

12주차 정리

1. 리스트와 그리드뷰

1.1 어댑터뷰(AdapterView)

- AdapterView 하위에 ListView, ExpandableListView, GridView, Spinner, Gallery 등을 묶어서 표현
- 어댑터뷰를 사용할 때 어댑터뷰의 모양을 설정하고 데이터를 채워주는 ArrayAdapter<T> 클래스를 함께 사용

```
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

1.2 리스트뷰(ListView)

데이터를 리스트 모양으로 보이며 리스트 중 하나를 선택하는 용도

XML로 리스트뷰 만들기

- ① 리스트뷰에 나열할 내용을 String 배열로 미리 만들
- ② 리스트뷰 변수를 생성하고 XML의 <ListView>에 대응시킴
- ③ ArrayAdapter<String>형의 변수를 선언하고, 리스트뷰의 모양과 내용을 ①번 배열로 채움
- ④ ③번에서 생성한 어레이어댑터를 ②번의 리스트뷰 변수에 적용
- ⑤ 리스트뷰의 항목을 클릭했을 때 동작하는 리스너 정의

예제 11-2 리스트뷰 기본 예제의 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4     setTitle("리스트뷰 테스트");
5
6     final String[] mid = { "히어로즈", "24시", "로스트", "로스트를", "스물뿔", "탕정웅크",
7         "빅뱅이론", "프렌즈", "덕스터", "글리", "가십걸", "테이큰", "슈퍼내추럴", "브이" };
8
9     ListView list = (ListView) findViewById(R.id.listView1);
10
11     ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
12         android.R.layout.simple_list_item_1, mid);
13     list.setAdapter(adapter);
14
15     list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
16         public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
17             long arg3) {
18             Toast.makeText(getApplicationContext(),
19                 mid[arg2], Toast.LENGTH_SHORT).show();
20         }
21     });
22 }
```



예제 11-2 리스트뷰 기본 예제의 Java 코드

```

1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4     setTitle("리스트뷰 테스트");
5
6     final String[] mid = { "히어로즈", "24시", "로스트", "로스트를", "스물뱀", "탕정웅크",
7         "빅뱅이론", "프렌즈", "덕스터", "글리", "가십걸", "테이큰", "슈퍼내추럴", "브이" };
8
9     ListView list = (ListView) findViewById(R.id.listView1);
10
11     ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
12         android.R.layout.simple_list_item_1, mid);
13     list.setAdapter(adapter);
14
15     list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
16         public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
17             long arg3) {
18             Toast.makeText(getApplicationContext(),
19                 mid[arg2], Toast.LENGTH_SHORT).show();
20         }
21     });
22 }

```



리스트뷰의 다양한 모양 설정

라디오 버튼: `simple_list_item_single_choice`로 바꾼다.

체크박스: `simple_list_item_multiple_choice`로 둔다.

`list.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE, SINGLE)`

```

1 ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
2     android.R.layout.simple_list_item_multiple_choice, mid);
3 list.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
4 list.setAdapter(adapter);

```



리스트 뷰의 동적 추가, 삭제

리스트 뷰의 항목을 동적으로 추가, 삭제하려면 `ArrayList<T>`를 정의 후 `add()`, `remove()`를 사용한다.

`ArrayAdapter`에 해당 `ArrayList`를 생성자의 매개변수로 넣어주고 **`ArrayList`를 `add()`, `remove()`할 때마다 `ArrayAdapter.notifyDataSetChanged()`를 호출한다.**

예제 11-5 리스트뷰 동적 추가·삭제의 Java 코드

```

1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4
5     final ArrayList<String> midList = new ArrayList<String>();
6     ListView list = (ListView) findViewById(R.id.listView1);
7
8     final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
9         android.R.layout.simple_list_item_1, midList);
10    list.setAdapter(adapter);
11
12    final EditText edtItem = (EditText) findViewById(R.id.edtItem);
13    Button btnAdd = (Button) findViewById(R.id.btnAdd);
14
15    btnAdd.setOnClickListener(new View.OnClickListener() {
16        public void onClick(View v) {
17            midList.add(edtItem.getText().toString());
18            adapter.notifyDataSetChanged();
19        }
20    });
21
22    list.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
23        public boolean onItemLongClick(AdapterView<?> parent, View view, int
24            position, long id) {
25            midList.remove(position);
26            adapter.notifyDataSetChanged();
27            return false;
28        }
29    });
30 }

```



1.3 그리드뷰(Gridview)

- 그리드뷰는 사진이나 그림을 격자 모양으로 배치한다.

- XML 파일에는 간단히 <GridView>를 넣으면 되지만, java 코드에는 필요한 내용을 반드시 코딩해야한다.
- 특히 <GirdView>의 속성 중 열 개수를 지정하는 **numColumns**는 꼭 넣어야 한다.

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>

```

android:numColumns="4"

Java 코드 작성

1. 메인 액티비티에서 BaseAdapter를 상속(extends)받는 MyGridAdatper를 정의한다.
2. activity_main.xml의 그리드뷰에 MyGridAdatper 변수를 적용한다.

```

final GrdiView gv = (GridView) findViewById(R.id.gridView);
MyGridAdapter gAdapter = new MyGridAdapter(this);
gv.setAdapter(gAdapter);

```

public int getCount() or public View getView()

- getCount(): 그리드뷰에 보여질 이미지의 개수를 반환하도록 수정한다.
- getView(): 영화 포스터를 각 그리드뷰의 칸마다 이미지뷰를 생성해서 보여주도록한다.

2. 갤러리와 스피너

2.1 갤러리

- 사진이나 이미지를 배치하고 좌우로 스크롤해서 볼 수 있도록 한다.
- 이미지 목록을 스크롤하는 기능만 존재한다.
→ 이미지를 클릭했을 때 큰 이미지가 나오게 하려면 Java 코드를 추가해서 사용한다.
- 그리드뷰와 비슷하지만 좀 더 부드럽고 고급스러운 느낌

Java 코드 작성

1. 메인 액티비티에서 BaseAdapter를 상속받는 MyGalleryAdapter 저으이
2. activtiy_main.xml의 그리드뷰에 MyGalleryAdapter 변수 적용

```

Gallery gallery = (Gallery) findViewById(R.id.gallery1);
MyGalleryAdapter galAdapter = new MyGalleryAdapter(this);
gallery.setAdapter(galAdapter);

```

onCreate() 함수 내부

- getCount()와 getView()사용

예제 11-14 메인 액티비티의 Java 코드 2

```

1 public int getCount() {
2     return posterID.length;
3 }
4
5 public View getView(int position, View convertView, ViewGroup parent) {
6     ImageView imageview = new ImageView(context);
7     imageview.setLayoutParams(new Gallery.LayoutParams(200, 300));
8     ~~~~ 생략([예제 11-10]의 8~11행과 동일) ~~~~
9
10    return imageview;
11 }

```

- Java코드로 포스트 클릭시 원래 크기로 보이도록하기 → setScaleType()

예제 11-15 메인액티비티의 Java 코드 3

```
1 final int pos = position;
2 imageView.setOnTouchListener(new View.OnTouchListener() {
3     public boolean onTouch(View v, MotionEvent event) {
4         ImageView ivPoster = (ImageView) findViewById(R.id.ivPoster);
5         ivPoster.setScaleType(ImageView.ScaleType.FIT_CENTER);
6         ivPoster.setImageResource(posterID[pos]);
7         return false;
8     }
9 });
```

setOnClickListener: 단순 click 이벤트를 받아와 동작을 구현함

setOnTouchListener: 손가락이 화면과 닿는(DOWN), 떨어지는(UP), 화면을 드래그하는(MOVE) 이벤트에 따라 여러 동작을 세부적으로 구현 가능함.

2.2 스피너

PC의 드롭다운 박스와 비슷한 기능

```
10 Spinner spinner = (Spinner) findViewById(R.id.spinner1);
11
12 ArrayAdapter<String> adapter;
13 adapter = new ArrayAdapter<String>(this,
14     android.R.layout.simple_spinner_item, movie);
15 spinner.setAdapter(adapter);
16 }
```



3. SQLite의 기본

3.1 데이터베이스 기본 개념

데이터베이스의 정의

대용량의 데이터 집합을 체계적으로 구성해놓은 것

데이터베이스 관리 시스템 - DB는 여러 사용자나 시스템이 서로 공유할 수 있어야한다. 이러한 DB를 관리해주는 시스템 또는 SW를 DBMS(DataBase Management System)라고 한다.

DBMS에는 계층형, 망형, 관계형, 객체지향형, 객체관계형이있다.

관계형 데이터베이스 - SQLite가 관계형 DBMS에 속하고 여러 유형 중 가장 많이 사용된다.

장점

1. 업무가 변화할 경우 다른 DBMS에 비해 변화게 쉽게 순응한다.
2. 유지 및 보수 측면에서 편리하다
3. 대용량 데이터 관리와 데이터 무결성을 보장한다.

데이터 무결성: 데이터에 여러 명의 사용자가 동시에 접근하고 조작하더라도 데이터의 정확성, 일관성을 유지하고 보증할 수 있는 성질

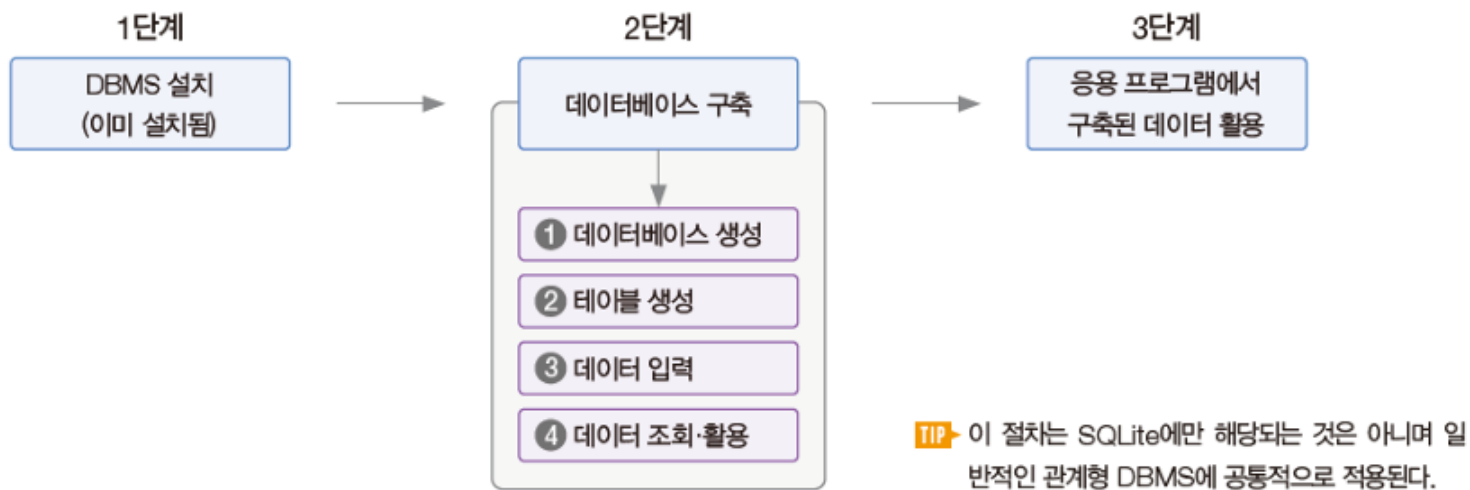
단점

1. 시스템 자원을 많이 차지해서 시스템이 전반적으로 느려진다.

SQL(Structured Query Language): 구조화된 질의 언어

데이터베이스 모델링: 현실 세계의 데이터를 DBMS 안에 어떻게 표현할 지 설계하는 과정.

- **데이터** : 하나하나의 단편적인 정보를 뜻함
- **테이블** : 회원 데이터가 표 형태로 표현된 것
- **데이터베이스(DB)** : 테이블이 저장되는 장소로 주로 원통 모양으로 표현
 - 각 데이터베이스는 서로 다른 고유한 이름이 있어야 함
- **DBMS** : 데이터베이스를 관리하는 시스템 또는 소프트웨어를 말함
 - 안드로이드에 포함된 SQLite 소프트웨어가 이에 해당
- **열(칼럼 또는 필드)** : 각 테이블은 1개 이상의 열로 구성됨
- **열 이름** : 각 열을 구분하는 이름, 열 이름은 각 테이블 안에서는 중복되지 않아야 함
- **데이터 형식** : 열의 데이터 형식을 뜻함
 - 테이블을 생성할 때 열 이름과 함께 지정해야 함
- **행(로우)** : 실제 데이터
- **SQL** : 사용자와 DBMS가 소통하기 위한 언어



SQLite의 활용

안드로이드 앱 개발을 위한 SQLite 동작 방식

- SQLiteOpenHelper 클래스, SQLiteDatabase 클래스, Cursor 인터페이스 활용

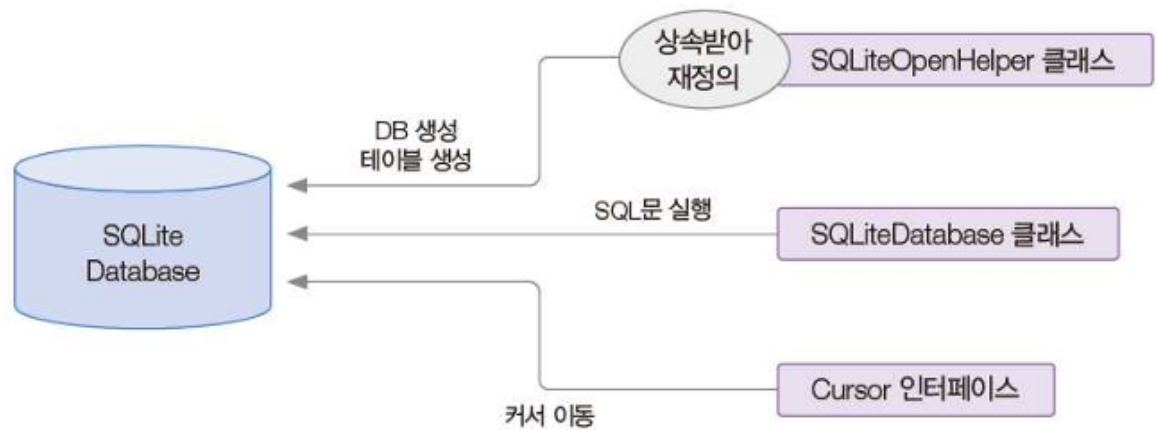


그림 12-9 SQLite 관련 클래스의 동작

DB생성 및 테이블 생성 - SQLiteOpenHelper 클래스를 상속받아 재정의 후 수행한다.

SQL문 실행 - SQLiteDatabase 클래스를 이용해 수행한다.

커서이동(다음 행 이동) - Cursor 인터페이스를 이용해 수행한다.

- 커서는 테이블을 조회할 때 사용하는 row(행, 실제 데이터 가로줄)을 가리키는 pointer 느낌

SQLiteOpenHelper 상속 받기

```

7 public class myDBHelper extends SQLiteOpenHelper {
8     public myDBHelper(Context context) {
9         super(context, "groupDB", null, 1);
10    }

```

onCreate() 메소드 - 테이블을 생성하는 기능을 코딩

onUpgrade() 메소드 - 테이블을 삭제한 후 다시 생성 → Drop Table 후 onCreate()한다.

예제 12-3 Java 코드 2

```
1  @Override
2  public void onCreate(SQLiteDatabase db) {
3      db.execSQL("CREATE TABLE groupTBL ( gName CHAR(20) PRIMARY KEY,
4          gNumber INTEGER);");
5  }
6  @Override
7  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
8      db.execSQL("DROP TABLE IF EXISTS groupTBL");
9      onCreate(db);
10 }
```

테이블 조회 **SELECT 열 이름1, 열이름 2 FROM 테이블명 WHERE 조건**
전체 열을 조회하려면 열 이름을 '*'로 표시

클래스 또는 인터페이스	메소드	주요 용도
SQLiteOpenHelper 클래스	생성자	DB 생성
	onCreate()	테이블 생성
	onUpgrade()	테이블 삭제 후 다시 생성
	getReadableDatabase()	읽기 전용 DB 열기, SQLiteDatabase 반환
	getWritableDatabase()	읽고 쓰기용 DB 열기, SQLiteDatabase 반환
SQLiteDatabase 클래스	execSQL()	SQL문(insert/Update/Delete) 실행
	close()	DB 닫기
	query(), rawQuery()	Select 실행 후 커서 반환
Cursor 인터페이스	moveToFirst()	커서의 첫 행으로 이동
	moveToLast()	커서의 마지막 행으로 이동
	moveToNext()	현재 커서의 다음 행으로 이동

Cursor(커서) 사용 예시

예제 12-7 Java 코드 6

```
1  btnSelect.setOnClickListener(new View.OnClickListener() {
2      public void onClick(View v) {
3          sqlDB = myHelper.getReadableDatabase();
4          Cursor cursor;
5          cursor = sqlDB.rawQuery("SELECT * FROM groupTBL;", null);
6
7          String strNames = "그룹 이름" + "\r\n" + "_____" + "\r\n";
8          String strNumbers = "인원" + "\r\n" + "_____" + "\r\n";
9
10         while (cursor.moveToNext()) {
11             strNames += cursor.getString(0) + "\r\n";
12             strNumbers += cursor.getString(1) + "\r\n";
13         }
14
15         edtNameResult.setText(strNames);
16         edtNumberResult.setText(strNumbers);
17
18         cursor.close();
19         sqlDB.close();
20     }
21 });
```

- 1. cursor를 SQL 실행 결과에 연결한다.
- 2. while(cursor.moveToNext())를 통해 다음 행으로 이동한다.
- 3. cursor.getString(0), cursor.getString(1)로 0열, 1열... 의 값을 꺼내 저장한다.

추가할 것)

같은 그룹 이름으로 두 번이상 입력 시, 토스트 메시지로 '그룹 이름이 중복됩니다'를 띄우고, DB에 추가되지 않도록 해야한다. try - catch 사용해서 예외처리 할 것.

예제 11-2 리스트뷰 기본 예제의 Java 코드

```

1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_main);
4     setTitle("리스트뷰 테스트");
5
6     final String[] mid = { "히어로즈", "24시", "로스트", "로스트룸", "스물빌", "탐정응크",
7         "빅뱅이론", "프렌즈", "텍스터", "글리", "가습걸", "테이큰", "슈퍼내추럴", "브이" };
8
9     ListView list = (ListView) findViewById(R.id.listView1);
10
11     ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
12         android.R.layout.simple_list_item_1, mid);
13     list.setAdapter(adapter);
14
15     list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
16         public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
17             long arg3) {
18             Toast.makeText(getApplicationContext(),
19                 mid[arg2], Toast.LENGTH_SHORT).show();
20         }
21     });
22 }

```

