

Android APP手機程式設計實務



講師：賴貴平



本節課程內容

- 本節課程內容將包含以下教學內容:

Android LBS與資料庫應用

- GPS系統
- 定位服務、Google Map
- 地圖解碼
- SQLite、資料庫應用



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- Android行動裝置結合定位功能和Google地圖可以建立「位置感知服務」（Location-based Service，LBS）
- LBS應用程式可以追蹤你的位置和提供一些額外服務，例如：找出附近的咖啡廳、停車場、自動櫃員機或加油站等。
- Android作業系統提供LocationManager類別的定位服務來存取行動裝置目前的定位資料
 - 緯度（latitude）
 - 經度（longitude）
 - 高度（altitude）



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- Android作業系統的定位提供者（**provider**）可以提供不同方式的定位服務，基本上，我們主要使用兩種定位提供者
 - **GPS定位提供者**：提供者名稱字串為"**gps**"，使用**GPS**（**Global Positioning System**）衛星訊號來定位，可以提供精確的位置資訊，但是看不到天空，無法收到衛星訊號的室內並無法使用。
 - **網路定位提供者**：提供者名稱字串為"**network**"，直接使用電信公司基地台來進行三角定位，其提供的位置資訊較不精確，但是可以在室內使用



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 定位服務最主要的目的是找出行動裝置目前位置的經緯度座標，經緯度是經度與緯度合稱的座標系統，也稱為地理座標系統，使用三度空間的球面來定義地球表面各點的座標系統，能夠標示地球表面上的任何一個位置



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 緯度 (**latitude**) : 地球表面某一點距離地球赤道以南或以北的度數，其值為**0**至**90**度，赤道以北的緯度叫北緯（符號為**N**）；赤道以南的緯度稱南緯（符號為**S**）
- 經度 (**longitude**) : 地球表面上某一點距離本初子午線（一條南北方向經過倫敦格林威治天文台舊址的子午線）以東或以西的度數，簡單的說，本初子午線的經度是**0**度，其他地點的經度是向東從**0**到**180**度，即東經（符號為**E**）或向西從**0**到**180**度，即西經（符號為**W**）



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 定位服務是Android作業系統的系統服務（ `system services` ），以定位服務來說，就是LocationManager物件，如下所示：

- `private LocationManager lc;`
- `lc = (LocationManager)
getSystemService(LOCATION_SERVICE);`

- 上述程式碼使用`getSystemService()`方法取得LocationManager物件lc



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 在取得系統服務的LocationManager物件後，可以使用if條件檢查行動裝置是否已經啟用GPS
 - `if (!lc.isProviderEnabled(LocationManager.GPS_PROVIDER)) {`
 - `AlertDialog.Builder builder = new AlertDialog.Builder(this);`
 - `builder.setTitle("定位管理")`
 - `.setMessage("GPS目前狀態是尚未啟用.\n"`
 - `+"請先啟用GPS?")`
 - `.setPositiveButton("確定", null).create().show();`
 - `}`
- if條件呼叫LocationManager物件的isProviderEnabled()方法檢查是否有啟用GPS，如果沒有，就顯示訊息視窗說明需要啟用GPS定位



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- Android作業系統的權限管理是透過請求使用者授予權限，應用程式需要取得使用者的授權才能執行指定任務。
- 舊版Android只有一種權限，Android 6.0以上版本的使用權限依危險層度分成正常和危險兩種權限，其說明如下所示：
- 正常權限（normal permissions）：我們需要在AndroidManifest.xml使用<use-permission>標籤宣告權限，當在裝置的Google Play下載安裝應用程式時，就會要求使用者授予這些權限。
- 危險權限（dangerous permissions）：對於執行Android 6.0以上版本的行動裝置，部分比較危險的權限，除了安裝時需要授予權限，在第一次啟動時，還需要再次要求使用者授權使用。



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- Android應用程式如果是在Android 6.0以上版本的裝置執行，而且需要使用危險權限時，我們需要額外撰寫程式碼在執行期請求權限，首先呼叫`checkSelfPermission()`方法檢查是否已經取得授權

- `ACCESS_FINE_LOCATION`的GPS定位權限

- `if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M &&`
 - `checkSelfPermission(`
 - `Manifest.permission.ACCESS_FINE_LOCATION) !=`
 - `PackageManager.PERMISSION_GRANTED) {`
 - `requestPermissions(new`
 - `String[]{Manifest.permission.ACCESS_FINE_LOCATION},`
 - `PERMISSIONS_REQUEST_GPS);`
 - `}`



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 如果沒有取得權限，就呼叫requestPermissions()方法在執行時再次請求權限
 - requestPermissions(new
 - String[]{Manifest.permission.ACCESS_FINE_LOCATION},
 - PERMISSIONS_REQUEST_GPS);
- 上述方法的第1個參數是請求的權限字串陣列，第2個參數是請求權限的請求碼常數，我們是在onRequestPermissionsResult()方法判斷是否是請求此權限



ANDROID LBS與資料庫應用

- **LBS 定位服務**
 - 外層if條件判斷請求碼常數是否是我們請求的權限，如果是，就在內層if/else條件判斷參數的grantResults[0]陣列值，若是PERMISSION_GRANTED，即可知道是否已經取得權限



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- `@Override`
- `public void onRequestPermissionsResult(int requestCode,`
- `String[] permissions, int[] grantResults) {`
- `if (requestCode == PERMISSIONS_REQUEST_GPS) {`
- `if (grantResults[0] ==`
- `PackageManager.PERMISSION_GRANTED) {`
- `output.setText("取得權限取得GPS資訊");`
- `} else {`
- `output.setText("直到取得權限, 否則無法取得GPS資訊");`
- `}`
- `}`
- `}`



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 定位服務需要定時更新座標位置，所以需要建立 **LocationListener** 傾聽者物件，這是實作 **LocationListener** 介面的類別，必須在 **MainActivity** 活動類別實作此介面

- `public class MainActivity extends AppCompatActivity`
 - `implements LocationListener {`
 - `...`
 - `}`

- **MainActivity** 類別實作 **LocationListener** 介面的4個方法



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 使用onLocationChanged()方法，當位置改變時呼叫的介面方法，參數是目前位置的Location物件

- @Override
 - public void onLocationChanged(Location location) {
 - double lat, lng;
 - if (location != null) {
 - // 取得經緯度
 - lat = location.getLatitude();
 - lng = location.getLongitude();
 - String p = "定位提供者:" + location.getProvider();
 - output.setText(p + "\n緯度:" + lat + "\n 經度:" + lng);
 - }
 - }



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 節省電力，需要在onResume()方法註冊服務的傾聽者物件； onPause()方法取消註冊服務的傾聽者物件
- 在onResume()方法使用requestLocationUpdates()方法註冊更新位置的傾聽者物件（此方法需要try/catch例外處理敘述）
 - @Override
 - protected void onResume() {
 - super.onResume();
 - int minTime = 1000; // 毫秒
 - float minDistance = 1; // 公尺
 - }



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- try {
 - String best = lc.getBestProvider(new Criteria(), true);
 - if (best != null) {
 - lc.requestLocationUpdates(best,minTime,minDistance,this);
 - }
 - else
 - output.setText("請確認開啟GPS");
 - }
- catch(SecurityException sex) {
 - output.setText("GPS權限失敗..." + sex.getMessage()); }
- }



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 在onPause()覆寫方法取消註冊更新的傾聽者物件，以節省電力的使用，如下所示：
- @Override
- protected void onPause() {
- super.onPause();
- try {
- lc.removeUpdates(this);
- }
- catch(SecurityException sex) {
- output.setText("GPS權限失敗..." + sex.getMessage());
- }
- }



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- Android應用程式可以顯示目前行動裝置的**GPS**位置座標，並且提供按鈕來啟動設定程式來設定**GPS**
- 注意模擬器的系統映像檔需選擇**Google APIs**，如果是**Android 6.0**以上版本，需要執行期授權
- 定位服務最常見的應用是整合**Google**地圖，可以顯示目前行動裝置所在座標附近的地圖資訊，在實作上，我們可以使用內建**Google**地圖程式，或在瀏覽器網頁顯示地圖資訊



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 按下按鈕，取得目前的**GPS**座標後，啟動內建地圖程式來顯示附近地圖
- 按鈕的事件處理方法是使用目前位置的座標來啟動**Google**地圖程式

- `current = lc.getLastKnownLocation(best);`
- `if (current != null) {`
- `double lat = current.getLatitude();`
- `double lng = current.getLongitude();`
- `String p = "定位提供者:" + current.getProvider();`
- `output.setText(p + "\n緯度:" + lat + "\n經度:" + lng);`
- `}`



ANDROID LBS與資料庫應用

- **LBS 定位服務**

- 使用Intent物件啟動地圖程式，首先使用Location物件current取得經緯度座標longitude和latitude
 - float latitude = (float) current.getLatitude();
 - float longitude = (float) current.getLongitude();
 - String label = "車站";
- 字串label是顯示的標籤文字，然後建立URI字串，字串uri是座標
 - String uri = "geo:" + latitude + "," + longitude;
 - String query = latitude + "," + longitude + "(" + label + ")";
 - query = Uri.encode(query);



ANDROID LBS與資料庫應用

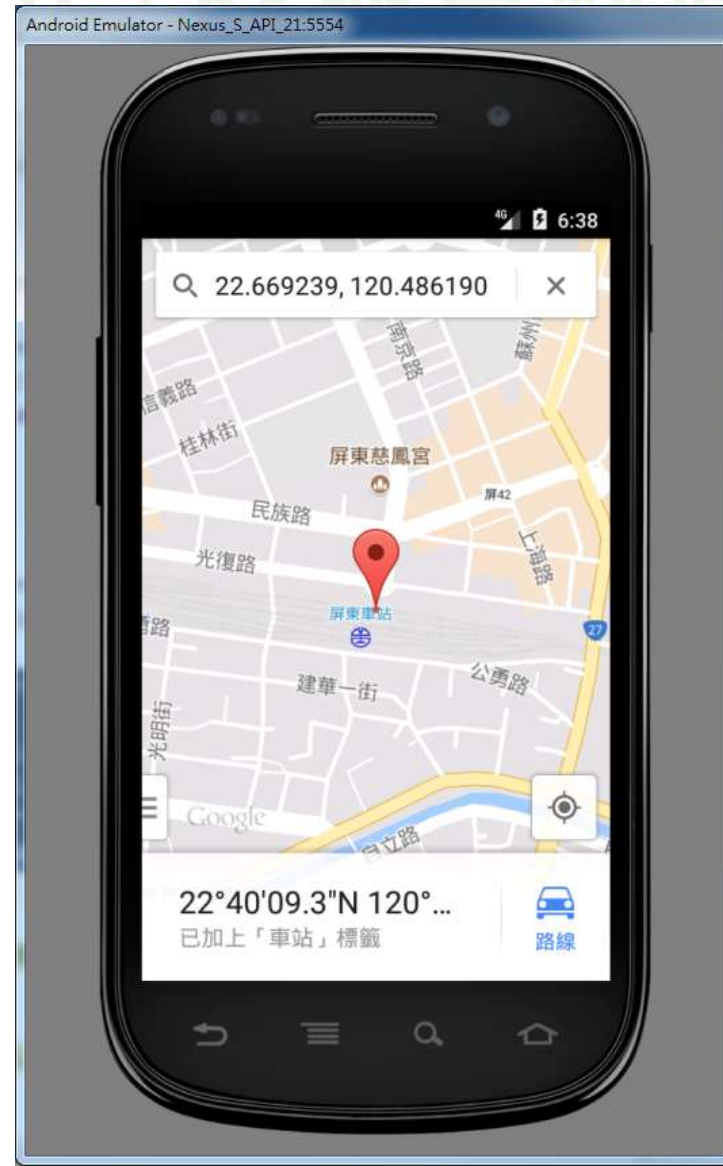
- **LBS 定位服務**

- **query**字串是在地圖上標示位置圖示和標籤文字，最後就可以組合成**URI**字串
 - `uri = uri + "?q=" + query + "?z=16";`
 - `Intent geoMap = new Intent(`
 - `Intent.ACTION_VIEW, Uri.parse(uri));`
 - `startActivity(geoMap);`
- 應用程式可以顯示目前行動裝置的**GPS**位置座標，並且提供按鈕啟動內建**Google**地圖程式，可以顯示目前座標附近的地圖資訊和標示位置



ANDROID LBS與資料庫應用

- 成果





ANDROID LBS與資料庫應用

- **LBS 地圖解碼服務**

- 地圖解碼服務（ **geocoding services** ）可以從位置名稱、郵遞區號等資訊來找出經緯度座標
- 也可以從經緯度座標找出位置名稱或地址
- 使用**Geocoder**類別，首先建立**Geocoder**物件
 - `Geocoder geocoder = new Geocoder(this, Locale.TAIWAN);`
- 建構子的參數1是活動類別自己，參數2是語系，**Local.TAIWAN**是指台灣，也可以使用**Local.getDefault()**方法傳回系統預設的語系



ANDROID LBS與資料庫應用

- **LBS 地圖解碼服務**

- Geocoder物件的getFromLocation()方法可以將經緯度座標轉換成地址資訊
- 經測試，台灣目前只能轉換成鄉鎮區和郵遞區號
 - `List<Address> listAddress = geocoder.getFromLocation(lat, lon, MAX);`
- 上述方法需要使用try/catch例外處理敘述，前2個參數是GPS座標的緯度和經度，參數3是最多傳回幾筆查詢結果的Address物件
- 傳回List<Address>集合物件，即Address地址物件的集合



ANDROID LBS與資料庫應用

- **LBS 地圖解碼服務**

- **List<Address>** 集合物件是呼叫 **get()** 方法——取出每一個 **Address** 地址物件，參數索引值是從 **0** 開始

- `Address ad = listAddress.get(i);`

- **Address** 物件常用方法

| 方法 | 定義 |
|-------------------------------|-----------|
| <code>getCountryName()</code> | 取得國家名稱 |
| <code>getCountryCode()</code> | 取得國碼 |
| <code>getPhone()</code> | 取得電話 |
| <code>getPostalCode()</code> | 取得郵遞區號 |
| <code>getAddressLine()</code> | 取得多行的地址資訊 |



ANDROID LBS與資料庫應用

- **LBS 地圖解碼服務**

- 從**Address**地址物件取出的地址可能有多行，需要使用**getMaxAddressLineIndex()**方法取得**Address**物件共有幾行地址資料
- 使用**for**迴圈重複呼叫**getAddressLine()**方法取出每一行，參數是行索引（從0開始）
 - ```
for (int j = 0; j < ad.getMaxAddressLineIndex(); j++) {
```
  - ```
    out += ad.getAddressLine(j) + " ";
```
 - ```
}
```
- 應用程式輸入經緯度座標，按【轉換成地址】鈕，可以在下方**TextView**元件顯示找到的地址資訊



# ANDROID LBS與資料庫應用

- 成果





# ANDROID LBS與資料庫應用

- **LBS 地圖解碼服務**

- 呼叫Geocoder物件的getFromLocationName()方法將地址或景點名稱字串轉換成經緯度座標
  - `List<Address> listGPSAddress = geocoder.getFromLocationName(addressName, 1);`
- 上述程式碼取得經緯度座標清單的List<Address>物件，參數1是地址或景點名稱字串，參數2最多傳回幾個座標，1就是1筆



# ANDROID LBS與資料庫應用

- **LBS 地圖解碼服務**

- 傳回座標，可以呼叫`getLatitude()`和`getLongitude()`方法取出緯度和經度
  - `double lat = listGPSAddress.get(0).getLatitude();`
  - `double lon = listGPSAddress.get(0).getLongitude();`
- `get(0)`方法取出List集合物件的第1個Address物件
- 應用程式的「景點或地址」欄輸入名稱，按「轉換成GPS座標」鈕，可以在下方顯示經緯度座標



# ANDROID LBS與資料庫應用

- LBS 地圖解碼服務

景點或地址: 台北101

轉換成GPS座標

緯度: 25.0339031  
經度: 121.5645099



# ANDROID LBS與資料庫應用

- **SQLite**資料庫

- SQLiteOpenHelper類別是幫助存取SQLite資料庫的幫助者類別（ helper class ），需要繼承此類別，透過繼承類別來建立SQLite資料表和版本管理，建立的資料庫是一個SQLiteDatabase類別的物件
- SQLiteDatabase類別已經實作存取SQLite資料庫的相關方法，可以直接透過這些方法來新增、更新和刪除資料表的記錄資料





# ANDROID LBS與資料庫應用

- **SQLite**資料庫

- 建立SQLite資料庫是繼承SQLiteOpenHelper類別來覆寫相關方法，目的是讓我們可以在SQLite資料庫新增資料表（因為資料庫的建立已經在父類別實作）
- 在StdDBHelper（自行命名）子類別需要新增建構子，覆寫onCreate()和onUpgrade()方法
  - ```
public class StdDBHelper extends SQLiteOpenHelper {
```
 - ```
 private static final String DATABASE_NAME = "Class";
```
  - ```
    private static final int DATABASE_VERSION = 1;
```
 - ```
 public StdDBHelper(Context context) {
```
  - ```
        super(context, DATABASE_NAME,
```
 - ```
 null, DATABASE_VERSION);
```
  - ```
    }
```



ANDROID LBS與資料庫應用

- SQLite資料庫

- 覆寫onCreate()和onUpgrade()方法

- @Override
 - public void onCreate(SQLiteDatabase db) {
 - db.execSQL("CREATE TABLE students (" +
 - "_id integer primary key, " +
 - "name text no null, grade real no null)");
 - }
 - @Override
 - public void onUpgrade(SQLiteDatabase db,
 - int oldVersion, int newVersion) {
 - db.execSQL("DROP TABLE IF EXISTS students");
 - onCreate(db);
 - }
 - }



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- SQLiteOpenHelper類別提供方法來開啟和關閉資料庫

| 方法 | 定義 |
|-----------------------|---|
| getReadableDatabase() | 建立或開啟（如果存在）一個唯讀資料庫 ，成功開啟傳回SQLiteDatabase物件 |
| getWritableDatabase() | 建立或開啟（如果存在）一個讀寫資料庫 ，成功開啟傳回SQLiteDatabase物件 |
| close() | 關閉開啟的資料庫 |



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- SQLiteOpenHelper類別提供方法開啟和關閉資料庫，然後使用SQLiteDatabase類別的相關方法來新增、更新和刪除資料表的記錄資料
- 在活動類別的onCreate()方法開啟資料庫
 - dbHelper = new StdDBHelper(this);
 - db = dbHelper.getWritableDatabase();
- 在建立StdDBHelper物件後，呼叫getWritableDatabase()方法取得SQLiteDatabase物件的資料庫



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- 在onStop()方法，呼叫SQLiteDatabase類別的close()方法關閉資料庫

- db.close();

- SQLiteDatabase物件使用insert()方法新增記錄，使用ContentValues類別建立欄位值

- long id;
 - ContentValues cv = new ContentValues();
 - cv.put("_id", Integer.parseInt(txtID.getText().toString()));
 - cv.put("name", txtName.getText().toString());
 - cv.put("grade", Double.parseDouble(txtGrade.getText().toString()));



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- 建立**ContentValues**物件**cv**，使用**put()**方法加入欄位值，參數**1**是欄位名稱字串，參數**2**是欄位值
- 建立後呼叫**insert()**方法新增記錄
 - `id = db.insert(DATABASE_TABLE, null, cv);`
- **SQLiteDatabase**物件使用**update()**方法更新記錄
- 需要**ContentValues**類別建立更新的欄位值
 - `int id = Integer.parseInt(txtID.getText().toString());`
 - `ContentValues cv = new ContentValues();`
 - `cv.put("grade", Double.parseDouble(`
 - `txtNewGrade.getText().toString()));`
 - `count = db.update(DATABASE_TABLE, cv, "_id=" + id, null);`



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- `update()`方法的傳回值是影響的記錄數，參數1是資料表名稱，參數2是ContentValues物件，參數3是WHERE子句的更新條件
- 如果參數擁有條件字串，最後一個參數是條件的參數值
- **SQLiteDatabase**物件可以使用`delete()`方法刪除記錄
 - `int id = Integer.parseInt(txtID.getText().toString());`
 - `count = db.delete(DATABASE_TABLE, "_id=" + id, null);`
- 方法的傳回值是影響的記錄數，參數1是資料表名稱，參數2是WHERE子句的刪除條件
- 如果參數擁有條件字串，最後一個參數是條件的參數值



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- SQLiteDatabase物件可以使用rawQuery()方法執行SQL指令來查詢記錄
 - Cursor c = db.rawQuery(sql, null);
- rawQuery()方法的參數1是SQL查詢指令
- 使用參數的SQL指令，可以在參數2指定參數值，方法傳回查詢結果記錄資料的Cursor物件
- 是一個查詢結果的記錄集合，擁有指標可以一筆筆取出每一筆記錄



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- 取得Cursor物件後，可以呼叫getColumnNames()方法取得欄位名稱
 - colNames = c.getColumnNames();
 - for (int i = 0; i < colNames.length; i++)
 - str += colNames[i] + "\t\t";
 - str += "\n";
- for迴圈可以顯示欄位名稱的標題列，移動記錄指標來取得查詢結果的每一筆記錄
 - c.moveToFirst();
 - for (int i = 0; i < c.getCount(); i++) {
 - str += c.getString(0) + "\t\t";
 - str += c.getString(1) + "\t\t";
 - str += c.getString(2) + "\n";
 - c.moveToNext();
 - }



ANDROID LBS與資料庫應用

- **SQLite**資料庫

- 呼叫**moveToFirst()**方法移至第1筆記錄，**getCount()**方法傳回記錄數
- 使用**for**迴圈來走訪每一筆記錄，**moveToNext()**方法可以移至下一筆
- 輸入學生資料的學號、姓名和成績後，按下方按鈕，可以新增、更新（利用學號條件，去更新成績）和刪除記錄資料（利用學號條件），按「顯示」鈕可以顯示資料表的所有記錄資料



ANDROID LBS與資料庫應用

- 成果





 **THE END**