Aggiungere la mapView nel layout:

<eu.trentorise.smartcampus.osm.android.views.MapView android:id=*"@+id/mapview"*/>

Creare un oggetto di tipo MapView nella classe dove vogliamo utilizzare la mappa e assegnare il layout creato sopra

**private** MapView mapView;  
mapView = (MapView) findViewById(R.id.*mapview*);

Ovviamente dare i permessi nel manifest per accedere a internet e per il GPS

<uses-permission android:name=*"android.permission.INTERNET"* />

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"* />

<uses-permission android:name=*"com.google.android.providers.gsf.permission.READ\_GSERVICES"* />

<uses-permission android:name=*"android.permission.ACCESS\_COARSE\_LOCATION"* />

<uses-permission android:name=*"android.permission.ACCESS\_FINE\_LOCATION"* />

<uses-permission android:name=*"android.permission.ACCESS\_WIFI\_STATE"* />

Per controllare la mappa abilitare i tasti per lo zoom(setBuiltInZoomControls) e lo zoom tramite il multitouch.

mapView.setBuiltInZoomControls(**true**);

mapView.setMultiTouchControls(**true**);

Per il controllo della mappa da codice creare un mapController collegato alla mappa, dal quale si potrà cambiare lo zoom (setZoom()) e anche centrare la mappa in un punto(tramite setCentre() o animateTo())

**private** MapController mapController;  
mapController = mapView.getController();  
  
mapController.setZoom(10);   
mapController.animateTo(Latitude,Longitude);

Per visualizzare sulla mappa un overlay mostrante la propria posizione, creare un MyLocationOverlay e aggiungerlo agli overlay della mappa, è anche possibile far visualizzare una bussola nell’angolo in alto a sinistra della mapview tramite l’istruzione myLoc.enableCompass();

MyLocationOverlay myLoc = **new** MyLocationOverlay(getApplicationContext(), mapView);

myLoc.enableMyLocation();

myLoc.enableCompass();

mapView.getOverlays().add(myLoc);

Per disegnare il percorso che passa per tutti i punti presenti in un arraylist di GeoPoint utilizzare il metodo drawPath(), passandogli come parametro l’arraylist.

mapView.drawPath(mPoints);

Se invece che disegnare il percorso sulla mapView si vuole avere solo la lista delle indicazioni per raggiungere uno o piu' punti, utilizzare il metodo

GetInstructions.*get*(ArrayList<GeoPoint>, Context);

Per aggiungere un marker ogni volta che si tiene premuto per 1 secondo nello stesso punto della mappa utilizzare il metodo addMarkersOnLongClick()

mapView.addMarkerOnLongClick();

Per aggiungere invece dei marker avendo già le posizioni esistono varie alternative:

- addMarkers(ArrayList<OverlayItem> items)

Da utilizzare se si hanno già gli OverlayItem con descrizione e titolo propri, oltre ad eventuale icona, da aggiungere.

- addMarkers(ArrayList<OverlayItem> items,OnItemGestureListener<OverlayItem> onGesture)

Da utilizzare se oltre agli OverlayItem si vuole impostare anche ciò che succede al tocco.

- addMarkersFromPoints(ArrayList<GeoPoint> points)

Da utilizzare se si hanno solo i punti e non si vuole dare titolo e descrizione ai propri marker, viene infatti usato di default “Title” e “Description”

E’ possibile fare in modo che la mapview non utilizzi la connessione anche se è disponibile, questo può essere utile se si dispone già della cache della mappa e non si vuole uilizzare la connessione.

mapView.setUseDataConnection();

La mappa salverà automaticamente tutte le tile scaricate per poterle utilizzare anche offline,questo può essere utile ma dopo un po’ di tempo questa occupa molta memoria nel dispositivo. Per cancellare la cache si può utilizzare la funzione deleteCache() della mapview.

mapView.deleteCache();

Per aggiungere un overlay di marker già clusterizzati automaticamente utilizzare uno dei metodi addClusteredMarkers della mapview

Se si vuole convertire un indirizzo in un punto geografico basta chiamare il metodo

-geocoding by android

Geocoding.FromAddressToPoint(address, context);

-geocoding by Nominatim

OSMGeocoding.FromAddressToPoint(address, context);

Il quale restituisce un ArrayList di GeoPoint, che punta a null in caso di un errore di connessione o di un errore di traduzione dell’indirizzo.

Se invece si vuole convertire un punto geografico in un indirizzo, basta chiamare il meotodo

-geocoding by android

Geocoding.FromPointToAddress(GeoPoint, context);

-geocoding by Nominatim

OSMGeocoding.FromPointToAddress (GeoPoint, context);

Il quale restituisce una List di Address, che punta a null in caso di un errore di connessione o di un errore di traduzione dell’indirizzo.