

11 어댑터뷰



학습목표

- ❖ 리스트뷰와 그리드뷰 활용법을 이해한다.
- ❖ 갤러리와 스피너 사용법을 익힌다.

차례

1. 리스트뷰와 그리드뷰
2. 갤러리와 스피너

01 어댑터뷰

- 어댑터뷰
 - ListView, GridView, ExpandableListView, Spinner, Gallery 등을 묶어서 칭함

```
java.lang.Object
└─ android.view.View
    └─ android.widget.ViewGroup
        └─ android.widget.AdapterView
            └─ android.widget.AbsListView
                └─ android.widget.GridView
                └─ android.widget.ListView
                    └─ android.widget.ExpandableListView
            └─ android.widget.AbsSpinner
                └─ android.widget.Spinner
                └─ android.widget.Gallery
```

어댑터뷰 계층도

01 어댑터

■ 어댑터(Adapter)란?

- Adapter는 하나의 Object(객체)로서, 보여지는 View와 그 View에 올릴 Data를 연결하는 일종의 Bridge다.
- 즉, 데이터의 원본을 받아 관리하고, 어댑터뷰가 출력할 수 있는 형태로 데이터를 제공하는 중간 객체 역할을 한다.
- 이러한 어댑터는 ArrayAdapter, CursorAdapter, SimpleAdapter 등등이 있다.
- 이때 어댑터와 연결된 원본 데이터가 변경되면 notifyDataSetChanged() 메서드를 호출하여 어댑터 뷰에 원본이 변경되었다고 알려주어 어댑터 뷰가 다시 그림을 그리도록 해야한다.
- ListAdapter, BaseAdapter, ArrayAdapter, SimpleAdapter등이 있다.
- ArrayAdapter 하나의 항목에 하나의 문자를 나열할때 사용

02 리스트뷰

- 리스트뷰(ListView)
 - 데이터를 리스트 모양으로 보여주고 그중 하나를 선택하는 용도로 사용

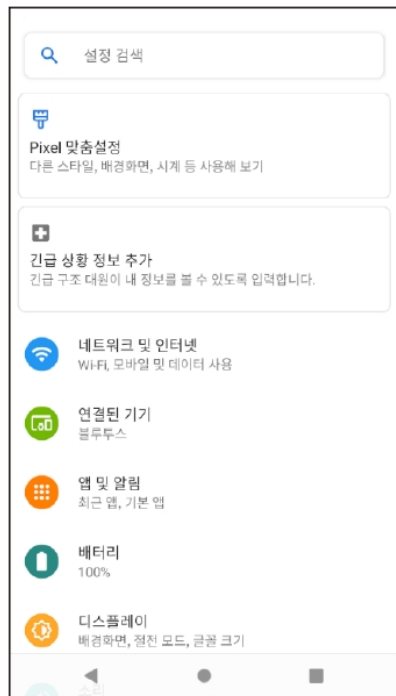


그림 11-1 안드로이드의 설정 기능

02 리스트뷰

- XML을 이용한 리스트뷰 만들기 (Exec11-1)
 - XML 파일의 <ListView> </ListView> 태그 부분에 리스트뷰를 생성
 - Kotlin 코드에서 리스트뷰에 실제 데이터 채움
 - ① 리스트뷰에 나열할 내용을 미리 String 배열로 만들어놓음
 - ② 리스트뷰 변수를 생성하고 XML의 <ListView>에 대응시킴
 - ③ ArrayAdapter<String> 형의 변수를 선언하고, 리스트뷰의 모양과 내용을 ①번의 배열로 채움
 - ④ ③번에서 생성한 어레이어댑터를 ②번의 리스트뷰 변수에 적용
 - ⑤ 리스트뷰의 항목을 클릭했을 때 동작하는 람다식 정의

예제 11-1 리스트뷰 기본 예제의 XML 코드

```

1 <LinearLayout>
2     <ListView
3         android:id="@+id/listView1"
4         android:layout_width="match_parent"
5         android:layout_height="wrap_content" >
6     </ListView>
7 </LinearLayout>

```

Item 1
Sub Item 1

Item 2
Sub Item 2

Item 3
Sub Item 3

Item 4
Sub Item 4

02 리스트뷰

예제 11-2 리스트뷰 기본 예제의 Kotlin 코드

```

1 public override fun onCreate(savedInstanceState: Bundle?) {
2     super.onCreate(savedInstanceState)
3     setContentView(R.layout.activity_main)
4     title = "리스트뷰 테스트"
5
6     var mid = arrayOf("히어로즈", "24시", "로스트",
7         "로스트룸", "스몰빌", "탐정몽크", "빅뱅이론",
8         "프렌즈", "텍스터", "클리", "가십걸", "테이큰",
9         "슈퍼내추럴", "브이")
10
11     var list = findViewById<View>(R.id.listView1) as ListView
12
13     var adapter : ArrayAdapter<String> = ArrayAdapter(this,
14         android.R.layout.simple_list_item_1, mid)
15     list.adapter = adapter
16
17     list.setOnItemClickListener { parent, view, position, id ->
18         Toast.makeText(applicationContext, mid[position], Toast.LENGTH_SHORT).show()
19     }
20 }

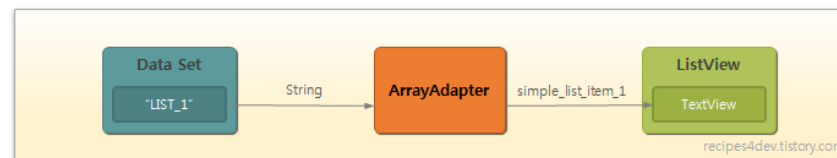
```



Java

Kotlin

```
val adapter = ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, LIST_MENU)
```



02 리스트뷰

- 리스트뷰의 다양한 모양 설정
 - 라디오버튼이나 체크박스로도 설정이 가능함

예제 11-3 모양을 변경한 리스트뷰의 Kotlin 코드

```
1 val adapter : ArrayAdapter<String> = ArrayAdapter( this,  
2           android.R.layout.simple_list_item_multiple_choice, mid )  
3 list.choiceMode = ListView.CHOICE_MODE_MULTIPLE  
4 list.adapter = adapter
```

히어로즈	<input checked="" type="checkbox"/>
24시	<input checked="" type="checkbox"/>
로스트	<input type="checkbox"/>
로스트룸	<input type="checkbox"/>

02 리스트뷰

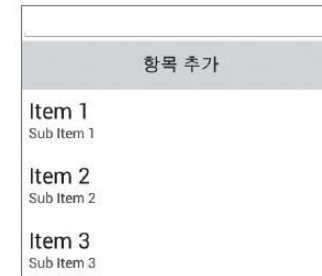
- 리스트뷰의 동적 추가·삭제 (Exec11-4)
 - 리스트뷰의 항목을 동적으로 추가하거나 삭제할 수도 있음
 - ArrayList <T>를 정의한 후 add()와 removeAt() 메소드 사용

예제 11-4 동적인 리스트뷰 항목 추가·삭제의 XML 코드

```

1 <LinearLayout>
2   <EditText
3     android:id="@+id/edtItem" />
4   <Button
5     android:id="@+id/btnAdd"
6     android:text="항목 추가" />
7   <ListView
8     android:id="@+id/listView1" />
9 </LinearLayout>

```



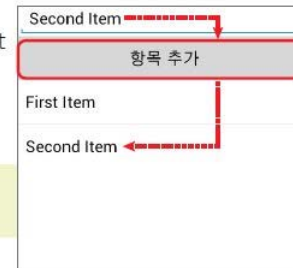
02 리스트뷰

예제 11-5 동적인 리스트뷰 항목 추가·삭제의 Kotlin 코드

```

1  public override fun onCreate(savedInstanceState: Bundle?) {
2      super.onCreate(savedInstanceState)
3      setContentView(R.layout.activity_main)
4
5      var midList = ArrayList<String>()
6      var list = findViewById<ListView>(R.id.listView1)
7
8      var adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, midList)
9      list.adapter = adapter
10
11     var editItem = findViewById(R.id.editItem) as EditText
12     var btnAdd = findViewById(R.id.btnAdd) as Button
13
14     btnAdd.setOnClickListener {
15         midList.add(editItem.text.toString())
16         adapter.notifyDataSetChanged()
17     }
18
19     list.setOnItemLongClickListener { parent, view, position, id ->
20         midList.removeAt(position)
21         adapter.notifyDataSetChanged()
22         false
23     }
24 }

```



03 그리드뷰

■ <실습 11-1> 영화 포스터 보기 1

- 그리드뷰를 이용하여 여러 사진(영화 포스터)을 격자 모양으로 배치
- 영화 포스터를 클릭하면 확대된 포스터가 대화상자로 표시

■ 안드로이드 프로젝트 생성

- (1) 새 프로젝트 만들기
 - 프로젝트 이름 : 'Project11_1'
 - 패키지 이름 : 'com.cookandroid.project11_1'
 - 그 외 규칙은 [실습 2-4]의 (1)~(4)를 따름

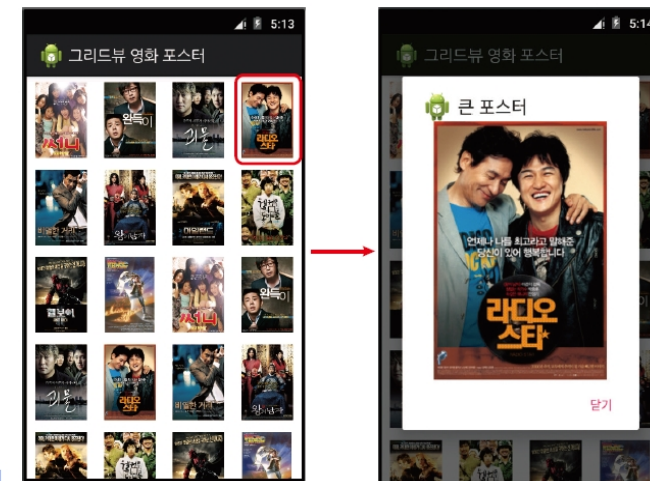


그림 11-2 그리드뷰 영화 포스터 앱 결과 화면

03 그리드뷰

- 화면 디자인 및 편집
 - (2) activity_main.xml에 그리드뷰 1개 추가

예제 11-6 activity_main.xml

```
1 <LinearLayout>
2     <GridView
3         android:id="@+id/gridView1"
4         android:layout_width="match_parent"
5         android:layout_height="wrap_content"
6         android:gravity="center"
7         android:numColumns="4" >
8     </GridView>
9 </LinearLayout>
```

03 그리드뷰

- (3) 그리드뷰의 작은 사진을 클릭하면 큰 사진을 보여주는 대화상자용 XML 파일 만들기
 - [res]-[layout]에서 [New]-[Layout resource file]을 선택하고 아래 정보 입력
 - » File name : 'dialog.xml' , Root element : 'LinearLayout'
- (4) dialog.xml에 이미지뷰를 1개 생성

예제 11-7 dialog.xml

```
1 <LinearLayout
2     android:gravity="center" >
3     <ImageView
4         android:id="@+id/ivPoster" />
5 </LinearLayout>
```

- (5) 영화 포스터 이미지 10개를 [res]-[drawable] 폴더에 복사

03 그리드뷰

- Kotlin 코드 작성 및 수정
 - (6) 메인 액티비티에서 BaseAdapter의 상속을 받는 MyGridAdapter를 정의하고 activity_main.xml의 그리드뷰에 MyGridAdapter 변수 적용

예제 11-8 메인 액티비티의 Kotlin 코드 1

```

1  override fun onCreate(savedInstanceState: Bundle?) {
2      super.onCreate(savedInstanceState)
3      setContentView(R.layout.activity_main)
4      title = "그리드뷰 영화 포스터"
5
6      var gv = findViewById<View>(R.id.gridView1) as GridView
7      var gAdapter = MyGridAdapter(this)
8      gv.adapter = gAdapter
9  }
10
11 inner class MyGridAdapter(var context: Context) : BaseAdapter() {
12
13     override fun getView(p0: Int, p1: View?, p2: ViewGroup?) : View {
14         TODO("not implemented")
15     }
16     override fun getItem(p0: Int) : Any {
17         TODO("not implemented")
18     }
19     override fun getItemId(p0: Int) : Long {
20         TODO("not implemented")
21     }
22     override fun getCount() : Int {
23         TODO("not implemented")
24     }
25 }

```

03 그리드뷰

- (7) MyGridAdapter를 계속 코딩
 - 영화 포스터 파일의 아이디를 배열로 지정

예제 11-9 메인 액티비티의 Kotlin 코드 2

```
1 var posterID = arrayOf(  
2     R.drawable.mov01, R.drawable.mov02, R.drawable.mov03, R.drawable.mov04,  
3     R.drawable.mov05, R.drawable.mov06, R.drawable.mov07, R.drawable.mov08,  
4     R.drawable.mov09, R.drawable.mov10,  
5     ~~~ 생략(2~4행 세 번 반복) ~~~  
6 )
```


03 그리드뷰

- (8) getCount(), getView() 메소드 수정
 - getCount() : 그리드뷰에 보일 이미지의 개수를 반환하도록 수정
 - getView() : 영화 포스터를 각 그리드뷰의 칸마다 이미지뷰로 생성

예제 11-10 메인 액티비티의 Kotlin 코드 3

```
1  override fun getView(p0: Int, p1: View?, p2: ViewGroup?) : View {
2      var imageview = ImageView(context)
3      imageview.layoutParams = ViewGroup.LayoutParams(200, 300)
4      imageview.scaleType = ImageView.ScaleType.FIT_CENTER
5      imageview.setPadding(5, 5, 5, 5)
6
7      imageview.setImageResource(posterID[p0])
8
9      return imageview
10 }
11 override fun getItem(p0: Int) : Any {
12     return 0
13 }
14 override fun getItemId(p0: Int) : Long {
15     return 0
16 }
17 override fun getCount() : Int {
18     return posterID.size
19 }
```

03 그리드뷰

- (9) 각각의 영화 포스터를 클릭하면 대화상자가 나오고 포스터의 원래 크기로 보이게 코딩

예제 11-11 메인 액티비티의 Kotlin 코드 4

```
1 imageView.setOnClickListener {
2     var dialogView = View.inflate(this@MainActivity, R.layout.dialog, null)
3     var dlg = AlertDialog.Builder(this@MainActivity)
4     var ivPoster = dialogView.findViewById<ImageView>(R.id.ivPoster)
5     ivPoster.setImageResource(posterID[p0])
6     dlg.setTitle("큰 포스터")
7     dlg.setIcon(R.drawable.ic_launcher)
8     dlg.setView(dialogView)
9     dlg.setNegativeButton("닫기", null)
10    dlg.show()
11 }
```

- 프로젝트 실행 및 결과 확인
 - (10) 프로젝트를 다시 실행하면 클릭한 영화 포스트가 확대되어 대화 상자로 보임

03 그리드뷰

▶ 작업 풀어보기 11-1

영화 포스터를 클릭하면 나오는 대화상자의 제목 창에 영화 제목이 보이도록 [실습 11-1]을 수정하라. 대화상자의 아이콘도 수정하라.

HINT posterID와 동일한 개수의 문자열 배열을 정의하고 영화 제목을 넣는다.



그림 11-3 수정된 그리드뷰 영화 포스터 앱

01 갤러리

- 갤러리(Gallery)
 - 사진이나 이미지를 배치하고 좌우로 스크롤하여 보여줌
 - 이미지 목록을 스크롤하는 기능만 있으므로 이미지를 클릭했을 때 큰 이미지가 나타나게 하려면 Kotlin 코드를 추가해야 함

01 갤러리

■ <실습 11-2> 영화 포스터 보기 2

- [실습 11-1]에서는 그리드뷰를 이용하여 작성한 영화 포스터 보기 앱을 이번에는 갤러리를 이용함
- 영화 포스터를 클릭하면 대화상자 대신 아래쪽 이미지뷰에 확대된 영화 포스터가 나옴

■ 안드로이드 프로젝트 생성

- (1) 새 프로젝트 만들기
 - 프로젝트 이름 : 'Project11_2'
 - 패키지 이름 : 'com.cookandroid.project11_2'
 - 그 외 규칙은 [실습 2-4]의 (1)~(4)를 따름



그림 11-4 갤러리 영화 포스터 앱 결과 화면

01 갤러리

- 화면 디자인 및 편집
 - (2) activity_main.xml에 갤러리와 이미지뷰 추가

예제 11-12 activity_main.xml

```
1 <LinearLayout>
2     <Gallery
3         android:id="@+id/gallery1"
4         android:layout_width="match_parent"
5         android:layout_height="wrap_content"
6         android:spacing="5dp" />
7     <ImageView
8         android:id="@+id/ivPoster"
9         android:layout_width="match_parent"
10        android:layout_height="match_parent"
11        android:padding="20dp" />
12 </LinearLayout>
```

- (3) 영화 포스터 이미지 10개를 [res]-[drawable] 폴더에 복사

01 갤러리

- Kotlin 코드 작성 및 수정
 - (4) 메인 액티비티에서 BaseAdapter를 상속받는 MyGalleryAdapter 정의
 - activity_main.xml의 갤러리에 MyGalleryAdapter 변수 적용

예제 11-13 메인 액티비티의 Kotlin 코드 1

```

1  override fun onCreate(savedInstanceState: Bundle?) {
2      super.onCreate(savedInstanceState)
3      setContentView(R.layout.activity_main)
4      title = "갤러리 영화 포스터"
5
6      var gallery = findViewById<Gallery>(R.id.gallery1)
7      var galAdapter = MyGalleryAdapter(this)
8      gallery.adapter = galAdapter
9  }
10
11  inner class MyGalleryAdapter(var context: Context) : BaseAdapter() {
12      var posterID = arrayOf(
13          R.drawable.mov11, R.drawable.mov12, R.drawable.mov13, R.drawable.mov14,
14          R.drawable.mov15, R.drawable.mov16, R.drawable.mov17, R.drawable.mov18,
15          R.drawable.mov19, R.drawable.mov20
16      )
17
18      ~~~ 생략(자동 완성된 추상 메소드 4개) ~~~
19
20  }

```

01 갤러리

- (5) getCount()와 getView() 메소드 수정

예제 11-14 메인 액티비티의 Kotlin 코드 2

```
1  override fun getView(p0: Int, p1: View?, p2: ViewGroup?) : View {  
2      var imageview = ImageView(context)  
3      imageview.layoutParams = Gallery.LayoutParams(200, 300)  
4      imageview.scaleType = ImageView.ScaleType.FIT_CENTER  
5      imageview.setPadding(5, 5, 5, 5)  
6      imageview.setImageResource(posterID[p0])  
7  
8      return imageview  
9  }  
10 override fun getItem(p0: Int) : Any {  
11     return 0  
12 }  
13 override fun getItemId(p0: Int) : Long {  
14     return 0  
15 }  
16 override fun getCount() : Int {  
17     return posterID.size  
18 }
```


01 갤러리

- (6) 갤러리의 영화 포스터를 클릭하면 아래쪽 이미지뷰에 포스터가 원래 크기로 보이게 함

예제 11-15 메인 액티비티의 Kotlin 코드 3

```
1 imageView.setOnTouchListener { v, event ->
2     var ivPoster = findViewById<ImageView>(R.id.ivPoster)
3     ivPoster.scaleType = ImageView.ScaleType.FIT_CENTER
4     ivPoster.setImageResource(posterID[p0])
5     false
6 }
```

- 프로젝트 실행 및 결과 확인

- (7) 프로젝트를 실행해하면 터치한 영화 포스터가 확대되어 아래쪽 이미지 뷰에 나타남

01 갤러리

▶ 작업 풀어보기 11-2

영화 포스터를 클릭하면 영화 제목이 토스트 메시지에 나타나도록 [실습 11-2]를 수정하라. 토스트 메시지가 아이콘과 함께 나타나게 한다.

HINT 고급 토스트는 [실습 7-3]을 참조한다.



그림 11-5 수정된 갤러리 영화 포스터 앱

01 스피너

- 스피너(Spinner)
 - PC의 드롭다운 박스와 비슷한 기능
 - 화면이 작은 스마트폰에서 여러 개 중 하나를 선택할 수 있도록 확장하는 용도

예제 11-16 스피너 기본 예제의 XML 코드

```
1 <LinearLayout>
2     <Spinner
3         android:id="@+id/spinner1"
4         android:layout_width="match_parent"
5         android:layout_height="wrap_content" />
6 </LinearLayout>
```

01 스피너

예제 11-17 스피너 기본 예제의 Kotlin 코드

```

1 public override fun onCreate(savedInstanceState: Bundle?) {
2     super.onCreate(savedInstanceState)
3     setContentView(R.layout.activity_main)
4     title = "스피너 테스트"
5
6     var movie = arrayOf("콩푸팬더", "짱구는 못말려", "아저씨",
7         "아바타", "대부", "국가대표", "토이스토리3",
8         "마당을 나온 암탉", "죽은 시인의 사회", "서유기")
9
10    var spinner = findViewById<Spinner>(R.id.spinner1)
11
12    var adapter: ArrayAdapter<String>
13    adapter = ArrayAdapter(this, android.R.layout.simple_spinner_item, movie)
14    spinner.adapter = adapter
15 }

```



01 스피너

▶ 직접 풀어보기 11-3

스피너에서 영화 제목을 선택하면 해당 영화 포스터가 아래쪽 이미지에 나오도록 [예제 11-17]을 수정하라.

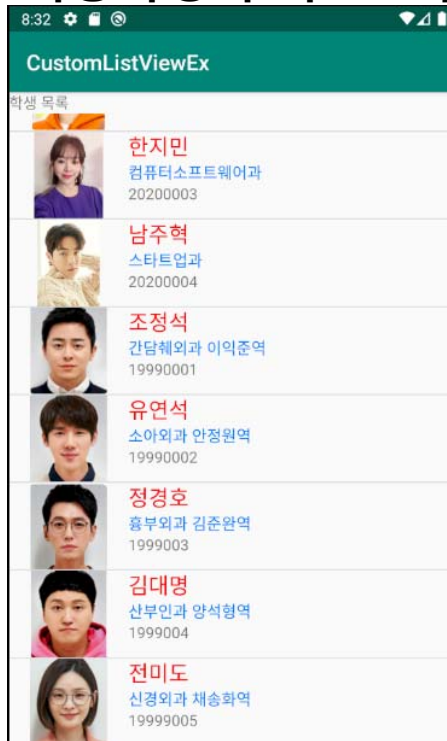
HINT Spinner의 `onItemSelectedListener`에 `AdapterView.OnItemClickListener` `SelectedListener` 람다식을 대입하는 방식을 사용한다.



그림 11-6 수정된 스피너 영화 포스터 앱

커스텀 리스트 뷰(Custom ListView)

■ 사용자정의 리스트 뷰



■ 한행(row) – rowitem.xml



커스텀 리스트 뷰(Custom ListView)

■ Student 클래스

- 이름
- 소속학과
- 학생번호
- 사진

■ 클래스구현방법

- `class StudentC (var name : String, var dept : String, var studentid: Int, var sajin : Int) { }`

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        var studentList : ArrayList<StudentC> = arrayListOf<StudentC>()  
        studentList.add( StudentC("배수지", "멀티미디어과", 2020001, R.drawable.suji))  
        studentList.add( StudentC("한지민", "컴퓨터소프트웨어과", 2020003, R.drawable.jimin) )  
        studentList.add( StudentC("남주혁", "스타트업과", 2020004, R.drawable.juhuk) )  
        studentList.add( StudentC("조정석", "간담체외과 이의준역", 19990001, R.drawable.talent1) )  
        studentList.add( StudentC("유연석", "소아외과 안정원역", 19990002, R.drawable.talent2) )  
        studentList.add( StudentC("정경호", "흉부외과 김준완역", 1999003, R.drawable.talent3) )  
        studentList.add( StudentC("김대명", "산부인과 양석형역", 1999004, R.drawable.talent4) )  
        studentList.add( StudentC("전미도", "신경외과 채송화역", 19999005, R.drawable.talent5) )  
  
        var adapter1 = ListViewAdapter(studentList)  
        listView1.adapter = adapter1  
        listView1.setOnItemClickListener { adapterView, view, i, l ->  
            Toast.makeText(applicationContext, studentList.get(i).name, Toast.LENGTH_SHORT).show()  
        }  
    }  
}
```

커스텀 리스트 뷰(Custom ListView)

```
7
8 class ListViewAdapter(var studentList : ArrayList<StudentC>) : BaseAdapter() {
9     override fun getView(p0: Int, p1: View?, p2: ViewGroup?): View {
10         var rowItemView : View? = p1
11         if (rowItemView == null) {
12             rowItemView = View.inflate(p2?.context, R.layout.rowitem, root: null)
13         }
14         val student1 : StudentC = studentList[p0]
15         rowItemView!!.itemImage.setImageResource(student1.sajin)
16         rowItemView.itemName.text = student1.name
17         rowItemView.itemDept.text = student1.dept
18         rowItemView.itemId.text = student1.studentId.toString()
19         return rowItemView
20     }
21
22     override fun getItem(p0: Int): Any {
23         return studentList[p0]
24     }
25
26     override fun getItemId(p0: Int): Long {
27         return p0.toLong()
28     }
29
30     override fun getCount(): Int {
31         return studentList.size
32     }
33
34 }
```