<Mobile Service>

디렉토리 구성

|  |  |
| --- | --- |
| Directory Name | Description |
| src | \*.java Files |
| gen | Auto Generated Java Files |
| lib | External Library |
| bin | \*.class Files |
| res | Resource Files |

패키지 구성

|  |  |
| --- | --- |
| Packege Name | Description |
| com.nhn.android.maps | Naver Map Api Toturial Packege |
| kr.android.hellodrinking | Application 작성 및 Application 과 Application Context와 통신에 사용하는 클래스의 작성에 필요한 클래스를 제공합니다. |
| kr.android.hellodrinking.activitiy | 사용자 인터페이스의 중 Activity 의 작성 및 Activity의 생성에 필요한 클래스를 제공합니다. |
| kr.android.hellodrinking.dialog | 사용자 인터페이스의 중 Dialog 의 작성 및 Dialog 의 생성에 필요한 클래스를 제공합니다. |
| kr.android.hellodrinking.sensor | 사용되는 Sensor 및 SensorManager 에 대해서 재정의한 클래스를 제공합니다. |
| kr.android.hellodrinking.utillity | 수치 연산 및 그래픽 연산, 파일 입출력에 관련한 클래스를 제공합니다. |
| kr.android.hellodrinking.view | 사용자 인터페이스 중 View 의 작성 및 View의 생성에 필요한 클래스를 제공합니다. |
| kr.android.hellodrinking.transmission | 네트워크 어플리케이션의 구현을 위한 클래스를 제공합니다. |
| kr.android.hellodrinking.transmission.dto | Bean 의 생성에 필요한 클래스를 제공합니다. |
| kr.android.hellodrinking.transmission.exception | 통신 중 일어날 수 있는 예외에 대해 정의한 클래스를 제공합니다. |

클래스 관계 개요도

Application

Activity / View

Sensor / Utils /   
Graphic Components

Data Model

서버

Responce

Request

|  |  |  |
| --- | --- | --- |
| Class | Description | 해당 Class |
| Application | 글로벌 변수 및 상태 정보를 가지고 있는 기초 클래스 | HelloDrinkingApplication |
| Activity / View | 데이터 모델을 뷰에 표현하기 위한 컨트롤을 담당하는 클래스  \*뷰에 관련한 정보는 .xml 파일이 담고 있다. | \*Activity, \*Dialog, \*View :  ARActivity, FrameActivity, LoginActivity, MapActivity, MemberinfoActivity, MemberJoinActivity, NMapFrameActivity, PostActivity, PostsActivity, ConfigureDialog, UserInfoDialog, CameraView, CompassView, MapContainerView |
| Utils | Activity 또는 View에서 사용하기 위한 연산을 담당하는 클래스 | Calculations, GraphicUtils |
| Sensor | SensorManager 의 역할을 확장하는 클래스 | Compass |
| Data Model | Activity 또는 View 에 표현되는 정보를 담고 있는 클래스  \* 서버와 동일한 클래스가 사용된다. | Bean, PostBean, UserBean |
| Request | 메서드를 호출하기 위해서 필요한 정보를 포함한 클래스  \* 서버와 동일한 클래스가 사용된다. | Request, RequestBeanPackege |
| Responce | Request를 받아 처리된 정보를 포함한 클래스  \* 서버와 동일한 클래스가 사용된다. | ResponceBeanPackege |

Application Classes

HelloDrinkingApplication

PostBean

UserBean

|  |  |
| --- | --- |
| Class Name | Description |
| HelloDrinkingApplication | PostBean 의 정보 및, 로그인 된 사용자의 상태 정보를 가지고 있는 기초 클래스 |

**class** HelloDrinkingApplication **extends** Application

|  |  |
| --- | --- |
| Constructor | Description |
| HelloDrinkingApplication() |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + setServerFromPreferences() | void | Preferences 객체에서 mServerIp, mServerPort 필드를 초기화한다. |
| + addPost(PostBean) | void | 지정 PostBean을 등록한다. |
| + removePost(PostBean) | void | 지정 PostBean을 삭제한다. |
| + getListPosts() | List<PostBean> | 저장된 PostBean 객체들을 List 형태로 반환한다. |
| + setListPosts(List<PostBean>) | void | 지정 List의 PostBean 객체들로 저장된 PostBean 객체들을 대체한다. |
| + setId(String) | void | 로그인된 User의 Id를 저장한다. |
| + getId() | String | 로그인된 User의 Id를 반환한다. |
| + refreshListPosts(NGeoPoint, int) | void | 지정 NGeoPoint 객체의 좌표로부터 지정한 거리 내의 PostBean 객체를 서버로부터 가져온다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| + (s) (f) DEFAULT\_SEARCH\_DISTANCE | int | 거리 조건 입력에 사용되는 기본 거리 : 10000 |
| + (s) mServerIp | String | 서버의 Ip를 가르킨다. |
| + (s) mServerPort | int | 서버의 Port를 가르킨다. |
| - id | String | 로그인 된 사용자의 Id를 가르킨다. |
| - mListPosts | List<PostBean> | 게시물 객체를 가르킨다. |

Activity / View Classes

FrameActivityActivity

NMapFrameActivitySensor

ARActivity

PostsActivity

MapActivityData Model

PostActivity

LoginActivity

MemberJoinActivity

MemberInfoActivity

ConfigureDialog

UserInfoDialog

CameraView

CompassView

MapContainerView

|  |  |
| --- | --- |
| Class Name | Description |
| ARActivity | 증강현실 Activity |
| FrameActivity | 레이아웃 및 동일 기능을 정의하는 추상 Activity |
| LoginActivity | 로그인 Activity |
| MapActivity | 지도 형태 Activity |
| MemberinfoActivity | 로그인된 회원의 정보 확인 및 수정을 위한 뷰의 Activity |
| MemberJoinActivity | 회원 가입을 위한 Activity |
| NMapFrameActivity | 레이아웃 및 동일 기능을 정의하는 추상 NMapActivity   * FrameActivity와 동일한 내용을 담고 있다. |
| PostActivity | 게시물 등록 Activity |
| PostsActivity | List 형태를 지닌 Activity |
| ConfigureDialog | Server의 Ip 및 Port 를 설정하는 Dialog |
| PostInfoDialog | 선택된 게시물의 상세 정보를 표현하는 Dialog |
| CameraView | 카메라에서 받아온 이미지를 표현하는 View |
| CompassView | 나침반의 이미지를 표현하는 View |
| MapContainerView | 지도의 이미지를 표현하는 View |

**class** ARActivity **extends** FrameActivity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| = loadContent() | void | ar.xml의 내용을 표현한다. 그리고 presentPosts()를 실행한다. |
| + onAccuracyChanged(Sensor, int) | void | Sensor의 정확도가 변경될 때 호출된다. 실제로는 어떠한 일도 하지 않는다. |
| = onResume() | void | Activity가 Resume 상태가 될 때 호출된다. Compass 객체의 리스너를 등록한다. |
| + onSensorChanged(SensorEvent) | void | Sensor의 값이 변경될 때 호출된다. Post의 위치를 계산하여 화면에 표현한다. |
| = onStop() | void | Activity가 Stop 상태가 될 때 호출된다. Compass 객체의 리스너를 해지한다. |
| = presentLocationChanged() | void | 자신의 위치가 변경될 때 호출된다. 실제로는 어떠한 일도 하지 않는다. |
| = presentPosts() | void | Application에 등록된 PostBean 객체의 정보를 바탕으로 View를 만든다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| + (s) (f) HORIZONAL\_VIEWING\_ANGLE | double | 화면에 표현되는 수평 시야각 : 30도 |
| + (s) (f) VERTICAL\_VIEWING\_ANGLE | double | 화면에 표현되는 수직 시야각 : 60도 |
| - (s) (f) STANDARD\_COORDINATE | doauble[] | 화면에 표현되는 좌표의 범위를 구하기 위한 단위 좌표 : (0,0,1) |
| - (s) (f) LEFT\_ON\_SCREEN | double | 수평 시야각으로부터 화면에 표현되는 최초의 왼쪽 좌표를 구한다.   * 화면 중앙을 0이라 칭한다. * 기준되는 계산에 이용되는 단위 좌표는 STANDARD\_COORDINATE 이다. |
| - (s) (f) UP\_ON\_SCREEN | double | 수직 시야각으로부터 화면에 표현되는 최초의 윗쪽 좌표를 구한다.   * 화면 중앙을 0이라 칭한다. * 기준되는 계산에 이용되는 단위 좌표는 STANDARD\_COORDINATE 이다. |
| - mCompass | Compass | Listener 등록 및 해지를 위한 객체 |
| - mCompassView | CompassView | 나침반을 표현하는 객체 |

**abstract class** FrameActivity **extends** Activity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | frame.xml의 내용을 표현한다. 그리고 loadContent()를 실행한다. |
| = onResume() | void | Activity가 Resume 상태가 될 때 호출된다. startMyLocation()을 실행한다. |
| = onStop() | void | Activity가 Stop 상태가 될 때 호출된다. stopMyLocation()을 실행한다. |
| = startMyLocation() | void | mMapLocationManager의 위치 탐색을 시작한다. 만약 사용이 불가능한 상태라면, GPS 설정 화면으로 넘어간다. |
| = stopMyLocation() | void | 위치 탐색을 종료한다. |
| + onLocationChanged( NMapLocationManager, NGeoPoint) | void | myLocatioin을 갱신한다. 그리고 presentLocationChanged()를 실핸한다. |
| + onLocationUpdateTimeout( NMapLocationManager) | void | stopMyLocation()을 실행한다. 그리고 에러 메세지를 출력한다. |
| = abstract loadContent() | void | 페이지가 만들어질 때 실행된다. |
| = abstract presentPosts() | void | 게시물을 표현한다. |
| = abstract presentLocationChanged() | Void | 위치가 변할 때 호출된다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| = mInflater | LayoutInFlater | Layout xml 파일을 뷰로 가져오기 위한 객체 |
| = mViewgroup | ViewGroup | 상속받은 클래스의 뷰를 담기 위한 객체 |
| = mButtonPosts | ImageButton | PostsActivity 로 인텐트시키는 객체 |
| = mButtonMap | ImageButton | MapActivity 로 인텐트시키는 객체 |
| = mButtonAR | ImageButton | ARActivity 로 인텐트시키는 객체 |
| = mButtonMember | ImageButton | MemberinfoActivity 로 인텐트시키는 객체 |
| = mButtonRefresh | Button | 게시물 리스트를 갱신하기 위한 객체 |
| = mButtonPost | Button | PostActivity 로 인텐트시키는 객체 |
| = mEditDistance | EditText | 게시물 리스트를 갱신하는 거리 조건을 담고 있는 객체 |
| = mMapLocationManager | NMapLocationManager | 현재 위치를 갱신하기 위해 사용 |
| = myLocation | NGeoPoint | 현재 위치 정보를 담고 있는 객체 |

**abstract class** NMapFrameActivity **extends** NMapActivity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | frame.xml의 내용을 표현한다. 그리고 loadContent()를 실행한다. |
| = onResume() | void | Activity가 Resume 상태가 될 때 호출된다. startMyLocation()을 실행한다. |
| = onStop() | void | Activity가 Stop 상태가 될 때 호출된다. stopMyLocation()을 실행한다. |
| = startMyLocation() | void | mMapLocationManager의 위치 탐색을 시작한다. 만약 사용이 불가능한 상태라면, GPS 설정 화면으로 넘어간다. |
| = stopMyLocation() | void | 위치 탐색을 종료한다. |
| + onLocationChanged( NMapLocationManager, NGeoPoint) | void | myLocatioin을 갱신한다. 그리고 presentLocationChanged()를 실핸한다. |
| + onLocationUpdateTimeout( NMapLocationManager) | void | stopMyLocation()을 실행한다. 그리고 에러 메세지를 출력한다. |
| = abstract loadContent() | void | 페이지가 만들어질 때 실행된다. |
| = abstract presentPosts() | void | 게시물을 표현한다. |
| = abstract presentLocationChanged() | Void | 위치가 변할 때 호출된다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| = mInflater | LayoutInFlater | Layout xml 파일을 뷰로 가져오기 위한 객체 |
| = mViewgroup | ViewGroup | 상속받은 클래스의 뷰를 담기 위한 객체 |
| = mButtonPosts | ImageButton | PostsActivity 로 인텐트시키는 객체 |
| = mButtonMap | ImageButton | MapActivity 로 인텐트시키는 객체 |
| = mButtonAR | ImageButton | ARActivity 로 인텐트시키는 객체 |
| = mButtonMember | ImageButton | MemberinfoActivity 로 인텐트시키는 객체 |
| = mButtonRefresh | Button | 게시물 리스트를 갱신하기 위한 객체 |
| = mButtonPost | Button | PostActivity 로 인텐트시키는 객체 |
| = mEditDistance | EditText | 게시물 리스트를 갱신하는 거리 조건을 담고 있는 객체 |
| = mMapLocationManager | NMapLocationManager | 현재 위치를 갱신하기 위해 사용 |
| = myLocation | NGeoPoint | 현재 위치 정보를 담고 있는 객체 |

**class** LoginActivity **extends** Activity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | login.xml의 내용을 표현한다. |
| = onActivityResult() | Void | ConfigureDialog의 결과로 실행된다. SharedPreferences에 Server의 Ip, Port를 저장한다. |
| + onCreateOptionsMenu( Menu) | boolean | 메뉴를 생성한다. |
| + onOptionsItemSelected( MenuItem) | void | 메뉴가 선택되면 호출된다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| = mEditId | EditText | 로그인 시 사용되는 아이디 값을 담는 객체 |
| = mEditPw | EditText | 로그인 시 사용되는 암호 값을 담는 객체 |
| = mButtonLogin | ImageButton | 로그인을 실행하기 위한 객체 |
| = mButtonJoin | ImageButton | MemeberJoinActivity 로 인텐트시키는 객체 |

**class** MapActivity **extends** NMapFrameActivity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| = loadContent() | void | map.xml의 내용을 표현한다. 그리고 startMyLocation(), presentPosts() 을 실행한다. |
| = startMyLocation() | void | 현재 위치를 파악하고, 현재 위치를 표현한다. 만약 사용이 불가능한 상태라면, GPS 설정 화면으로 넘어간다. |
| = stopMyLocation() | void | 위치 탐색을 종료한다. 그리고 현재 위치 표현을 삭제한다. |
| = presentLocationChanged() | void | 현재 위치가 변함을 표현한다. 실제로는 어떠한 일도 하지 않는다. |
| = presentPosts() | void | 게시물을 표현한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - (s) (f) API\_KEY | String |  |
| - mMapContainerView | MapContainerView |  |
| - mMapView | NMapView |  |
| - mMapController | NMapController |  |
| - (s) (f) NMAP\_LOCATION\_DEFAULT | NGeoPoint |  |
| - mOverlayManager | NMapOverlayManager |  |
| - mMyLocationOverlay | NMapMyLocationOverlay |  |
| - mMapCompassManager | NMapCompassManager |  |
| - mMapViewerResourceProvider | NMapViewerResourceProvider |  |
| - mPoiDataOverlay | NMapPOIdataOverlay |  |
| - mCompassView | CompassView |  |
| - onCalloutOverlayListener | NMapOverlayManager. OnCalloutOverlayListener |  |
| - onMapViewStateChangeListener | NMapView. OnMapStateChangeListener |  |
| - onMapViewTouchDelegate | NMapView. OnMapViewDelegate |  |
| - onPOIdataStateChangeListener | NMapPOIdataOverlay. OnStateChangeListener |  |

**class** MemberinfoActivity **extends** Activity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | memberinfo.xml의 내용을 표현한다. 그리고 loadUser()을 실행한다. |
| + onClick(View) | void | 클릭 액션에 따른 이벤트를 정의한다. |
| = onActivityResult(int, int, Intent) | void | android.provider.MediaStore.*ACTION\_IMAGE\_CAPTURE*의 결과로 실행된다. 카메라에서 얻어온 이미지를 출력한다. |
| - loadUser() | void | 서버로부터 로그인된 사용자의 정보를 불러와 출력한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mButtonSend | Button | 입력된 결과를 서버로 전송시키는 객체 |
| - mButtonTakePhoto | Button | android.provider.MediaStore.*ACTION\_IMAGE\_CAPTURE*로 인텐트시키는 객체 |
| - mEditId | EditText | 아이디 정보를 담는 객체 |
| - mEditPw | EditText | 암호 정보를 담는 객체 |
| - mEditPw2 | EditText | 암호 확인 정보를 담는 객체 |
| - mEditName | EditText | 이름 정보를 담는 객체 |
| - mEditAge | EditText | 나이 정보를 담는 객체 |
| - mEditSex | EditText | 성별 정보를 담는 객체 |
| - mEditPhone | EditText | 전화번호 정보를 담는 객체 |
| - mEditJob | EditText | 직업 정보를 담는 객체 |
| - mImagePhoto | ImageView | 사진을 표현하는 객체 |

**class** MemberJoinActivity **extends** Activity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | memberjoin.xml의 내용을 표현한다. |
| + onClick(View) | void | 클릭 액션에 따른 이벤트를 정의한다. |
| = onActivityResult(int, int, Intent) | void | android.provider.MediaStore.*ACTION\_IMAGE\_CAPTURE*의 결과로 실행된다. 카메라에서 얻어온 이미지를 출력한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mButtonSend | Button | 입력된 결과를 서버로 전송시키는 객체 |
| - mButtonTakePhoto | Button | android.provider.MediaStore.*ACTION\_IMAGE\_CAPTURE*로 인텐트시키는 객체 |
| - mEditId | EditText | 아이디 정보를 담는 객체 |
| - mEditPw | EditText | 암호 정보를 담는 객체 |
| - mEditPw2 | EditText | 암호 확인 정보를 담는 객체 |
| - mEditName | EditText | 이름 정보를 담는 객체 |
| - mEditAge | EditText | 나이 정보를 담는 객체 |
| - mEditSex | EditText | 성별 정보를 담는 객체 |
| - mEditPhone | EditText | 전화번호 정보를 담는 객체 |
| - mEditJob | EditText | 직업 정보를 담는 객체 |
| - mImagePhoto | ImageView | 사진을 표현하는 객체 |

**class** PostActivity **extends** Activity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | post.xml의 내용을 표현한다. |
| + onClick(View) | void | 클릭 액션에 따른 이벤트를 정의한다. |
| = onActivityResult(int, int, Intent) | void | android.provider.MediaStore.*ACTION\_IMAGE\_CAPTURE*의 결과로 실행된다. 카메라에서 얻어온 이미지를 출력한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mButtonSend | Button | 입력된 결과를 서버로 전송시키는 객체 |
| - mButtonTakePhoto | Button | android.provider.MediaStore.*ACTION\_IMAGE\_CAPTURE*로 인텐트시키는 객체 |
| - mEditId | EditText | 아이디 정보를 담는 객체 |
| - mEditComment | EditText | 코멘트 정보를 담는 객체 |
| - mImagePhoto | ImageView | 사진을 표현하는 객체 |
| - longitude | double | 현재 위치의 경도 정보를 가진다. |
| - latitude | double | 현재 위치의 위도 정보를 가진다. |

**class** PostsActivity **extends** FrameActivity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| = loadContent() | void | posts.xml의 내용을 표현한다. 그리고 presentPosts()를 실행한다. |
| = presentPosts() | void | 게시물을 리스트로 표현한다. |
| = presentLocationiChanged() | void | 각 게시물과의 거리를 갱신한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mListViewPosts | ListView | 게시물을 표현하기 위한 리스트 객체 |

|  |
| --- |
| Nested Class |
| class PostAdapter extends BaseAdapter   |  |  | | --- | --- | | Constuctor | Description | | PostAdapter(Context, in, List<PostBean>) |  |  |  |  |  | | --- | --- | --- | | Method | Return Type | Description | | + getCount() | int | 리스트의 총 개수를 반환한다. | | + getItem(int) | Object | 해당 인덱스에 해당하는 행의 객체를 반환한다. | | + getItemId(int) | long | 해당 인덱스에 해당하는 행의 아이디를 반환한다. | | + getView(int, View, ViewGroup) | View | 해당 인덱스에 해당하는 행의 뷰를 반환한다. |  |  |  |  | | --- | --- | --- | | Field | Type | Description | | context | Context | ListView가 표현되는 Context | | id | int | ListView의 id | | posts | List<PostBean> | ListView의 표현 객체 | |

**class** ConfigureDialog **extends** Activity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | configure.xml의 내용을 표현한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mEditIp | EditText | 서버의 Ip 정보를 담는 객체 |
| - mEditPort | EditText | 서버의 Port 정보를 담는 객체 |
| - mButtonSave | Button | 서버 정보를 저장하는 객체 |

**class** PostInfoDialog **extends** Activity

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + onCreate(Bundle) | void | postinfo.xml의 내용을 표현한다. |
| + onClick(View) | voic | 클릭 액션에 따른 이벤트를 정의한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mTextId | TextView | 게시물의 ID 정보를 담는 객체 |
| - mTextComment | TextView | 게시물의 Comment 정보를 담는 객체 |
| - mImage | ImageView | 게시물의 Image 정보를 담는 객체 |
| - mButtonCall | Button | 게시물의 사용자에게 전화를 거는 Intent를 실행하는  객체 |
| - mButtonLetter | Button | 게시물의 사용자에게 문자를 보내는 Intent를 실행하는 객체 |

**class** CameraView **extends** SurfaceView

|  |  |
| --- | --- |
| Constructor | Description |
| + CameraView(Context) |  |
| + CameraView(Context, AttributeSet) |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + surfaceCreated(SurfaceHolder) | Void | Camera 객체의 Preview를 시작한다. |
| + surfaceChanged( SurfaceHolder, int, int, int) | void | 변경된 크기로 Camera 객체의 Preview를 시작한다. |
| + surfaceDestroyed(SurfaceHolder) | void | Camera 객체의 Preview를 종료한 후, 자원을 반환한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mCamera | Camera | Camera 를 이용하기 위한 객체 |
| - mSurfaceHolder | SurfaceHolder | SurfaceView의 상태를 캐치하기 위한 객체 |

**class** CompassView **extends** ImageButton

|  |  |
| --- | --- |
| Constructor | Description |
| + CompassView(Context) |  |
| + CompassView(Context, AttributeSet) |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + setCompass(Compass) | void | Compass에 Listener를 등록한다. |
| + setBackgroundBitmap(Bitmap) | void | 나침반의 배경이 될 이미지를 링크한다. |
| + setOrientationBitmap(Bitmap) | void | 나침반이 가르키는 방향을 나타낼 이미지를 링크한다. |
| = onDraw(Canvas) | void | 나침반을 그린다. |
| + onSensorChanged(SensorEvent) | void | mValues 를 수정한다. |
| + onAccuracyChanged(Sensor, int) | void |  |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mBitmapBackground | Bitmap | 나침반의 배경이 되는 이미지 객체 |
| - mBitmapOrient | Bitmap | 나침반의 바늘이 되는 이미지 객체 |
| - mValues | double[] | 나침반의 각도 정보를 가지고 있는 배열 |

Sensor Classes

|  |  |
| --- | --- |
| Class Name | Description |
| Compass | Orientation 센서를 등록하고 SensorEventListener 를 관리하는 클래스 |

**class** Compass

|  |  |
| --- | --- |
| Constructor | Description |
| + Compass(Context) |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + addListener( SensorEventListener) | void | mListSensorListeners 에 SensorEventListener 를 추가한다. |
| + removeListener( SensorEventListener) | void | mListSensorListeners 에 SensorEventListener 를 삭제한다. |
| + start() | Void | 모든 SensorEventListener 를 등록한다. |
| + stop() | void | 모든 SensorEventListener 를 제거한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - mSensorManager | SensorManager | 장비의 센서에 접근하게 해주는 객체 |
| - mSensor | Sensor | Orientation 센서를 나타내는 객체 |
| - mListSensorListeners | List<SensorEventListener> | Sensor 의 값이 바뀔 때, SensorManager로부터 알려주는 리스너의 집합 |

Utillity Classes

|  |  |
| --- | --- |
| Class Name | Description |
| Calculations | 좌표 계산을 위한 클래스 |
| GraphicUtils | Image File, Bitmap, Drawable 변환을 위한 클래스 |

Class Calculations

|  |  |
| --- | --- |
| Constructor | Description |
| - Calculations () |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + (s) distanceBetween3DPoint( double[], double[]) | double | 두 좌표 사이의 거리를 반환한다. |
| + (s) distanceBetweenTwoGeoPoint( NGeoPoint, NGeoPoint) | int | 두 좌표 사이의 실제 거리를 반환한다. |
| + (s) distanceFromZero( double[]) | double | 원점에서 해당 좌표까지의 거리를 반환한다. |
| + (s) getLocationOfViewOnScreen( float[], NGeoPoint, NGeoPoint) | double[] | 보는 방향을 기준으로 목표체의 새로운 좌표를 반환한다. |
| + (s) xRotationConvert(double[], double) | double[] | x축으로 회전한 후의 변환된 좌표를 반환한다. |
| + (s) yRotationConvert(double[], double) | double[] | y축으로 회전한 후의 변환된 좌표를 반환한다. |
| + (s) zRotationConvert(double[], double) | double[] | z축으로 회전한 후의 변환된 좌표를 반환한다. |

Class GraphlicUtils

|  |  |
| --- | --- |
| Constructor | Description |
| - GraphlicUtils() |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + createBitmapFromImage(File) | Bitmap | 파일 객체로부터 Bitmap 객체를 만들어 반환한다. |
| + createImageFile(Context, byte[]) | File | 배열을 Context에서 지정한 캐시 폴더에 파일로 저장한 후 반환한다. |
| + createTempImageFile(Context, Drawable) | File | Drawable 객체을 Context에서 지정한 캐시 폴더에 파일로 저장한 후 반환한다. |
| + getOverlayDrawableFromBitmap( Bitmap) | Drawable | Bitmap 객체를 Drawable 객체로 변환 후, Overlay 형식에 맞게 이미지를 조정 후, 반환한다. |
| + setDefaultNoSearchImage(Resources, int) | Void | getOverlayDrawableFromBitmap(Bitmap) 메서드의 반환 중, Bitmap이 Null 이 경우 반환되는 Drawable 의 형태를 결정한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| + (s) (f) TEMP\_IMAGE\_FILE\_NAME | String | 임시 파일을 생성할 때 사용되는 파일 이름 : “temp.jpg” |
| + (s) DEFAULT\_NO\_SEARCH\_IMAGE | Drawable | getOverlayDrawableFromBitmap(Bitmap) 메서드의 반환 중, Bitmap이 Null 이 경우 반환되는 객체 |
| - (s) (f) POIOVERLAY\_IMAGE\_WIDTH | int | Overlay Drawable을 생성할 때, 사용되는 Overlay의 넓이 : 150 |
| - (s) (f) POIOVERLAY\_IMAGE\_HEIGHT | int | Overlay Drawable을 생성할 때, 사용되는 Overlay의 높이 : 150 |