

**모바일 소프트웨어 설계**

HOMEWORK #2

학부: 정보통신공학과

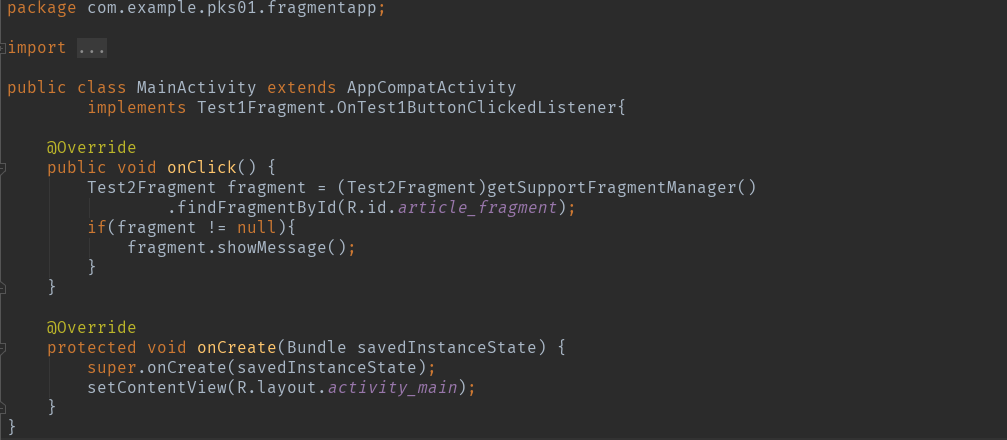
학년: 4학년

성명: 박광석

학번: 12141693

**< 실 습 >**

**1. MainActivity**

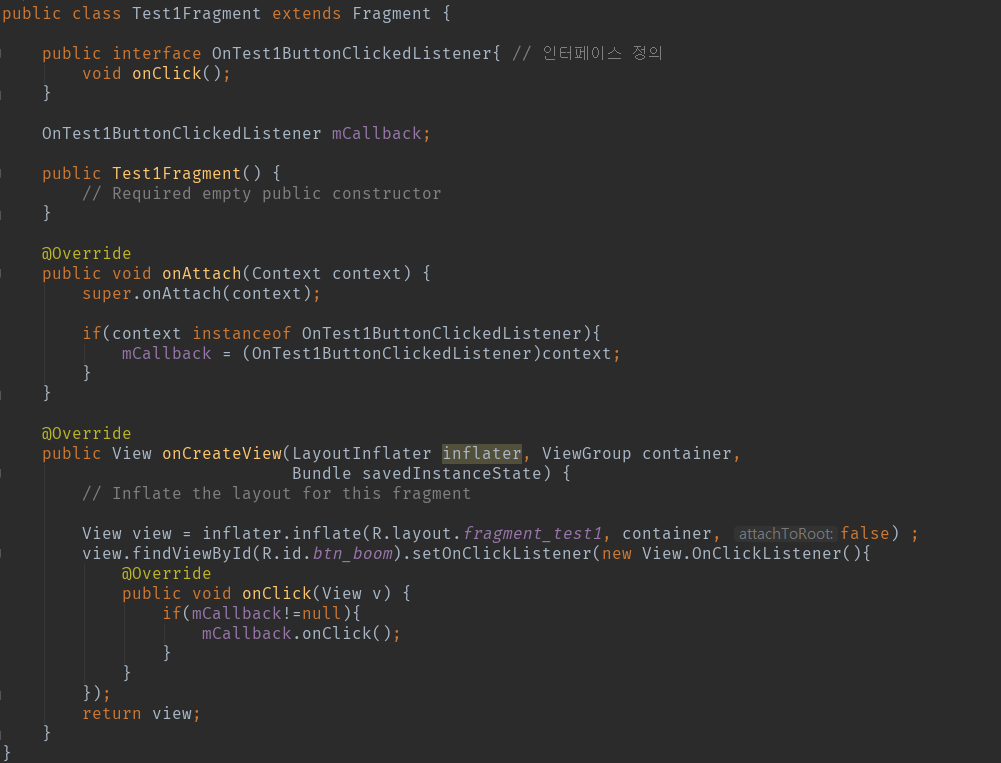


Fragment는 Activity에서 UI의 일부를 나타내고 여러 Activity에서 Fragment를 재사용 할 수도 있다. Fragment는 Activity의 모듈이라고 생각할 수 있고, Activity가 동작하는 동안 추가하고 제거할 수 있다. 그리고 하나의 Fragment에서 다른 Fragment와 통신할 때는 반드시 부모 Activity를 통해서 해야 한다. 직접 통신은 불가능하다.

Fragment와 Activity와 통신하기 위해서는 Fragment 클래스에 Interface를 정의해야 하며, Fragment로부터 이벤트 Callback을 받기 위해서, 부모 Activity에서 Interface를 구현해야 하고 내용은 위와 같다. Test1Fragment 클래스에서 정의한 OnTest1ButtonClickedListener Interface를 implements 하고 Test1Fragment OnTest1ButtonClickedListener Interface에서 선언한 onClick() 함수를 override 정의했다.

getSupportFragmentManager().findFragmentById(R.id.article\_fragment)를 통해 activity\_main.xml에서 선언한 Fragment를 불러오고 Test2Fragment 인스턴스를 하나 생성했다. 그리고 Test2Fragment에서 정의한 showMessage() 함수를 호출하여 알림창이 나오도록 하였다.

**2. Test1Fragment**



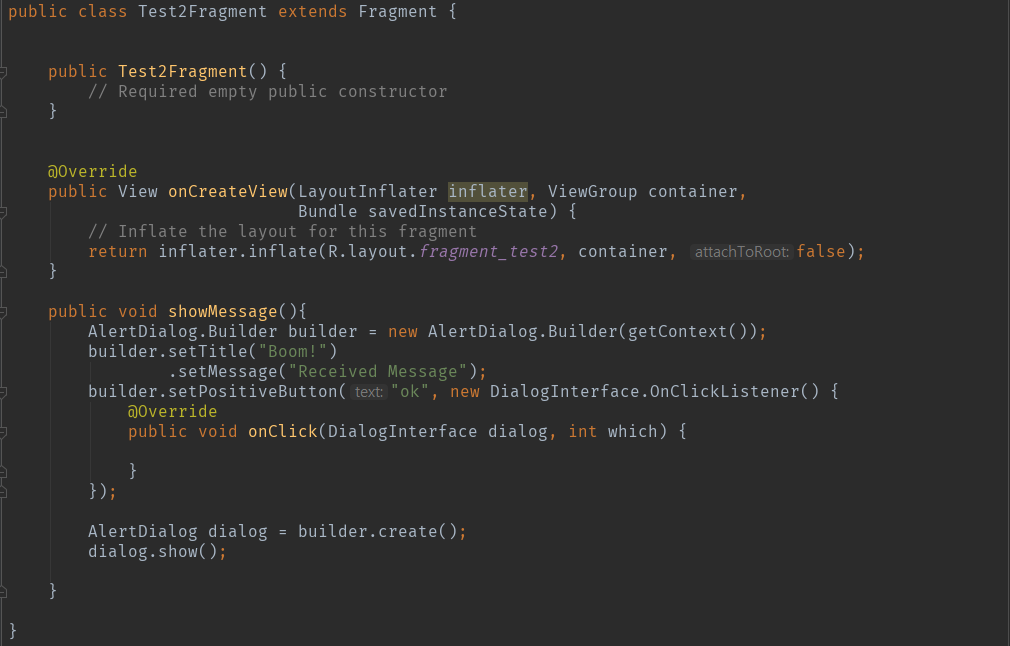
Fragment와 Activity와 통신하기 위해서는 Fragment 클래스에 Interface를 정의해야 하며, OnTest1ButtonClickedListener라는 Interface를 만들었다. 그리고나서

Fragment의 UI를 만들거나 제거하기 위해 Activity와 상호작용하는 기존의 Fragment 클래스의 Callback 함수(life cycle 함수)들을 override 해야 한다.

onAttach() 함수는 Fragment가 Activity에 연결될 때 호출되고 Activity 객체가 전달된다. OnTest1ButtonClickedListener 인스턴스를 생성 후 Activity 객체인 context를 저장했다. Activity와 통신할 땐 저장한 mCallback Interface 함수를 호출하면 된다.

OnCreateView() 함수는 Fragment의 view를 생성할 때 호출된다. 그리고 반드시 Layout을 초기화 해야 한다. Inflate 함수는 xml에 정의해 놓은 레이아웃파일을 View Object로 바꿔주는 역할을 한다. 그리고 id로 view를 찾은 후 클릭에 대한 이벤트 처리를 해주었다. onClick() 함수가 실행될 것이며 함수 내용은 MainActicity에서 이미 정의하였다.

**3. Test2Fagment**

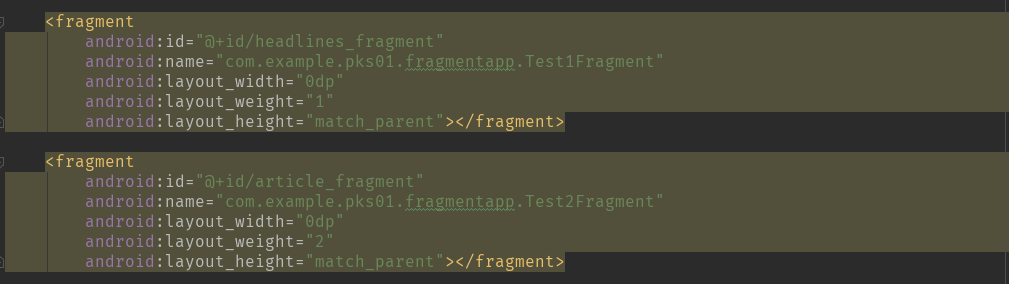


MainActivity에서 호출한 showMessage() 함수 정의한 부분이다.

Title과 Message을 Dialog를 통해 보여주는 내용이다.

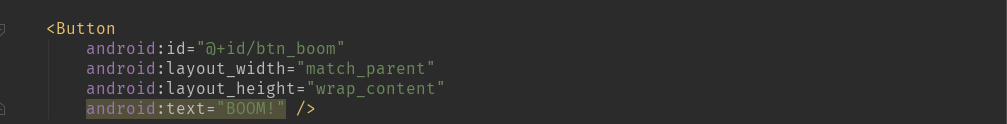
Ok 버튼 클릭 시 이벤트 처리는 처리해주지 않아서 클릭 시 창만 사라진다.

**4. activity\_main.xml**



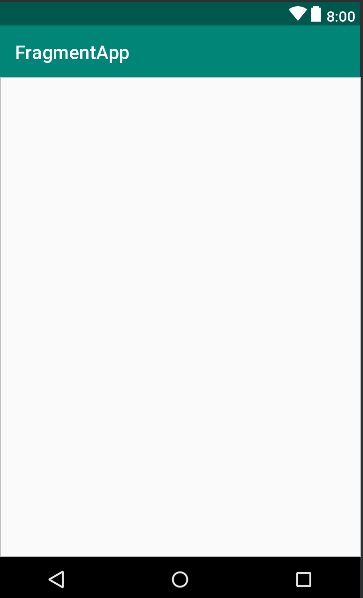
Fragment는 재사용 가능한 modular UI이지만, 각각의 Fragment 인스턴스는 부모 Activity인 MainActivity와 연결되어야 한다. name attribute를 통해서 2개의 Fragment와 연결하고 각각 Activity와 연결하였다.

**5. fragment\_test1.xml**

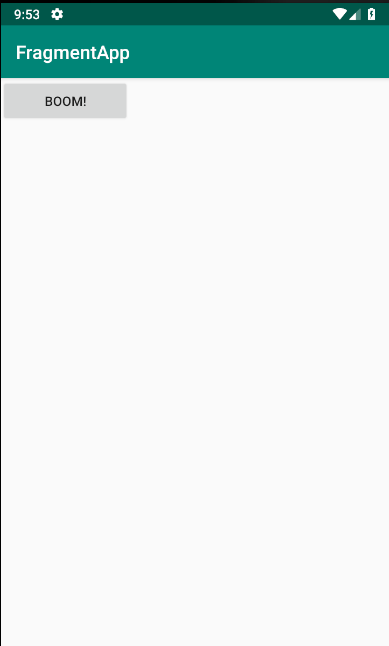
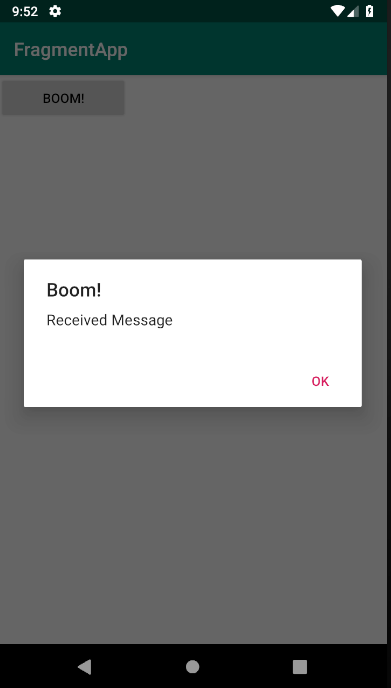


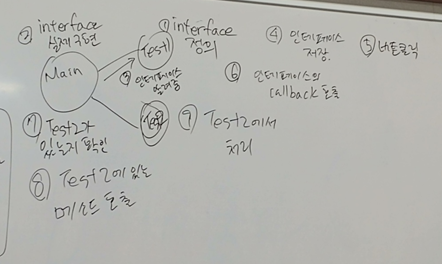
버튼 한 개를 생성하였고 이는 첫 번째 Fragment에 출력될 것이다.

**6. fragment\_test2.xml**

해당 xml은 알림창만 나타날 것이기 때문에 view는 아무것도 선언하지 않았다.

**7. Result**



내용을 정리하면 아래와 같다.

1. Fragment에서 Interface 정의

2. 부모 Activity에서 Interface 실제 기능 구현

3. onAttach() 함수를 통해 Interface(context)를 Fragment에게 알려준다.

4. onAttach() 함수에서 Interface(mCallback)를 저장한다.

5. 저장한 Interface(mCallback) 함수를 호출한다.

6. Interface의 Callback을 호출한다.

7. 다른 Fragment가 있는지 확인한다.

8. 해당 Fragment의 Method를 호출한다.

9. 해당 Fragment에서 정의한 함수 처리 및 실행한다.