



2장. 안드로이드 프로젝트 이해

2.1 안드로이드 앱 개발 특징

2.1.1. 안드로이드의 특징

- 안드로이드는 공개 운영체제인 리눅스 기반이다.
- 안드로이드 앱 개발은 자바, 코틀린 언어를 이용해 개발한다.
- 운영체제의 핵심 코드, 라이브러리 코드, 스마트폰에 기본으로 탑재된 구글에서 만든 앱 코드 등 대부분 코드가 오픈되어 있다.
- 안드로이드 플랫폼 기반의 스마트폰을 여러 제조업체에서 만들 수 있다.
- 개발자가 만든 앱은 구글 Play 스토어뿐만 아니라 다양한 방법으로 사용자에게 배포할 수 있다.
- 안드로이드 기반의 모든 애플리케이션은 평등하다는 사상으로 스마트폰에 기본 탑재된 앱과 개발자들이 만드는 앱이 같은 환경에서 같은 API를 이용한다.
- 파편화 (Fragmentation)
- 멀티태스킹(multitasking)

2.1 안드로이드 앱 개발 특징

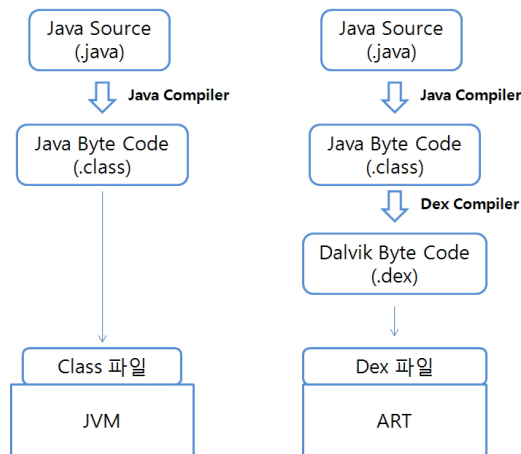
2.1.2. 안드로이드 플랫폼 아키텍처

안드로이드 아키텍처

- 안드로이드 플랫폼은 리눅스 커널 기반
- HAL(Hardware Abstraction Layer)은 자바 API 프레임워크에 하드웨어 기능을 이용하는 표준 인터페이스를 제공.
- 안드로이드 런타임(Android Runtime)은 ART 가상 머신을 이용하며 그 위에 일반 애플리케이션 개발 시 이용할 수 있는 자바 API 프레임워크를 제공

안드로이드 런타임(ART)

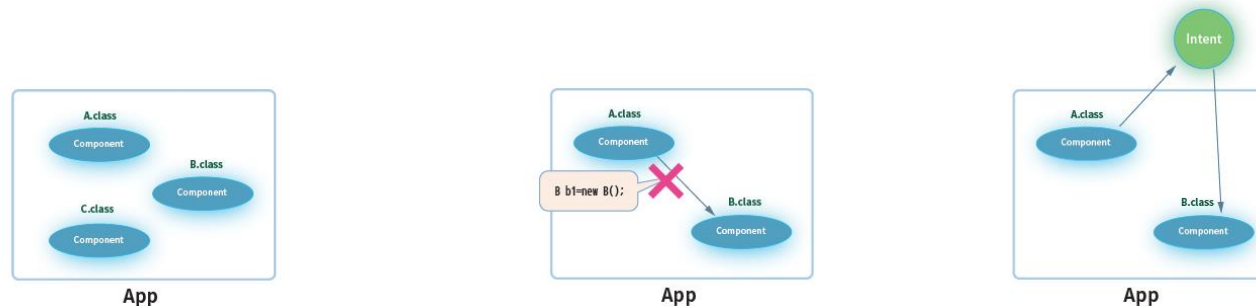
- 안드로이드의 VM은 ART(Android Runtime)
- ART는 앱을 실행할 때 DEX 파일을 실행



2.1 안드로이드 앱 개발 특징

2.1.3. 컴포넌트 기반 개발

- 컴포넌트는 앱의 구성 단위이며, 컴포넌트 여러 개를 조합하여 하나의 앱을 만든다
- 컴포넌트의 물리적인 모습은 클래스
- 컴포넌트는 앱 내에서 독립적인 실행 단위이다
- 실행하고자 하는 클래스가 안드로이드 컴포넌트 클래스라면 개발자 코드로 생성해서 실행할 수 없다.
- 인텐트(Intent)라는 것을 매개로 하여 결합하지 않은 상태에서 독립적으로 실행하는 구조



- main 함수 같은 애플리케이션의 진입 지점이 따로 없다
- 애플리케이션 라이브러리 개념이 있다



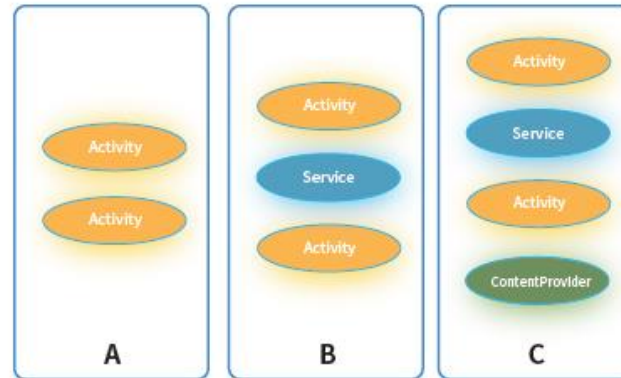
2.1 안드로이드 앱 개발 특징

안드로이드 컴포넌트 종류

- 액티비티(Activity): UI를 구성하기 위한 컴포넌트
- 서비스(Service): UI 없이 백그라운드에서 장시간 수행되는 컴포넌트
- 콘텐츠 프로바이더(ContentProvider): 애플리케이션 간 데이터를 공유하기 위한 컴포넌트
- 브로드캐스트 리시버(BroadcastReceiver): 이벤트 모델로 수행되는 컴포넌트

컴포넌트를 이용한 앱 구성

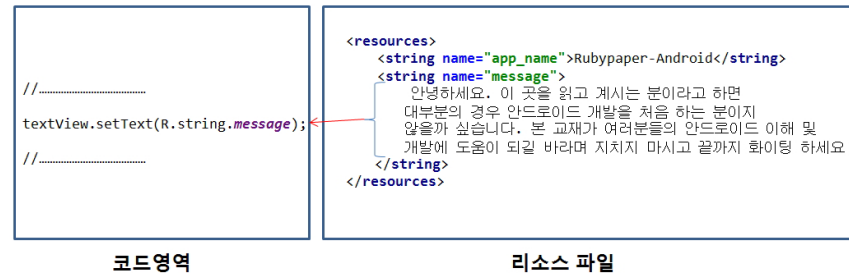
앱의 업무와 화면 설계에 따라 적절하게 컴포넌트를 결정하여 앱을 개발



2.1 안드로이드 앱 개발 특징

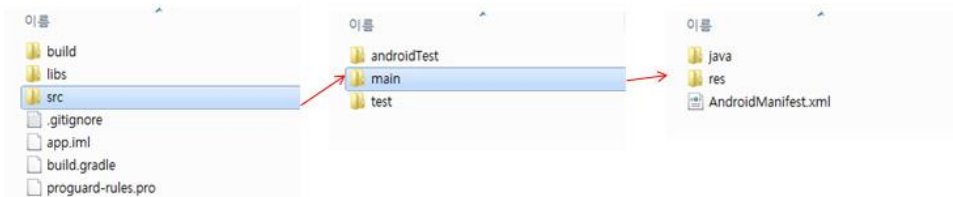
2.1.4. 리소스를 이용한 개발

- 안드로이드 앱 개발의 또 하나의 큰 특징 중 하나가 리소스 외부화를 극대화해서 개발한다는 것
- 정적인 문자열의 리소스 외부화뿐만 아니라, 문자열, 크기, 색상, 레이아웃, 메뉴, 애니메이션 등 정적인 많은 부분을 리소스로 만들어 개발하는 특징

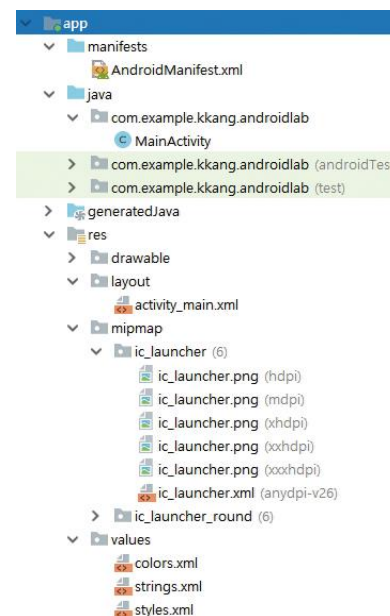


2.2 개발 디렉토리와 파일 구조

2.2.1. 앱의 디렉터리와 파일



- java/MainActivity.java: 화면 구성을 위한 액티비티 컴포넌트로 실 제 이 파일이 수행되어 화면에 UI가 출력된다.
- res: 앱의 모든 리소스 파일은 res 폴더 하위에 위치한다.
- res/drawable: 리소스 중 이미지 파일을 저장하기 위한 폴더이다.
- res/layout: 리소스 중 UI 구성을 위한 레이아웃 XML 파일을 위한 폴 더이다.
- res/mipmap: 리소스 중 앱의 아이콘 이미지를 위한 폴더이다.
- res/values: 리소스 중 문자열 값 등을 위한 폴더이다.
- AndroidManifest.xml: 앱의 메인 환경 파일이다.

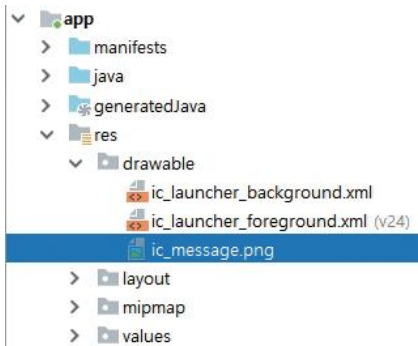


2.2 개발 디렉토리와 파일 구조

2.2.2. R.java의 이해

```
public final class R {  
    public static final class anim {  
        public static final int abc_fade_in=0x7f050000;  
        public static final int abc_fade_out=0x7f050001;  
        public static final int abc_grow_fade_in_from_bottom=0x7f050002;  
        public static final int abc_popup_enter=0x7f050003;  
    }
```

- res 폴더의 리소스들을 식별하기 위 한 변수
- res 밑에 임의의 폴더를 생성 불가
- 리소스 파일명은 자바 명명규칙을 위배할 수 없다.
- 파일 명에 대문자를 사용할 수 없다.

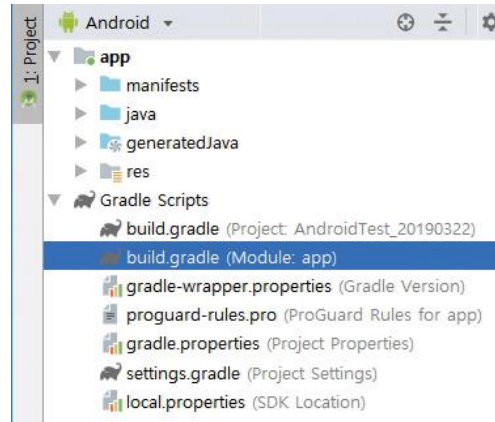


```
public static final class drawable {  
    //...  
    public static final int ic_message=0x7f020053;  
    //...  
}
```


2.3 그레이들(gradle) 파일

안드로이드의 빌드 도구

build.gradle 파일에서 그레이들 관련 설정



2.3.1. settings.gradle

그레이들에 모듈을 포함하여 그레이들이 모듈을 관리하고 빌드하게 설정

```
include ':app'
```

2.3.2. 프로젝트 수준의 그레이들

- 'Gradle Scripts' 영역의 최상위에 있는 build.gradle
- 모듈을 위한 최상위 설정을 목적

2.3 그레들(gradle) 파일

2.3.3. 모듈 수준의 그레이들

```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.user.androidlab"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {.....}
}
dependencies {
    .....
}
```

- applicationId "com.example.user.androidlab": 앱의 식별자
- minSdkVersion: 최소 지원 범위
- targetSdkVersion: 사용하고 있는 SDK 버전
- versionCode: 앱의 버전

2.4 HelloWorld 앱 코드 분석

2.4.1. AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.user.androidlab">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportRtl="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>
```

2.4 HelloWorld 앱 코드 분석

2.4.2. MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

2.4.3. activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/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="com.example.user.androidlab.MainActivity">  
    <TextView android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Hello World!"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintLeft_toLeftOf="parent"  
        app:layout_constraintRight_toRightOf="parent"  
        app:layout_constraintTop_toTopOf="parent" />  
</android.support.constraint.ConstraintLayout>
```

2.5 안드로이드 버전

버전	코드네임	배포	API 레벨
1.0	Apple Pie	2008/9	1
1.1	Banana Bread	2009/2	2
1.5	CupCake	2009/4	3
1.6	Donut	2009/9	4
2.0.x	Eclair	2009/10	5~6
2.1	Eclair	2010/1	7
2.2	Froyo	2010/5	8
2.3.x	Gingerbread	2010/12	9~10
3.x	HoneyComb	2011/2	11~13
4.0.x	Icecream Sandwich	2011/11	14~15
4.1, 4.2, 4.3	Jelly Bean	2012/6~2013/10	16~18
4.4.x	KitKat	2013/10	19~20
5.0, 5.1	Lollipop	2014/10	21~22
6.0	marshmallow	2015/5	23
7.0, 7.1	nougat	2016/9	24~25
8.0, 8.1	Oreo	2017/9	26~27
9.0	Pie	2018/8	28