

구글 안드로이드 고급 프로그래밍

강대기

동서대학교 컴퓨터정보공학부

학습 목표 (1)

- GPS 장치를 통해 위치를 인식하는 방법에 대해서 알아본다.
- 가속도계에 대해서 알아본다.
- 지도를 나타내는 맵 뷰에 대해 알아본다.
- 웹 뷰와 맵 뷰를 결합함으로써, 여러 서비스들을 결합하는 매시업(mashup)에 대해 알아본다.

학습 목표 (2)

- SQLite가 뭔지 알고, 이를 사용할 줄 안다.
- Structured Query Language (SQL)의 기본적인 사용법들을 안다.
- SQLite을 이용해 기본적인 데이터베이스 응용 프로그램을 작성할 수 있다.
- 행을 추가하는 기본적인 데이터베이스 응용 프로그램을 작성할 수 있다.
- 쿼리를 실행하는 기본적인 데이터베이스 응용 프로그램을 작성할 수 있다.
- 쿼리 결과를 보여주는 기본적인 데이터베이스 응용 프로그램을 작성할 수 있다.

학습 목표 (3)

- 데이터 바인딩을 통해 데이터 소스에 해당하는 데이터 베이스와 뷰에 해당하는 액티비티를 연결한 데이터베이스 응용 프로그램을 작성할 수 있다.
- 안드로이드 내의 다른 어플리케이션의 데이터에 접근하기 위해 제공되는 ContentProvider를 사용할 수 있다.
- 자신의 어플리케이션에서 다른 어플리케이션으로의 데이터 제공을 위한 ContentProvider를 구현할 수 있다.

차례

- 위치, 위치, 위치
- 센서를 최대한으로 설정하기
- 조감도
- 웹 뷰와 맵 뷰
- SQLite 소개
- Structured Query Language (SQL) 기본
- 헬로, 데이터베이스
- 데이터 바인딩
- ContentProvider 사용하기
- ContentProvider 구현하기
- 프로그램의 배포 및 안드로이드 마켓
- 요약
- 퀴즈
- 연습문제

위치, 위치, 위치

- 위치 정보
 - Global Positioning System (GPS) – GPS 소스가 어떤 것인가에 따라 해상도가 다르지만, 2009년 현재 기본적으로 10~25m 정도의 오차를 가짐
 - 내비게이션은 자체 알고리즘으로 이를 보정함
 - 그 외에도 Differential GPS 등과 같은 다양한 구조 개선이 있음
 - 2000년 5월 이전에는, 미국의 군사적 보안을 이유로, 민간용은 일부러 100m 오차를 가지게 했었음 (Selective Availability)
 - 한국에서 위치정보를 제공하는 사업자에게는 국가 기관이 위치 정보 내역을 요구할 수 있음 (아이폰 출시 문제, 개인정보 보호 문제)
 - 인근 휴대폰 기지국
 - Wifi 핫스팟
- LocationTest 프로젝트 (org.example.locationtest)
- AndroidManifest.xml 에 다음 추가
 - `<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />`
 - `<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />`

res/layout/main.xml

```
<ScrollView
  xmlns:android="http://schemas.android.com/a
  pk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <TextView
    android:id="@+id/output"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
</ScrollView>
```

LocationTest.onCreate()

@Override

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

```
  
    mgr = (LocationManager) getSystemService(LOCATION_SERVICE);  
    output = (TextView) findViewById(R.id.output);
```

```
  
    log("Location providers:");  
    dumpProviders();
```

```
  
    Criteria criteria = new Criteria();  
    best = mgr.getBestProvider(criteria, true);  
    log("\nBest provider is: " + best);
```

```
  
    log("\nLocations (starting with last known):");  
    Location location = mgr.getLastKnownLocation(best);  
    dumpLocation(location);
```

```
}
```


LocationTest.onCreate()

- getSystemService(LOCATION_SERVICE)는 LocationManager 클래스 반환
- dumpProviders() 는 위치 정보 제공자들 출력
- getBestProvider(criteria, true)에서 원래는 비용, 전력, 정확성 등에 대해 선정 기준을 조정할 수 있음

위치 업데이트

- `LocationManager.requestLocationUpdates()` 호출하면 위치 변화를 안드로이드가 알려줌
- 배터리 절약을 위해 프로그램이 foreground 인 경우에만 위치 업데이트 - `onResume()`과 `onPause()`

@Override

```
protected void onResume() {  
    super.onResume();  
    // Start updates (doc recommends delay >= 60000 ms)  
    mgr.requestLocationUpdates(best, 15000, 1, this);  
}
```

@Override

```
protected void onPause() {  
    super.onPause();  
    // Stop updates to save power while app paused  
    mgr.removeUpdates(this);  
}
```

리스너 객체 vs. 액티비티에서 구현

- 리스너 객체를 새로 만드는 대신, 액티비티에 참조를 넘기면 1KB 메모리 절약

```
public void onLocationChanged(Location location) {  
    dumpLocation(location);  
}
```

```
public void onProviderDisabled(String provider) {  
    log("\nProvider disabled: " + provider);  
}
```

```
public void onProviderEnabled(String provider) {  
    log("\nProvider enabled: " + provider);  
}
```

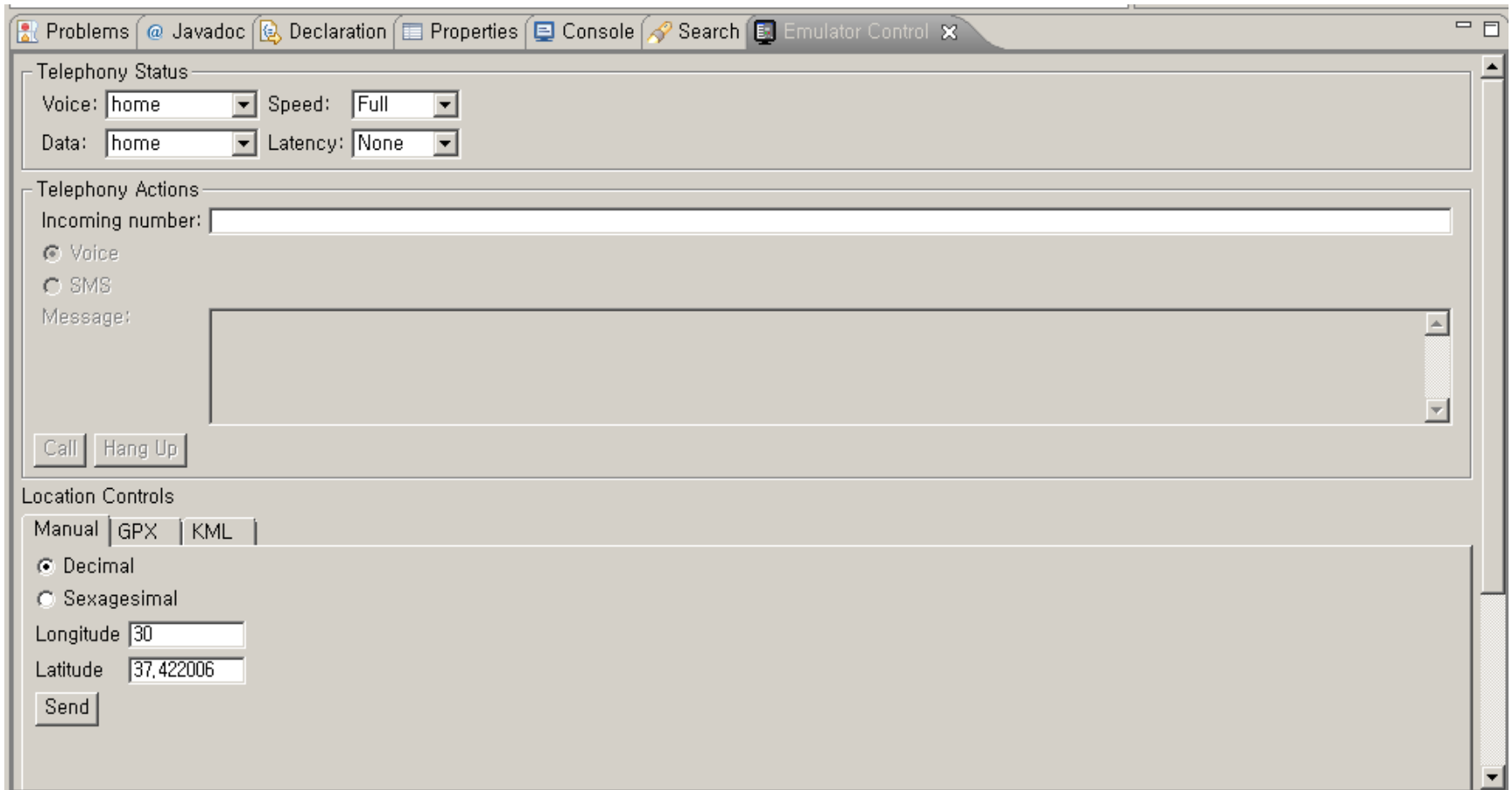
```
public void onStatusChanged(String provider, int status,  
    Bundle extras) {  
    log("\nProvider status changed: " + provider + ", status=" +  
        + S[status] + ", extras=" + extras);  
}
```

- 장치 위치 변경을 제공자가 알아챌 때마다, onLocationChanged() 메서드가 호출됨

에뮬레이터는 어떻게 하나?

- 근본적으로 에뮬레이터는 Fake GPS 제공자를 사용함
- Dalvik Debug Monitor Service (DDMS) 를 이용하는 에뮬레이터 컨트롤에서 경도와 위도 입력
 - 창 > 보기 뷰 > 기타 > 안드로이드 > 에뮬레이터 컨트롤
 - Window > Show View > Other > Android > Emulator Control
- 안드로이드 에뮬레이터 콘솔 (telnet 127.1 5554)
- 에뮬레이터 컨트롤에서 Google Earth 에서 출력된 KML 파일 입력
- DDMS 를 이용하는 외부 프로그램 사용

에뮬레이터 컨트롤에서 GPS 값 입력

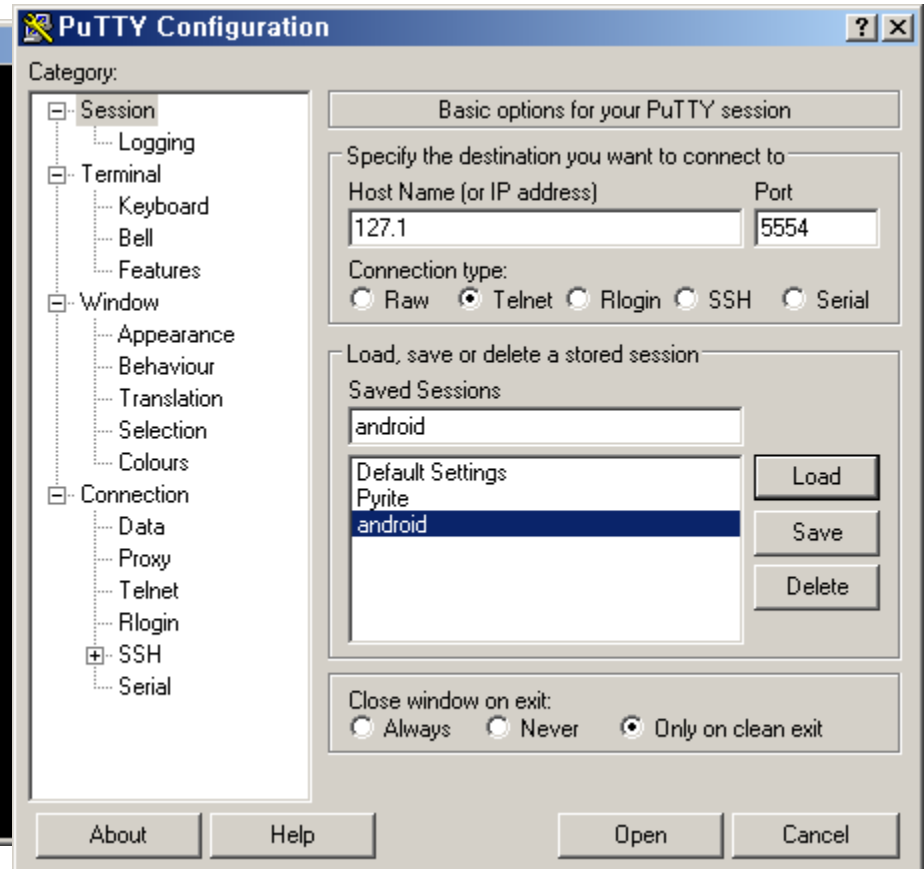


에뮬레이터 콘솔에서 GPS 값 입력

```
127.1 - PuTTY
OK
help
KO: unknown command, try 'help'
help
Android console command help:

  help|h|?      print a list of commands
  event        simulate hardware events
  geo          Geo-location commands
  gsm          GSM related commands
  kill         kill the emulator instance
  network      manage network settings
  power        power related commands
  quit|exit    quit control session
  redir        manage port redirections
  sms          SMS related commands
  avd          manager virtual device state
  window       manage emulator window

try 'help <command>' for command-specific help
OK
geo fix 10 10
OK
```



LocationTest 실행 결과

5554:BigAVD

LocationTest

Location[unknown]

Location[mProvider=gps, mTime=1253672703000, mLatitude=37.422006, mLongitude=-122.084095, mHasAltitude=true, mAltitude=0.0, mHasSpeed=false, mSpeed=0.0, mHasBearing=false, mBearing=0.0, mHasAccuracy=false, mAccuracy=0.0, mExtras=null]

Location[mProvider=gps, mTime=1253672718000, mLatitude=37.422006, mLongitude=30.0, mHasAltitude=true, mAltitude=0.0, mHasSpeed=false, mSpeed=0.0, mHasBearing=false, mBearing=0.0, mHasAccuracy=false, mAccuracy=0.0, mExtras=null]

Location[mProvider=gps, mTime=1253664000000, mLatitude=10.001666666666667, mLongitude=10.001666666666667, mHasAltitude=false, mAltitude=0.0, mHasSpeed=false, mSpeed=0.0, mHasBearing=false, mBearing=0.0, mHasAccuracy=false, mAccuracy=0.0, mExtras=null]

오후 5:51

Navigation pad: Camera, Mute, Volume, Power, Call, Home, MENU, Back, Search

1 !	2 @	3 #	4 \$	5 %	6 ^	7 &	8 *	9 (0)
Q	W ~	E "	R `	T {	Y }	U -	I _	O + =
A S \	D ' /	F [G]	H <	J >	K ;	L :	DEL
Home	Z X	C V	B N	M .	↵			
ALT	SYM	@		→	/ ?	,		ALT

센서를 최대한으로 설정하기

- 레이싱 게임의 경우의 예
 - 닌텐도 DS – 버튼을 이용해서 좌회전, 우회전
 - 안드로이드폰, 아이폰, 닌텐도 Wii – Accelerometer(가속도계) 사용
- 참여를 유발하는 센서
 - SENSOR_ACCELEROMETER – x,y,z 좌표의 가속 측정
 - SENSOR_LIGHT – 주위 영역의 밝기를 알려줌
 - SENSOR_MAGNETIC_FIELD – x,y,z, 축에 전자 극을 반환함
 - SENSOR_ORIENTATION – 장치가 한쪽으로 쏠리고, 던져지고, 돌려지는 것을 측정함
 - SENSOR_ORIENTATION_RAW – SENSOR_ORIENTATION에서 필터링이 빠진 것
 - SENSOR_PROXIMITY – 센서와 객체 사이의 거리를 알려줌
 - SENSOR_TEMPERATURE – 주위의 온도를 측정함
 - SENSOR_TRICORDER – 장치를 영화 스타트렉의 트라이코더와 같이 만듦
- SensorManager 클래스는, LocationManager와 유사하나, 업데이트가 몇백분의 일 초 정도로 빠르게 됨.

SensorTest.java

- 센서에 접근하기 위해서는 getSystemService() 메서드 호출함

```
private SensorManager mgr;
private TextView output;
private List<Sensor> sensorList;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    // ...
    mgr = (SensorManager)
    getSystemService(SENSOR_SERVICE);
    output = (TextView) findViewById(R.id.output);
}
```

SensorTest.java

- 센서 서비스는 값이 변경될 때마다 `onSensorChanged()`를 호출함

```
for (int i = 0; i < event.values.length; i++) {  
  
    builder.append(event.values[i]);  
  
}
```

센서들을 사용하는 프로그래밍의 경우

- 모든 센서는 부동 소수점 명령을 반환함
- 반환되는 배열의 크기는 센서의 특성에 따라 다름
- 센서, 특히 가속도계의 값을 쓸모있는 정보로 하려면 많은 어려움이 있음
 - 가속도계는 수치가 불안함. 평활화(smoothing)를 통해 부드럽게 만들어야 하지만, 지나치면 인터페이스가 처지는 느낌이 남.
 - 센서 숫자는 임의의 개수가 함께 나오는 데, 한 번에 여러 개 나오기도 하고, 잠시 멈췄다가 다시 여럿이 전송되기도 함
 - 사용자가 다음에 무엇을 입력할지 미리 예측해 한 발 앞서야 함. 연이은 세 개의 값이 오른쪽이라면, 다음 값도 어느 정도 예측하고 있어야 함.
- 센서의 가장 난이도 있는 사용의 예는, 사용자가 장치를 움직이는 것과 스크린에서 일어나는 반응이 일대일로 연결되는 액션 게임 (에뮬레이터 상에서는 테스트가 어려움)

Sensor Simulator

- SensorTest 프로그램을 그냥 실행하면 아무런 결과도 나오지 않음
- www.openintents.org 에서 대체 센서 API를 제공함
- www.openintents.org 의 Sensor Simulator 를 다운받아 에뮬레이터와 연결하면, 시뮬레이터에서는 가상 폰의 이미지를 보여주고, 마우스로 움직이게 해주며, 그 움직임을 에뮬레이터에 있는 안드로이드 프로그램에 넘김
- 애플 맥북에는 센서가 내장되어 있으며, 센서가 없는 컴퓨터의 경우, 닌텐도 Wii를 연결해도 됨 - 대신 프로그램 소스를 고쳐야 함

Sensor Simulator 설치 방법

- 최신 버전의 sensorsimulator-버전.zip 을 다운받아 임의의 디렉토리에 풀음
(<http://code.google.com/p/openintents/downloads/list>)
- 다운받아 툰 시뮬레이터의 bin 디렉토리에 있는 일반 자바 응용 프로그램인 SensorSimulator.jar 를 실행한 후, IP 주소, 포트 및 센서 옵션들을 설정
- SensorSimulatorSettings.apk 를 에뮬레이터에 설치한 후, SensorSimulatorSettings에서 동일하게 IP 주소, 포트, 및 센서들의 옵션 설정
- samples 디렉토리에 있는 SensorDemo (OISensorDemo) 를 이클립스에 임포트(import)하여, 버전을 1.5 로 설정하고, 안드로이드에서 실행함

Sensor Simulator 사용예

The image shows two windows from a desktop environment. The left window is titled "SensorSimulator" and contains the "OpenIntents Sensor Simulator" interface. It features three sliders for Yaw, Pitch, and Roll, each ranging from -180 to 180 degrees. Below the sliders is a "Settings" panel with two sections: "Supported sensors" and "Enabled sensors". In the "Supported sensors" section, "accelerometer", "magnetic field", and "orientation" are checked, while "temperature" is unchecked. In the "Enabled sensors" section, "accelerometer" and "magnetic field" are checked. To the left of the sliders, there are radio buttons for "yaw & pitch", "roll & pitch", and "move", with "yaw & pitch" selected. A "Socket" field contains the value "8011" and a "Set" button. A log window at the bottom left shows the following text: "ALL SENSORS DISABLED!", "Incoming connection opened.", "Incoming connection opened.", "accelerometer: 0.00, -9.22, -3.35", "magnetic field: 9.82, -33.22, -34.97", and "orientation: 350.00, -70.00, 0.00".

The right window is titled "5554:Android 1.5" and displays an Android emulator. The emulator screen shows a "Sensor demo" application with the following text: "Hello World, SensorDemoActivity!", "Accelerometer: 0.0, -9.215237, -3.3540719", "Compass: 9.821311, -33.2246, -34.967487", and "Orientation: 350.0, -70.0, 0.0". The emulator also shows a virtual keyboard and a red phone icon.

조감도

- Google 맵스 - Ajax 기술을 통해 (자바스크립트와 XMLHttpRequest 객체를 이용해 플러그인 없이) 어떤 브라우저에서도 드래그할 수 있고, 확대/축소할 수 있는, 지도 뷰어
- 맵뷰 컨트롤 임베딩
 - MyMap / org.example.mymap / MyMap / MyMap
 - Build target 은 Google API 1.6 으로 설정
 - AVD도 Google API 1.6 용으로 디바이스 하나 만듦
- 레이아웃을 맵 뷰로 대체함(main.xml)
 - MapAPIKey 부분은 사용자가 스스로 API Key를 받아서 바꾸어 주어야 함

main.xml

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/
android"

android:id="@+id/frame"

android:orientation="vertical"

android:layout_width="fill_parent"

android:layout_height="fill_parent">

<com.google.android.maps.MapView

android:id="@+id/map"

android:apiKey="MapAPIKey"

android:layout_width="fill_parent"

android:layout_height="fill_parent"

android:clickable="true" />

</LinearLayout>

Google MAP API 디버깅용 Key 얻기

- 디버깅용 API Key
 - 이클립스를 이용할 경우 debug.keystore의 위치는 Window > Preference > Android > Build 탭의 default debug keystore 에 있음
 - `keytool -list -keystore debug.keystore` 위치 를 입력하여 MD5 fingerprint 생성
 - <http://code.google.com/intl/ko/android/maps-api-signup.html> 에 가서 Terms and Conditions 를 체크하고, MD5 fingerprint 를 입력하고, Generate API Key 버튼 누름

```
C:\Android\android-sdk-windows-1.5_r3\tools>keytool -list -keystore
C:\Users\DK\.android\debug.keystore
keystore 암호를 입력하십시오:
```

```
Keystore 유형: JKS
Keystore 공급자: SUN
```

Keystore에는 1 항목이 포함되어 있습니다.

```
androiddebugkey, 2009. 9. 10, PrivateKeyEntry,
인증서 지문(MD5): CB:76:29:11:B5:72:8F:08:E6:70:86:92:DF:23:EC:C9
```

```
C:\Android\android-sdk-windows-1.5_r3\tools>
```

AndroidManifest에 MapView 추가

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.example.mymap"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
        android:name="android.permission.INTERNET" />
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".MyMap"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <uses-library android:name="com.google.android.maps" />
    </application>
    <uses-sdk android:minSdkVersion="4" />
</manifest>
```

MyMap 실행 화면



MyMap 클래스 개요

- MapActivity 확장
- findViewById()로 지도에 대한 접근을 얻고, mapView.getController()로 컨트롤러를 얻음
- mapView.setBuiltInZoomControls()로 확대/축소 컨트롤 설정
- initMyLocation() 내에서 MyLocationOverlay 포인터를 얻고 MyLocationOverlay.enableMyLocation()로 현재 위치의 업데이트를 받도록 함
- MyLocationOverlay.runOnFirstFix() 는 위치 제공자로부터 처음 정보를 제공받으면 오버레이가 무엇을 해야 하는지 지정해 줌
- 실행하면, 폰의 경우에는 붉은 점이 사용자의 위치를 따라다님. 에뮬레이터의 경우에는 GPS 정보를 입력해 줘야 함.
- 구글 맵스에서 지질학적 정보와 이미지들은 안드로이드 코어와 같이 오픈 소스로 제공되는 것이 아니라, 유료 지도 데이터 제공자들로부터 구글을 통해 다른 방식으로 제공되는 것이므로 예를 들면 android.maps 패키지 아래 있지 않고, com.google.android.maps 패키지로 제공됨

MyMap.java (1 / 2)

```
package org.example.mymap;

import android.os.Bundle;

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

public class MyMap extends MapActivity {
    private MapView map;
    private MapController controller;

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

    @Override
    protected boolean isRouteDisplayed() {
        // Required by MapActivity
        return false;
    }
}
```

MyMap.java (2/2)

```
/** Find and initialize the map view. */
private void initMapView() {
    map = (MapView) findViewById(R.id.map);
    controller = map.getController();
    map.setSatellite(true);
    map.setBuiltInZoomControls(true);
}

/** Start tracking the position on the map. */
private void initMyLocation() {
    final MyLocationOverlay overlay = new MyLocationOverlay(this, map);
    overlay.enableMyLocation();
    //overlay.enableCompass(); // does not work in emulator
    overlay.runOnFirstFix(new Runnable() {
        public void run() {
            // Zoom in to current location
            controller.setZoom(8);
            controller.animateTo(overlay.getMyLocation());
        }
    });
    map.getOverlays().add(overlay);
}

}
```

웹 뷰와 맵뷰

- 맵 뷰에 웹 뷰를 추가함
- main.xml 에서 `android:layout_height="150px"` 으로 고정해서 설정 후, 하단에 “뷰 안의 웹” 예제를 가져다 삽입
- `MyMapWeb.onCreate()`에 “뷰 안의 웹” 예제를 가져다 삽입
- “뷰 안의 웹” 예제의 `openBrowser()` 추가

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/frame" android:orientation="vertical"
  android:layout_width="fill_parent" android:layout_height="fill_parent">
  <com.google.android.maps.MapView
    android:id="@+id/map" android:apiKey=" MapAPIKey"
    android:layout_width="fill_parent" android:layout_height="150px"
    android:clickable="true" />
  <LinearLayout android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <EditText android:id="@+id/url_field" android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1.0" android:lines="1" />
    <Button android:id="@+id/go_button" android:layout_width="wrap_content"
      android:layout_height="wrap_content" android:text="@string/go_button" />
  </LinearLayout>
  <WebView
    android:id="@+id/web_view" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:layout_weight="1.0" />
</LinearLayout>
```


MyMapWeb.java (1/2)

```
package org.example.mymap;
...
public class MyMapWeb extends MapActivity {
    private MapView map;
    private MapController controller;
    private EditText urlText;
    private Button goButton;
    private WebView webView;

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

        urlText = (EditText) findViewById(R.id.url_field);
        goButton = (Button) findViewById(R.id.go_button);
        webView = (WebView) findViewById(R.id.web_view);

        // Setup event handlers
        goButton.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {        openBrowser();    }
        });
        urlText.setOnKeyListener(new OnKeyListener() {
            public boolean onKey(View view, int keyCode, KeyEvent event) {
                if (keyCode == KeyEvent.KEYCODE_ENTER) {        openBrowser();        return true;    }
                return false;
            }
        });
    }
}
```

MyMapWeb.java (2/2)

```
/** Open a browser on the URL specified in the text box */
private void openBrowser() { webView.loadUrl(urlText.getText().toString()); webView.requestFocus(); }

/** Find and initialize the map view. */
private void initMapView() {
    map = (MapView) findViewById(R.id.map);
    controller = map.getController();
    map.setSatellite(true);
    map.setBuiltInZoomControls(true);
}

/** Start tracking the position on the map. */
private void initMyLocation() {
    final MyLocationOverlay overlay = new MyLocationOverlay(this, map);
    overlay.enableMyLocation();
    //overlay.enableCompass(); // does not work in emulator
    overlay.runOnUiThread(new Runnable() {
        public void run() {
            // Zoom in to current location
            controller.setZoom(8);
            controller.animateTo(overlay.getMyLocation());
        }
    });
    map.getOverlays().add(overlay);
}

@Override
protected boolean isRouteDisplayed() { // Required by MapActivity
    return false;
}
}
```

MyMapWeb 실행 화면



SQLite 소개

- 리처드 힙 (D. Richard Hipp) 박사가 2000년에 개발한 작지만 강력한 데이터베이스 엔진
- 안드로이드, 아이폰, 심비안폰, 파이어폭스 브라우저, 스카이프, PHP, 어도비 AIR, 맥 OSX, 솔라리스 등에 사용됨
- 인기 있는 이유
 - 무료, 작은 크기, 설치나 관리가 필요없음
- SQLite 데이터베이스는 하나의 파일임 - 실은 대부분의 임베디드 데이터베이스가 그러함
- /data/data/package_name/databases 에 파일 저장
- adb 나 이클립스의 파일 익스플로러 (창 > 뷰 보기 > 기타 > 안드로이드 > 파일 익스플로러) 에서 파일을 보고, 이동시키고, 삭제할 수 있음
- 프로그램에서 이 데이터베이스 파일을 액세스하려면 자바 입출력 루틴을 호출하는 대신, SQL 구문을 실행시키면 됨
- 안드로이드는 도우미 클래스와 메서드를 통해 SQL의 복잡한 부분을 숨기긴 하지만, 제대로 사용하려면 SQL을 알아야 함

감동적인 SQLite 라이선스

1. 선을 행하고 악을 멀리하라
 - May you do good and not evil.
 2. 자신을 용서하고 남을 용서하라
 - May you find forgiveness for yourself and forgive others.
 3. 서로 나누며 자신이 베푼 만큼만 취하라
 - May you share freely, never taking more than you give.
- SQLite은 오픈 소스가 아니라 공개 도메인 소프트웨어
 - 소프트웨어 산업의 현재 관행으로는 오픈 소스가 되려면 오픈 소스를 위한 라이선스를 가지고 있어야 함
 - <http://www.opensource.org/licenses/index.html>
 - 오픈 소스 사용자는 그 라이선스에 동의해야 사용할 수 있음
 - 공개 도메인 소프트웨어인 SQLite는 동의해야 하는 라이선스라는 최소한의 제약 자체도 없음
 - 따라서 엄밀히 말하면 오픈 소스조차도 넘어서는 말 그대로 공기처럼 공짜로 사용할 수 있는 소프트웨어임

SQL 기본

- SQL 구문에는 다음과 같은 것들이 있음
 - Data Definition Language (DDL)
 - 수정 (Modification)
 - 쿼리 (Query)

Data Definition Language (DDL)

- 데이터베이스에는 여러 개의 테이블들이 있음
- 테이블에는 여러 행들이 있음
- 행들에는 여러 열들이 있음
- 각 열들은 이름과 데이터 타입을 가짐
- DDL로 테이블과 열의 명칭들을 정의함
- 다음 구문은 열이 세 개인 테이블 생성

```
CREATE TABLE mytable (  
  _id INTEGER PRIMARY KEY AUTOINCREMENT,  
  name TEXT,  
  phone TEXT  
);
```

Data Definition Language (DDL)

- 한 열은 PRIMARY KEY 로 지정되는 데, 대부분의 데이터베이스 실제 응용에서는 그 행을 식별하는 고유 숫자가 사용됨
- AUTOINCREMENT 는 각 레코드의 키에 1을 더해 고유의 키 값을 가지게 해 줌
- 안드로이드에서 SQLite이 쓰일 때, 관습적으로 첫 열은 _id라고 명명함 - 안드로이드의 ContentProvider 에서 사용할 때 필요함
- SQLite의 열 타입은 강제적인 것이 아닌 힌트에 불과함 - 즉 타입 체킹을 안한다는 뜻으로 정수 열에 문자를 저장하거나, 그 반대도 아무 문제 없음 (이건 일부러 그렇게 한 게 아니라 기능을 구현하지 않은 것 뿐으로 보이나, 임베디드 데이터베이스에서 타입 체킹의 비용이 비싼 점을 감안하면 이해가 되기도 함)

수정 (Modification)

- SQL 데이터베이스에 레코드를 삽입, 삭제, 업데이트 하는 구문들을 의미함
- 몇 개의 전화 번호를 추가하는 구문은 다음과 같음
 - `INSERT INTO TABLE mytable VALUES (null, '강대기', '320-1724');`
 - `INSERT INTO TABLE mytable VALUES (null, '강준서', '320-1725');`
 - `INSERT INTO TABLE mytable VALUES (null, 'Ian Kang', '320-1726');`
- `CREATE TABLE` 구문에서 사용된 순서대로 값이 입력됨
- `_id`는 SQLite에서 직접 값을 찾아 넣을 것이므로 널 (NULL) 값을 넣어 둬

쿼리 (Query)

- 데이터가 일단 테이블에 로드되고 나면, SELECT 구문을 사용해 쿼리를 테이블에 실행시킴
- 예를 들어 세번째 값을 원한다면, 다음과 같이 함
 - `SELECT * FROM mytable WHERE (_id=3);`
- 데이터베이스가 content addressable 한 특징이 중요하므로, 위와 같이 특정 위치보다는 이름으로 번호를 검색하게 됨
 - `SELECT name, phone FROM mytable WHERE (name LIKE "%Ian%");`
- SQL은 대소문자 구분을 하지 않음

헬로, 데이터베이스

- 데이터베이스에 레코드를 저장한 뒤 이를 나중에 다시 보여주는 Events 라는 작은 응용 프로그램
- 프로젝트 생성
 - Events, org.example.events, Events, Events
- 데이터베이스를 설명하는 상수 – Constants 인터페이스
- SQLiteOpenHelper 도우미 클래스를 확장한 EventData 클래스로 데이터베이스의 생성과 버전 관리
- 이벤트를 저장하고 그 이벤트를 TextView로 보여주는 메인 프로그램 정의

Constants 인터페이스

- 데이터베이스를 설명하는 상수 몇 개를 저장할 장소로 Constants 인터페이스 사용 (자바에서 상수를 정의하는 방법)
- 각 이벤트는 events 테이블 안의 행으로 저장되며, 각 행은 _id, time, title 열이 있음
- _id는 primary key 이며 BaseColumns 인터페이스에 정의되어 있음
- time과 title은 타임스탬프와 이벤트 제목으로 각기 사용됨
- Java 5부터는 정적 임포트가 가능
- 정적 임포트에 대한 이클립스에서의 지원은 미약할 수도 있음

```
/src/org/example/events/Constants.j
```

```
ava
```

```
import android.net.Uri;
```

```
import android.provider.BaseColumns;
```

```
public interface Constants extends BaseColumns {  
    public static final String TABLE_NAME = "events";
```

```
    // Columns in the Events database
```

```
    public static final String TIME = "time";
```

```
    public static final String TITLE = "title";
```

```
}
```

SQLiteOpenHelper 사용하기

- EventsData 라는 helper 클래스를 만들어 데이터베이스 응용 프로그램을 구현 (데이터베이스 생성과 버전 관리)
- SQLiteOpenHelper 클래스를 확장
 - 생성자와 두 개의 메서드 구현
- DATABASE_NAME – 우리가 사용할 데이터베이스 이름
- DATABASE_VERSION – 우리가 사용할 데이터베이스 버전
- 최초에는 데이터베이스에 접근을 시도하면 SQLiteOpenHelper는 데이터베이스가 없다는 걸 감지하고 onCreate()를 호출해 데이터베이스를 만듦 – CREATE TABLE 구문 실행
- 접근한 데이터베이스가 버전 숫자에 근거해 오래된 것이면, onUpgrade() 메서드 호출 – 여기서는 간단히 테이블을 삭제해 버리지만, ALTER TABLE 명령으로 존재하는 테이블에 열을 추가할 수도 있음

/src/org/example/events/EventsData.

java

```
public class EventsData extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "events.db";
    private static final int DATABASE_VERSION = 1;

    /** Create a helper object for the Events database */
    public EventsData(Context ctx) {
        super(ctx, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + " (" + _ID
            + " INTEGER PRIMARY KEY AUTOINCREMENT, " + TIME
            + " INTEGER," + TITLE + " TEXT NOT NULL);");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion,
        int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```

메인 프로그램 정의하기 (Events.java)

- Events 프로그램은 로컬 SQLite 데이터베이스를 이용하여 이벤트를 저장하고 그 이벤트를 TextView 에 문자열로 보여줌
- 화면에 너무 많은 이벤트가 있는 경우에 대비해서 ScrollView로 감쌈

main.xml

```
<ScrollView
  xmlns:android="http://schemas.android.com/a
  pk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <TextView
    android:id="@+id/text"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
</ScrollView>
```

Events.java의 onCreate()

- setContentView로 메인 뷰 설정
- EventsData로 데이터베이스 인스턴스 설정
- addEvent()로 새 이벤트 추가
- getEvents()로 데이터베이스에서 이벤트 목록을 Cursor로 받음
- showEvents()로 이벤트들을 화면에 보여줌
- 데이터베이스에 행을 추가하고 읽어들이어 보여주는 부분은 try-catch-finally 블록으로 묶음
- finally 블록에서 데이터베이스를 SQLiteDatabase.close()로 닫음

Events.java

```
public class Events extends Activity {
    private static String[] FROM = { _ID, TIME, TITLE, };
    private static String ORDER_BY = TIME + " DESC";
    private EventsData events;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        events = new EventsData(this);
        try {    addEvent("Hello, Android!");    Cursor cursor = getEvents();    showEvents(cursor);    } finally {    events.close();    }
    }

    private void addEvent(String string) {
        SQLiteDatabase db = events.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(TIME, System.currentTimeMillis());
        values.put(TITLE, string);
        db.insertOrThrow(TABLE_NAME, null, values);
    }

    private Cursor getEvents() {
        SQLiteDatabase db = events.getReadableDatabase();
        Cursor cursor = db.query(TABLE_NAME, FROM, null, null, null, null, ORDER_BY);
        startManagingCursor(cursor);
        return cursor;
    }

    private void showEvents(Cursor cursor) {
        StringBuilder builder = new StringBuilder( "Saved events:\n");
        while (cursor.moveToNext()) {
            long id = cursor.getLong(0);
            long time = cursor.getLong(1);
            String title = cursor.getString(2);
            builder.append(id).append(": ");
            builder.append(time).append(": ");
            builder.append(title).append("\n");
        }
        TextView text = (TextView) findViewById(R.id.text);
        text.setText(builder);
    }
}
```

행 추가하기 (addEvents())

- addEvents() 메서드를 통해 데이터베이스에 새로운 행을 추가함
- 시간은 현재 시간
- 제목은 주어진 string 변수
- SQLiteOpenHelper를 확장한 EventsData가 멤버로 가지고 있는 SQLiteDatabase 객체를 받아서, insertOrThrow() 메서드로 레코드를 추가함
- 다만 쓰기 작업을 해야 하므로, getWritableDatabase() 메서드로 객체를 받아야 함
- 레코드에 해당하는 ContentValues를 구성하는 방식은 동일함

쿼리 실행하기 (getEvents())

- 역시 행 추가하기와 비슷한 방법으로 SQLiteDatabase 객체를 받음. 다만 읽기 작업을 하므로, getReadableDatabase()로 받음
- SQLiteDatabase.query()를 통해 SELECT 문 수행. ORDER_BY로 레코드를 새 것부터 옛날 것 순으로 정렬해서 반환함
- Activity.startManagingCursor() 호출 - 액티비티의 수명주기에 맞춰 커서의 수명 주기도 같이 관리해 줌. 프로그래머가 액티비티의 수명주기에 따라 데이터 소스에 연결된 커서를 비활성화하거나 쿼리를 다시 호출할 필요가 없게 함
- Cursor는 Java Iterator, JDBC ResultSet, ASP.NET의 DataSet 과 비슷함

쿼리 결과 보여주기(showEvents())

- Cursor를 받아서, 사용자가 볼 수 있도록 출력함
- 이벤트를 큰 문자열로 만들어서 모든 이벤트를 저장하고 줄바꿈으로 구분함 - 모든 이벤트들을 하나의 문자열로 만듦 - 바보같은 방법
- Cursor.moveToNext() 메서드는 데이터베이스의 다음 행으로 넘어가게 함
- Cursor에 대해 getLong(*column index*)과 getString(*column index*)을 통해 해당 열의 데이터를 뽑아냄
- Cursor.getColumnIndexOrThrow()로 열 인덱스 숫자를 찾아낼 수 있으나, 실행 속도가 느려짐

데이터 바인딩

- 몇 줄의 코드로 데이터(또는 모델)과 뷰를 연결
 - 모델은 상황에 따라 다르게 해석되나, (예를 들어 패션 모델, 모델 하우스, 성능 평가 모델, 기계 학습 모델 등) 데이터베이스에서는 데이터 모델, 즉 스키마와 비슷하게 해석됨
- 데이터 바인딩을 위해, Events 예제를 수정하여 데이터베이스 쿼리 결과에 엮인 ListView 를 사용하도록 함

Cursor-Adapter-Activity 간의 연결에 대한 개인적인 견해

- ASP.NET이나 C#.NET의 데이터베이스 응용을 보면, DataTable 방식과 DataSet 방식이 있는데, 지난 시간의 예제 코드는 DataTable 방식에 가까우며, DataSet 방식은 프로그래머가 연결 관리를 고민할 필요가 없도록 더 진일보된 방식이며, 이번 시간의 ContentProvider를 사용하는 예제 코드는 프로세스 간 통신까지 포괄적으로 고려한 더 진일보된 방식
- 데이터-컨트롤-뷰의 3-tier 구조와 비슷하게, 뷰에 해당하는 Activity에 대해 데이터를 의미하는 SQLiteOpenHelper 등을 지정해 주는 이러한 방식은, 이미, 예를 들어, ASP.NET 등에서 활발히 사용되고 있음
- 본 안드로이드 예제에서 소개된 방식은 오히려 진정한 3-tier 분할이 매끄럽게 구현되지 않았음. 예를 들어 ASP.NET 이나 Java 웹 서비스 진영의 방식이 더 편하고 진보된 방식임.
 - ASP.NET에선 ASP를 나타내는 .aspx 파일(뷰)과 C#을 나타내는 .cs 파일(컨트롤)이 연결되어 있으며, 데이터 소스를 .aspx에서 직접 지정할 수 있음
 - 이는 차기 안드로이드 설계에 반영할 필요가 있음. 이에 따라, Java 소스에서 뿐만 아니라, 뷰를 의미하는 XML 파일에서 직접 데이터 소스를 지정할 수 있도록 하는 게 좋음
 - 또한 컨트롤 또는 뷰에서 데이터를 매끄럽게 가져올 수 있게 하기 위해 .NET에선 통합언어쿼리 (LINQ), 그리고 Java 진영에선 iBATIS 를 제공하고 있음
 - 역시 차기 안드로이드 설계에 반영할만 하지만, 자원이 부족한 임베디드 시스템의 경우 부담이 될 수도 있음
 - SQLiteOpenHelper 같은 도우미 클래스들을 LINQ 와 같은 개념으로 완전히 재코딩하는 것도 생각해 볼 만함
- 또한, ListActivity 와 ListView 의 경우가 다른 개념으로 사용되는 데, 이는 차기 안드로이드 설계에서 일관성 있게 통합시키는 것도 관찰을 것으로 봄

Events.showEvents()를 수정

- Events 는 Activity를 확장한 ListActivity 로 선언
- 데이터베이스 테이블에는, Cursor 클래스를 통해 레코드들에 접근함
- Cursor를 받아서, 사용자가 볼 수 있도록 출력함
- ListActivity에는 임의의 데이터 소스를 Adapter 를 통해 연결해 줄 수 있음.
 - 이를 위해 setListAdapter() 메서드 사용
 - 여기서는 Adapter를 확장한 SimpleCursorAdapter 사용

SimpleCursorAdapter

- 어댑터는 뷰와 소스를 연결하는 중간 다리 역할
- 웹 서비스 이용하기의 Translate.setAdapters() 에서 이미 사용했었음 - 그 예제에서 데이터 소스는 XML 배열이므로 ArrayAdapter 를 사용했음
- 여기서는 데이터베이스 쿼리에서 나온 Cursor 객체이므로 SimpleCursorAdapter 사용
- 매개 변수
 - context - 현재 Activity의 참조
 - layout - 하나의 목록 아이템을 정의하는 리소스 (layout/item.xml)
 - cursor - 데이터 집합 커서
 - from - 데이터가 나오는 열 이름 목록
 - to - 데이터가 들어갈 뷰 목록

layout/item.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:orientation="horizontal"
    android:padding="10sp">
    <TextView android:id="@+id/rowid" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView android:id="@+id/rowidcolon" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=": "
        android:layout_toRightOf="@id/rowid" />
    <TextView android:id="@+id/time" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_toRightOf="@id/rowidcolon"
        />
    <TextView android:id="@+id/timecolon" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=": "
        android:layout_toRightOf="@id/time" />
    <TextView android:id="@+id/title" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:ellipsize="end"
        android:singleLine="true" android:textStyle="italic"
        android:layout_toRightOf="@id/timecolon" />
</RelativeLayout>
```

layout/main.xml

- list 와 empty 라는 built-in ID 사용
- 목록에 아이템이 있다면, @android:id/list 뷰가 보이고, 아니면 @android:id/empty 뷰가 보임
- 데이터 바인딩을 사용하는 본 예제에서 이벤트 데이터베이스에 데이터를 추가하거나, 데이터베이스를 직접 보는 방법은? – ContentProvider 를 사용하는 것

layout/main.xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/
  res/android"  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
<ListView      android:id="@android:id/list"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"/>
<TextView      android:id="@android:id/empty"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/empty" />
</LinearLayout>
```

ContentProvider 사용하기

- 안드로이드 보안 모델에선 원칙적으로 한 어플리케이션이 자신의 데이터 디렉토리에 작성한 파일은 다른 어플리케이션에 의해 읽히고 수정될 수 없음
- 각 프로그램은 개별 리눅스 사용자 ID와 데이터 디렉토리(/data/data/<package name>)와 보안된 메모리 공간을 가짐
- 안드로이드 프로그램들은 다음 두가지로 소통함
 - Inter-Process Communication – Android Interface Definition Language (AIDL) 와 IBinder 인터페이스
 - ContentProvider – 프로세스는 시스템에 자신이 어떤 종류 데이터의 제공자인지 등록함. 그 정보가 요청되면 안드로이드는 적합한 방식으로 콘텐츠를 쿼리 또는 수정하기 위해 고정 API를 통해 호출함

ContentProvider URI

- ContentProvider에 의해 관리되는 정보는 다음과 같은 URI를 통해 나타내짐
 - `content://<authority>/<path>/<id>`
 - `content`
 - `authority`
 - `path`
 - `id`
- 안드로이드는 다음과 같은 내장된 ContentProvider가 있음
 - `content://browser`
 - `content://contacts`
 - `content://media`
 - `content://settings`
- 본 예제의 Event 제공자로는 다음의 URI를 사용
 - `content://org.example.events/events/3` - `_id=3` 인 단일 이벤트
 - `content://org.example.events/events/` - 모든 이벤트

Constants.java

```
public interface Constants extends BaseColumns {  
    public static final String TABLE_NAME =  
        "events";  
  
    public static final String AUTHORITY =  
        "org.example.events";  
    public static final Uri CONTENT_URI =  
        Uri.parse("content://"  
            + AUTHORITY + "/" + TABLE_NAME);  
  
    // Columns in the Events database  
    public static final String TIME = "time";  
    public static final String TITLE = "title";  
}
```


Events.onCreate()

- 추적할 데이터베이스 객체가 없어짐
- try catch 블록 필요 없으며, EventsData 참조 없어짐

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    addEvent("Hello, Android!");  
    Cursor cursor = getEvents();  
    showEvents(cursor);  
}
```

행 추가하기 (addEvent())

- addEvent() 메서드를 통해 데이터베이스에 새로운 행을 추가함
- 시간은 현재 시간
- 제목은 주어진 string 변수
- CONTENT_URI 는 Constants.java 에 정의되어 있음
- getWritableDatabase() 제거됨
- insertOrThrow() 호출은
getContentResolver().insert()로 대체됨

쿼리 실행하기 (getEvents())

- Cursor를 채워주기 위한(populate)
Activity.managedQuery() 실행으로 충분함
- 데이터베이스에 대한 모든 참조를 제거함으로써,
Events 클라이언트를 Events 데이터 제공자(여기
서는 EventsData)로부터 분리함

Events.java

```
public class Events extends ListActivity {
    private static String[] FROM = { _ID, TIME, TITLE, };
    private static int[] TO = { R.id.rowid, R.id.time, R.id.title, };
    private static String ORDER_BY = TIME + " DESC";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        addEvent("Hello, Android!");
        Cursor cursor = getEvents();
        showEvents(cursor);
    }

    private void addEvent(String string) {
        ContentValues values = new ContentValues();
        values.put(TIME, System.currentTimeMillis());
        values.put(TITLE, string);
        getContentResolver().insert(CONTENT_URI, values);
    }

    private Cursor getEvents() {
        return managedQuery(CONTENT_URI, FROM, null, null, ORDER_BY);
    }

    private void showEvents(Cursor cursor) {
        // Set up data binding
        SimpleCursorAdapter adapter = new SimpleCursorAdapter(this, R.layout.item, cursor, FROM, TO);
        setListAdapter(adapter);
    }
}
```

ContentProvider 구현하기

- AndroidManifest.xml 에서 ContentProvider 추가
- `<provider android:name="EventsProvider" android:authorities="org.example.events" />`
 - `android:name` – 클래스 이름
 - `android:authorities` – 콘텐츠 URI에 사용되는 문자열
- ContentProvider를 확장하는 EventsProvider 클래스 생성 – 관습에 따라 MIME 타입에 org.example 대신 vnd.example 사용
- EventsProvider는 두 종류의 데이터를 다룸
 - `CONTENT_TYPE` – 이벤트들의 디렉토리의 MIME 타입
 - `CONTENT_ITEM_TYPE` – 단일 이벤트의 MIME 타입, ID 수준까지 상세히 지정할 수 있음

EventsProvider.java

```
public class EventsProvider extends ContentProvider {  
    private static final int EVENTS = 1;  
    private static final int EVENTS_ID = 2;  
  
    /** The MIME type of a directory of events */  
    private static final String CONTENT_TYPE  
        = "vnd.android.cursor.dir/vnd.example.event";  
  
    /** The MIME type of a single event */  
    private static final String CONTENT_ITEM_TYPE  
        = "vnd.android.cursor.item/vnd.example.event";  
  
    private EventsData events;  
    private UriMatcher uriMatcher;  
    // ...
```

프로그램의 배포 및 안드로이드 마켓

요약 (1)

- GPS 장치를 통해 위치를 인식하는 방법에 대해서 알아보았다.
- 가속도계에 대해서 알아보았다.
- 지도를 나타내는 맵 뷰에 대해 알아보고, 실제 프로그래밍에 적용해 보았다.
- 웹 뷰와 맵 뷰를 결합한 프로그램을 작성함으로써, 여러 API들을 연결하여 정보들을 결합하는 매시업(mashup)의 가능성에 대해 알아보았다.

요약 (2)

- SQLite가 뭔지 알아 보았고, 이를 사용할 줄 알았다.
- SQL의 기본적인 사용법들을 알았다.
- SQLite을 이용해 기본적인 데이터베이스 응용 프로그램을 작성할 수 있었다.
- 행을 추가하는 기본적인 데이터베이스 응용 프로그램을 작성할 수 있었다.
- 쿼리를 실행하는 기본적인 데이터베이스 응용 프로그램을 작성할 수 있었다.
- 쿼리 결과를 보여주는 기본적인 데이터베이스 응용 프로그램을 작성할 수 있었다.

요약 (3)

- 데이터 바인딩을 통해 데이터 소스에 해당하는 데이터 베이스와 뷰에 해당하는 액티비티를 연결한 데이터베이스 응용 프로그램을 작성해 보았다.
- 안드로이드 내의 다른 어플리케이션의 데이터에 접근하기 위해 제공되는 ContentProvider를 사용하는 방법에 대해 알아 보았다.
- 자신의 어플리케이션에서 다른 어플리케이션으로의 데이터 제공을 위한 ContentProvider를 구현해 보았다.

퀴즈 (1)

- 위치 정보를 위해서는 AndroidManifest.xml 에 어떤 것을 추가해야 하는가?
- 인터넷 사용을 위해서는 AndroidManifest.xml 에 어떤 것을 추가해야 하는가?
- 액티비티 내에서 LocationManager 클래스를 반환받기 위해서는 어떤 메서드를 사용하는가?
- LocationManager.getBestProvider() 의 선정 기준에는 어떠한 것들이 있는가?
- 위치 업데이트에는 배터리가 많이 소모된다. 이를 어떻게 줄일 수 있는가?
- 에뮬레이터가 돌아가는 경우, GPS 정보가 제공되지 않을 수 있다. 이를 해결하는 방법들에는 어떤 것들이 있는가?
- 액티비티 내에서 SensorManager 클래스를 반환받기 위해서는 어떤 메서드를 사용하는가?
- 역시 에뮬레이터의 경우, 센서가 없을 수 있다. 이럴 때, 사용자가 임의로 센서의 값들을 입력하기 위해서는 어떻게 해야 하는가?
- 구글 맵을 실행하려면 처음에 프로젝트를 어떻게 생성해야 하는가?
- 왜 구글 맵은 안드로이드 코어의 일부가 아닌가?
- 구글 맵 API 키에는 어떠한 것들이 있고, 이를 얻기 위해서는 어떻게 해야 하는가?
- 안드로이드 운영체제에서 Home의 배경 화면을 어떻게 바꾸는가? 만일 자신만의 사진으로 배경 화면을 하고 싶다면 어떻게 해야 하는가?

퀴즈 (2)

- 데이터베이스 테이블에 레코드를 삽입하는 메서드는 무엇인가?
- 데이터베이스에 대해 쿼리를 실행하는 메서드들을 열거해 보라. `query()`는 어떤 메서드이며, `execSQL()`은 어떤 메서드인가?
- `ORDER BY`는 어떤 옵션인가?
- 커서란 무엇인가?
- 데이터베이스에서 컬럼 인덱스 숫자를 찾아내는 방법은 무엇인가?

퀴즈 (3)

- 어댑터란 무엇인가? 그리고 커서란 무엇인가?
- SimpleCursorAdapter는 무엇을 위한 클래스인가? 무엇을 위한 어댑터인가? XML 입력을 위해서는 어떤 어댑터를 사용해야 하는가?
- 안드로이드에서 다른 어플리케이션이 사용하는 파일이 그 어플리케이션의 고유의 디렉토리에 있다면, 그 파일에 접근하려면 어떻게 해야 하는가?
- ContentProvider에서 사용하는 URI는 어떤 구조인가?
- 안드로이드가 기본적으로 제공하는 ContentProvider 용 URI는 무엇이 있는가?
- ContentProvider 를 구현하기 위해 해야 할 일들은 무엇이 있는가?
- MIME는 무슨 뜻인가?

연습문제 (1)

- 웹 뷰와 맵뷰를 같이 쓰는 예제를 스스로 처음부터 구현해 보자. 맵뷰에서 사용자가 지도의 특정 부분에 가거나, 특정 부분을 클릭했을 때, 이에 대한 이벤트나 정보를 받으려면 어떻게 해야 하는지, 그게 가능한지 아닌지를 MapView 레퍼런스 매뉴얼에서 검색해 보자.
- 웹 뷰와 맵 뷰를 같이 쓰는 예제에서, 웹 뷰 대신 배경에 컬러를 서로 다르게 하는 텍스트 뷰나 다른 위젯들을 사용해 보자.
- 다음 지도 API나 네이버 지도 API를 이용하는 안드로이드 응용 프로그래밍에 시도해 보자.
- 센서의 값들에는 잡음이 들어가기도 하고 시간적으로 불규칙으로 들어오는 경우가 많다. 이러한 센서의 값을 제대로 받고 해석하기 위해서는 어떠한 방법이 요구되는가? 평활화는 어떻게 수행하는 게 효과적일까?

연습문제 (2)

- 학생 이름, 국어, 영어, 수학에 대한 데이터베이스 프로그램을 작성하라. 총점, 평균, Grade 등을 계산해서 디스플레이 하라.
- 자동차를 운전하는 사람을 위한 차계부 프로그램을 구상해 보라. 차계부 데이터베이스에 저장해야할 데이터는 무엇이 있는지 열거해 보라.
- 도서 대여점의 도서 관리 프로그램을 작성하라. 도서를 표현하고 저장하기 위해서는 어떤 데이터가 있어야 하고, 고객이 도서를 대여하기 위해서는 어떤 데이터가 필요한가?
- 점심이나 저녁 외식을 위한 음식점을 추천하는 서비스를 생각해 보자. 이러한 추천 서비스를 위해, 음식점을 데이터베이스에 저장해야 한다. 음식점을 저장하기 위해 기본적으로 어떤 데이터가 필요하고, 사용자에게 효과적인 서비스를 제공하려면 어떤 데이터가 더 필요할까?

연습문제 (3)

- 본 장의 예제를 확장하여 이벤트 데이터 레코드에 새로운 컬럼을 추가하고, 사용자가 이벤트를 선택하면, 이벤트에 대한 자세한 정보를 보여주는 뷰어를 여는 프로그램을 작성하라.
- 사용자가 이벤트를 선택하면, 그 이벤트 정보를 특정 이메일로 보내는 프로그램을 작성하라.
- 각각의 이벤트에 대해 사용자가 별점을 줄 수 있는 프로그램을 작성하라.
- 사용자가 이벤트를 삭제할 수 있는 프로그램을 작성하라.
- 안드로이드에서 데이터를 저장하는 DBMS인 db4o 에 대해 알아보자.