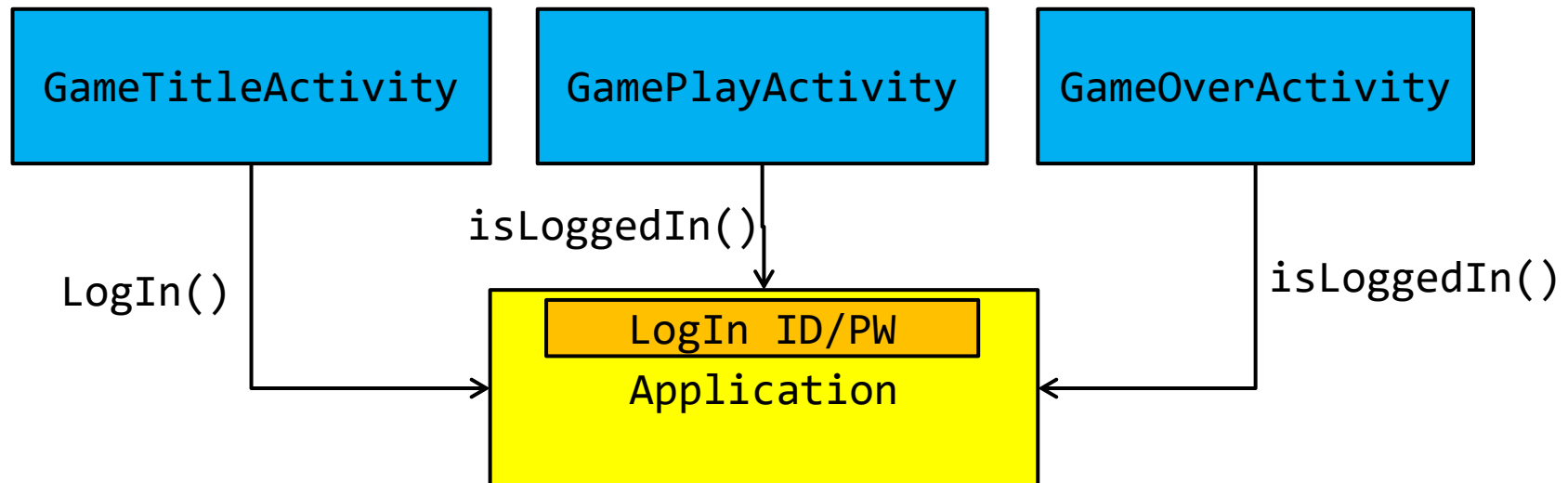


안드로이드 SDK 개발 플랫폼 활용하기

APPLICATION 클래스 활용하기

● Application Class

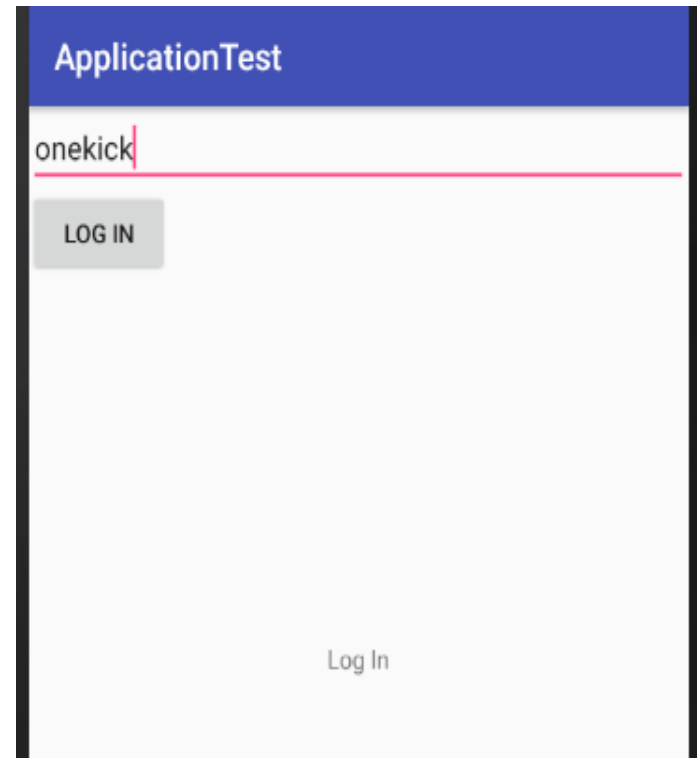
- VM이 사라지지 않는 한 **가장 먼저 생성되고 마지막까지 살아있는 클래스**
 - Activity는 수시로 생성, 소멸되므로 공통 정보는 여기에 맡기자.
- 모든 Activity들은 Application에 언제든지 접근 가능
- 공동으로 사용하는 데이터는 Application에게 맡기자.
 - 로그인정보, 데이터베이스 연결정보, 네트워크 통신 소켓 등을 맡김



- <Application 클래스 사용 순서>
 1. Application 클래스를 extends하는 클래스를 생성
 - android.app.Application Class 상속 받아 사용
 2. AndroidManifest 파일 열기
 - AndroidManifest.xml 에 있는 <application> 의 name 속성에 클래스 등록
 3. 모든 Activity에서는 언제나 Application에 접근 가능
 - Context는 getApplicationContext() 메소드 제공
 - 단, 형변환을 해서 사용해야 함
 - 예시) `MyApp myApp = (MyApp) getApplicationContext();`

- Application class 상속 시 Override 해야 하는 메소드
 - onCreate() : 앱 생성될 때 호출
 - 모든 상태변수와 리소스의 초기화는 이곳에서 구현
 - onLowMemory()
 - 시스템의 리소스가 부족할 때 처리할 일 구현
 -
 - onConfigurationChanged()
 - 컴포넌트가 실행되는 동안 기기의 설정 정보가 변경될 때 시스템에서 호출

- 여러 개의 화면이 어플리케이션 클래스를 통해 데이터를 공유하게 한다.
 1. 로그인 화면 구현
 2. 한 화면에서 한번 로그인
 - 다른 모든 화면들은 로그인 표시
 3. 로그인 정보
 - application 클래스 객체에 저장
 - intent에 정보를 담지 않는다.



- main_activity.xml -1/2

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="안드로이드 APK 파일의
스키마 경로는 생략합니다."
    xmlns:app="안드로이드 APK 파일의 스키마 경로는 생략"
    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="Log In"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

- main_activity.xml 2/2

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Log In"
    app:layout_constraintTop_toBottomOf="@id/editText"
    />

</android.support.constraint.ConstraintLayout>
```

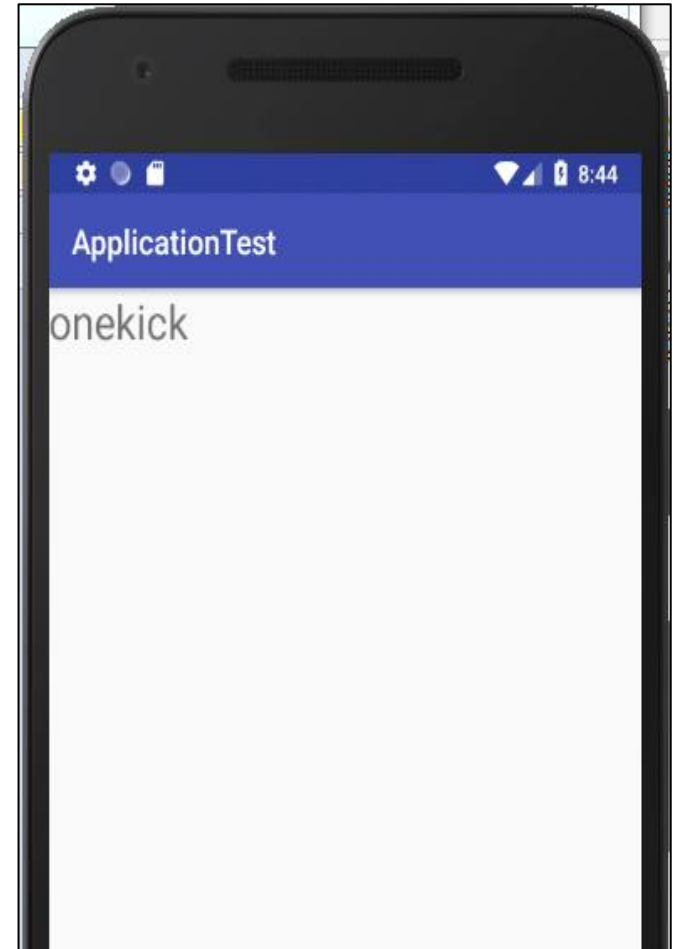
- sub_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="안드로이드 APK 파일의
스키마 경로는 생략합니다."

    xmlns:app="안드로이드 APK 파일의 스키마 경로는 생략"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SubActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ID"
        android:textSize="28dp"
    />
</android.support.constraint.ConstraintLayout>
```


- 여러 개의 액티비티를 구현
 - 각 화면들은 로그인 정보를 Application 클래스 객체에게 요청
 - 로그인된 아이디를 화면 표시
- 오른쪽 화면
 - 로그인 기능을 구현하기 위한 결과 화면
 - 로그인이 성공 시
 - 사용자의 아이디를 TextView에 표시



- MainActivity.java

```
package com.iot.applicationtest;

import android.app.Application;
import android.content.Intent; // 인텐트 활용
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View; // 뷰계열 최상위 클래스
import android.widget.Button; // 버튼 뷰 사용
import android.widget.EditText; // 글자 입력창

public class MainActivity extends AppCompatActivity {
    private EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); 표시
        editText = (EditText)findViewById(R.id.editText);
```

- MainActivity.java

```
Button button = (Button)findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        String id = editText.getText().toString();

        Application app = getApplication();
        MyApp myApp = (MyApp) app;           // 형변환
        myApp.setId(id);

        Intent intent = new Intent(MainActivity.this, SubActivity.class);
        startActivity(intent);
    }
});
}
```

- Applicaton을 상속 받아 MyApp 클래스 구현
 - 어플리케이션 객체는 유일하게 생성
 - 매니페스트 파일의 **application name**에 반드시 등록해야 함

```
package com.iot.applicationtest;
import android.app.Application;
import android.util.Log; // 로그 출력 목적

public class MyApp extends Application {
    private String id = null;
    @Override
    public void onCreate() {
        super.onCreate();
        Log.i("test", "Application onCreate() called");
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
}
```

- 두 번째 화면
 - 아이디를 표시하는 화면
 - application 객체에 접근해서 데이터를 얻어오기

```
package com.iot.applicationtest;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import org.w3c.dom.Text;

public class SubActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sub);

        String id = ( (MyApp)getApplication() ).getId(); // 한 줄로 처리
        ((TextView)findViewById(R.id.textView)).setText(id);
    }
}
```

- 화면의 순서를 바꾸는 앱을 구현
 - 새로운 액티비티를 생성한 경우 반드시 이름을 등록해야 함

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="안드로이드 APK 파일의 스키마 경로는 생략합니다."
    package="com.iot.applicationtest">
    <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"
        android:name=".MyApp"
    >
        <activity android:name=".MainActivity">
```

- 여러 화면에서 Application을 통해 데이터를 공유할 수 있는지 확인해보자
 - Application에 그 밖에 여러 데이터를 추가
 - 이를 맡기고 찾아와 표시하는 기능을 더 추가