

Mobile Software

Everything of GoogleMap

3조 급한사람들



INDEX

Introduce of Googlemap

Install of Googlemap

Evolution of Googlemap

QnA



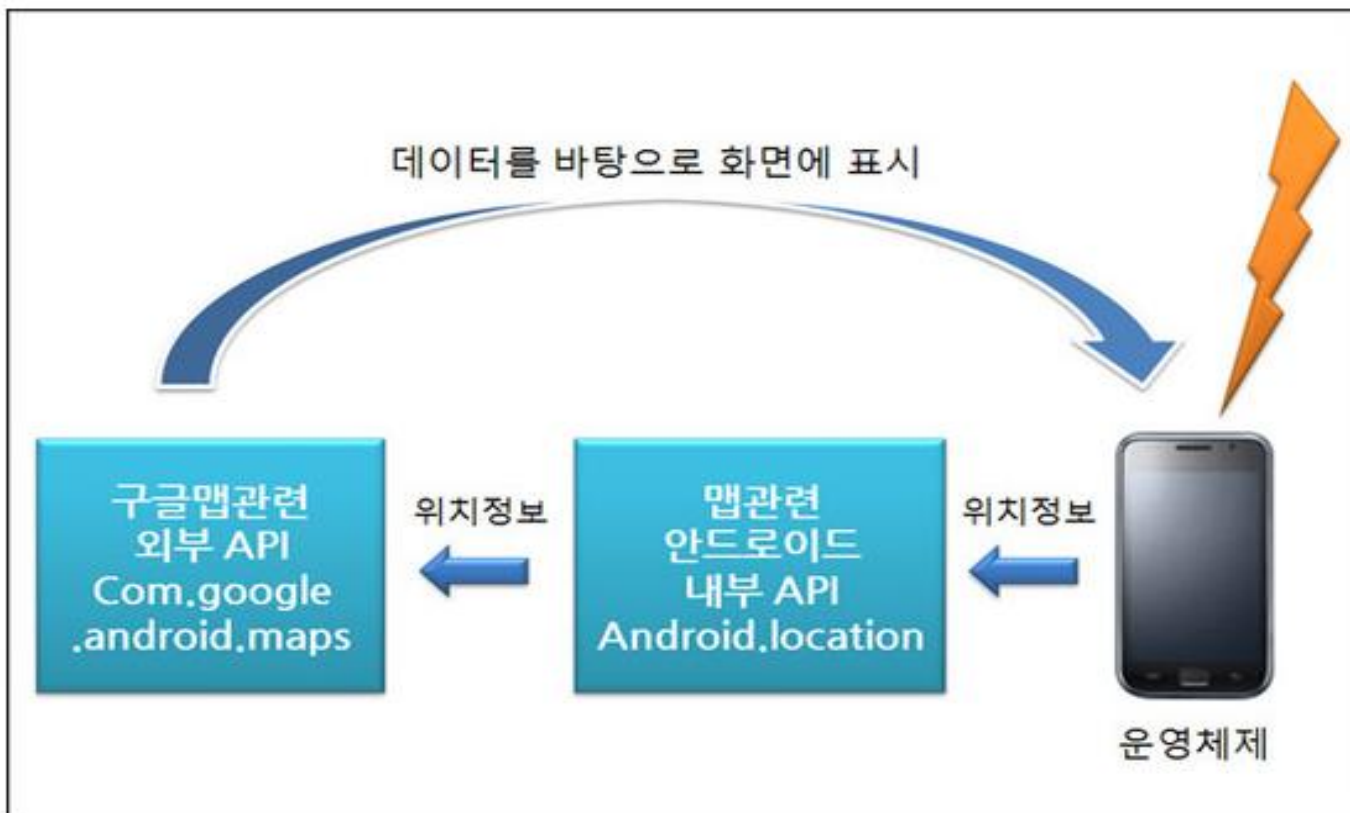
A modern, white, egg-shaped chair with a wide, flat base sits on a light-colored wooden floor. The chair's interior is lined with bright orange cushions, including a backrest and a seat. The background is a solid, muted olive-green wall. The entire scene is framed within a rounded rectangle.

Introduce of Googlemap

01



(1) 위치기반 서비스 처리의 전체구조 및 처리 절차



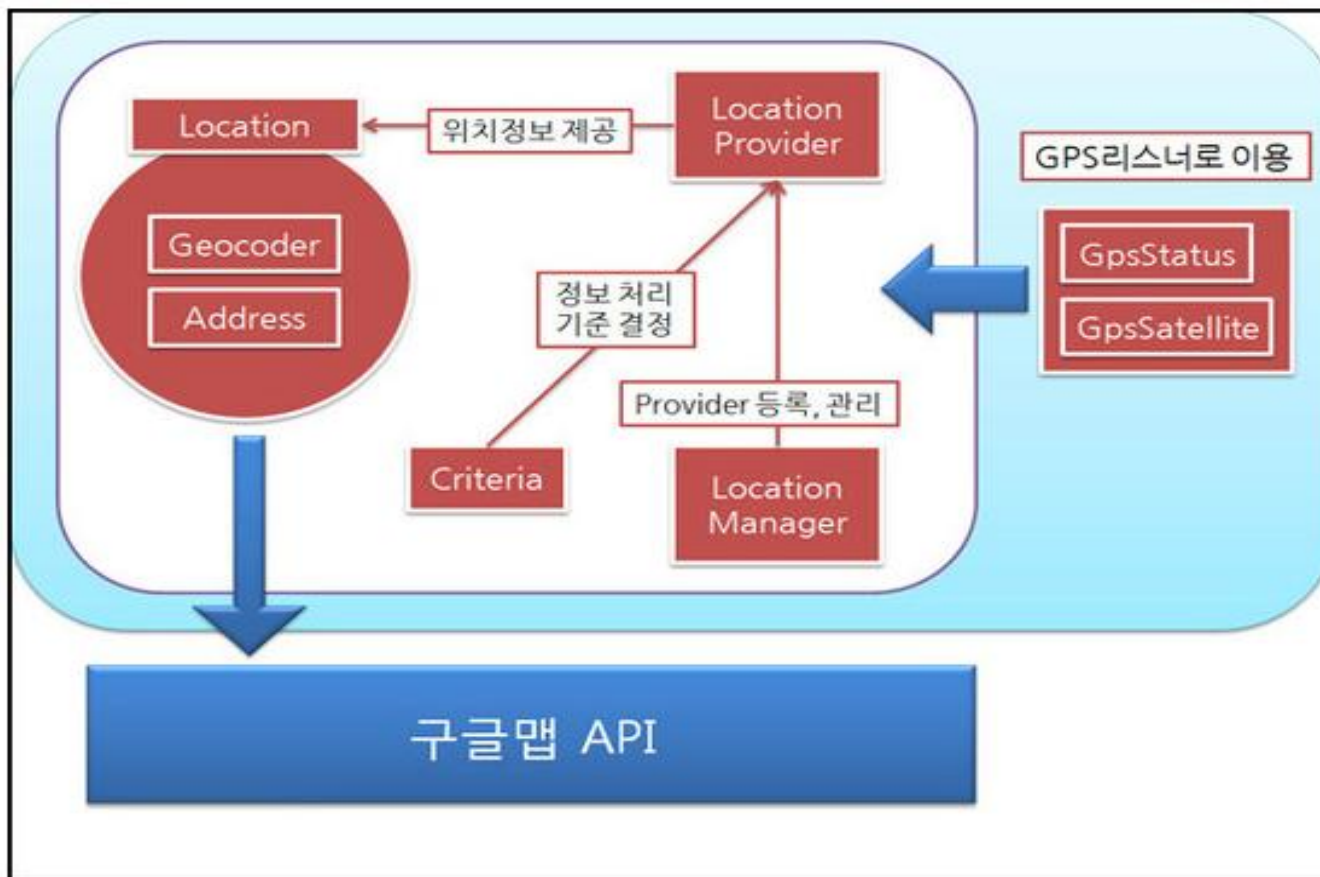


- 1) 기기가 GPS에 대한 정보를 받음
- 2) 받은 위치 정보를 안드로이드 내부 API를 통해서
Android.location 패키지에 있는 클래스로 가공함
- 3) 가공된 정보를 구글맵 API를 통해서 재 가공 함
- 4) 재 가공된 정보를 구글맵 API를 통해서 재 가공함





(2) 안드로이드 내부 API - android.location





- (1) 안드로이드 API 문서 위치 : 안드로이드 설치폴더
WdocsWreferenceWclasses.html
- (2) 내부 API는 총 8개의 클래스(location, geocoder, adress, location provider, criteria, location manager, gpsstatus, gpssatellite) 와 3개의 인터페이스(GPS.Listener, GpsStatus.NmeaListener, LocationListener)로 구성되어 있음





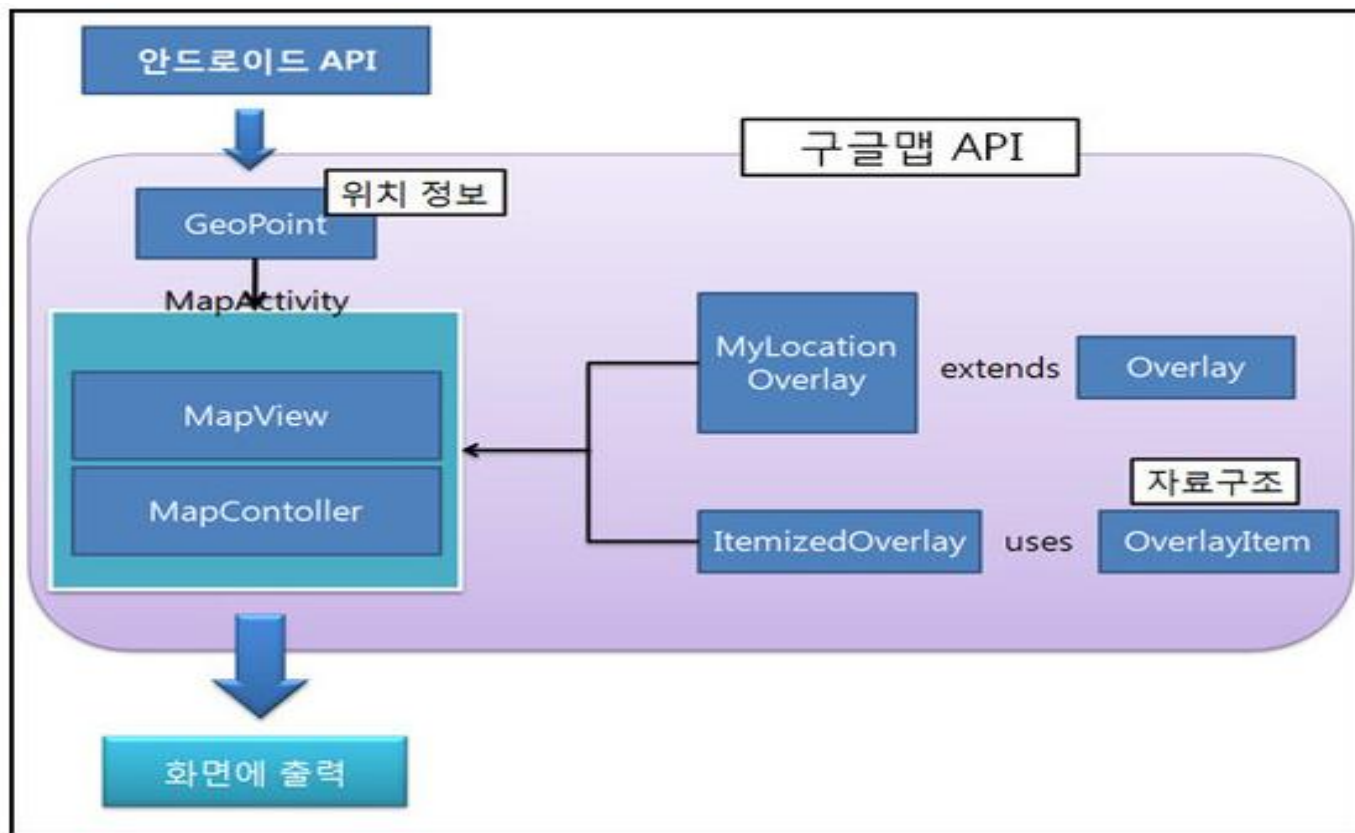
(3) Class 설명

- 1) Location manager : 위치 정보를 제공해주는 제공자 (Location provider) 관리역할을 함
- 2) Criteria : 정보처리 결정 기준
- 3) Location Provider : Location형(ex - Gps)의 위치정보를 제공하고 location에는 위도,경도,정확도,속도,고도,방향 등이 제공되는데 주로 다루는 것은 위도와 경도임
- 4) Address : Location Provider에서 받은 위도,경도를 통해서 Geocoder를 거쳐 주소지를 얻음





(3) 구글맵 API : Com.google.android.maps





- (1) 구글맵 API 문서의 위치 : 안드로이드 설치폴더
 `Wadd-onsWaddon_google_apis_google_inc_10`
 `WdocsWreferenceWindex.html`
- (2) 구글맵 API는 10개 클래스(GeoPoint ItemizedOverlay
 <Item extends OverlayItem>, MapActivity, MapController
 MapView, MapView.LayoutParams, MyLocationOverlay,
 Overlay, OverlayItem, TrackballGestureDetector)와 3개의
 인터페이스(ItemizedOverlay.OnFocusChangeListener,
 Overlay.Snappable Projection)가 존재





(3) 구글맵에서는 너무 많은 정보를 가지고 있어 좌표를 location 그대로 사용하지 않고 위도,경도 좌표만 가지고 있는 GeoPoint를 이용

(4) MapView를 통해 맵이 담긴 뷰를 다루고, Mapcontroller를 통해 맵의 줌이나 맵으로 보여 줄 위치를 지정하고 Map Activity를 사용하여 이 뷰를 표시

(5) 맵에 뭔가를 표시할 때는 Overlay를 사용





@ 참고문헌

- * Naver Blog : 녹차의 공장
(<http://blog.naver.com/crowdark7>)





Install of Googlemap

02



1. 구글맵 띄우기

(1) 안드로이드 프로젝트를 생성

1) File – Android Project

(2) 구글맵 API 발급 받기

1) MD5 문자열 생성하기

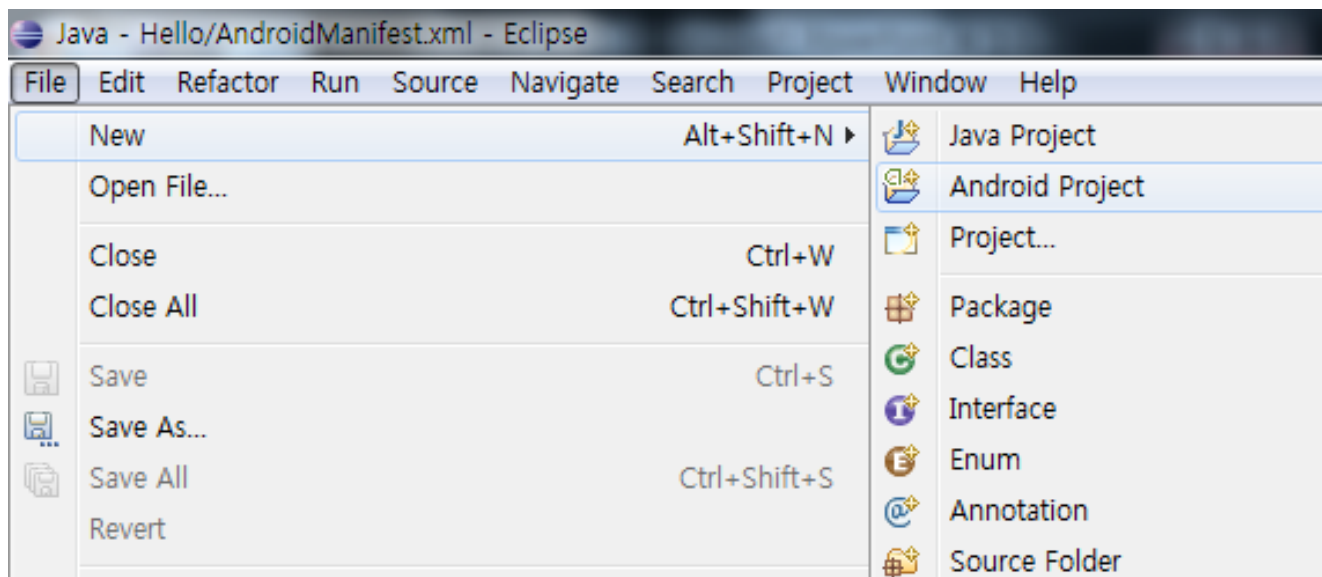
(3) 소스코드 작성

(4) 예제화면





1) File – Android Project





* 주의 점 : 프로젝트 생성 과정에서 Platform에 맞는 Google APIs를 선택해준다.(2번째사진)

The image displays three sequential screenshots of the 'New Android Project' wizard in Android Studio, illustrating the steps to create a new project.

Step 1: Create Android Project
Select project name and type of project.
Project Name: Mapview1
Options: ☒ Create new project in workspace, ☐ Create project from existing source, ☐ Create project from existing sample
☒ Use default location
Location: C:/Users/Java/workspace/Mapview1
Working sets: ☐ Add project to working sets
Working sets: [Dropdown] [Select...]

Step 2: Select Build Target
Choose an SDK to target
Build Target table:

Target Name	Vendor	Platform	API ...
☐ Android 2.1	Android Open Source Project	2.1	7
☒ Google APIs	Google Inc.	2.1	7

Android + Google APIs

Step 3: Application Info
Configure the new Android Project
Application Name: Mapview1
Package Name: com.Mapview1
☒ Create Activity: Mapview1Activity
Minimum SDK: 7
☐ Create a Test Project
Test Project Name: Mapview1Test
Test Application: Mapview1Test
Test Package: com.Mapview1.test

여기까지 프로젝트 생성 완료!

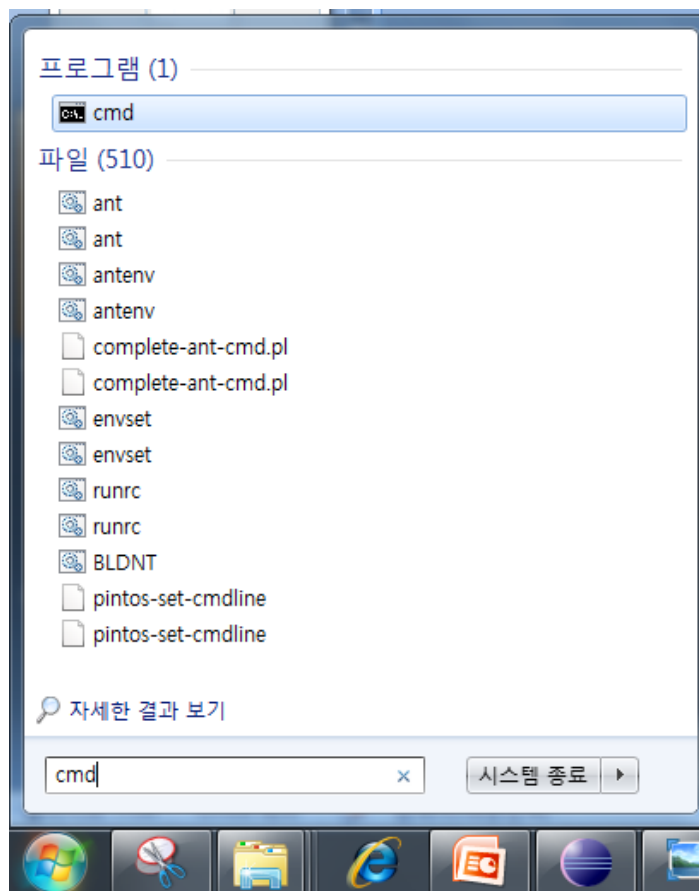




구글 맵 API를 발급받기 위해서는 MD5라는 .android에 생성된 문자가 필요하다.

방법은 다음과 같다.

시작에서 cmd를 쳐서 실행시킨다



Cmd창에서 내 컴퓨터에 있는 .android 폴더로 갑니다.

저 같은 경우에는 c:\Wandroid\W.android의 위치입니다.



* 윈도우 7기준



CMD 창에서 `keytool -list -alias androiddebugkey -v -keystore debug.keystore -storepass android -keypass 입력`

```
C:\> 관리자: C:\windows\system32\cmd.exe

일련 번호: 640fa2dd
적합한 시작 날짜: Mon Mar 19 15:23:05 KST 2012, 종료 날짜: Wed Mar 12 15:23:05 K
ST 2042
인증서 지문:
MD5: EF:DB:3B:4E:7C:B8:9A:11:9C:02:60:14:E0:1...
SHA1: 5A:01:B2:E6:53:5B:F9:F0:69:BA:DF:4D:F6:A5:5C:01:77:65:09:7D

버전: 3

확장:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 89 DB CF 36 66 8B 58 88    C7 6B 8B 4E DE A5 B1 07    ...6f.X...k.N....
0010: 34 E6 52 42                      4.RB
]
]

C:\android\android>
C:\android\android>^U
```

* MD5 발급 받은것을 복사하여 메모장에 옮겨두자

* MD5 발급시 기타 에러사항





MD5까지 받았으면 아래 웹사이트에 들어가
<http://code.google.com/intl/ko-kr/android/maps-api-signup.html>

구글 맵 API를 발급받는다.

체크박스에 체크를 하고 화살표에 발급 받은 MD5키를 넣어준다.



Sign Up for the Android Maps API - Android Maps API - Google Code - Windows Internet Explorer

<http://code.google.com/intl/ko-kr/android/maps-api-signup.html>

즐거찾기 | 추천 사이트 | 웹 조각 갤러리

Sign Up for the Android Maps API - Android M...

for more information about application signing. To get a Maps API key for your certificate, you will need to provide its Linux or Mac OSX, you would examine your debug keystore like this:

```
$ keytool -list -keystore ~/.android/debug.keystore
...
Certificate fingerprint (MD5): 94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98
```

If you use different keys for signing development builds and release builds, you will need to obtain a separate Maps API key and the corresponding certificate.

You also need a [Google Account](#) to get a Maps API key, and your API key will be connected to your Google Account

this document (the "Maps APIs Terms"); 2) the Legal Notices (http://www.google.com/intl/en-us/help/legalnotices_maps.html); and 3) the Privacy Policy (<http://www.google.com/privacy.html>). Before you use the Maps APIs, you should read each of the documents comprising the Terms, and print or save a local copy for your records.

1.3. If you use the Maps APIs in conjunction with any other Google products or services, including any other Google API, (collectively, the "Services"), your agreement with Google will also include the terms applicable to those Services. All of these are referred to as the "Additional Terms." If Additional Terms apply, they will be accessible to you either within or through your use of that Service. If there is any contradiction between what any Additional Terms say and what the Maps APIs Terms say, then the Maps APIs Terms will take precedence only as it relates to the Maps APIs,

☒ I have read and agree with the terms and conditions ([printable version](#))

My certificate's MD5 fingerprint:

[Generate API Key](#)





아래의 박스 부분을 메모장에 옮겨 복사해둔다.



Google ?? API

[Google ?? ?](#) > [Google ?? API](#) > Google ?? API ????

????? ?? API ?? ?????? ??????.

??? ?.

```
00sFmwrNfka4Lax_9D8h32_AQJs5mrpWaM-fkYg
```

? ??

```
EF:DB:3B:4E:7C:B8:9A:11:9C:02:60:14:E0:B9:57:E0
```

??? ??? ??? ????? ?? ?? ?????????? ??? ? ?????.

??? ?? ??? ????? ??? ?? xml ??? ??????

```
<com.google.android.maps.MapView                android:layout_width="fill_parent"
android:layout_height="fill_parent"                android:apiKey="00sFmwrNfka4Lax_9D8h32_AQJs5mrpWaM-
fkYg"                                                />
```

??? ??? [API](#) ????? ??????



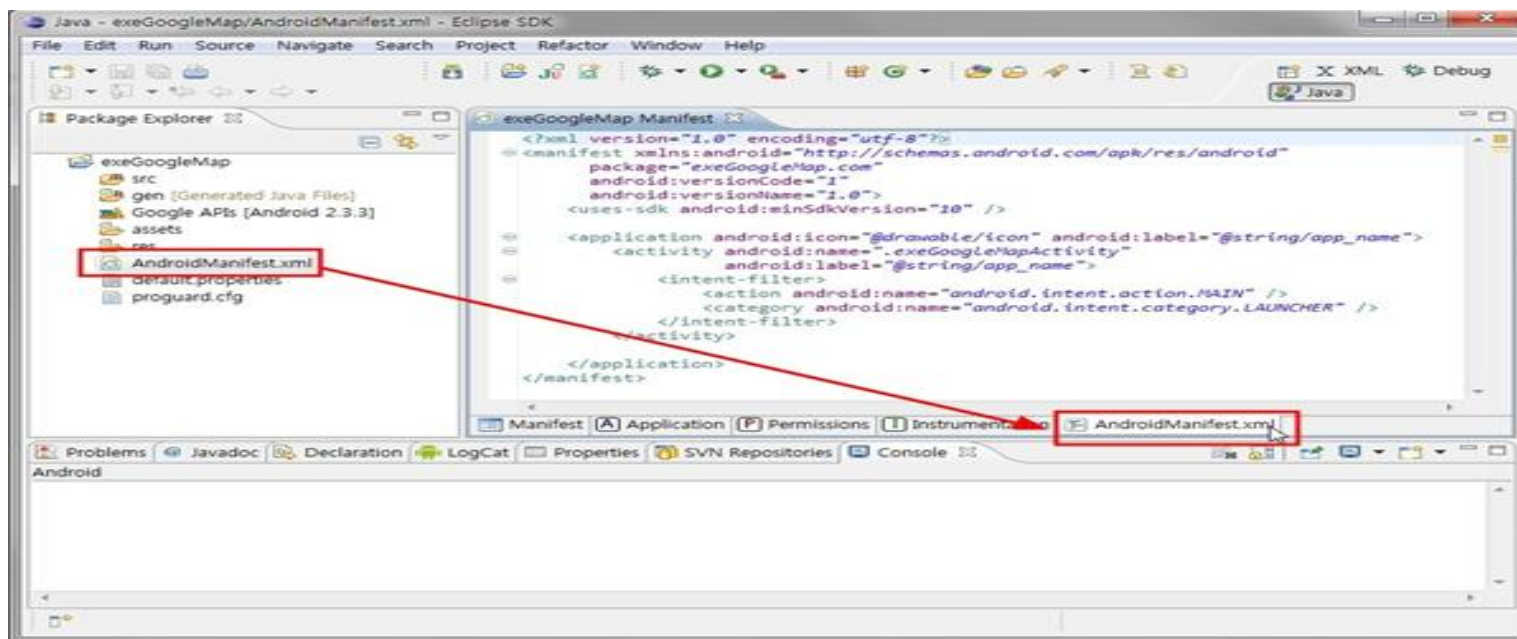
화살표에 발급받은 소스역시 메모장에 복사
이 소스가 바로 구글 맵 API(?)인 것이다.
이제 다음 단계로!





(3) 소스코드 작성

1) AndroidManifest.xml





1) 구글맵 라이브러리 등록

```
<!-- google library 등록 -->  
<uses-library android:name="com.google.android.maps" android:required="true"/>
```

구글 맵 라이브러리를 사용하기 위해 라이브러리 등록

2) 인터넷 사용권한 부여

```
<!-- 인터넷 권한 부여 -->  
<uses-permission android:name="android.permission.INTERNET"/>
```

역시 인터넷을 사용하기 위해 사용권한 부여

코드를 작성합니다.

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="exeGoogleMap.com"  
    android:versionCode="1"  
    android:versionName="1.0">  
    <uses-sdk android:minSdkVersion="10" />  
  
    <application android:icon="@drawable/icon" android:label="@string/app_name">  
        <activity android:name=".exeGoogleMapActivity"  
            android:label="@string/app_name">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
  
        <!-- google library 등록 -->  
        <uses-library android:name="com.google.android.maps" android:required="true"/>  
    </application>  
  
    <!-- 인터넷 권한 부여 -->  
    <uses-permission android:name="android.permission.INTERNET"/>  
</manifest>
```

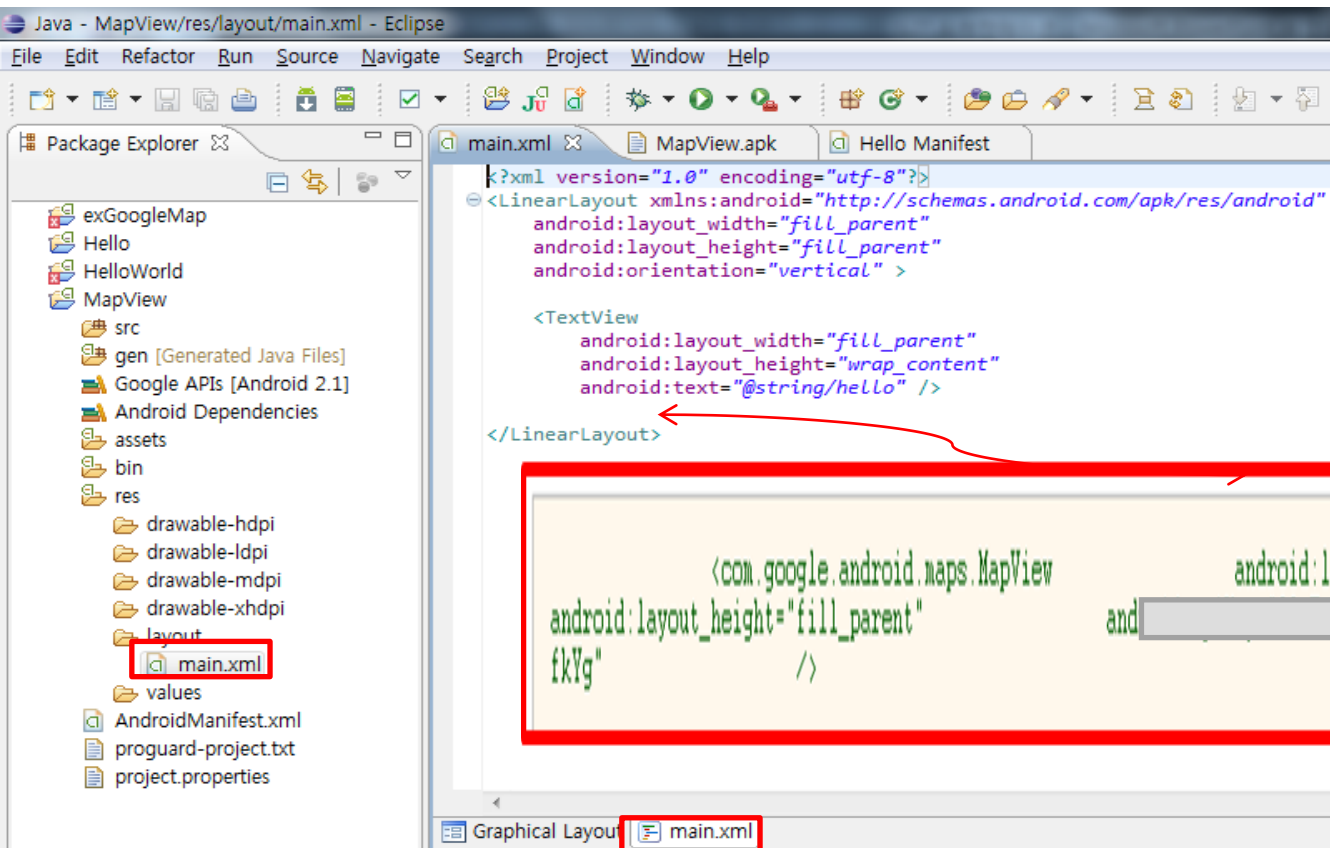
다음과 같이 추가시켜준다.





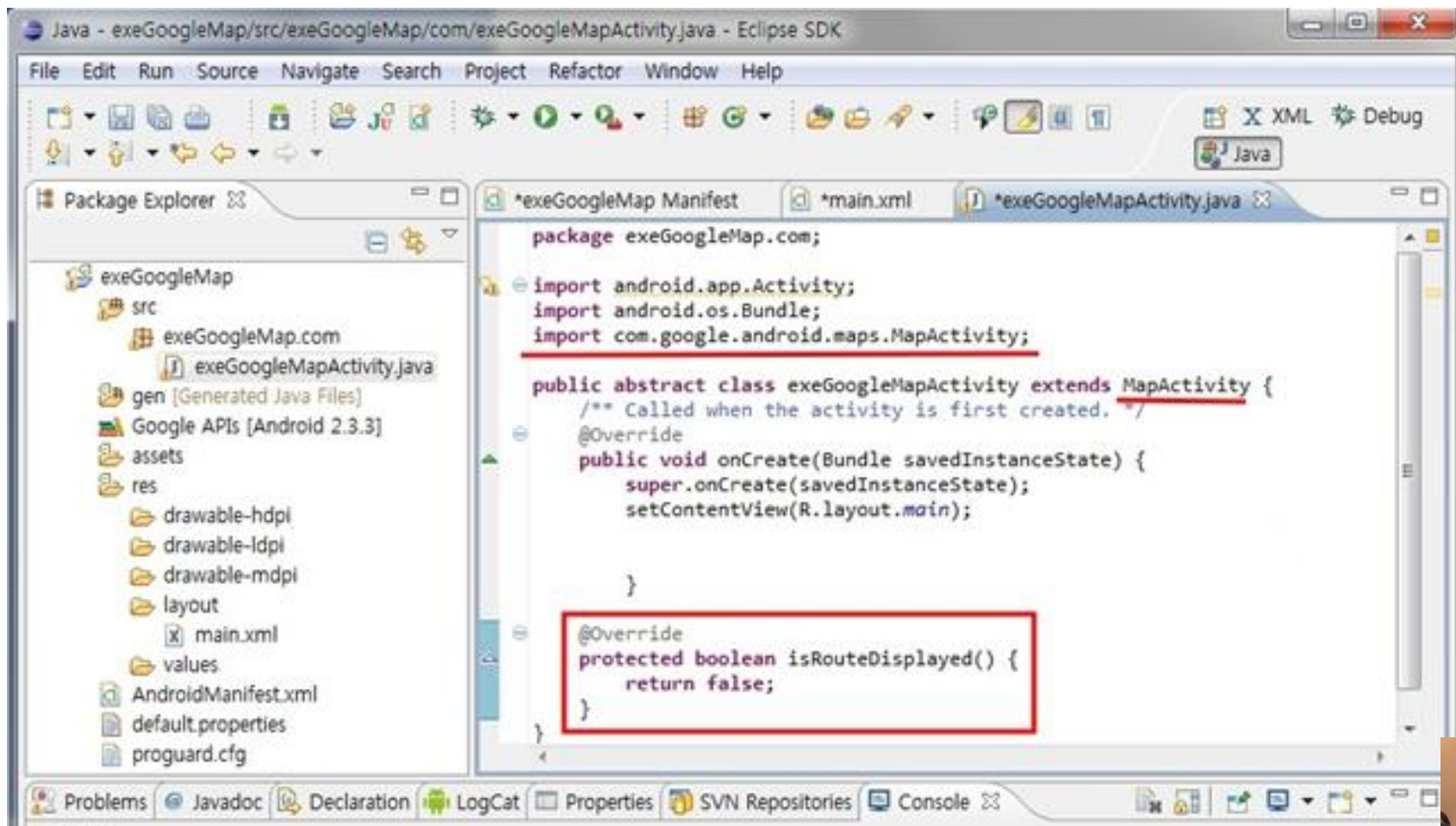
3) res/ layout/ main.xml

아까 받은 GoogleAPI를 화살표에 C. V한다.





4) JAVA코드 작성

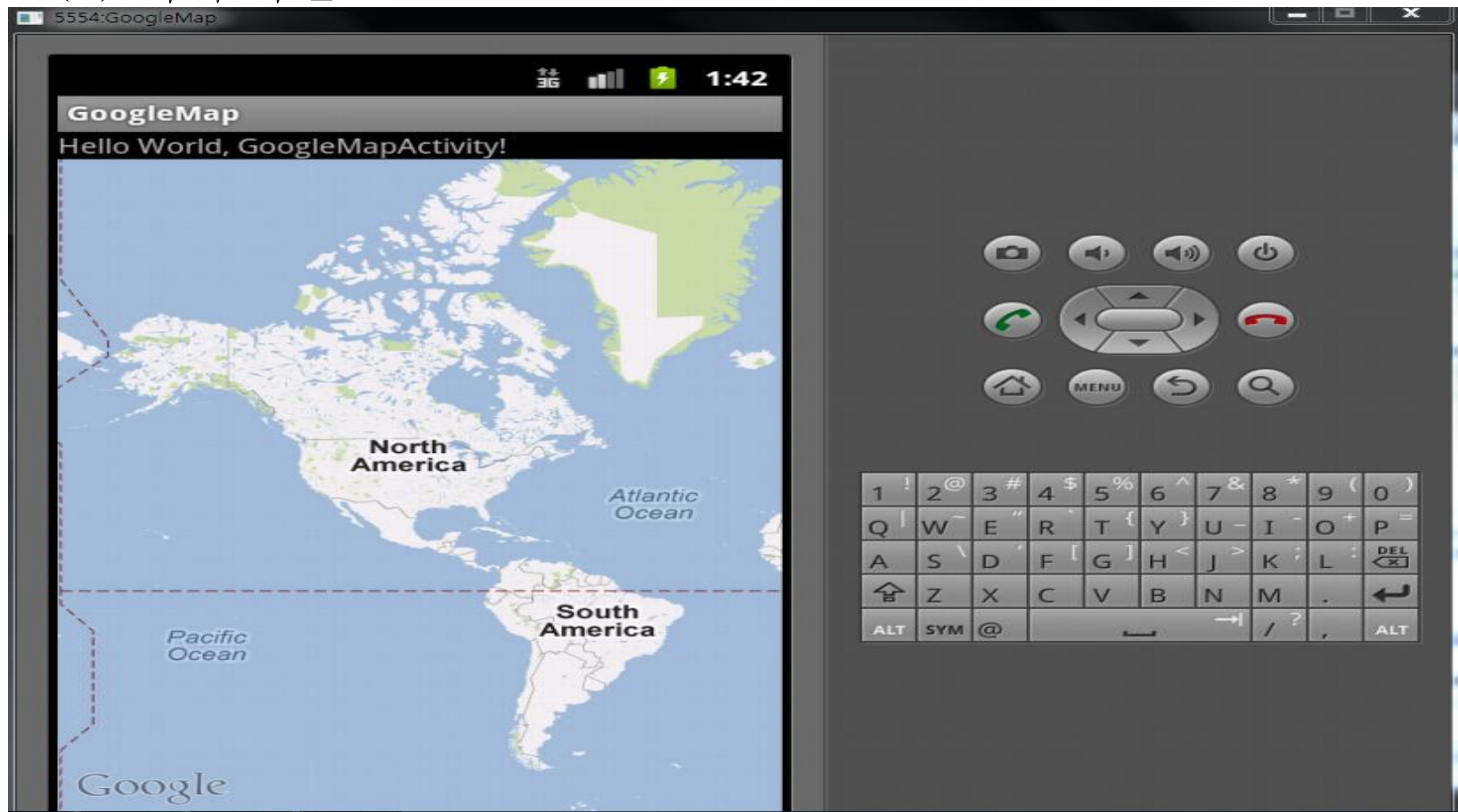


마지막으로 간단한 선언과 이름을 변경해주고 알수없는것을 protected boolean 해주면 끝! 실행창에서 확인해보자





(4) 예제 화면



따끈따끈한 예제화면! 미국을 보여주고있다. 화면은 이동이 가능하지만 아직 구글 맵이 진화 하기 전 화면이랑 확대 / 축소 및 자기 위치등은 찾아볼수가 없다.
이상으로 Install of Googlemap을 마칩니다.





* 주의 사항

MD5를 받을 때 에러 때문에 고생한 일이 많아서 따로 주의 사항을 따로 기재해보았습니다.

```
관리자: C:\windows\system32\cmd.exe

-rfc                RFC 스타일의 출력
-alias <alias>      처리할 항목의 별칭 이름
-keystore <keystore> 키 저장소 이름
-storepass <arg>     키 저장소 비밀번호
-storetype <storetype> 키 저장소 유형
-providername <providername> 제공자 이름
-providerclass <providerclass> 제공자 클래스 이름
-providerarg <arg>   제공자 인수
-providerpath <pathlist> 제공자 클래스 경로
-v                상세 정보 출력
-protected        보호되는 메커니즘을 통한 비밀번호

사용 가능한 모든 명령에 "keytool -help" 사용

C:\Windows\android>keytool -list -alias androiddebugkey -v -keystore debug.keystore -storepass android -keypass android
keytool 오류: java.lang.Exception: <androiddebugkey> 별칭이 존재하지 않습니다.
java.lang.Exception: <androiddebugkey> 별칭이 존재하지 않습니다.
    at sun.security.tools.KeyTool.doPrintEntry(KeyTool.java:1667)
    at sun.security.tools.KeyTool.doCommands(KeyTool.java:1062)
    at sun.security.tools.KeyTool.run(KeyTool.java:338)
    at sun.security.tools.KeyTool.main(KeyTool.java:331)

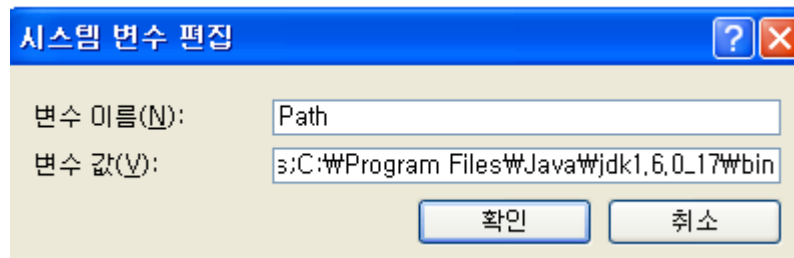
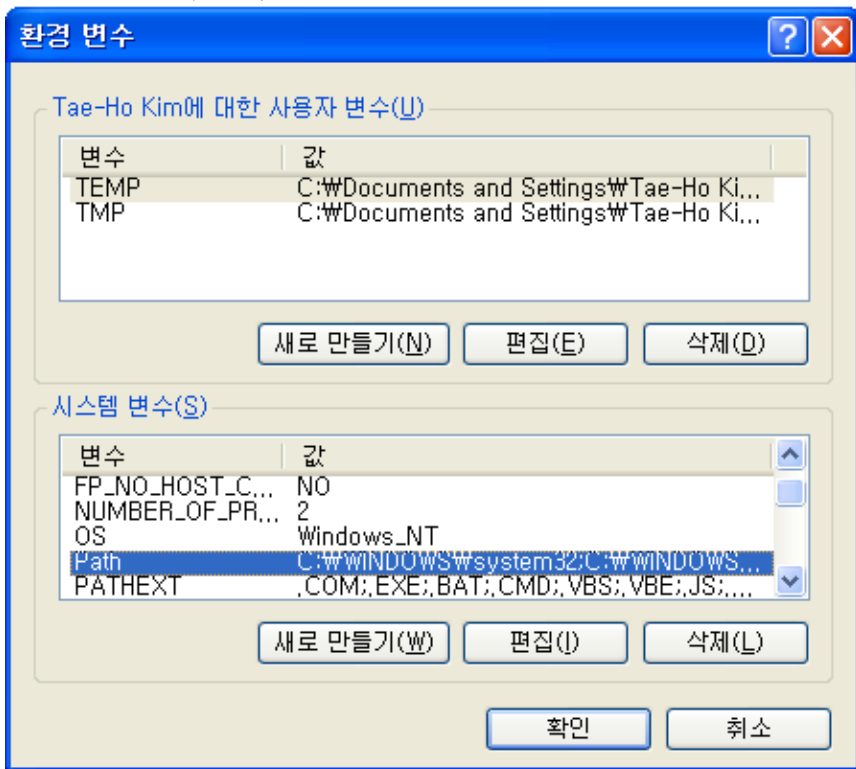
C:\Windows\android>
```

다음과 같이 별칭이 존재하지 않습니다 라는 에러일 때
아무곳에서 이 명령어를 실행하려면 PATH등록이 있어야 함
이 점을 해결하려면 내 컴퓨터 - 속성 - 고급탭의 환경 변수 클릭





* 주의 사항



첫번째 사진의 시스템 변수 목록 중 Path 항목을 선택한 후, 편집을 눌러 아래와 같이 맨 마지막 항목 각 항목을 구분지어주는 문자인 세미콜론(;)을 입력한 후, JDK의 bin 폴더의 경로를 입력한 후, 확인을 눌러주시면 됩니다.





* 주의 사항

찾다보니 다른 에러사항도 워낙 많이 있어 다 정리하지는 못하고 참고한 곳을 링크해드립니다.

<http://androidhuman.tistory.com/entry/>

구글 맵을 띄우면서 참고했던 사이트

<http://godffs.tistory.com/1393>

<http://gxgsung.blog.me/140120805038>

이제 다음으로 넘어가겠습니다.





Evolution of Googlemap

03



Evolution of Googlemap

1st evolution

확대/축소



2ed evolution

자기위치 알아내기



3th evolution

오버레이 사용하기





1. 맵의 확대/축소 기능 추가하기

(1) setBuiltinZoomControls() 메소드 활용하기

(2) ZoomIn(),ZoomOut() 메소드 활용하기





1. 맵의 확대/축소 기능 추가하기

```
Java - ZoomEx/src/com/hotack/www/ZoomExActivity.java - Eclipse
File Edit Run Source Refactor Navigate Search Project Window Help

Package Explorer
LocationEx
OutlayEx
OverlayEx
ZoomEx
src
com.hotack.www
ZoomExActivity.java
gen [Generated Java Files]
Google APIs [Android 2.1]
Android Dependencies
assets
bin
res
AndroidManifest.xml
proguard-project.txt
project.properties

ZoomExActivity.java
package com.hotack.www;

import android.app.Activity;

public class ZoomExActivity extends MapActivity {
    /** Called when the activity is first created. */
    MapView mapView;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        @Override
        protected boolean isRouteDisplayed() {
            // TODO Auto-generated method stub
            return false;
        }
    }
}
```

프로젝트의 .java 파일의 메인액티비티 클래스안에 작업을 하시면 됩니다.





1. 맵의 확대/축소 기능 추가하기

Package Explorer

- LocationEx
- OutlayEx
- OverlayEx
- ZoomEx
 - src
 - com.hotack.www
 - ZoomExActivity.java
- gen [Generated Java Files]
- Google APIs [Android 2.1]
- Android Dependencies
- assets
- bin
- res
 - drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - layout
 - main.xml**
 - values
- AndroidManifest.xml
- proguard-project.txt
- project.properties

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <com.google.android.maps.MapView
        android:id="@+id/mapview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:apiKey="0KQTz5kclGrT_P99M0nG4m5fwfLjCYAy_qHnYEQ"
        android:clickable="true"
        android:focusable="true" >
    </com.google.android.maps.MapView>
</LinearLayout>
```

앞서 main.xml에 추가한 내용에서 map을 java에서 사용하기 위해서 id를 추가로 작성해 줍니다.
예제에서의 아이디는 mapview입니다.

Graphical Layout | main.xml

Problems | Javadoc | Declaration | Console | LogCat | Progress

Saved Filters: All messages (no filters)

Search for messages. Accepts Java regexes. Prefix with pid:

L...	Time	PID	Application	Tag
------	------	-----	-------------	-----





(1) setBuiltinZoomControls() 메소드 활용하기

1) 다음 항목을 import 한다.

```
import com.google.android.maps.MapActivity;  
import com.google.android.maps.MapController;  
import com.google.android.maps.MapView;
```

2) MapView객체를 얻은후 MapView의 setBuiltinZoomControls 메소드를 호출

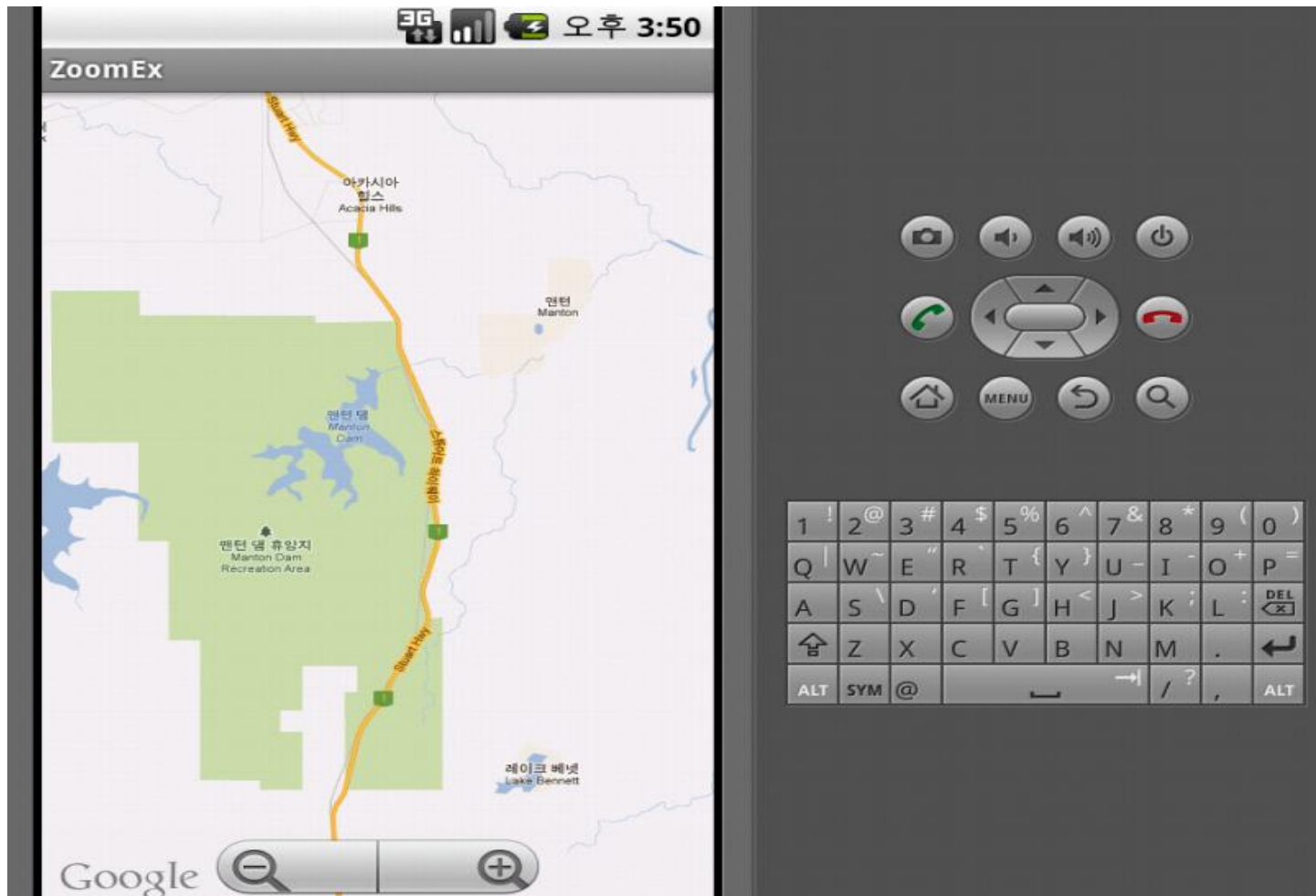
```
public class ZoomExActivity extends MapActivity {  
    /** Called when the activity is first created. */  
    MapView mapView;  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
  
        mapView = (MapView)findViewById(R.id.mapview);  
        mapView.setBuiltinZoomControls(true);  
    }  
}
```

Xml에서 작성한 구글맵의 id를 이용하여 MapView객체를 얻습니다.
그 후 mapview의 객체의 setBuiltinZoomControls 메소드를 호출합니다.





(1) setBuiltinZoomControls() 메소드 활용하기





(2) ZoomIn(),ZoomOut() 메소드 활용하기

1) 메뉴키 이벤트를 이용하여 기능 구현

1) MainActivity안에 Menu옵션을 만든다.

```
Java - ZoomEx/src/com/hotack/www/ZoomExActivity.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help

Package Explorer
LocationEx
OutlayEx
OverlayEx
ZoomEx
src
  com.hotack.www
    ZoomExActivity.java
gen [Generated Java Files]
Google APIs [Android 2.1]
Android Dependencies

*ZoomExActivity.java
main.xml

package com.hotack.www;

import android.app.Activity;

public class ZoomExActivity extends MapActivity {
    /** Called when the activity is first created. */
    MapView mapView;
    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        super.onCreateOptionsMenu(menu);
        MenuItem zoomIn = menu.add(0, 1, Menu.NONE, "지도 확대");
        MenuItem zoomOut = menu.add(0, 2, Menu.NONE, "지도 축소");
        return true;
    }
}
```

매개변수는 순서대로 그룹ID, 메뉴ID, 메뉴 정렬방식, 메뉴항목의 이름입니다.





(2) ZoomIn(),ZoomOut() 메소드 활용하기

1) 메뉴키 이벤트를 이용하여 기능 구현

2) Menu 옵션 키를 눌렀을 때 발생하는 이벤트를 만든다.

Package Explorer

- LocationEx
- OutlayEx
- OverlayEx
- ZoomEx
 - src
 - com.hotack.www
 - ZoomExActivity.java**
- gen [Generated Java Files]
- Google APIs [Android 2.1]
- Android Dependencies
- assets
- bin
- res
 - drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - layout
 - main.xml
 - values
- AndroidManifest.xml

```
public class ZoomExActivity extends MapActivity {
    /** Called when the activity is first created */
    MapView mapView;
    @Override
    public boolean onOptionsItemSelected() {
        super.onOptionsItemSelected(item);
        MapController mc=mapView.getController();
        Context context = getApplicationContext();
        Toast toast=null;

        switch(item.getItemId()){
            case 1:
                if(!mc.zoomIn()){
                    toast.makeText(context, "최대 줌 레벨에 도달했습니다.", Toast.LENGTH_SHORT).show();
                    break;
                }
                break;
            case 2:
                if(!mc.zoomOut()){
                    toast.makeText(context, "최소 줌 레벨에 도달했습니다.", Toast.LENGTH_SHORT).show();
                    break;
                }
            }
        }
        return true;
    }
}
```

Menu를 눌렀을때 호출되는 메소드

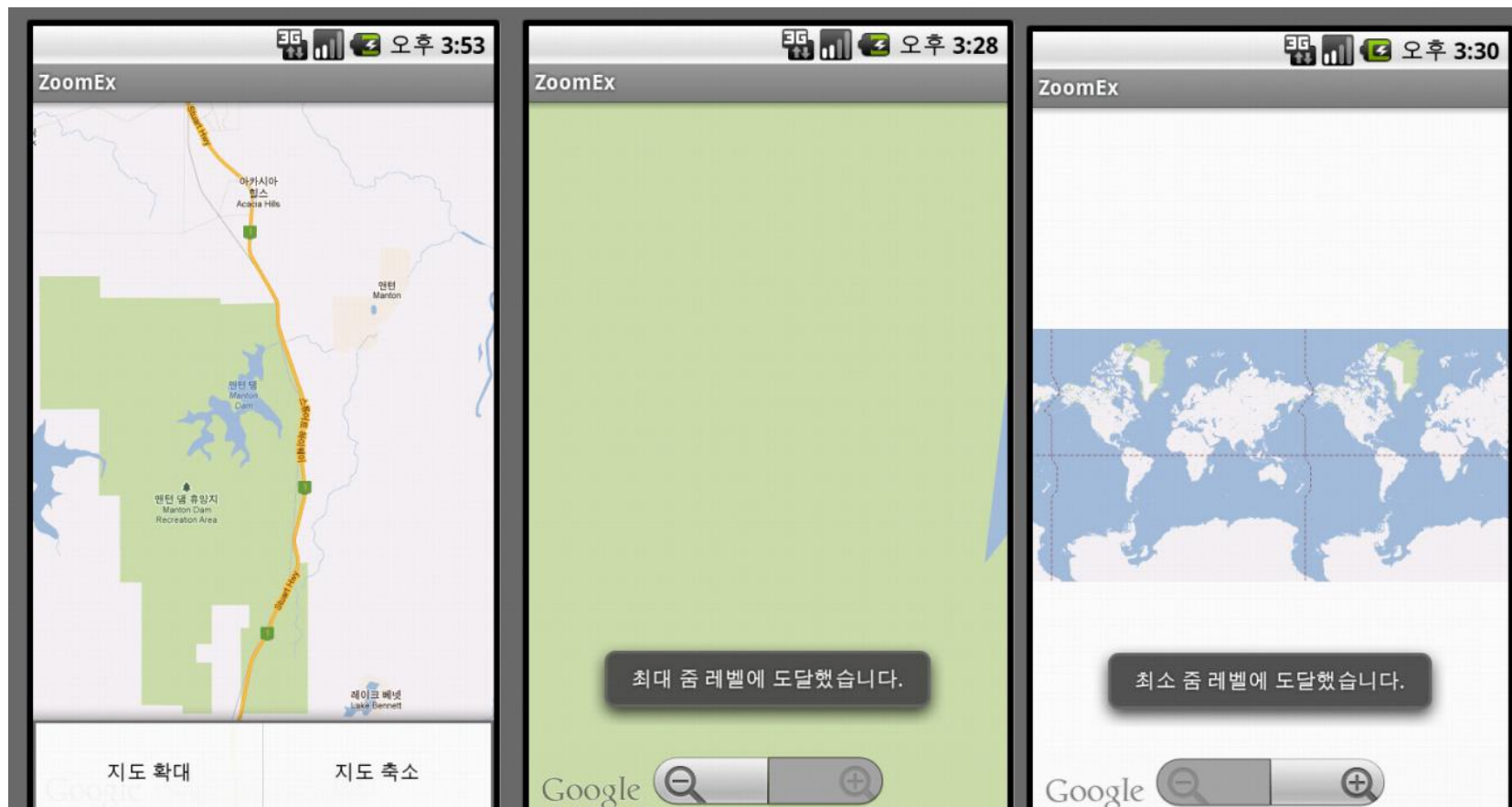
mapview의 컨트롤러를 얻음

컨트롤러의 zoomIn, zoomOut 메소드 호출





(2) ZoomIn(),ZoomOut() 메소드 활용하기





2. 자기위치 확인하기

(1) GeoCode란?

(2) GPS기능 이용하기





(1) GeoCode란?

GEOCODE(Geospatial Entity Object Code)은 특허 받은 지역 정보 변환 알고리즘과 특허지리 좌표 표현 형식을 포함하는 시스템입니다. 지구의 표면 위의 정확한 위치 또는 지리 지점의 세부 사항을 제공합니다. 위도나 경도 외의 여러 가지 정보를 가질 수 있습니다.





2. 자기위치 확인하기

Java - LocationEx/src/com/hotack/LocationExActivity.java - Eclipse

File Edit Run Source Navigate Search Project Refactor Window Help

Package Explorer

- LocationEx
 - src
 - com.hotack
 - LocationExActivity.java**
 - gen [Generated Java Files]
 - Google APIs [Android 2.1]
 - Android Dependencies
 - assets
 - bin
 - res
 - AndroidManifest.xml
 - proguard.cfg
 - project.properties
- OutlayEx
- OverlayEx
- ZoomEx

```
package com.hotack;

import android.location.*;

public class LocationExActivity extends MapActivity implements LocationListener{
    /** Called when the activity is first created. */
    private LocationManager locationManager;
    private MapController mc;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mapView=(MapView)findViewById(R.id.mapview);
        mapView.setClickable(true);
        mapView.setBuiltInZoomControls(true);

        mc=mapView.getController();
        mc.setZoom(16);
    }
}
```

프로젝트의 .java 파일의 메인액티비티 클래스안에 작업을 하시면 됩니다.





(2) GPS기능 이용하기

1) 다음 항목을 import 한다.

```
import android.location.*;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.GeoPoint;
```

2) LocationListener 인터페이스를 implements 한다.

```
public class LocationExActivity extends MapActivity implements LocationListener{
    /** Called when the activity is first created. */
    private LocationManager locManager;
    private MapController mc;
    private MapView mapView;

    Toast myToast=null;
```

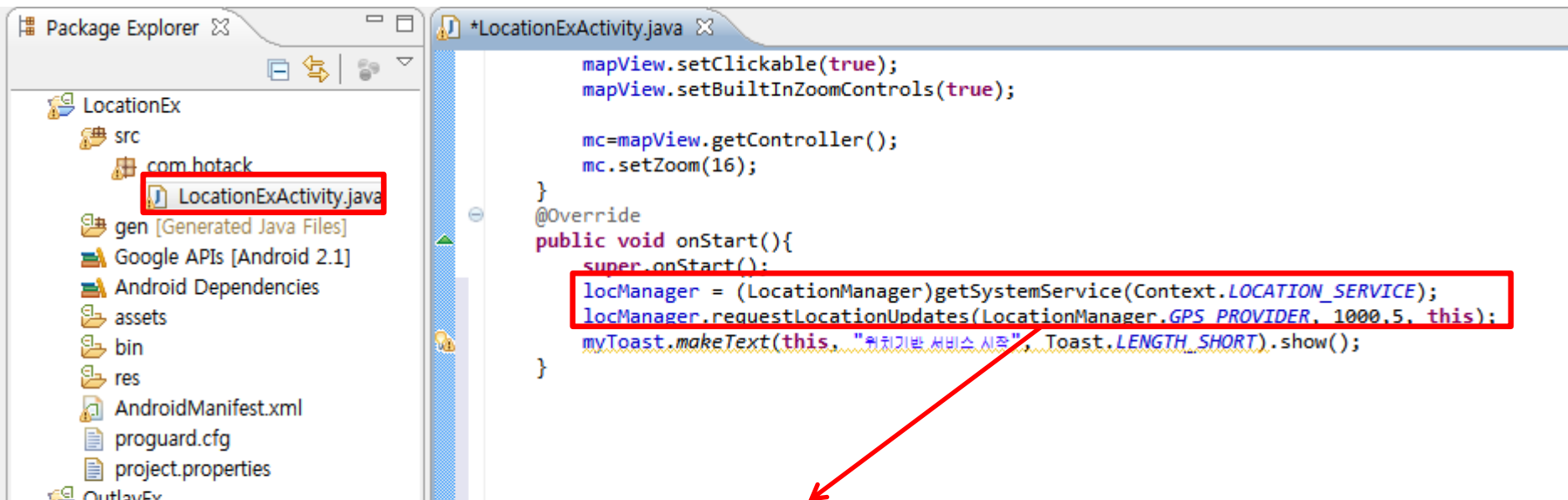
LocationManager객체를 생성





(2) GPS기능 이용하기

3) LocationService를 시작하고 위치정보를 받아온다.



매개변수는 순서대로 위치정보 제공자, 최소 업데이트 타임(milliseconds), 최소 업데이트 거리(meter), 위치정보를 받을 리스너를 나타냅니다.





(2) GPS기능 이용하기

4) LocationListener 인터페이스의 추상화 메소드들을 정의한다.

```
public void onLocationChanged(Location location) {  
    // TODO Auto-generated method stub  
    GeoPoint pos=new GeoPoint((int)(location.getLatitude()*1E6),(int)(location.getLongitude()*1E6));  
    mc.setCenter(pos);  
    myToast.makeText(this, "위치변경", Toast.LENGTH_SHORT).show();  
}  
public void onProviderDisabled(String provider)  
    // TODO Auto-generated method stub  
}  
public void onProviderEnabled(String provider)  
    // TODO Auto-generated method stub  
}  
public void onStatusChanged(String provider, int status, Bundle extras) {  
    // TODO Auto-generated method stub  
}  
}  
주어진 위치로 맵을 이동시킨다.
```

Location 객체의 getLatitude와 getLongitude 함수를 이용하여 위도와 경도를 얻는다. 반환값이 long형인데 int형으로 캐스팅하고 데이터손실을 막기 위해 1E6을 곱한다.





(2) GPS기능 이용하기

4) LocationListener 인터페이스의 추상화 메소드들을 정의한다.

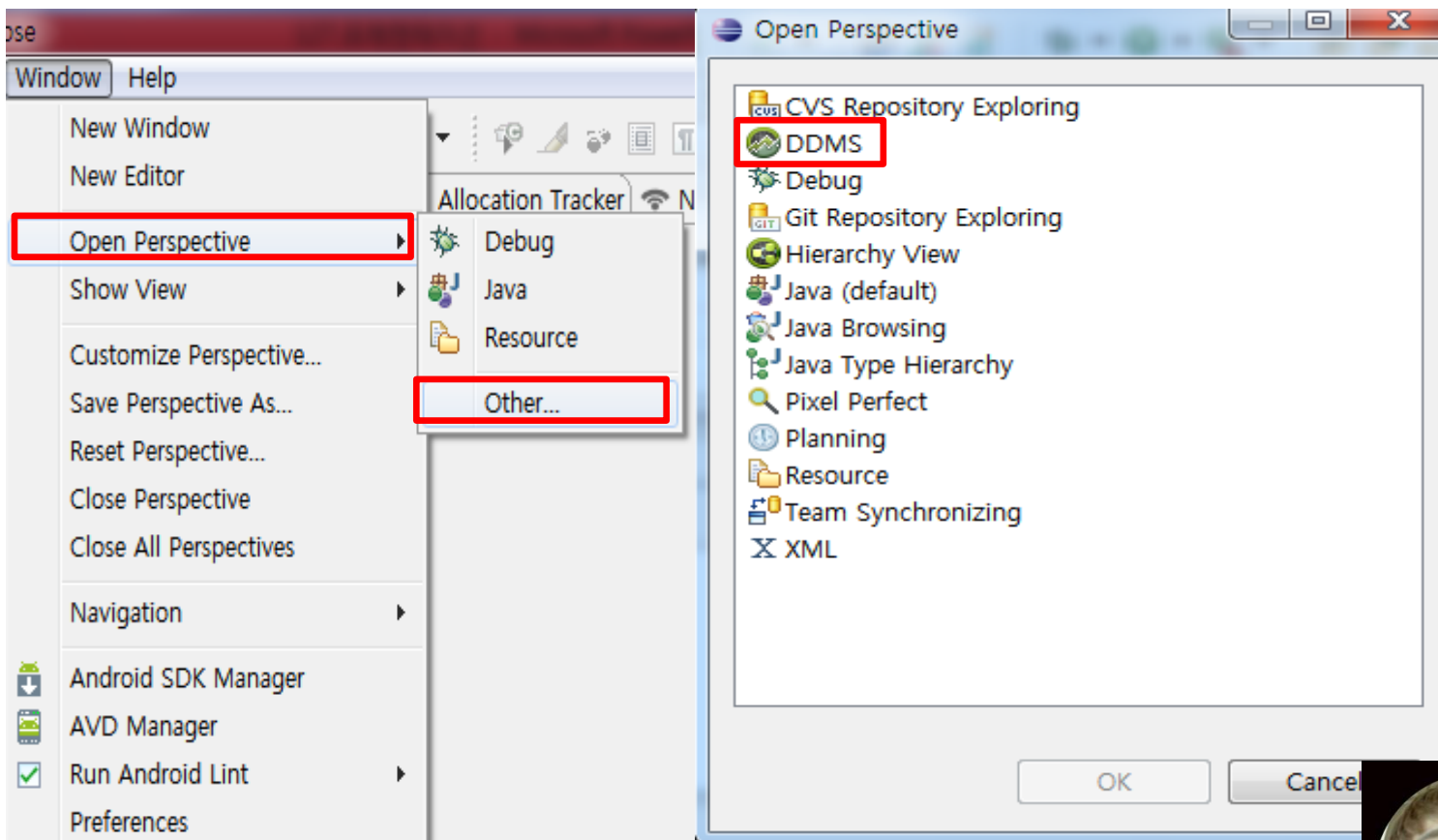
```
public void onProviderDisabled(String provider) {  
    // TODO Auto-generated method stub  
    —————> 위치정보 제공자가 유저에 의해 이용 불가할 때 호출된다.  
}  
public void onProviderEnabled(String provider) {  
    // TODO Auto-generated method stub  
    —————> 위치정보 제공자가 유저에 의해 이용 가능할 때 호출된다.  
}  
public void onStatusChanged(String provider, int status, Bundle extras) {  
    // TODO Auto-generated method stub  
    —————> 위치정보 제공자의 상태가 변화될 때 호출된다.  
}
```





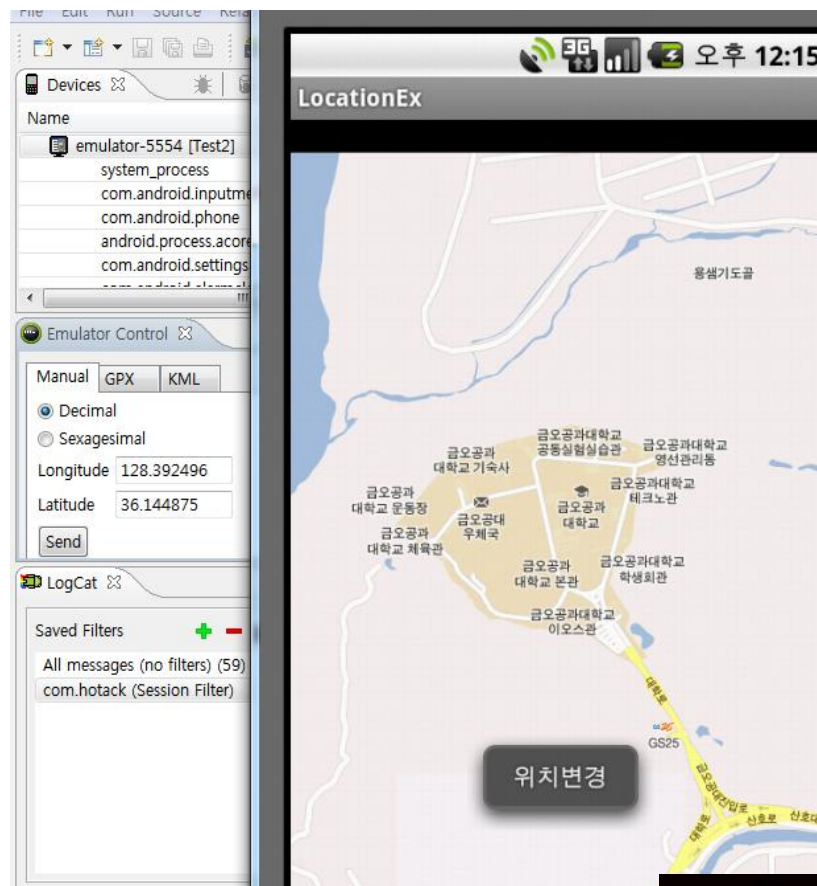
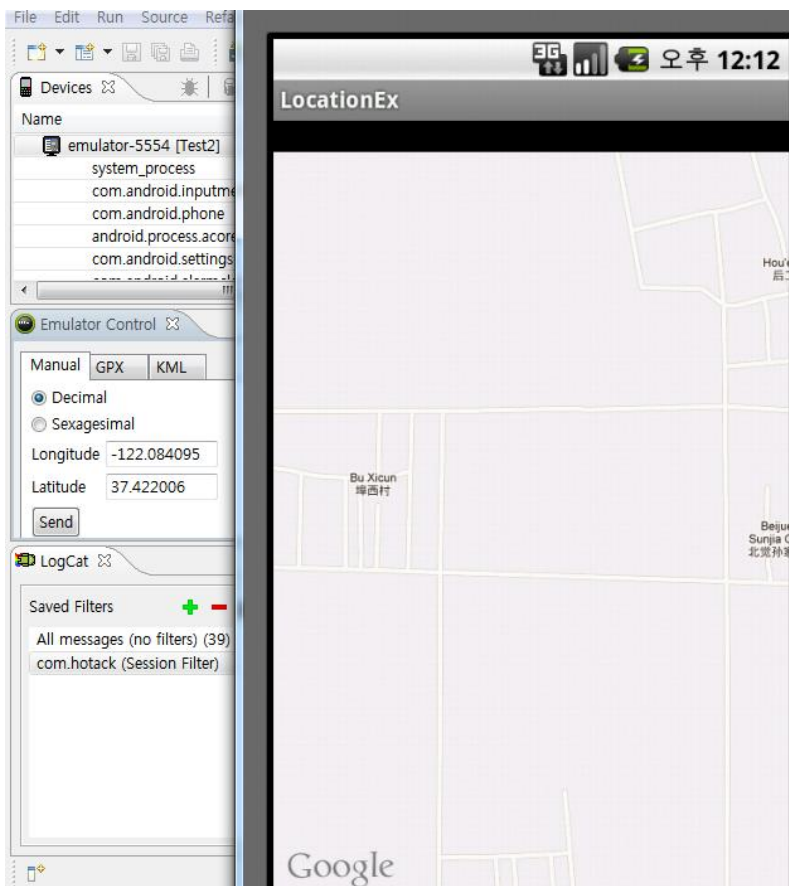
(2) GPS기능 이용하기

5) 에플레이터 상에서 좌표를 넘겨주기 위해 DDMS를 사용한다.





(2) GPS기능 이용하기





3. 오버레이 사용하기

(1) MyLocationOverlay Class 사용하기

(2) ItemizedOverlay Class 사용하기





3. 오버레이 사용하기

(1) MyLocationOverlay Class 사용하기

프로젝트의 .java 파일의 메인액티비티 클래스안에 작업을 하시면 됩니다.





(1) MyLocationOverlay Class 사용하기

1) MyLocationOverlay 객체를 생성한다.

```
import com.google.android.maps.MapView;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MyLocationOverlay;

import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Toast;

public class OverlayExActivity extends MapActivity implements LocationListener{
    /** Called when the activity is first created. */
    private MapView mapView;
    private LocationManager locManager;
    private MapController mc;
    private MyLocationOverlay mLocationOverlay;
    private Toast myToast = null;
```





(1) MyLocationOverlay Class 사용하기

2) 생성된 MyLocationOverlay을 지도 위에 오버레이한다.

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    mapView = (MapView)findViewById(R.id.mapview);  
    mapView.setBuiltInZoomControls(true);  
    mc = mapView.getController();  
    mc.setZoom(16);  
  
    initLocationOverlay();  
  
    mLocationOverlay.runOnFirstFix(new Runnable(){  
        public void run(){  
            mc.animateTo(mLocationOverlay.getMyLocation());  
        }  
    });  
}  
  
private void initLocationOverlay(){  
    mLocationOverlay = new MyLocationOverlay(this, mapView);  
    mLocationOverlay.enableMyLocation();  
    mLocationOverlay.enableCompass();  
    mapView.getOverlays().add(mLocationOverlay);  
}
```

MyLocationOverlay 생성

MyLocationOverlay에 자기위치를 표시해주는
함수를 호출한다.

현재 Map에 Overlay 추가





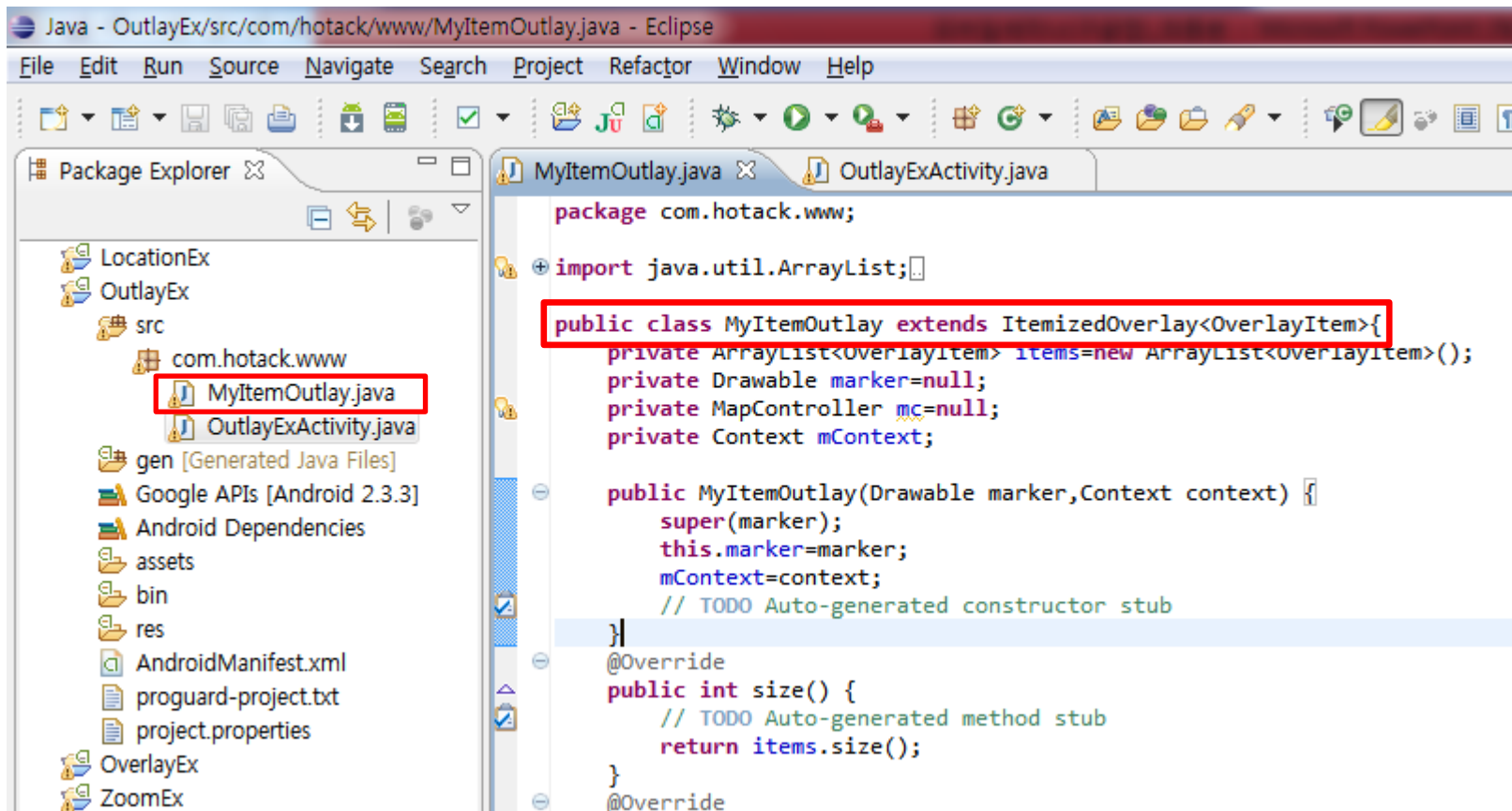
(1) MyLocationOverlay Class 사용하기





(2) ItemizedOverlay Class 사용하기

1) ItemizedOverlay<OverlayItem> Class를 상속한 Class를 만든다.





(2) ItemizedOverlay Class 사용하기

2) OverlayItem을 저장할 ArrayList와 Map위에 그릴 Drawable변수 생성

```
Java - OutlayEx/src/com/hotack/www/MyItemOutlay.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help

Package Explorer
LocationEx
OutlayEx
src
com.hotack.www
MyItemOutlay.java
OutlayExActivity.java
gen [Generated Java Files]
Google APIs [Android 2.3.3]
Android Dependencies
assets
bin
res
AndroidManifest.xml
proguard-project.txt
project.properties
OverlayEx
ZoomEx

MyItemOutlay.java
OutlayExActivity.java

package com.hotack.www;

import java.util.ArrayList;

public class MyItemOutlay extends ItemizedOverlay<OverlayItem>{
    private ArrayList<OverlayItem> items=new ArrayList<OverlayItem>();
    private Drawable marker=null;
    private MapController mc=null;
    private Context mContext;

    public MyItemOutlay(Drawable marker,Context context) {
        super(marker);
        this.marker=marker;
        mContext=context;
        // TODO Auto-generated constructor stub
    }

    @Override
    public int size() {
        // TODO Auto-generated method stub
        return items.size();
    }

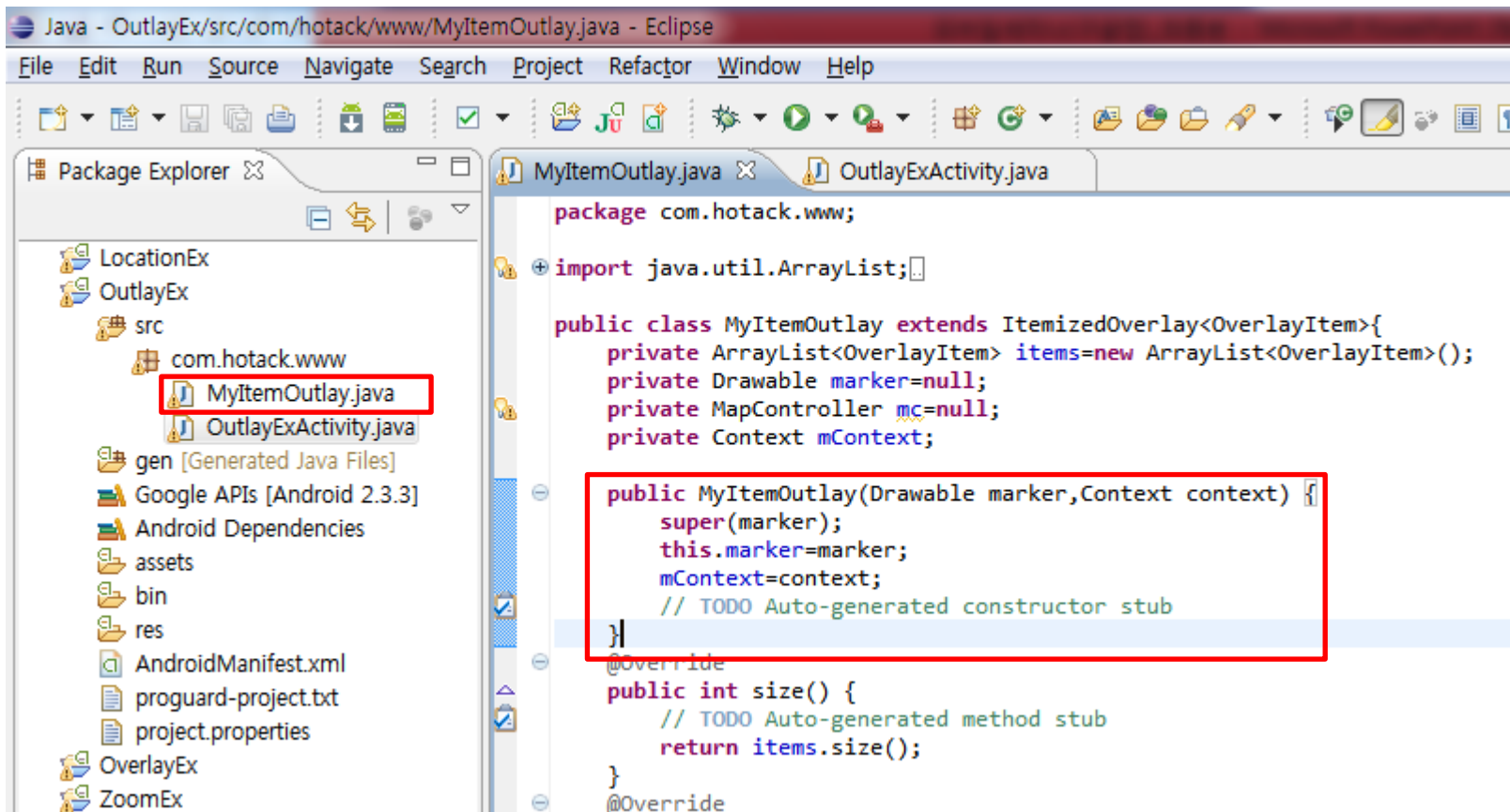
    @Override
```





(2) ItemizedOverlay Class 사용하기

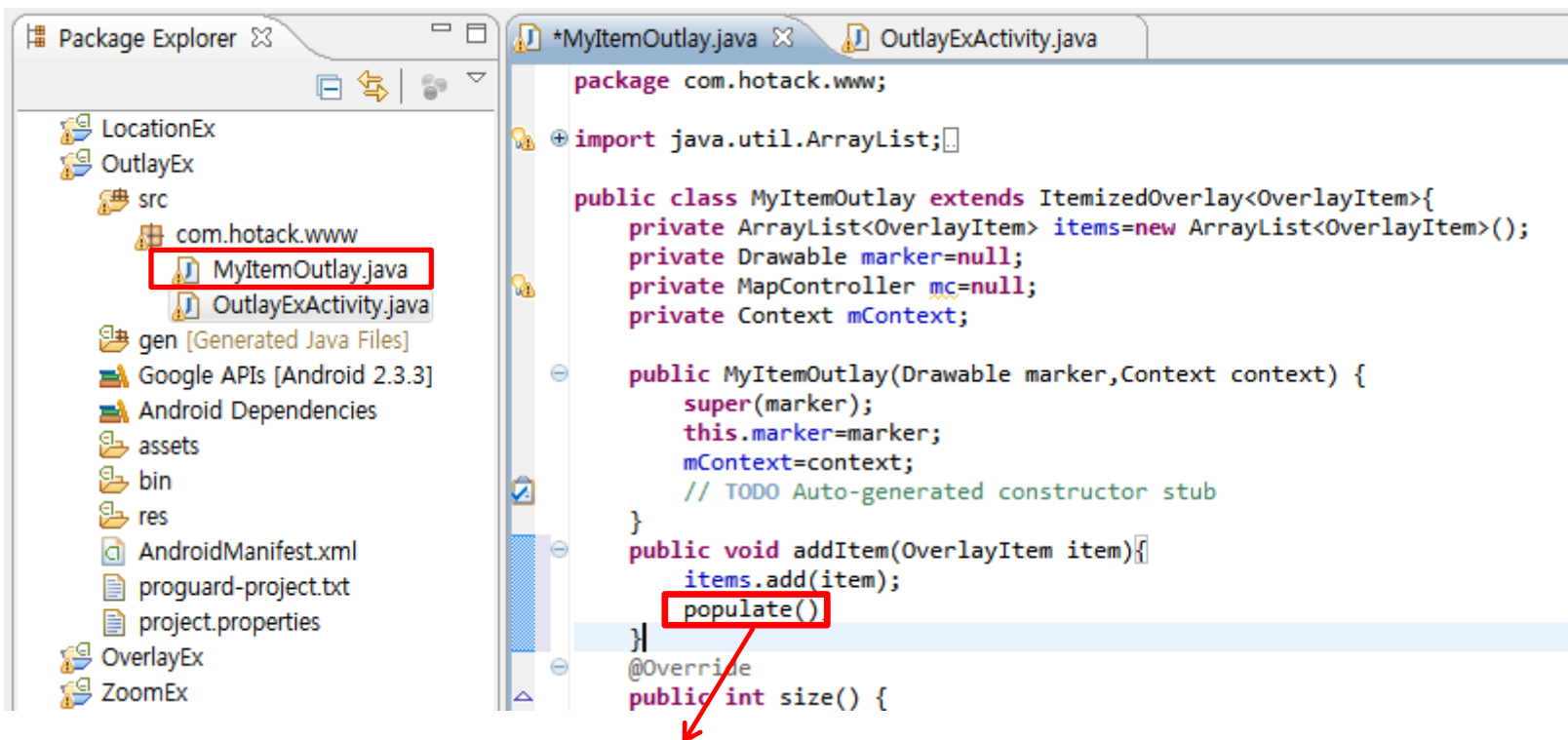
3) 생성자를 만든다.





(2) ItemizedOverlay Class 사용하기

4) Overlay Item을 List에 추가하는 메소드를 만든다.



populate 메소드가 호출되면 ItemizedOverlay 클래스의 추상화 메소드인 size(), createItem(int)을 순서대로 호출하게 된다.





(2) ItemizedOverlay Class 사용하기

5) 추상화 메소드들을 재정의한다.

```
public class MyItemOutlay extends ItemizedOverlay<OverlayItem>{
    private ArrayList<OverlayItem> items=new ArrayList<OverlayItem>();
    private Drawable marker=null;
    private MapController mc=null;
    private Context mContext;

    public MyItemOutlay(Drawable marker,Context context) {
        super(marker);
        this.marker=marker;
        mContext=context;
        // TODO Auto-generated constructor stub
    }
    @Override
    public int size(){
        // TODO Auto-generated method stub
        return items.size();
    }
    @Override
    protected OverlayItem createItem(int i){
        // TODO Auto-generated method stub
        return (items.get(i));
    }
    @Override
    public void draw(Canvas canvas, MapView mapView, boolean shadow){
        super.draw(canvas, mapView, shadow);
        boundCenterBottom(marker);
    }
}
```

→ List의 현재 크기를 반환한다.

→ List에 item을 삽입한다.

그림자를 사용하기 위한 메소드(사용을 원하지않으면 재정의하지 않아도된다.)





(2) ItemizedOverlay Class 사용하기

6) MainActivity Class에 ItemizedOverlay를 상속한 클래스의 객체를 만든다.

```
package com.hotack.www;

import android.location.*;

public class OutlayExActivity extends MapActivity implements LocationListener{
    /** Called when the activity is first created. */
    private final static String API_KEY = "0KQTz5kclGrT_P99M0nG4m5fwfLjCYAy_qHnYEQ";

    private LocationManager locManager;
    private MapController mc;
    private MapView mapView;
    private MyItemOutlay mItemOverlay;
    private MyLocationOverlay mLocationOverlay;

    Toast myToast=null;

    setContentView(R.layout.main);

    mapView=new MapView(this,API_KEY);
    mapView.setClickable(true);
```

메인 액티비티 Class안에 앞서 작성한 Class의 객체를 만든다.





(2) ItemizedOverlay Class 사용하기

7) Overlay에 사용할 marker를 지정하고 Overlay를 Mapview에 추가시킨다.

```
mc=mapView.getController();
mc.setZoom(16);

initLocationOverlay();
initItemOverlay();
addOverlayItem(36148520,128392391,"Test", "금오금대입수관");
addOverlayItem(36143944,128392084,"Test", "금오금대부관");
addOverlayItem(36146056,128386417,"Test", "금오금대체육관");

mLocationOverlay.runOnFirstFix(new Runnable(){
    public void run(){
        mc.animateTo(mLocationOverlay.getMyLocation());
    }
});

private void initItemOverlay(){
    Drawable marker=getResources().getDrawable(R.drawable.change_mvface);
    marker.setBounds(0, 0, marker.getIntrinsicWidth(), marker.getIntrinsicHeight());

    mItemOverlay=new MyItemOutlay(marker,this);
    mapView.getOverlays().add(mItemOverlay);
}

private void initLocationOverlay(){}
}
```

사용할 marker의 이미지는 프로젝트 탐색기에서 /res/drawable-mdip폴더 아래에 복사하시면 됩니다.





(2) ItemizedOverlay Class 사용하기

7) Overlay에 사용할 marker를 지정하고 Overlay를 Mapview에 추가시킨다.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    mapView=new MapView(this,API_KEY);
    mapView.setClickable(true);
    mapView.setBuiltInZoomControls(true);
    setContentView(mapView);

    mc=mapView.getController();
    mc.setZoom(16);

    initLocationOverlay();
    initItemOverlay();
}

private void initItemOverlay(){
    Drawable marker=getResources().getDrawable(R.drawable.change_myface);

    mItemOverlay=new MyItemOutlay(marker,this);
    mapView.getOverlays().add(mItemOverlay);
}
```

onCreate메소드안에서 실행될수 있게함

마커등록

등록한 마커를 사용하는 오버레이 생성

생성한 오버레이를 mapview에 추가
Map위에 오버레이가 추가되었지만 아직 Item이 없으므로
아무것도 나타나지 않음





(2) ItemizedOverlay Class 사용하기

8) 메인 액티비티에서 오버레이 item을 추가하는 메소드를 생성한다.

The screenshot shows an IDE with two tabs: `MyItemOutlay.java` and `OutlayExActivity.java`. The `Package Explorer` on the left shows the project structure with `src` and `gen` folders. In `src`, `com.hotack.www` contains `MyItemOutlay.java` and `OutlayExActivity.java`. In `gen`, `MyItemOutlay.java` is highlighted.

OutlayExActivity.java code:

```
mLocationOverlay = new MyLocationOverlay(this, mapView);
mLocationOverlay.enableMyLocation();
mLocationOverlay.enableCompass();

mapView.getOverlays().add(mLocationOverlay);

private void addOverlayItem(int lat, int lon, String title, String snip){
    mItemOverlay.addItem(new OverlayItem(new GeoPoint(lat, lon), title, snip));
}
```

MyItemOutlay.java code:

```
@Override
public void draw(Canvas canvas, MapView mapView, boolean shadow){
    super.draw(canvas, mapView, shadow);
    boundCenterBottom(marker);
}

public void addItem(OverlayItem item){
    items.add(item);
    populate();
}
```

A red arrow points from the `addItem` call in `addOverlayItem` to the `addItem` method in `MyItemOutlay`.

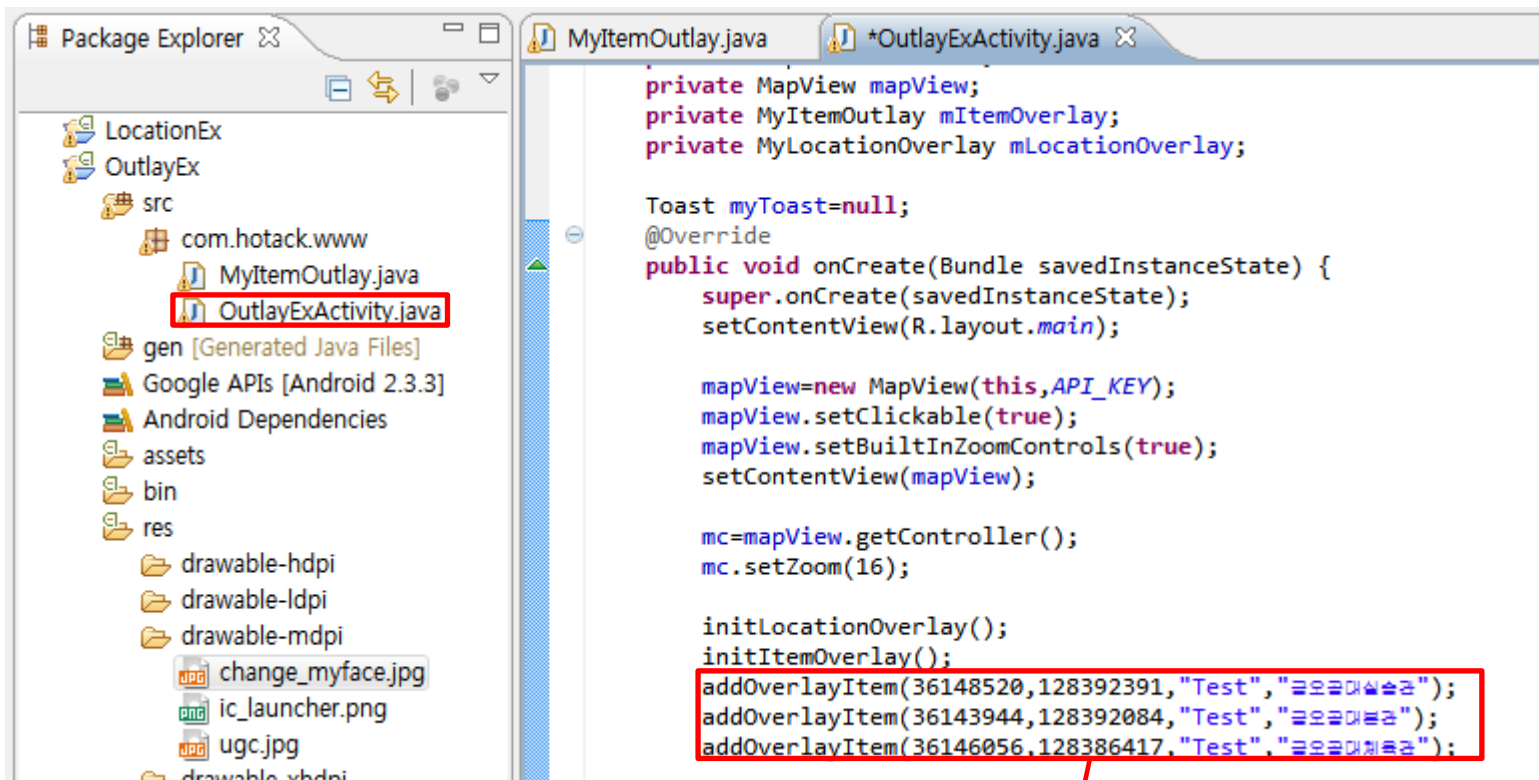
앞서 작성한 클래스의 객체를 이용하여 addItem 함수를 호출하여서 매개변수로 들어온 데이터를 OverlayItem으로 만들어 List에 저장한다.





(2) ItemizedOverlay Class 사용하기

9) Testing을 위해 메인 액티비티에서 Overlay 아이템을 추가한다.



앞서 만든 item을 추가하는 메소드를 호출한다.
넘겨지는 매개변수는 순서대로 위도,경도,타이틀,내용이다.





(2) ItemizedOverlay Class 사용하기

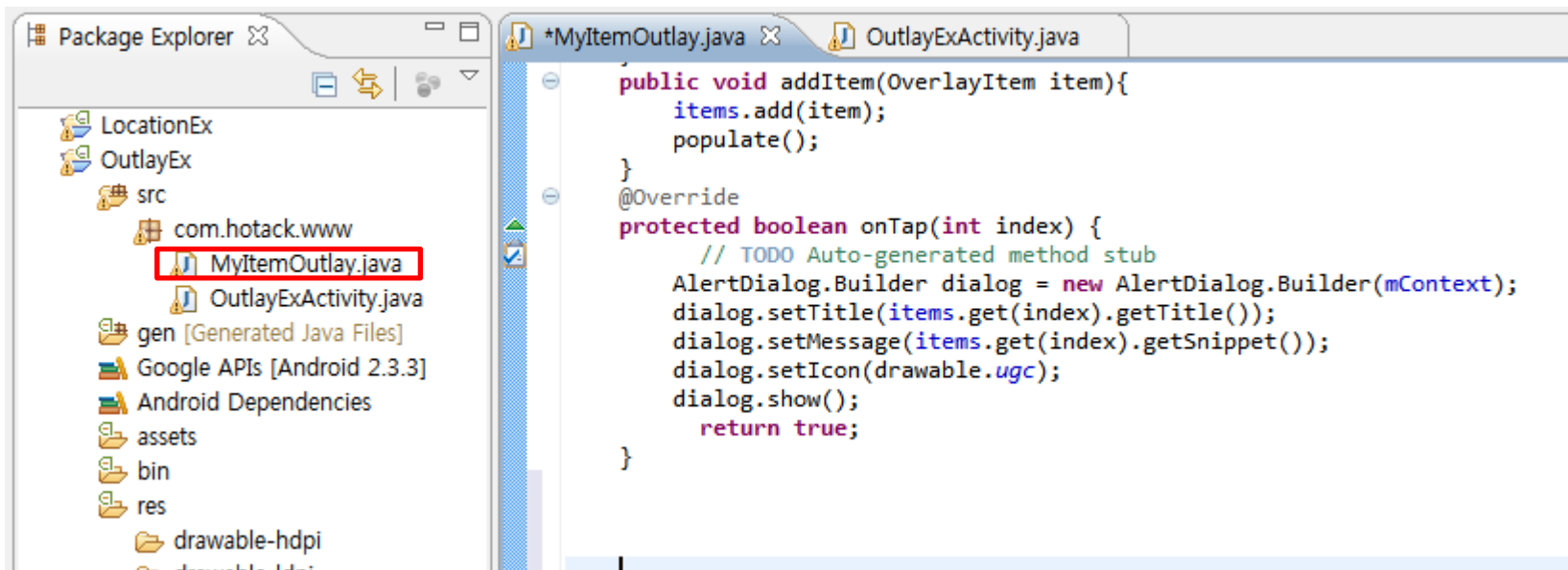




(2) ItemizedOverlay Class 사용하기

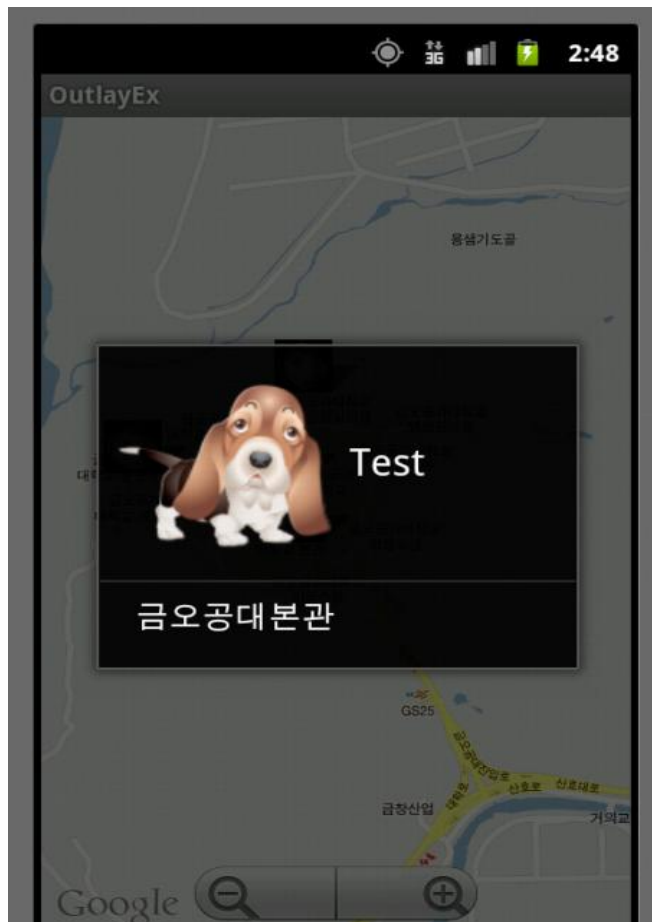
10) onTap(int)메소드 사용하기

- Map위에 나타난 marker를 터치하였을 경우 발생하는 이벤트메소드 (필수적으로 추가해야 되는 메소드는 아닙니다.)





(2) ItemizedOverlay Class 사용하기



THANK YOU

Speaker- 3조 급한 사람들



Q&A

