
Project location : D:\Android\workspace\ch04\5-ListFooterExam

### <실행 결과>

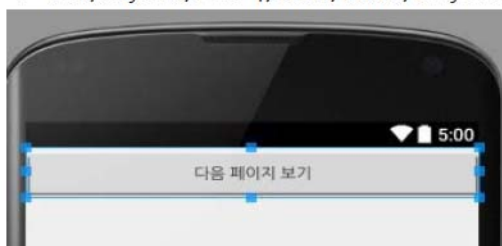


### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 ListView를 드래그하여 1개 추가한다.
- /res/drawable/ 폴더에 그림을 복사해 놓는다.  
=> job01.png~job14.png
- /res/layout/ 폴더에 list\_item.xml 파일을 추가한다.  
=> res/layout/오\_버/new/XML/Layout XML File 선택



- /res/layout/ 폴더에 list\_footer.xml 파일을 추가한다.  
=> res/layout/오\_버/new/XML/Layout XML File 선택



- Java Beans 만들기
  - => 패키지 만들기 : com.example.listfooterexam/**model**
  - => Job 클래스 만들기
  - => 멤버 변수 : title, description, image
- ListView에 사용할 Adapter 만들기
  - => 패키지 만들기 : com.example.listfooterexam/**adapter**
  - => JobAdapter 클래스 만들기
  - => 상속받기 : ArrayAdapter<Job>
  - => 생성자 추가하기
    - ① 메뉴 -> Code -> Generate 클릭 (단축키 : Alt + Insert) -> Constructor 선택
    - ② ArrayAdapter(context:Context, resource:int, objects:List<T>) 생성자를 선택
  - => 멤버 변수 : activity, resource 추가
  - => getView함수 추가
    - ① 메뉴 -> Code -> Generate 클릭 (단축키 : Alt + Insert) -> Override Methods... 선택
    - ② getView 메소드를 선택
- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.

## ⑩ 리소스 파일

- 1) app/res/drawable/
  - > job01.jpg ~ job14.jpg
- ① app/res/layout/activity\_main.xml
- ② app/res/layout/list\_item.xml
- ③ app/res/layout/list\_footer.xml
- ④ app/java/com.example.listfooterexam/model/Job.java
- ⑤ app/java/src/com.example.listfooterexam/adapter/JobAdapter.java
- ⑥ app/java/com.example.listfooterexam/MainActivity.java

## 6. 동적 List 데이터

### 예제6. 동적 List 데이터

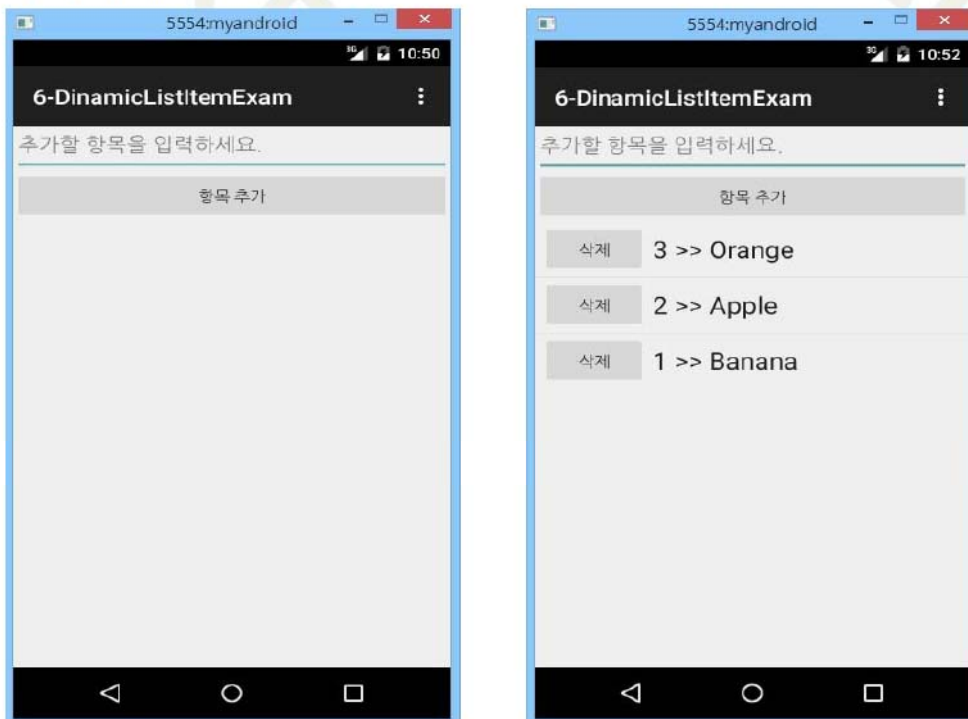
Application Name : 6-DinamicListItemExam

Company Domain : .example.com

Package Name : com.example.dinamiclistitemexam

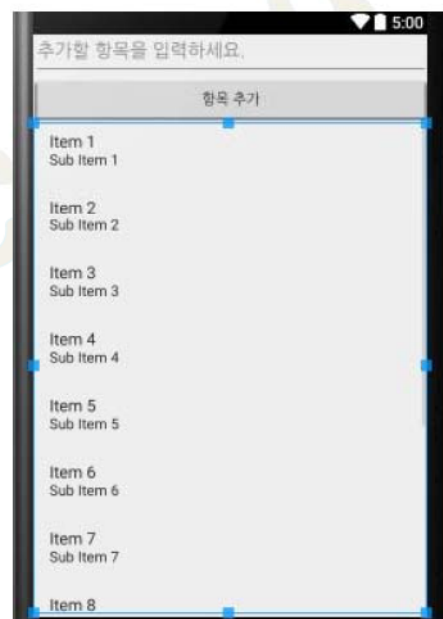
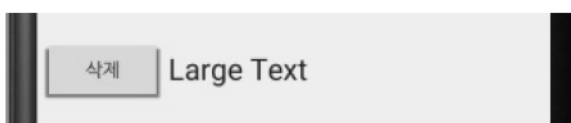
Project location : D:\Android\workspace\ch04\6-DinamicListItemExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 Plain Text를 드래그하여 1개 추가한다.
- Button을 드래그하여 1개 추가한다.
- ListView를 드래그하여 1개 추가한다. =====>  
=> 옆에처럼 설정을 수정한다.
- /res/layout/ 폴더에 list\_item.xml 파일을 추가한다.  
=> Button과 Large Text를 추가한다.  
=> Button에 아래 내용을 추가한다.  
android:focusable="false"  
android:focusableInTouchMode="false"





- Java Beans 만들기
  - => 패키지 만들기 : `com.example.dinamiclistitemexam/model`
  - => Item 클래스 만들기
  - => 멤버 변수 : `num, name`
- ListView에 사용할 Adapter 만들기
  - => 패키지 만들기 : `com.example.dinamiclistitemexam/adapter`
  - => ItemAdapter 클래스 만들기
  - => 상속받기 : `ArrayAdapter<Item>`
  - => 생성자 추가하기
    - ① 메뉴 -> Code -> Generate 클릭 (단축키 : `Alt + Insert`) -> Constructor 선택
    - ② `ArrayAdapter(context:Context, resource:int, objects:List<T>)` 생성자를 선택
  - => 멤버 변수 : `activity, resource` 추가
  - => `getView` 함수 추가
    - ① 메뉴 -> Code -> Generate 클릭 (단축키 : `Alt + Insert`) -> Override Methods... 선택
    - ② `getView` 메소드를 선택
- 실행 결과처럼 설정을 수정한다.
- `MainActivity.java`에서 이벤트 처리를 한다.

- ① `app/res/layout/activity_main.xml`
- ② `app/res/layout/list_item.xml`
- ③ `app/java/com.example.dinamiclistitemexam/model/Item.java`
- ④ `app/java/com.example.dinamiclistitemexam/adapter/ItemAdapter.java`
- ⑤ `app/java/com.example.dinamiclistitemexam/MainActivity.java`

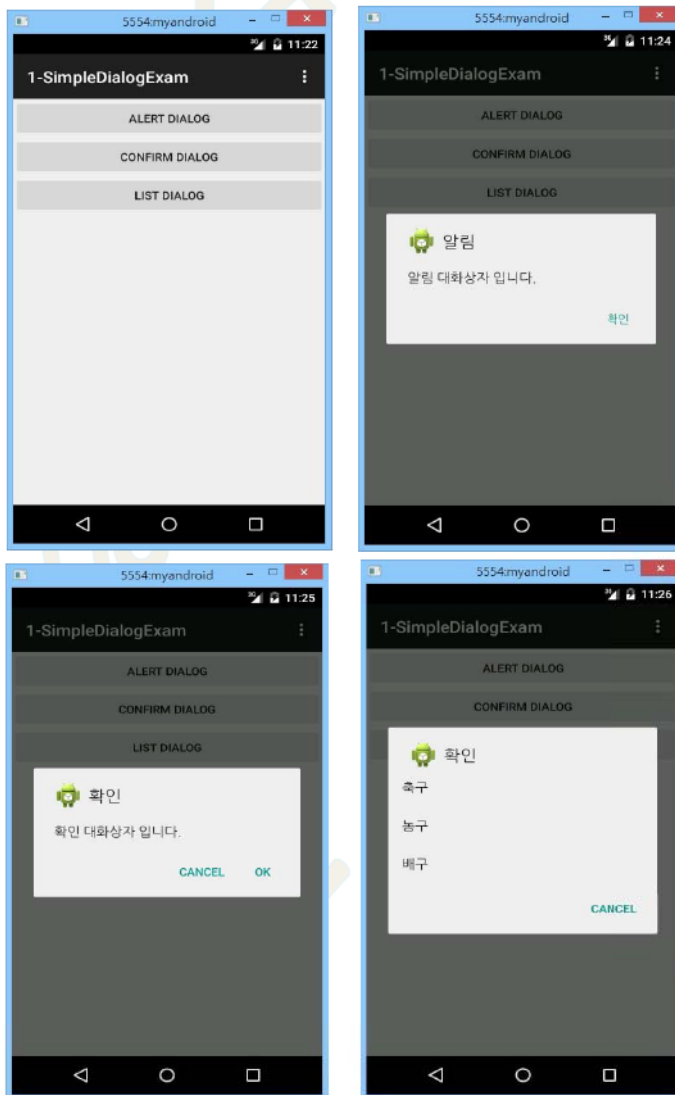
## 5장. 다이얼로그

### 1. 기본 다이얼로그의 사용

#### 예제1. 기본 다이얼로그의 사용

Application Name : 1-SimpleDialogExam  
Company Domain : .example.com  
Package Name : com.example.simpdialogexam  
Project location : D:\Android\workspace\ch05\1-SimpleDialogExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 Button을 3개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.

① app/res/layout/activity\_main.xml

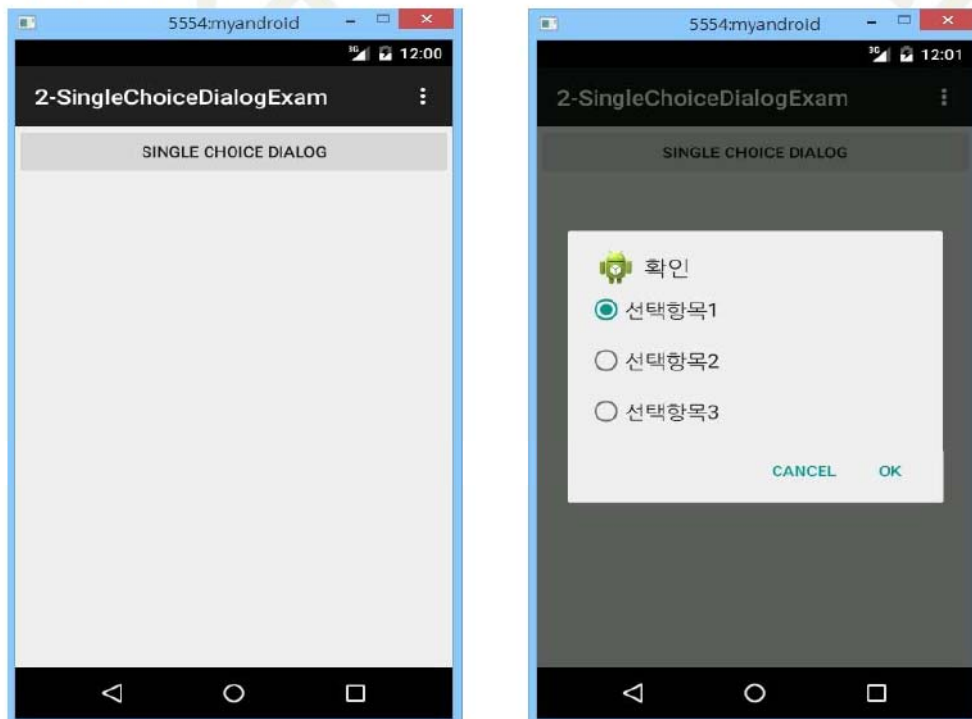
② app/java/com.example.simpdialogexam/MainActivity.java

## 2. SingleChoice 다이얼로그

### 예제2. SingleChoice 다이얼로그

Application Name : 2-SingleChoiceDialogExam  
Company Domain : .example.com  
Package Name : com.example.singlechoicedialogexam  
Project location : D:\Android\workspace\ch05\2-SingleChoiceDialogExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 Button을 1개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.

① app/res/layout/activity\_main.xml

② app/java/com.example.singlechoicedialogexam/MainActivity.java

### 3. MultiChoice 다이얼로그

#### 예제3. MultiChoice 다이얼로그

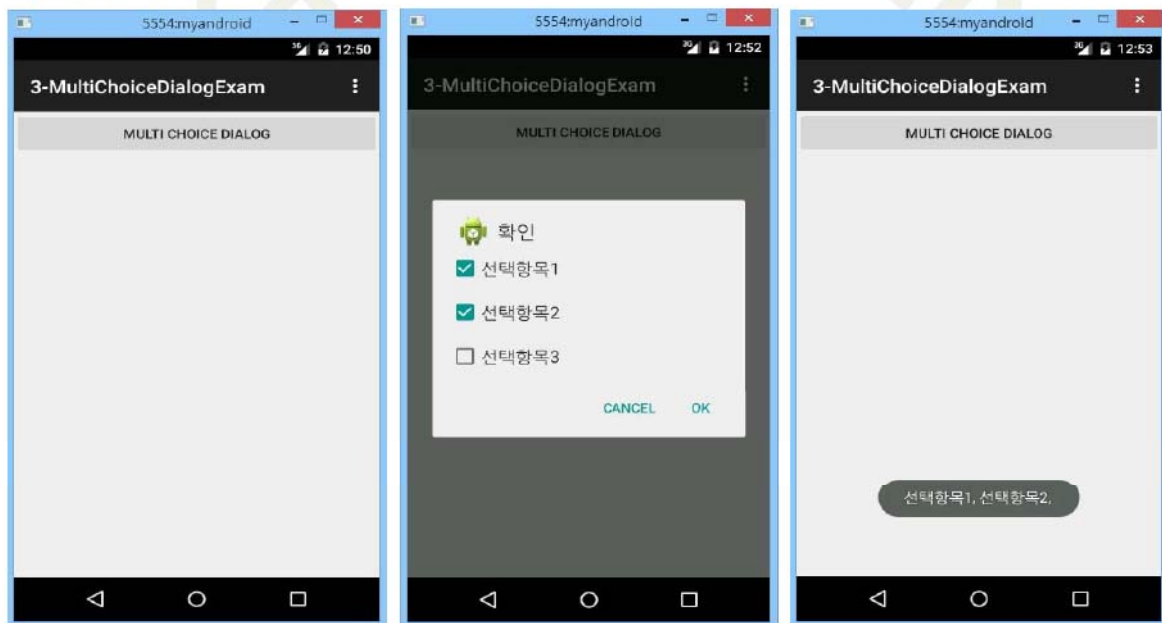
Application Name : 3-MultiChoiceDialogExam

Company Domain : .example.com

Package Name : com.example.multichoicedialogexam

Project location : D:\Android\workspace\ch05\3-MultiChoiceDialogExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 Button을 1개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.

① app/res/layout/activity\_main.xml

② app/java/com.example.multichoicedialogexam/MainActivity.java



## 4. DatePicker 다이얼로그

### 예제4. DatePicker 다이얼로그

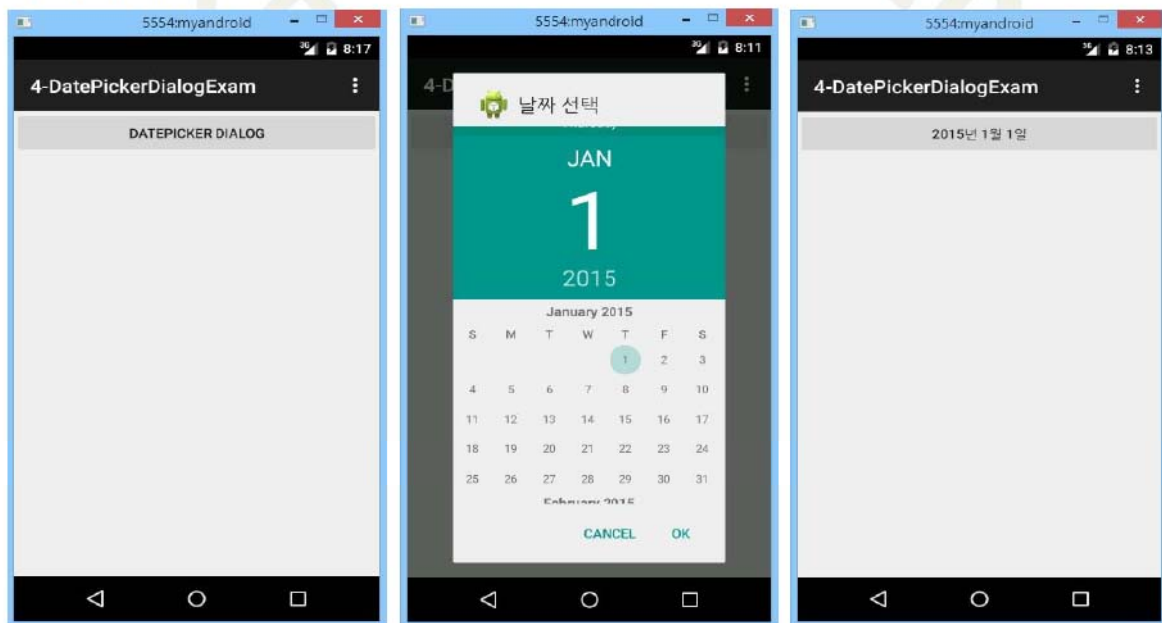
Application Name : 4-DatePickerDialogExam

Company Domain : .example.com

Package Name : com.example.datepickerdialogexam

Project location : D:\Android\workspace\ch05\4-DatePickerDialogExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 Button을 1개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- DateTimeHelper 클래스 추가하기
  - => 패키지 만들기 : app/java/com.example.datepickerdialogexam/helper
  - => DateTimeHelper 클래스 만들기
  - => 싱글톤으로 만들기
- MainActivity.java에서 이벤트 처리를 한다.

① app/res/layout/activity\_main.xml

② app/java/com.example.datepickerdialogexam/helper/DateTimeHelper.java

③ app/java/com.example.datepickerdialogexam/MainActivity.java

## 5. TimePicker 다이얼로그

### 예제5. TimePicker 다이얼로그

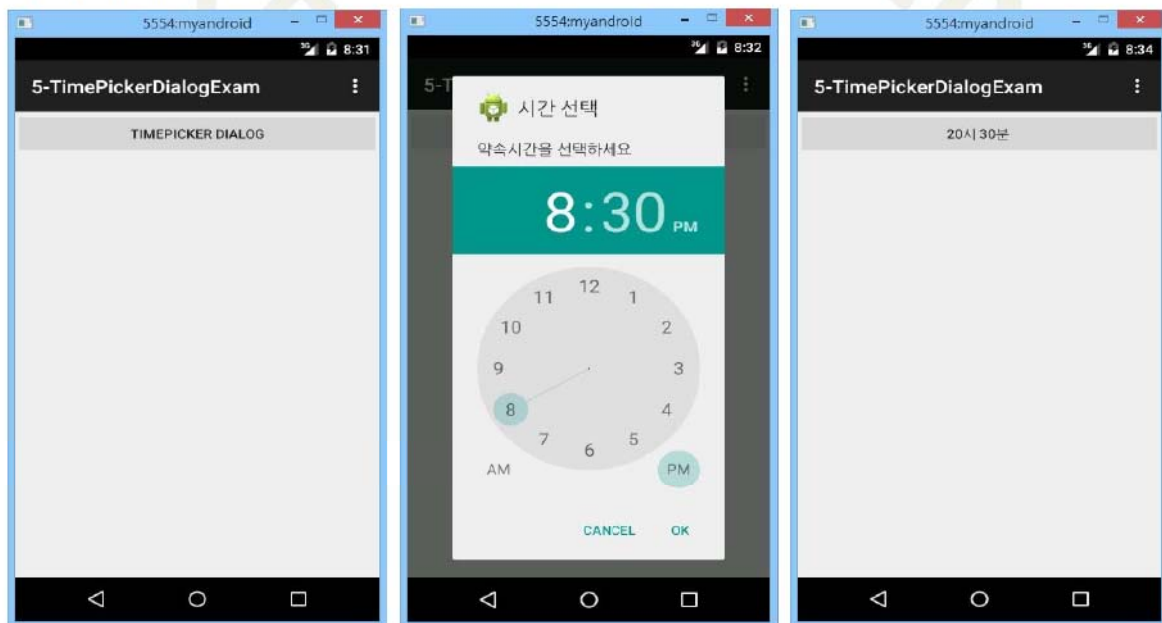
Application Name : 5-TimePickerDialogExam

Company Domain : .example.com

Package Name : com.example.timepickerdialogexam

Project location : D:\Android\workspace\ch05\5-TimePickerDialogExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 Button을 1개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- DateTimeHelper 클래스 추가하기
  - => 패키지 만들기 : app/java/com.example.datepickerdialogexam/helper
  - => DateTimeHelper 클래스 만들기
  - => 싱글톤으로 만들기
- MainActivity.java에서 이벤트 처리를 한다.

① app/res/layout/activity\_main.xml

② app/java/com.example.timepickerdialogexam/helper/DateTimeHelper.java

③ app/java/com.example.timepickerdialogexam/MainActivity.java

## 6. Progress 다이얼로그

### 예제6. Progress 다이얼로그

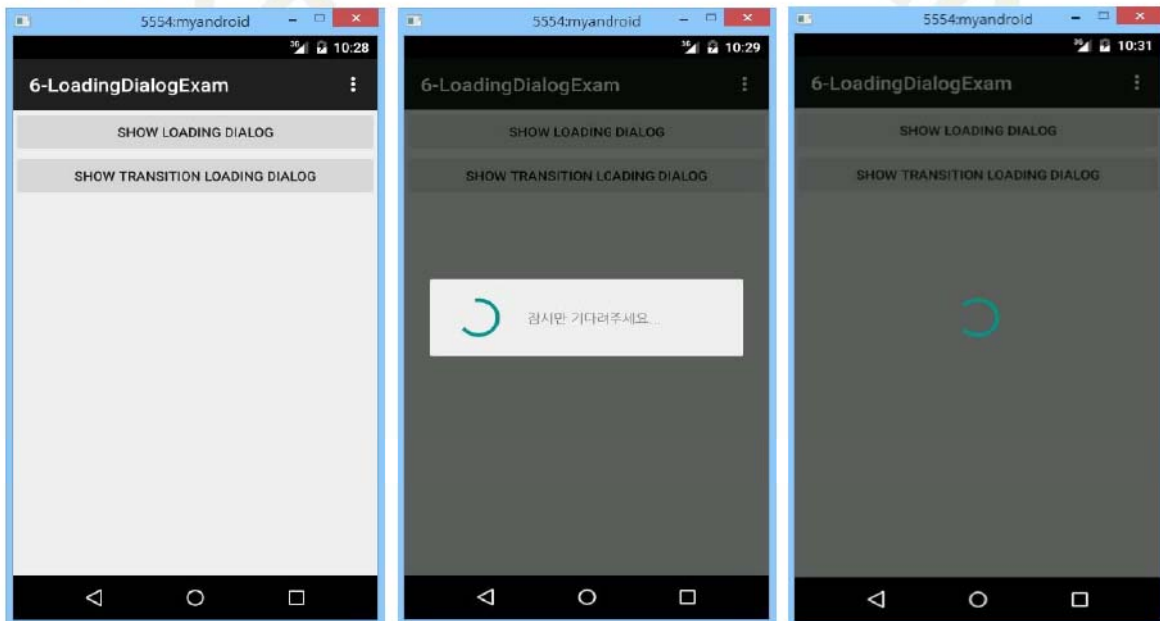
Application Name : 6-LoadingDialogExam

Company Domain : .example.com

Package Name : com.example.loadingdialogexam

Project location : D:\Android\workspace\ch05\6-LoadingDialogExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Design에서 Button을 2개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- 원으로 된 프로그레스바 만들기
  - => app/res/values/폴더에 loading\_dialog.xml 추가한다.
  - => /res/values/오\_버/new/XML/Values XML file 선택
  - => Values File Name : loading\_dialog
  - => Finish
  - => 아이템을 추가한다.
- MainActivity.java에서 이벤트 처리를 한다.

① app/res/layout/activity\_main.xml

② app/res/values/loading\_dialog.xml

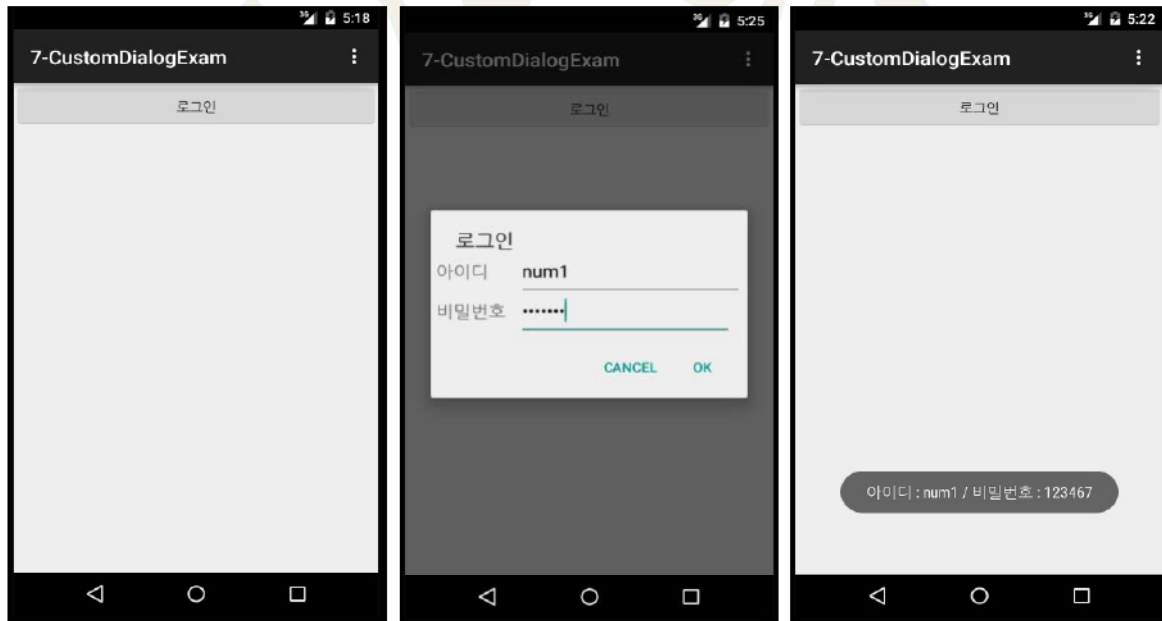
③ app/java/com.example.loadingdialogexam/MainActivity.java

## 7. 커스텀 다이얼로그

### 예제7. 커스텀 다이얼로그

Application Name : 7-CustomDialogExam  
Company Domain : .example.com  
Package Name : com.example.customdialogexam  
Project location : D:\Android\workspace\ch05\7-CustomDialogExam

#### <실행 결과>



#### <작업 순서>

- activity\_main.xml 파일 기본 설정
- Button을 1개 추가한다.
- 실행 결과처럼 설정을 수정한다.
- 커스텀 다이얼로그 만들기
  - => app/res/layout/ 폴더에 dialog.xml를 추가한다.
  - => /res/layout/오\_버/new/XML/Layout XML file 선택
  - => Values File Name : dialog
  - => Finish
  - => LinearLayout 2개, Medium Text 2개, EditText 2개를 추가한다.



- MainActivity.java에서 이벤트 처리를 한다.

- ① app/res/layout/activity\_main.xml
- ② app/res/layout/dialog.xml
- ③ app/java/com.example.customdialogexam/MainActivity.java