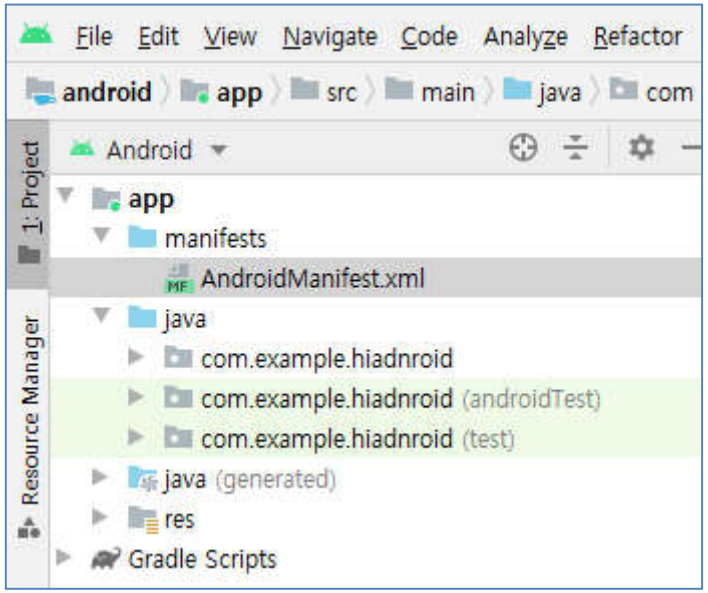
프로젝트 및 앱 분해

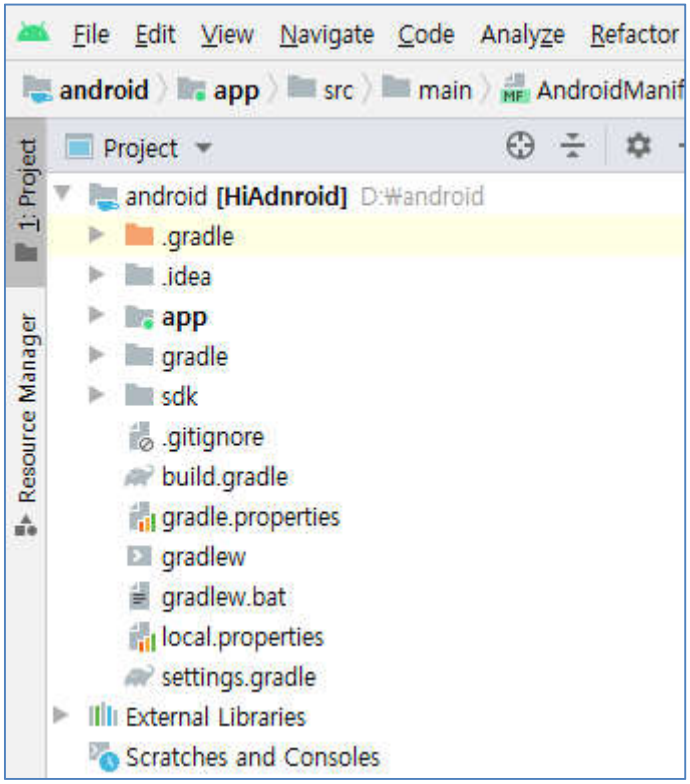
성신여자대학교 컴퓨터공학과
우 종 정 교수

프로젝트 분석

프로젝트 도구창

- 프로젝트를 구성하는 파일의 목록을 보여준다.
- 도구창 모드에 따라 보여주는 방식이 다르다.
- Android 모드는 논리적 구조를 보여주며 Project 모드는 물리적인 구조를 보여준다.





프로젝트 분석

프로젝트 구성 파일

- 종류에 따라 여러 폴더에 파일이 저장되어 있다.

파일, 폴더	설명
AndroidManifest.xml	프로젝트의 이름, 아이콘, 테마, 자동 백업 여부 등에 대한 전역 설정 정보와 앱에 소속되는 구성 요소의 속성에 대한 정보를 가진다. 디폴트로 액티비티 하나가 포함되어 있다.
MainActivity.java	메인 액티비티의 소스 파일이다. 여기에 코드를 작성하여 동작을 기술한다.
activity_main.xml	메인 액티비티의 레이아웃을 정의하며 여기에 위젯을 배치한다.
menu 폴더	메뉴 리소스이다. 메뉴가 있는 템플릿의 경우 Settings 항목 하나를 가지는 메뉴를 만들어 준다.
mipmap 폴더	앱의 아이콘 파일을 밀도별로 생성해 준다.
drawable 폴더	앱에서 사용할 이미지 파일을 저장한다. 디폴트로 비어 있는데 필요한 이미지를 이 폴더에 저장하면 된다.
values 폴더	앱에서 사용하는 문자열, 수치값, 색상값, 스타일 등을 정의한다. strings.xml 파일에 문자열이 정의되어 있다.
build.gradle	그래들의 빌드 파일이다. 프로젝트 전역 스크립트가 있고 각 모듈별 스크립트가 있다.

프로젝트 분석

매니페스트

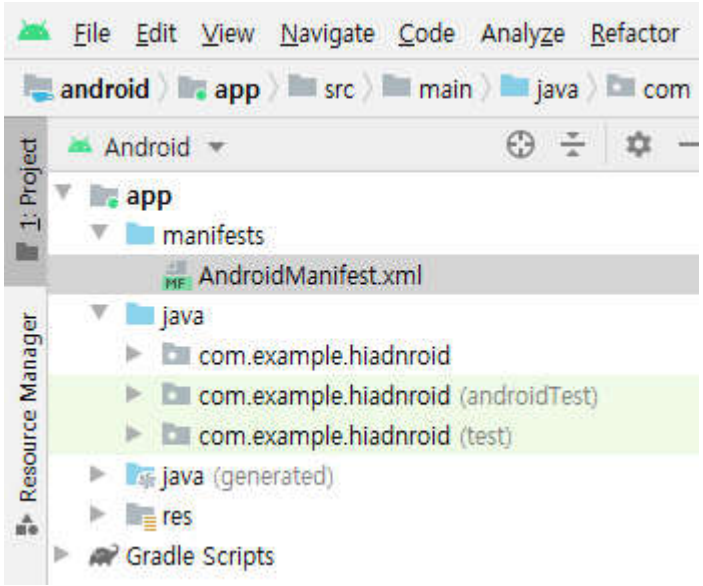
- 프로젝트의 구성 정보와 컴포넌트 목록, 권한 등이 작성된다.
- XML 포맷의 텍스트 파일이며 직접 편집 가능하다.
- 디폴트로 만든 프로젝트에는 액티비티 하나가 포함되어 있다.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hiadnroid">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

프로젝트 분석

앱의 구성 요소

- 자바 소스 파일(안드로이드 코드)
- 리소스
  - XML 파일
  - 이미지, 사운드 등



Hello 앱 분석

메인 액티비티

- 액티비티를 띄우고 내용물을 채운다.
- onCreate에서 부모 클래스의 onCreate메서드를 호출하여 기본 초기화를 수행하고 setContentView 메서드를 호출하여 레이아웃을 액티비티에 채운다.

```
package com.example.hiadnroid;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Hello 앱 분석

레이아웃 채우기

- setContentView 메서드는 다음 두 가지 원형이 있다.
  - void Activity.setContentView (int layoutResID)
  - void Activity.setContentView (View view [, ViewGroup.LayoutParams params])

- 레이아웃을 또는 뷰를 채운다.
- 레이아웃의 ID는 R.java에 정의되어 있다.
- 개발툴에서 관리하므로 직접 편집하지 않는다.
- R 클래스는 코드와 리소스를 연결시키는 역할

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found.  It
 * should not be modified by hand.
 */
package com.example.hiandroid;

public final class R {
    ....
    public static final class drawable {
        ....
    }

    public static final class id {
        ....
    }

    public static final class layout {
        ....
        public static final int activity_main=0x7f040018;
    }
```

Hello 앱 분석

레이아웃

- 그래픽 편집기보다는 텍스트 형태로 편집하는 것이 원리 이해에 더 도움이 된다.
- android 네임스페이스는 안드로이드의 속성을 의미한다.
- 레이아웃 안에 텍스트 뷰 하나가 배치되어 있다.
- 텍스트 뷰의 text 속성이 뷰에 표시된다.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android=“ ... "
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hi Android!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Hello 앱 분석

코드 수정하기

- 레이아웃을 쓰지 않고 코드로 위젯을 만들어 배치할 수도 있다.
- 사용할 클래스에 대해 import 선언을 한다.
- new 연산자로 객체를 생성하고 원하는 속성을 대입한다.
- 유니코드 기반으로 한글도 문제없이 출력된다.

```
package com.example.hiadnroid;

...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView myText = new TextView(this);
        myText.setText("코드로 문자열 출력하기");
        setContentView(myText);
    }
}
```

Hello 앱 분석

■ 코드와 리소스의 분리

- 코드에서 모든 것을 처리할 수 있지만 관리가 어렵다.
- 리소스를 분리하면 여러 가지 이점이 생긴다.

① 코드와 데이터가 완벽하게 분리되므로 개발자와 디자이너의 분담 작업이 용이하다.

② 조건에 따라 레이아웃을 교체할 수 있으므로 호환성 확보, 국제화에 유리하다.

③ 레이아웃만 수정할 때는 코드를 컴파일하지 않아도 되므로 개발 속도가 빨라진다.

④ 구조와 속성을 함축적으로 기술할 수 있으며 레이아웃 재활용도 가능하다.

- 정적 레이아웃은 XML을 사용하며 동적 레이아웃은 코드를 사용한다.

Hello 앱 분석

에러 처리

- 에러 발생시 편집기가 에러 위치와 내용을 보여 준다.
- 에러를 찾아 수정하고 다시 빌드하면 문제가 해결된다.
- 편집창과 프로젝트 도구창의 파일 목록에 에러가 발생한 파일에 빨간 줄을 그어 보여준다.