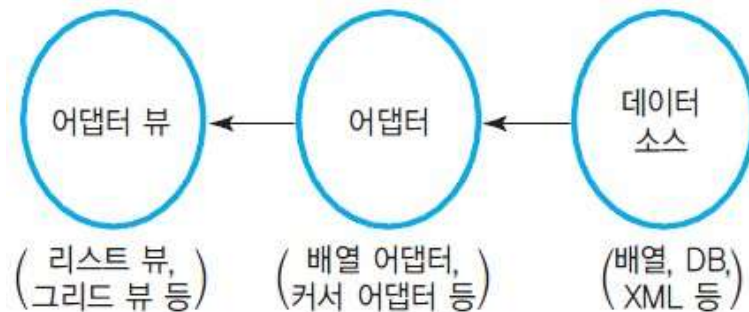


어댑터와 어댑터뷰

어댑터

■ 의미

- 두 개의 상이한 부분을 연결시키는 데 사용하는 장치
- 외부 데이터 소스와 어댑터뷰 사이의 연결 수단



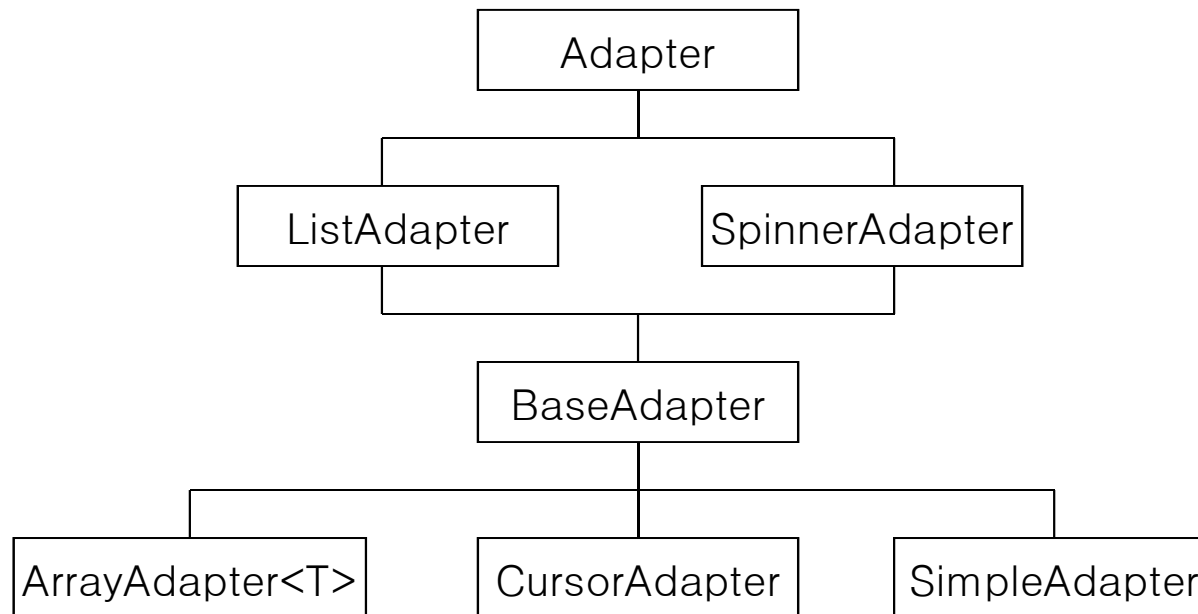
- 데이터 집단을 편리하게 화면에 표시할 수 있는 인터페이스

■ 역할

- 데이터 소스를 접근하여 데이터 항목을 읽어오고,
- 각 데이터 항목을 위한 뷰를 생성한다.

어댑터

■ 어댑터 계층구조



어댑터

- 데이터 소스를 접근하여 데이터 항목을 읽어오는 역할
 - `public abstract int getCount ()`
 - `public abstract Object getItem (int position)`
 - `public abstract long getItemId (int position)`
- 뷰를 생성하기 위한 메소드
 - `public abstract View getView (int position, View convertView, ViewGroup parent)`

어댑터

■ 배열 어댑터

- ArrayAdapter
- 배열 어댑터는 문자열 집합과 같은 배열로 구성된 데이터 소스를 어댑터뷰와 연관시키기 위한 어댑터
- 객체 생성
 - 생성자
`public ArrayAdapter (Context context, int textViewResourceId, List<T> objects 혹은 T[] objects)`
 - 데이터 소스가 외부 리소스로 구성된 배열이라면
`public static ArrayAdapter<CharSequence> createFromResource (Context context, int textArrayResId, int textViewResId)`

어댑터

■ 배열 어댑터

- 대표적인 레이아웃 리소스 ID:
platforms\android-<버전>\data\res\layout 폴더에 있음

리소스 ID	의미
simple_list_item_1	하나의 텍스트 뷰로 구성된 레이아웃
simple_list_item_checked	체크가 표시되는 레이아웃
simple_list_item_single_choice	라디오 버튼이 표시되는 레이아웃
simple_list_item_multiple_choice	체크 버튼이 표시되는 레이아웃

- 데이터 소스의 변화가 발생하면 다음과 같은 메소드를 사용하여 어댑터뷰에게 알리고 재생 필요
public void notifyDataSetChanged ()

어댑터

■ 배열 어댑터

- simple_list_item_1 : 속성이 좀 많을 뿐 단지 TextView

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:gravity="center_vertical"
    android:paddingLeft="?android:attr/listPreferredItemPaddingLeft"
    android:paddingRight="?android:attr/listPreferredItemPaddingRight"
    android:minHeight="?android:attr/listPreferredItemHeightSmall"
/>
```

- 커스텀 레이아웃을 작성하면 항목을 임의의 모양으로 출력 가능

어댑터

■ 단순 어댑터

- SimpleAdapter
- 정적인 데이터를 XML 파일에서 정의한 뷰로 사상해주는 어댑터
- 키와 값의 쌍으로 구성된 Map 객체를 포함하는 ArrayList 타입의 데이터 소스인 경우 적합
- 객체 생성: 생성자 이용

```
public SimpleAdapter (Context context, List<? extends  
    Map<String, ?>> data, int resource, String[] from, int[] to)
```


어댑터뷰

■ 의미

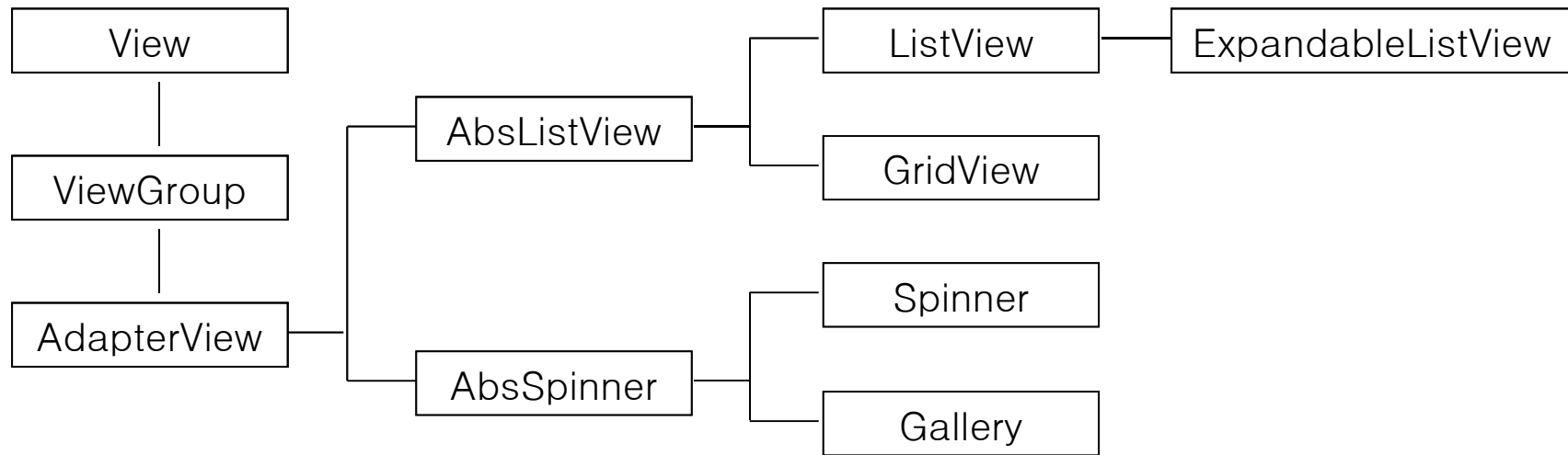
- ViewGroup의 서브클래스로써 다른 뷰를 담는 컨테이너 역할을 수행하는 것은 당연
- ViewGroup의 자손인 레이아웃과는 달리 일반적인 위젯처럼 사용자와 상호작용 가능

■ 역할

- 어댑터로부터 공급받은 뷰를 레이아웃에 채우고
- 또한 사용자에게 의한 항목 선택 이벤트를 처리한다.

어댑터뷰

■ 어댑터뷰 계층구조



어댑터뷰

■ 어댑터와 바인딩

- 어댑터는 코드에서 제공하는 리스트나 디바이스의 데이터베이스에 대한 쿼리 결과와 같은 외부 리소스로부터 데이터를 가져온다
- `public abstract void setAdapter (T adapter)`

■ 사용자의 데이터 항목 선택에 대한 이벤트를 처리하기 위한 핸들러 등록

- `public void setOnItemClickListener (AdapterView.OnItemClickListener listener)`
- `public void setOnItemLongClickListener (AdapterView.OnItemLongClickListener listener)`

어댑터뷰

■ 리스트뷰

- 일반적으로 가장 많이 사용하는 어댑터뷰
- 리스트뷰는 배열 어댑터와 매칭 양호

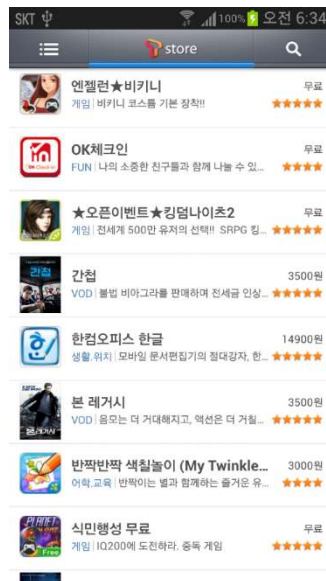
■ 기타 어댑터뷰

- 리스트뷰 외에도 스피너, 갤러리, 그리드뷰 등과 같은 어댑터뷰가 존재
- 제한된 항목에 대한 선택 혹은 사진 이미지의 관리 등과 같은 경우 다른 종류의 어댑터뷰를 사용
- 다른 어댑터뷰는 다른 종류의 어댑터 - SimpleAdapter 혹은 BaseAdapter - 를 사용하는 경우가 다양

리스트뷰

■ 의미

- 어댑터뷰 중에서 가장 자주 사용되는 위젯
- 화면에 표시할 내용들이 리스트 형태로 되어 있는 경우 수직으로 펼쳐서 표시하기 위하여 사용
- 데이터 항목은 ListAdapter에서 공급
- 예: 주소록, 설정창, 통합 예제, 뉴스, 쇼핑 목록 등



리스트뷰

■ 속성

- choiceMode: 리스트뷰의 선택 특성을 의미. 속성값으로 none, singleChoice, 그리고 multipleChoice를 지정.
- divider: 리스트 항목 사이를 구분하기 위하여 그려질 Drawable 객체나 색상을 의미.
- dividerHeight: 구분선의 높이를 의미

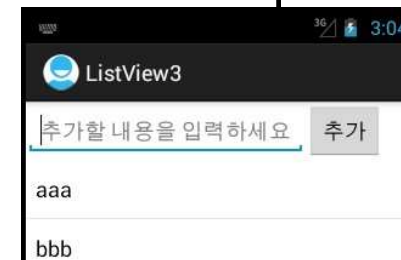
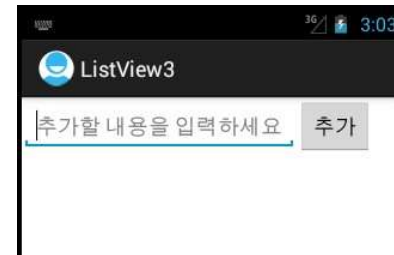
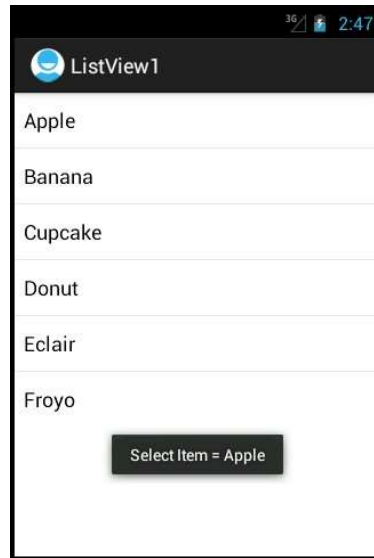
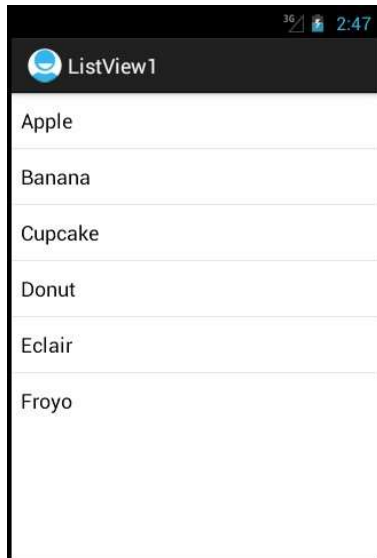
■ 메소드

- public void setChoiceMode (int choiceMode)
- public void setDivider (Drawable divider)
- public void setDividerHeight (int height)
- public void clearChoices ()
- public int getCheckedItemPosition ()

리스트뷰

■ 실습 1 & 2

- ListView1Demo: 코드에서 배열 선언([activity](#), [layout](#))
- ListView2Demo: XML 파일로 배열 선언([activity](#), [layout](#), [arrays](#))
- ListView3Demo: 데이터 추가 기능([activity](#), [layout](#))



리스트뷰

■ 참고

- InputMethodManager와 Edit 객체 생성

```
EditText edit = findViewById(R.id.edit);  
InputMethodManager im =  
    (InputMethodManager) getSystemService(INPUT_METHOD_SERVICE);
```

- 키보드 올리기

```
im.showSoftInput(edit, 0)
```

- 키보드 내리기

```
im.hideSoftInputFromWindow(edit.getWindowToken(), 0)
```


리스트 액티비티

■ 의미

- 리스트뷰 + 액티비티
- 배열이나 커서와 같은 데이터 소스를 바인딩함으로써 데이터 항목의 리스트를 표시하고 하나의 항목을 선택하면 이벤트 핸들러를 호출하는 액티비티
- ListView를 포함하며 관련된 이벤트 리스너가 이미 등록되어 있음
- 리스트 뷰로 액티비티를 가득 채운다면 ListActivity를 사용하는 것이 편리

■ 주요 메소드

- `public ListView getListView ()`
- `public long getSelectedItemId ()`
- `public int getSelectedItemPosition ()`
- `public void setListAdapter (ListAdapter adapter)`
- `protected void onItemClick (ListView l, View v, int position, long id)`

리스트 액티비티

■ 실습 1

- ListActivity1Demo ([activity](#))
- ListView1Demo와 동일한 결과

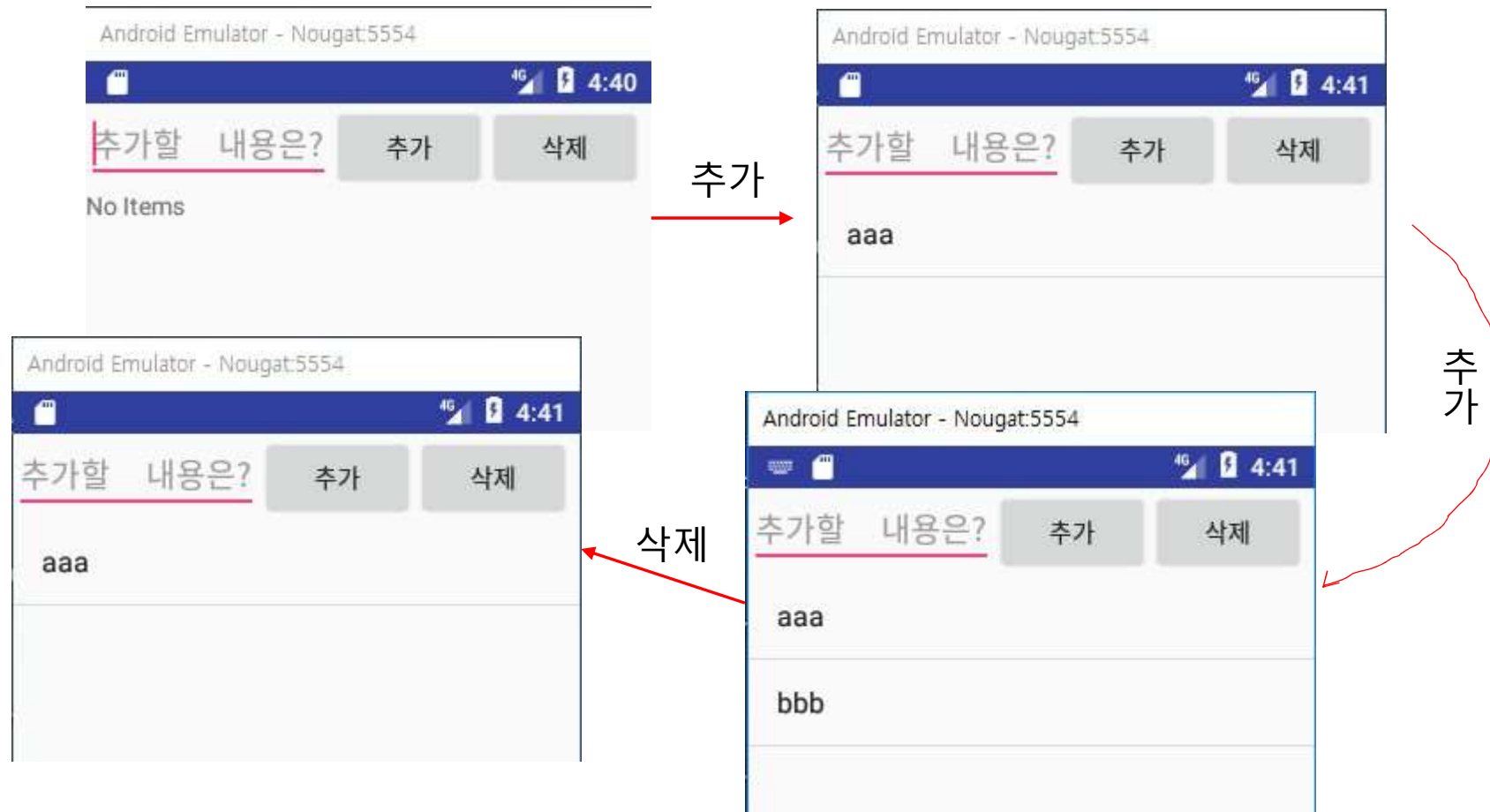
■ 실습 2

- ListActivity2Demo ([activity](#), [layout](#))
- ListView3Demo와 유사한 결과

리스트 액티비티

■ 실습 3

- ListActivity3Demo ([activity](#), [layout](#))
- 데이터 추가 및 삭제 기능 추가: 삭제는 마지막 항목



기타 어댑터

■ 단순 어댑터

- 키와 값을 가진 데이터 집합을 리스트뷰로 표시
- 하나의 행이 두 개 이상의 열을 가진 데이터 소스라면 ArrayAdapter보다 SimpleAdapter가 편리
- 예제: SimpleAdapterDemo([activity](#), [layout](#), [row](#))

기타 어댑터

■ 베이스 어댑터

- 원본 데이터로부터 항목 뷰를 생성하는 핵심 메서드
 - View getView (int position, View convertView, ViewGroup parent)
 - position : 항목의 순서값
 - convertView : 이미 생성된 항목 뷰. null이면 새로 생성한다.
 - parent : 항목 뷰의 부모
- 항목 뷰를 생성하고 레이아웃에 대응되는 정보를 출력
- getView는 사용이 끝난 뷰를 convertView로 전달받는다.

기타 어댑터

■ 베이스 어댑터

● 예제

- [activity](#)
- [layout](#)
- 항목 뷰 ([MyItem](#), [icontext](#))
- 커스텀 어댑터([MyListAdapter](#))

