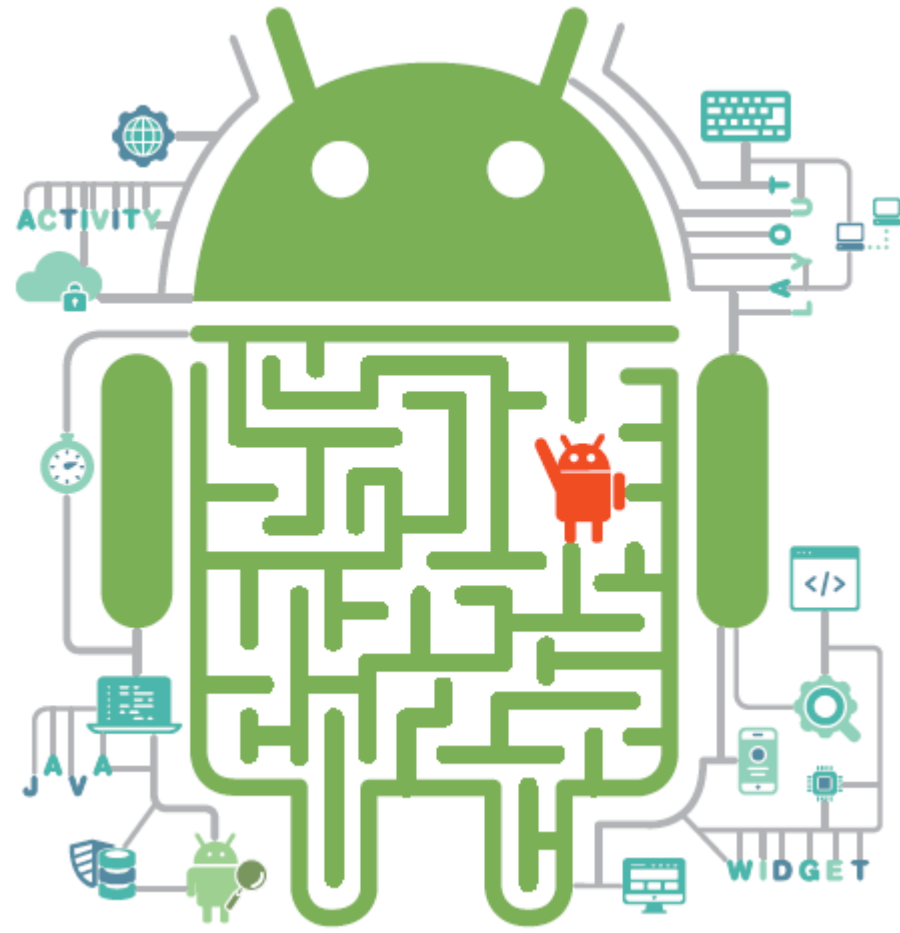


단계 별로 배우는 안드로이드 프로그래밍

[강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 최고 5년 이하의 징역 또는 5천만원 이하의 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.



단계별로 배우는

안드로이드 프로그래밍

Chapter 13. 네트워크와 지도

목차

01 TCP/IP 소켓 통신

02 위치 서비스

03 구글 지도

학습목표

- TCP 서버의 구조와 핵심 코드를 이해한다.
- TCP 클라이언트의 구조와 핵심 코드를 이해한다.
- 위치 서비스로 기기의 위치를 얻을 수 있다.
- 구글 지도를 사용하기 위한 절차를 파악한다.
- 구글 지도 응용 앱의 구조와 핵심 코드를 이해한다

■ 소켓(Socket)

- 다양한 운영 체제와 언어에서 지원되는 네트워크 프로그래밍 API
- 저수준이므로 통신의 세부 사항을 개발자가 일일이 설계하고 구현해야 함

■ 소켓 통신 기초

- 서버(Server): 먼저 실행하여 기다리는 프로그램
- 클라이언트(Client): 통신을 하고자 서버에 접속하는 프로그램
- 서버와 클라이언트는 서로 다른 IP 주소를 가지는 시스템에서 실행
- 해당 시스템 내에서 고유한 포트 번호를 가짐

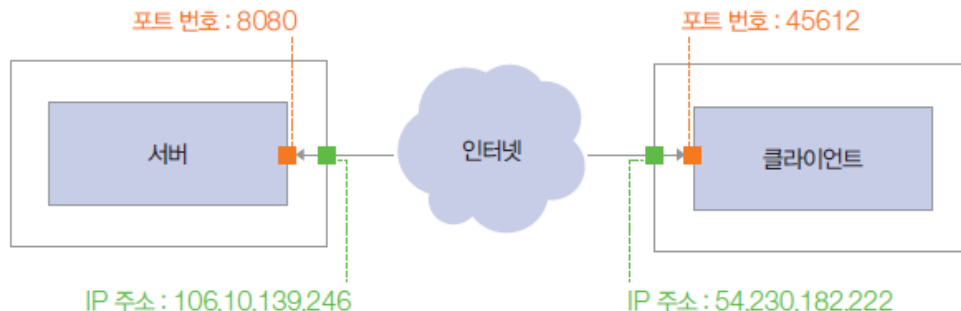


그림 13-1 서버/클라이언트 통신

■ TCP 서버의 기본 구조

- 자바 언어가 제공하는 소켓 클래스를 이용하여 작성

TCP 서버

```
1  ServerSocket servSock = null;
2  try {
3      // 서버 소켓을 초기화한다.
4      servSock = new ServerSocket(9000);
5      while (true) {
6          // 클라이언트의 접속을 기다린다.
7          Socket sock = servSock.accept();
8          InputStream inStream = sock.getInputStream();
9          OutputStream outStream = sock.getOutputStream();
10         // 클라이언트와 데이터를 주고받는다.
11         while (true) {
12             // 데이터 수신
13             inStream.read(...);
14             // 데이터 송신
15             outStream.write(...);
16         }
17         sock.close();
18     }
19 }
20 catch (IOException e) {
21     e.printStackTrace();
22 }
23 finally {
24     servSock.close();
25 }
```

■ 소켓을 이용한 서버 작성 절차

- ① ServerSocket 객체 생성
- ② ServerSocket 객체에 대해 accept()를 호출하면 대기하다가 클라이언트가 접속하면 리턴
- ③ accept()가 리턴한 Socket 객체에 대해 getInputStream()과 getOutputStream()을 각각 호출하여 InputStream과 OutputStream 객체 얻기
- ④ InputStream 객체에 대해 read()를 호출하면 클라이언트가 보낸 데이터를 읽을 수 있고, OutputStream 객체에 대해 write()를 호출하면 클라이언트에 데이터를 보낼 수 있음
- ⑤ 클라이언트와 데이터 통신이 끝나면 Socket 객체 닫기
- ⑥ 서버 작동을 끝내려면 ServerSocket 객체 닫기

■ TCP 클라이언트의 기본 구조

- 자바 언어가 제공하는 소켓 클래스를 이용하여 작성

TCP 클라이언트

```
1  Socket sock = null;
2  try {
3      // 서버에 접속한다.
4      sock = new Socket("127.0.0.1", 9000);
5      InputStream inStream = sock.getInputStream();
6      OutputStream outStream = sock.getOutputStream();
7      // 서버와 데이터를 주고받는다.
8      while (true) {
9          // 데이터 송신
10         outStream.write(...);
11         // 데이터 수신
12         inStream.read(...);
13     }
14 }
15 catch (IOException e) {
16     e.printStackTrace();
17 }
18 finally {
19     sock.close();
20 }
```


■ 소켓을 이용한 클라이언트 작성 절차

- ① Socket 객체를 생성하여 서버에 접속
- ② Socket 객체에 대해 `getInputStream()`과 `getOutputStream()`을 각각 호출하여 `InputStream`과 `OutputStream` 객체 얻기
- ③ `OutputStream` 객체에 대해 `write()`를 호출하면 서버에 데이터를 보낼 수 있고, `InputStream` 객체에 대해 `read()`를 호출하면 클라이언트가 보낸 데이터 읽기 가능
- ④ 서버와 데이터 통신이 끝나면 Socket 객체 닫기

01 TCP/IP 소켓 통신 ► TCP 서버

■ 소켓 API의 특성을 고려한 서버 구조

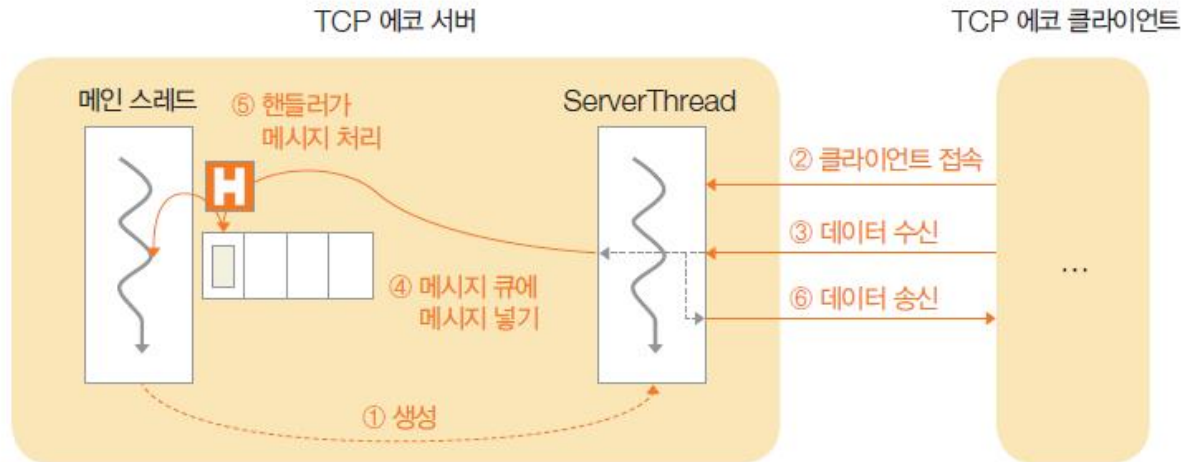


그림 13-2 TCP 에코 서버 구조

- ① 메인 스레드는 ServerThread를 생성
- ② ServerThread는 클라이언트의 접속을 받아들임
- ③ 접속한 클라이언트가 보낸 데이터 수신
- ④ ServerThread는 메인 스레드의 메시지 큐에 메시지(= 수신 데이터)를 넣음
- ⑤ 메인 스레드에서 수행되는 핸들러는 메시지(= 수신 데이터)를 꺼내서 그대로 화면에 출력
- ⑥ 서버 스레드는 수신 데이터를 그대로 클라이언트에 보냄

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

- ▼ LinearLayout (vertical)
 - ▼ LinearLayout (horizontal)
 - OK btnStart (Button) - "시작"
 - OK btnQuit (Button) - "종료"
 - ▼ ScrollView
 - Ab textOutput (TextView)

그림 13-3 컴포넌트 트리

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <LinearLayout
7          android:layout_width="match_parent"
8          android:layout_height="wrap_content"
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
9         android:orientation="horizontal">
10     <Button
11         android:id="@+id/btnStart"
12         android:layout_width="0dp"
13         android:layout_height="wrap_content"
14         android:layout_weight="1"
15         android:onClick="mOnClick"
16         android:text="시작"/>
17     <Button
18         android:id="@+id/btnQuit"
19         android:layout_width="0dp"
20         android:layout_height="wrap_content"
21         android:layout_weight="1"
22         android:onClick="mOnClick"
23         android:text="종료"/>
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
23         android:text="종료"/>
24     </LinearLayout>
25     <ScrollView
26         android:layout_width="match_parent"
27         android:layout_height="0dp"
28         android:layout_weight="1">
29         <TextView
30             android:id="@+id/textOutput"
31             android:layout_width="match_parent"
32             android:layout_height="wrap_content"/>
33     </ScrollView>
34 </LinearLayout>
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ MainActivity 클래스: 서버 스레드 관리, 데이터 출력 코드 작성

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {
2
3      private ServerThread mServerThread; // 서버 통신 담당
4      private TextView mTextOutput;
5
6      @Override
7      protected void onCreate(Bundle savedInstanceState) {
8          super.onCreate(savedInstanceState);
9          setContentView(R.layout.activity_main);
10         mTextOutput = (TextView) findViewById(R.id.textOutput);
11     }
12
13     @Override
14     protected void onDestroy() {
15         super.onDestroy();
16         android.os.Process.killProcess(android.os.Process.myPid());
17     }
18
19     public void mOnClick(View v) {
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ MainActivity 클래스: 서버 스레드 관리, 데이터 출력 코드 작성

```
20     switch (v.getId()) {
21     case R.id.btnStart:
22         if (mServerThread == null) {
23             mServerThread = new ServerThread(this, mMainHandler);
24             mServerThread.start();
25         }
26         break;
27     case R.id.btnQuit:
28         finish();
29         break;
30     }
31 }
32
33 private Handler mMainHandler = new Handler() {
34     @Override
35     public void handleMessage(Message msg) {
36         switch (msg.what) {
37             case 1: // 화면에 데이터 출력
38                 mTextOutput.append((String) msg.obj);
39                 break;
40         }
41     }
42 };
43 }
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ ServerThread 클래스: MainActivity 클래스 아래쪽에 코드 추가

MainActivity.java

```
45  class ServerThread extends Thread {
46
47      private Context mContext;
48      private Handler mMainHandler;
49
50      public ServerThread(Context context, Handler mainHandler) {
51          mContext = context;
52          mMainHandler = mainHandler;
53      }
54
55      @Override
56      public void run() {
57          ServerSocket servSock = null;
58          try {
59              // 서버 소켓을 초기화한다.
60              servSock = new ServerSocket(9000);
61              // 서버의 IP 주소와 포트 번호를 출력한다.
62              WifiManager wifiMgr = (WifiManager) mContext.getSystemService()
```


01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ ServerThread 클래스: MainActivity 클래스 아래쪽에 코드 추가

```
63         .getSystemService(Context.WIFI_SERVICE);
64         WifiInfo wifiInfo = wifiMgr.getConnectionInfo();
65         int serverip = wifiMgr.isWifiEnabled() ? wifiInfo.getIpAddress() : 0x0100007F;
66         doPrintln(">>> 서버 시작! " + ipv4ToString(serverip) + "/" + servSock.getLocalPort());
67
68         while (true) {
69             // 클라이언트 접속을 기다린다.
70             Socket sock = servSock.accept();
71             // 접속한 클라이언트 정보를 출력한다.
72             String ip = sock.getInetAddress().getHostAddress();
73             int port = sock.getPort();
74             doPrintln(">>> 클라이언트 접속: " + ip + "/" + port);
75             // 접속한 클라이언트와 통신한다.
76             InputStream in = sock.getInputStream();
77             OutputStream out = sock.getOutputStream();
78             // 클라이언트와 데이터를 주고받는다.
79             byte[] buf = new byte[1024];
80             while (true) {
81                 try {
82                     // 데이터 수신
83                     int nbytes = in.read(buf);
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ ServerThread 클래스: MainActivity 클래스 아래쪽에 코드 추가

```
84         if (nbytes > 0) {
85             // 데이터 출력
86             String s = new String(buf, 0, nbytes);
87             doPrintln("[ " + ip + "/" + port + " ] " + s);
88             // 데이터 송신
89             out.write(buf, 0, nbytes);
90         } else {
91             // 정상 종료
92             doPrintln(">> 클라이언트 종료: " + ip + "/" + port);
93             break;
94         }
95     } catch (IOException e) {
96         // 강제 종료
97         doPrintln(">> 클라이언트 종료: " + ip + "/" + port);
98         break;
99     }
100 } // end of inner while-loop
101 sock.close();
102 } // end of outer while-loop
103 } catch (IOException e) {
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

■ ServerThread 클래스: MainActivity 클래스 아래쪽에 코드 추가

```
104         doPrintln(e.getMessage());
105     } finally {
106         try {
107             if (servSock != null) {
108                 servSock.close();
109             }
110             doPrintln(">> 서버 종료!");
111         } catch (IOException e) {
112             doPrintln(e.getMessage());
113         }
114     }
115 } // end of run()
116
117 private void doPrintln(String str) {
118     Message msg = Message.obtain();
119     msg.what = 1;
120     msg.obj = str + "\n";
121     mMainHandler.sendMessage(msg);
122 }
123
124 private String ipv4ToString(int ip) {
```

01 TCP/IP 소켓 통신 ► TCP 서버



실습 13-1

EchoServer

- ServerThread 클래스: MainActivity 클래스 아래쪽에 코드 추가

```
125         int a = (ip) & 0xFF, b = (ip >> 8) & 0xFF;  
126         int c = (ip >> 16) & 0xFF, d = (ip >> 24) & 0xFF;  
127         return Integer.toString(a) + "." + Integer.toString(b) + "."  
128                + Integer.toString(c) + "." + Integer.toString(d);  
129     }  
130 }
```

01 TCP/IP 소켓 통신 ▶ TCP 서버



실습 13-1

EchoServer

■ 매니페스트에 퍼미션 추가

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="andbook.example.echoserver">
4      <uses-permission android:name="android.permission.INTERNET"/>
5      <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
6      <application
...      ...
```

01 TCP/IP 소켓 통신 ▶ TCP 서버



실습 13-1

EchoServer

■ 실행 화면



(a) 초기 화면



(b) 서버 시작

그림 13-4 실행 화면

01 TCP/IP 소켓 통신 ► TCP 클라이언트

■ 소켓 API의 특성을 고려한 클라이언트의 구조

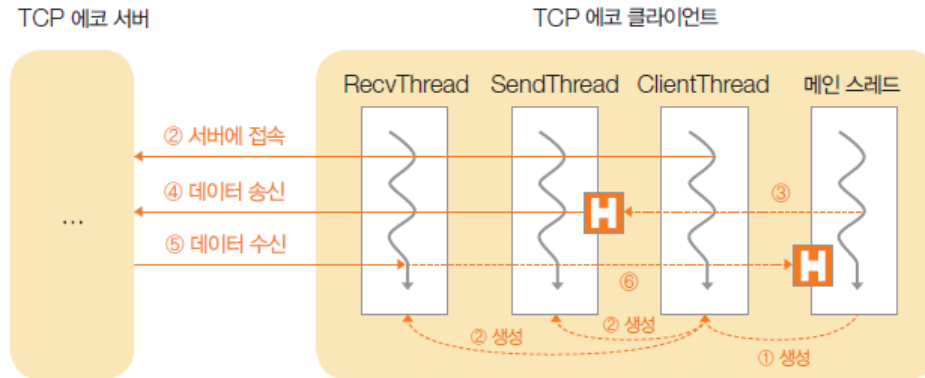


그림 13-5 TCP 에코 클라이언트 구조

- ① 메인 스레드 ClientThread 생성
- ② 서버에 접속 성공하면 ClientThread는 SendThread와 RecvThread 생성
- ③ 사용자가 데이터를 입력하면 SendThread의 메시지 큐에 메시지 넣음
- ④ SendThread에서 수행되는 핸들러는 메시지를 꺼내서 그대로 서버에 송신
- ⑤ 서버가 응답으로 보내준 데이터를 RecvThread가 수신
- ⑥ 메인 스레드의 메시지 큐에 메시지 넣음.
메인 스레드에서 수행되는 핸들러는 메시지를 꺼내서 화면에 출력



실습 13-2

EchoClient

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

- ▼ LinearLayout (vertical)
 - ▼ LinearLayout (horizontal)
 - abc editIP (EditText) - "127.0.0.1"
 - OK btnConnect (Button) - "연결"
 - OK btnQuit (Button) - "종료"
 - ▼ LinearLayout (horizontal)
 - abc editData (EditText)
 - OK btnSend (Button) - "보내기"
 - ▼ ScrollView
 - Ab textOutput (TextView)

그림 13-6 컴포넌트 트리

activity_main.xml

```
1  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      android:layout_width="match_parent"
3      android:layout_height="match_parent"
4      android:orientation="vertical">
5      <LinearLayout
6          android:layout_width="match_parent"
7          android:layout_height="wrap_content"
```




실습 13-2

EchoClient

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
8         android:orientation="horizontal">
9         <EditText
10             android:id="@+id/editIP"
11             android:layout_width="0dp"
12             android:layout_height="wrap_content"
13             android:layout_weight="1"
14             android:digits="0123456789."
15             android:hint="IPv4 주소 입력"
16             android:inputType="number"
17             android:text="127.0.0.1"/>
18         <Button
19             android:id="@+id/btnConnect"
20             android:layout_width="100dp"
21             android:layout_height="wrap_content"
22             android:onClick="mOnClick"
23             android:text="연결"/>
24         <Button
25             android:id="@+id/btnQuit"
26             android:layout_width="100dp"
27             android:layout_height="wrap_content"
28             android:onClick="mOnClick"
29             android:text="종료"/>
```



실습 13-2

EchoClient

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
30     </LinearLayout>
31     <LinearLayout
32         android:layout_width="match_parent"
33         android:layout_height="wrap_content"
34         android:orientation="horizontal">
35         <EditText
36             android:id="@+id/editData"
37             android:layout_width="0dp"
38             android:layout_height="wrap_content"
39             android:layout_weight="1"
40             android:hint="데이터 입력"
41             android:singleLine="true"/>
42         <Button
43             android:id="@+id/btnSend"
44             android:layout_width="100dp"
45             android:layout_height="wrap_content"
46             android:enabled="false"
```

01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
47         android:onClick="mOnClick"
48         android:text="보내기"/>
49     </LinearLayout>
50     <ScrollView
51         android:layout_width="match_parent"
52         android:layout_height="0dp"
53         android:layout_weight="1">
54         <TextView
55             android:id="@+id/textOutput"
56             android:layout_width="match_parent"
57             android:layout_height="wrap_content"/>
58     </ScrollView>
59 </LinearLayout>
```



실습 13-2

EchoClient

■ MainActivity.java 파일에 클라이언트 통신 코드 작성

- MainActivity 클래스

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {  
2  
3      private ClientThread mClientThread; // 클라이언트 통신 담당  
4      private EditText mEditIP, mEditData;  
5      private Button mBtnConnect, mBtnSend;  
6      private TextView mTextOutput;  
7  
8      @Override  
9      protected void onCreate(Bundle savedInstanceState) {  
10         super.onCreate(savedInstanceState);  
11         setContentView(R.layout.activity_main);  
12  
13         mEditIP = (EditText) findViewById(R.id.editIP);  
14         mEditData = (EditText) findViewById(R.id.editData);  
15         mBtnConnect = (Button) findViewById(R.id.btnConnect);  
16         mBtnSend = (Button) findViewById(R.id.btnSend);  
17         mTextOutput = (TextView) findViewById(R.id.textOutput);  
18     }
```

01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

▪ MainActivity 클래스

```
19
20  @Override
21  protected void onDestroy() {
22      super.onDestroy();
23      android.os.Process.killProcess(android.os.Process.myPid());
24  }
25
26  public void mOnClick(View v) {
27      switch (v.getId()) {
28          case R.id.btnConnect:
29              if (mClientThread == null) {
30                  String str = mEditIP.getText().toString();
31                  if (str.length() != 0) {
32                      mClientThread = new ClientThread(str, mMainHandler);
33                      mClientThread.start();
34                      mBtnConnect.setEnabled(false);
35                      mBtnSend.setEnabled(true);
36                  }
37              }
38              break;
```



실습 13-2

EchoClient

■ MainActivity 클래스

```
39         case R.id.btnQuit:
40             finish();
41             break;
42         case R.id.btnSend:
43             if (SendThread.mHandler != null) {
44                 Message msg = Message.obtain();
45                 msg.what = 1;
46                 msg.obj = mEditData.getText().toString();
47                 SendThread.mHandler.sendMessage(msg);
48                 mEditData.selectAll();
49             }
50             break;
51     }
52 }
53
54 private Handler mMainHandler = new Handler() {
55     @Override
56     public void handleMessage(Message msg) {
57         switch (msg.what) {
58             case 1: // 화면에 데이터 출력
59                 mTextOutput.append((String) msg.obj);
60                 break;
61         }
62     }
63 };
64 }
```

01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

- ClientThread 클래스

MainActivity.java

```
66 class ClientThread extends Thread {
67
68     private String mServAddr;
69     private Handler mMainHandler;
70
71     public ClientThread(String servAddr, Handler mainHandler) {
72         mServAddr = servAddr;
73         mMainHandler = mainHandler;
74     }
75
76     @Override
77     public void run() {
78         Socket sock = null;
79         try {
80             sock = new Socket(mServAddr, 9000);
81             doPrintln(">> 서버와 연결 성공!");
82             SendThread sendThread = new SendThread(this, sock.getOutputStream());
83             RecvThread recvThread = new RecvThread(this, sock.getInputStream());
84             sendThread.start();
85             recvThread.start();
86             sendThread.join();
87             recvThread.join();

```

01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

▪ ClientThread 클래스

```
88         } catch (Exception e) {
89             doPrintln(e.getMessage());
90         } finally {
91             try {
92                 if (sock != null) {
93                     sock.close();
94                     doPrintln(">> 서버와 연결 종료!");
95                 }
96             } catch (IOException e) {
97                 doPrintln(e.getMessage());
98             }
99         }
100     }
101
102     public void doPrintln(String str) {
103         Message msg = Message.obtain();
104         msg.what = 1;
105         msg.obj = str + "\n";
106         mMainHandler.sendMessage(msg);
107     }
108 }
```


01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

- SendThread 클래스

MainActivity.java

```
110 class SendThread extends Thread {
111
112     private ClientThread mClientThread;
113     private OutputStream mOutputStream;
114     public static Handler mHandler;
115
116     public SendThread(ClientThread clientThread, OutputStream outputStream) {
117         mClientThread = clientThread;
118         mOutputStream = outputStream;
119     }
120
121     @Override
122     public void run() {
123         Looper.prepare();
124         mHandler = new Handler() {
125             @Override
126             public void handleMessage(Message msg) {
127                 switch (msg.what) {
128                     case 1: // 데이터 송신 메시지
```

01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

■ SendThread 클래스

```
129         try {
130             String s = (String) msg.obj;
131             mOutputStream.write(s.getBytes());
132             mClientThread.doPrintln("[보낸 데이터] " + s);
133         } catch (IOException e) {
134             mClientThread.doPrintln(e.getMessage());
135         }
136         break;
137     case 2: // 스레드 종료 메시지
138         getLooper().quit();
139         break;
140     }
141 }
142 };
143 Looper.loop();
144 }
145 }
```

```

147 class RecvThread extends Thread {
148
149     private ClientThread mClientThread;
150     private InputStream mInStream;
151
152     public RecvThread(ClientThread clientThread, InputStream inStream) {
153         mClientThread = clientThread;
154         mInStream = inStream;
155     }
156
157     @Override
158     public void run() {
159         byte[] buf = new byte[1024];
160         while (true) {
161             try {
162                 int nbytes = mInStream.read(buf);
163                 if (nbytes > 0) {
164                     String s = new String(buf, 0, nbytes);

```

01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

▪ RecvThread 클래스

```
165         mClientThread.doPrintln("[받은 데이터] " + s);
166     } else {
167         mClientThread.doPrintln(">> 서버가 연결 끊음!");
168         if (SendThread.mHandler != null) {
169             Message msg = Message.obtain();
170             msg.what = 2;
171             SendThread.mHandler.sendMessage(msg);
172         }
173         break;
174     }
175 } catch (IOException e) {
176     mClientThread.doPrintln(e.getMessage());
177 }
178 }
179 }
180 }
```

01 TCP/IP 소켓 통신 ► TCP 클라이언트



실습 13-2

EchoClient

■ 매니페스트에 퍼미션 추가

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="andbook.example.echoclient">
4      <uses-permission android:name="android.permission.INTERNET"/>
5      <application
...      ...
```

01 TCP/IP 소켓 통신 ▶ TCP 클라이언트

실습 13-2

EchoClient

■ 실행 화면



(a) 초기 화면



(b) [연결] 클릭



(c) 데이터 입력 후 [보내기] 클릭

그림 13-7 실행 화면

■ 하드웨어(GPS, WiFi, 이동통신 등)

- 기기의 현재 위치를 파악할 수 있도록 대부분의 안드로이드 시스템에 내장

■ 위치 제공자(Location Provider)

- 하드웨어를 기반으로 위치 정보를 알려주는 소프트웨어

■ 위치 서비스(Location Service)

- 위치 제공자로부터 기기의 위치를 얻는 등의 작업을 함

■ 위치 관리자(Location Manager)

- 위치 서비스에 접근하기 위한 응용 프로그램



그림 13-8 안드로이드의 앱과 위치 서비스 구성 요소

■ 안드로이드 기기의 현재 위치를 얻는 절차

① 위치 서비스 사용에 필요한 위치 관리자 얻기

```
LocationManager mLocMgr = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
```

② LocationListener 객체 준비

```
LocationListener mLocListener = new LocationListener() {  
    @Override  
    public void onLocationChanged(Location location) {  
        // 기기의 위치가 변할 때마다 호출된다.  
    }  
    @Override  
    public void onStatusChanged(String provider, int status, Bundle extras) {  
        // 위치 제공자의 상태가 변할 때마다(활성화, 비활성화, ...) 호출된다.  
    }  
    @Override  
    public void onProviderEnabled(String provider) {  
        // 위치 제공자가 활성화되면 호출된다.  
    }  
    @Override  
    public void onProviderDisabled(String provider) {  
        // 위치 제공자가 비활성화되면 호출된다.  
    }  
};
```


■ 안드로이드 기기의 현재 위치를 얻는 절차

- ③ 기기의 위치 정보를 얻고 싶을 때 LocationListener 객체를 등록

```
String provider = mLocMgr.getBestProvider(new Criteria(), true);  
mLocMgr.requestLocationUpdates(provider, 5000, 10, mLocListener);
```

- ④ 기기의 위치 정보가 필요 없을 때 LocationListener 등록 해제

```
mLocMgr.removeUpdates(mLocListener);
```



실습 13-3

GetLocation

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

▼ LinearLayout (vertical)
Ab textStatus (TextView) - "[현재 상태]"
Ab textLocation (TextView) - "[현재 위치]"

그림 13-9 컴포넌트 트리

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      android:padding="8dp">
8      <TextView
9          android:id="@+id/textStatus"
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:text="[현재 상태]"
13         android:textSize="16dp"/>
14     <TextView
15         android:id="@+id/textLocation"
16         android:layout_width="match_parent"
17         android:layout_height="wrap_content"
18         android:text="[현재 위치]"
19         android:textSize="16dp"/>
20 </LinearLayout>
```



실습 13-3

GetLocation

- 매니페스트에 퍼미션 추가
- 'build.gradle (Module: app)' 열기 -> targetSdkVersion을 22로 수정 -> 우측 상단의 'Sync Now' 클릭

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="andbook.example.getlocation">
4      <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
5      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
6      <application
7          android:allowBackup="true"
8      ...
9      ...
```



실습 13-3

GetLocation

■ MainActivity 클래스에 코드 작성

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity {  
2  
3      private TextView mTextStatus;  
4      private TextView mTextLocation;  
5  
6      private LocationManager mLocMgr;  
7      private int mLocCnt = 0;  
8  
9      @Override  
10     protected void onCreate(Bundle savedInstanceState) {  
11         super.onCreate(savedInstanceState);  
12         setContentView(R.layout.activity_main);  
13  
14         mTextStatus = (TextView) findViewById(R.id.textStatus);  
15         mTextLocation = (TextView) findViewById(R.id.textLocation);  
16  
17         mLocMgr = (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
18         showLocationStatus();  
19     }  
20 }
```



실습 13-3

GetLocation

■ MainActivity 클래스에 코드 작성

```
21  @Override
22  protected void onStart() {
23      super.onStart();
24      if (!getLocationStatus()) {
25          Toast.makeText(this, "위치 서비스를 켜십시오.", Toast.LENGTH_SHORT).show();
26          finish();
27      }
28      String provider = mLocMgr.getBestProvider(new Criteria(), true);
29      mLocMgr.requestLocationUpdates(provider, 2000, 0, mLocListener);
30  }
31
32  @Override
33  protected void onStop() {
34      super.onStop();
35      mLocMgr.removeUpdates(mLocListener);
36  }
37
38  private void showLocationStatus() {
39      boolean net = mLocMgr.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
40      boolean gps = mLocMgr.isProviderEnabled(LocationManager.GPS_PROVIDER);
41      String str = String.format(Locale.KOREAN,
42          "[현재 상태] Network=%b, GPS=%b", net, gps);
```



실습 13-3

GetLocation

■ MainActivity 클래스에 코드 작성

```
43     m textStatus.setText(str);
44 }
45
46 private boolean getLocationStatus() {
47     boolean net = mLocMgr.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
48     boolean gps = mLocMgr.isProviderEnabled(LocationManager.GPS_PROVIDER);
49     return (net || gps);
50 }
51
52 LocationListener mLocListener = new LocationListener() {
53
54     @Override
55     public void onLocationChanged(Location location) {
56         String str = String.format(Locale.KOREAN,
57             "[현재 위치] 경도 %.4f 위도 %.4f (%d)",
58             location.getLongitude(), location.getLatitude(), ++mLocCnt);
59         mTextLocation.setText(str);
60     }
61 }
```



실습 13-3

GetLocation

■ MainActivity 클래스에 코드 작성

```
62         @Override
63         public void onStatusChanged(String provider, int status, Bundle extras) {
64         }
65
66         @Override
67         public void onProviderEnabled(String provider) {
68         }
69
70         @Override
71         public void onProviderDisabled(String provider) {
72         }
73     };
74 }
```

02 위치 서비스



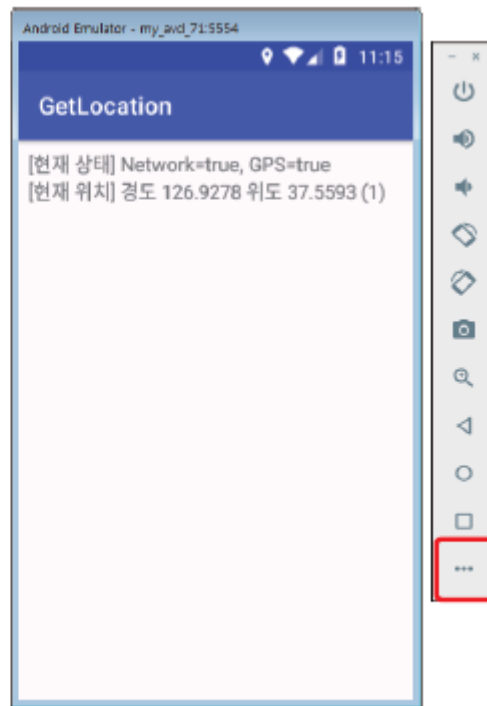
실습 13-3

GetLocation

■ 실행 화면



(a) 초기 화면



(b) 에뮬레이터 제어 대화상자 열기

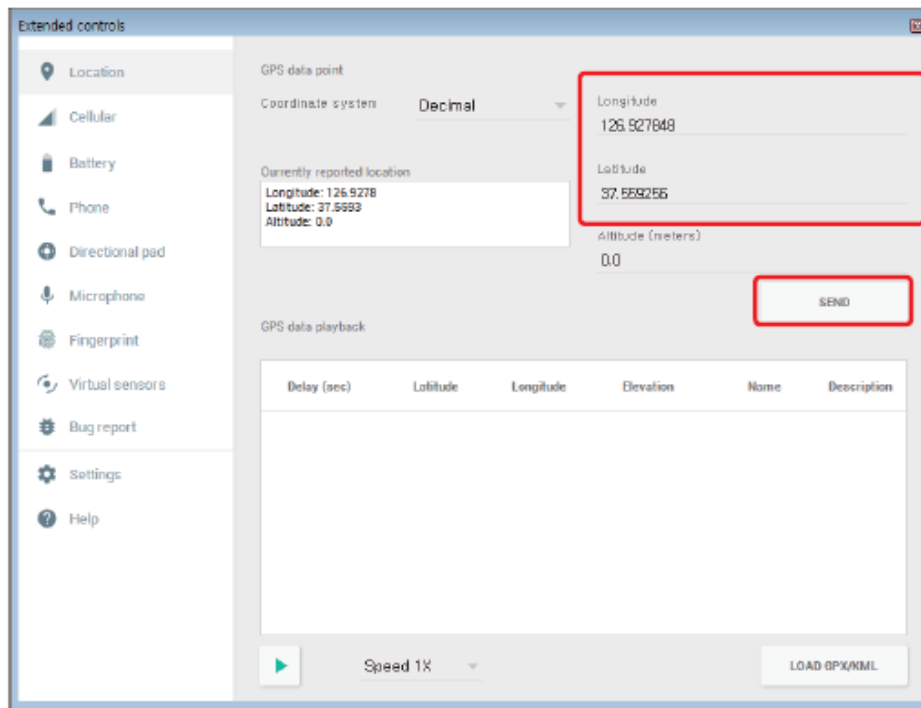
02 위치 서비스



실습 13-3

GetLocation

■ 실행 화면



(c) 위치 정보 입력 후 [SEND] 클릭

그림 13-10 실행 화면

03 구글 지도 ▶ 구글 지도 기초

■ 구글 지도를 사용하는 앱의 개발과 테스트를 위한 조건

- 최신 구글 플레이 서비스(Google Play Service)가 테스트 기기에 설치됨
- 기기에서 인터넷 사용 가능
 - 구글 서버에서 지도 타일(map tile-직사각형 영역의 지도 데이터)을 받아와야 하기 때문
- 안드로이드 SDK 구성 요소 중 Google Repository가 설치됨

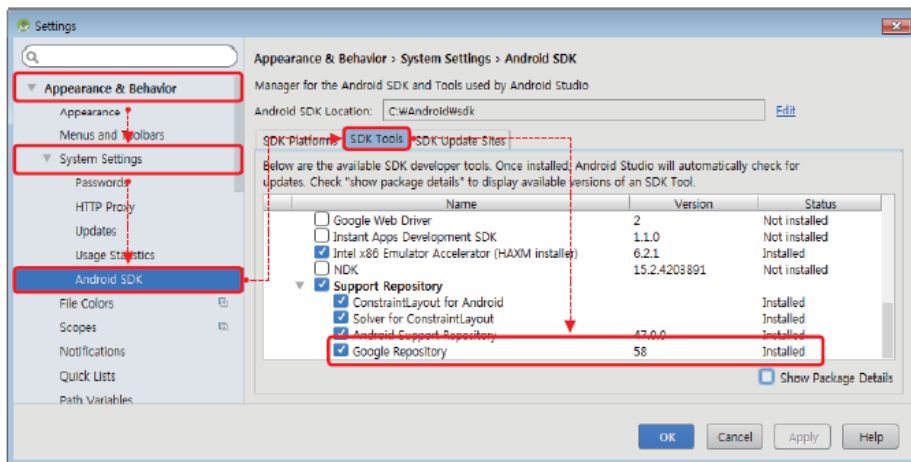


그림 13-11 Google Repository 설치와 최신 버전 여부 확인

- 안드로이드 에뮬레이터를 사용하는 경우, PC의 그래픽 카드가 OpenGL 2.0 이상 지원

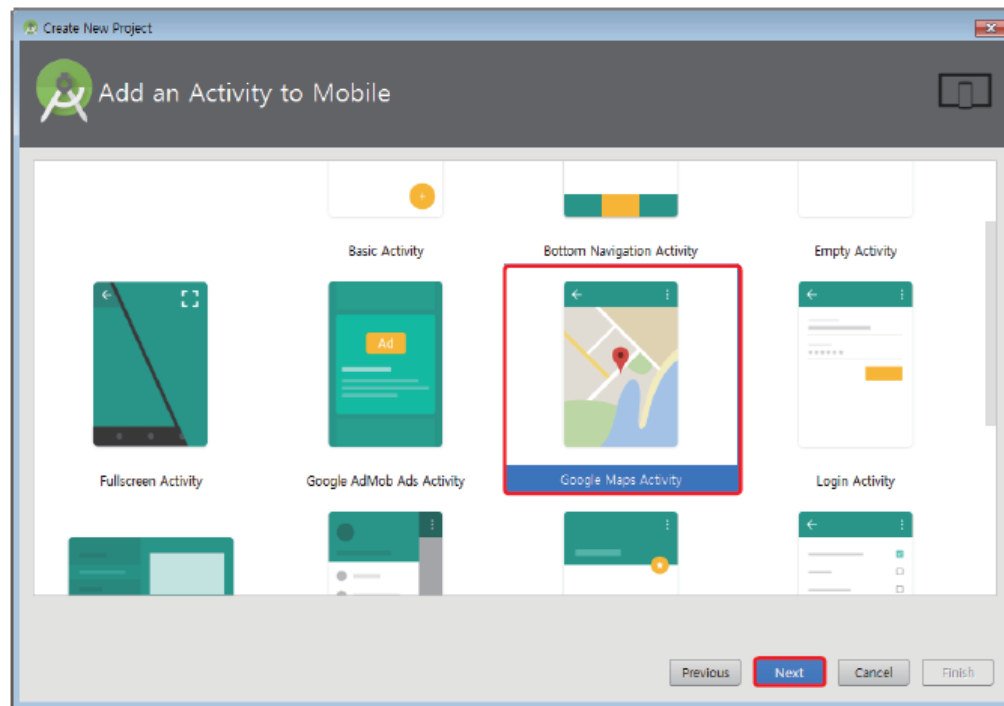
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 프로젝트 생성 및 수정



(a) 'Empty Activity' 대신 'Google Maps Activity' 선택

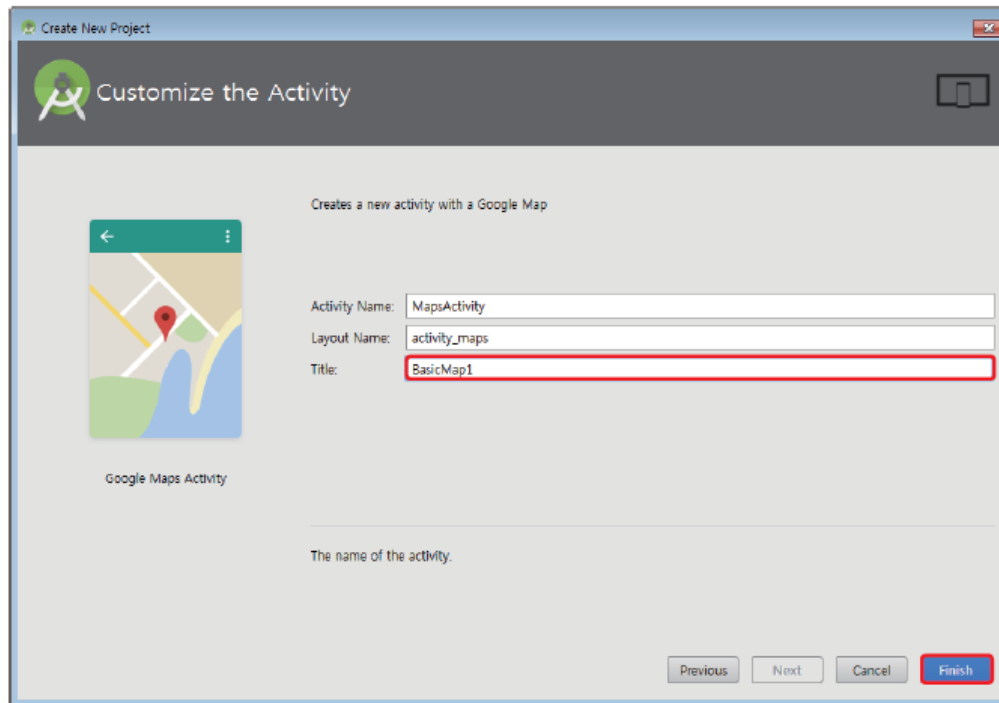
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 프로젝트 생성 및 수정



(b) 'Title'을 "BasicMap1"로 수정

그림 13-12 프로젝트 생성 단계

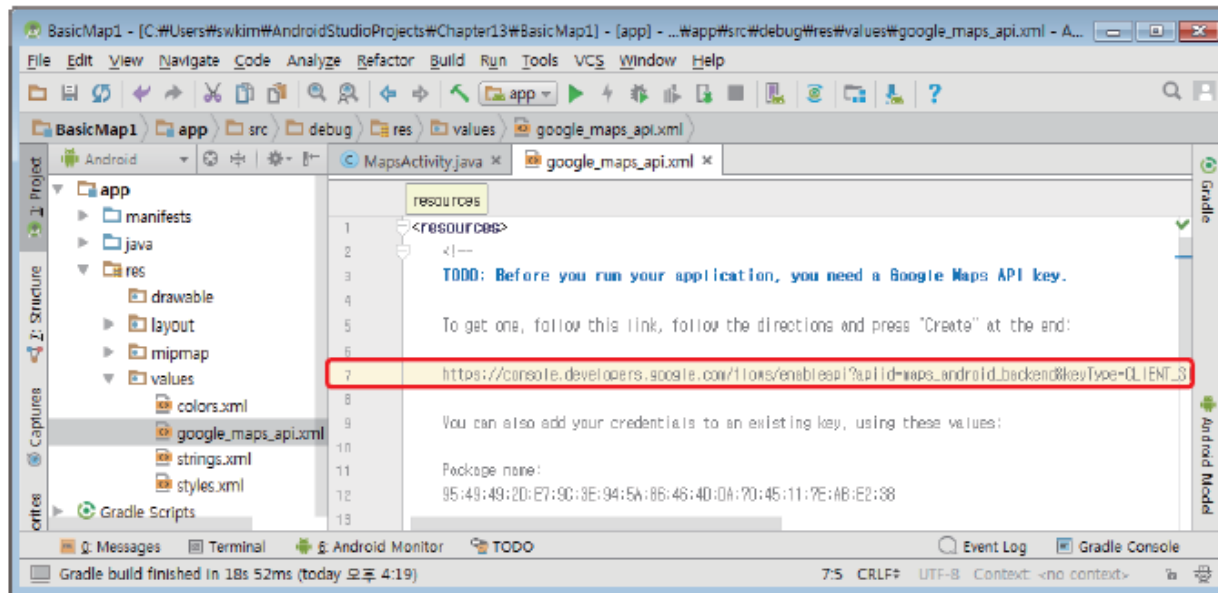
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 구글 웹 사이트에서 구글 지도 API 키 발급, 프로젝트 수정



(a) 웹사이트 주소 복사

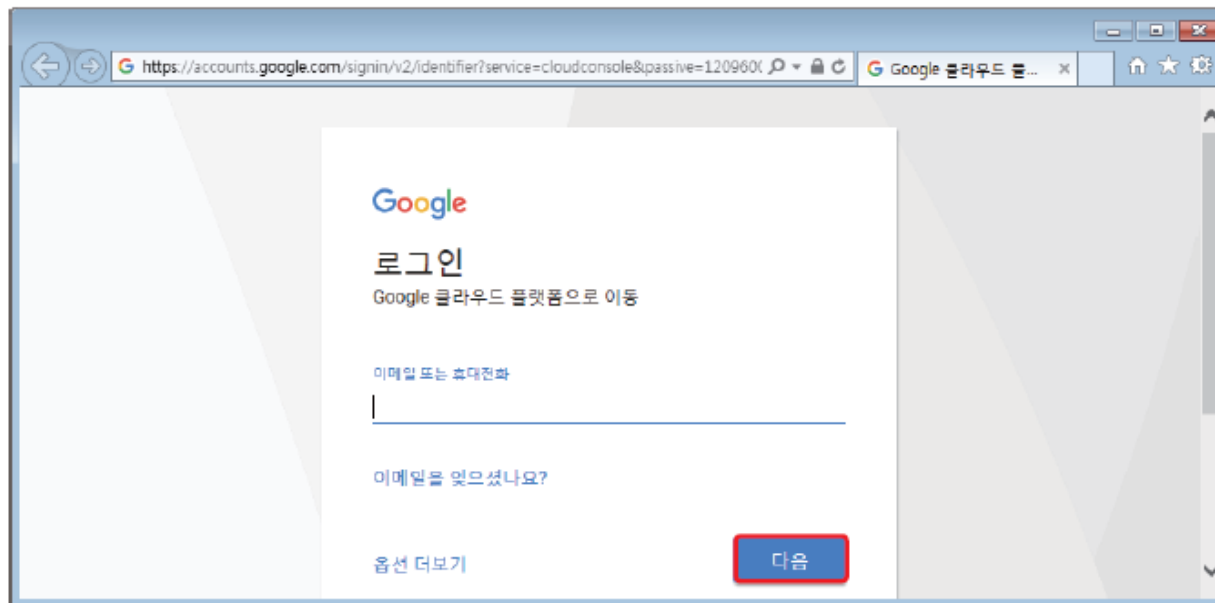
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 구글 웹 사이트에서 구글 지도 API 키 발급, 프로젝트 수정



(b) 구글 계정 로그인

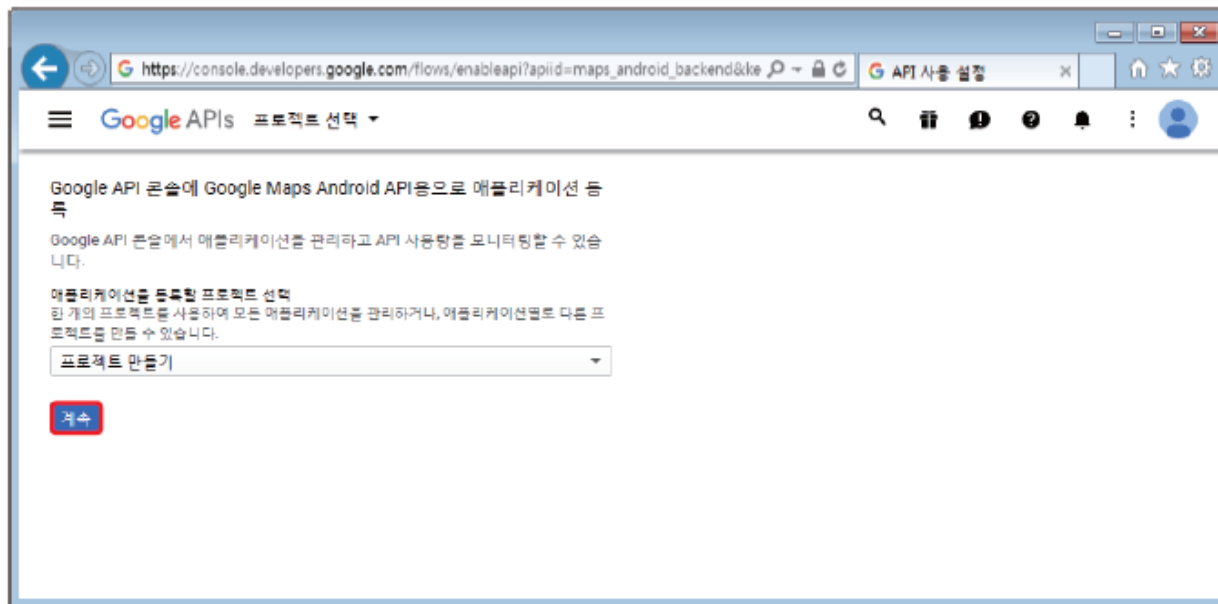
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 구글 웹 사이트에서 구글 지도 API 키 발급, 프로젝트 수정



(c)[계속] 클릭

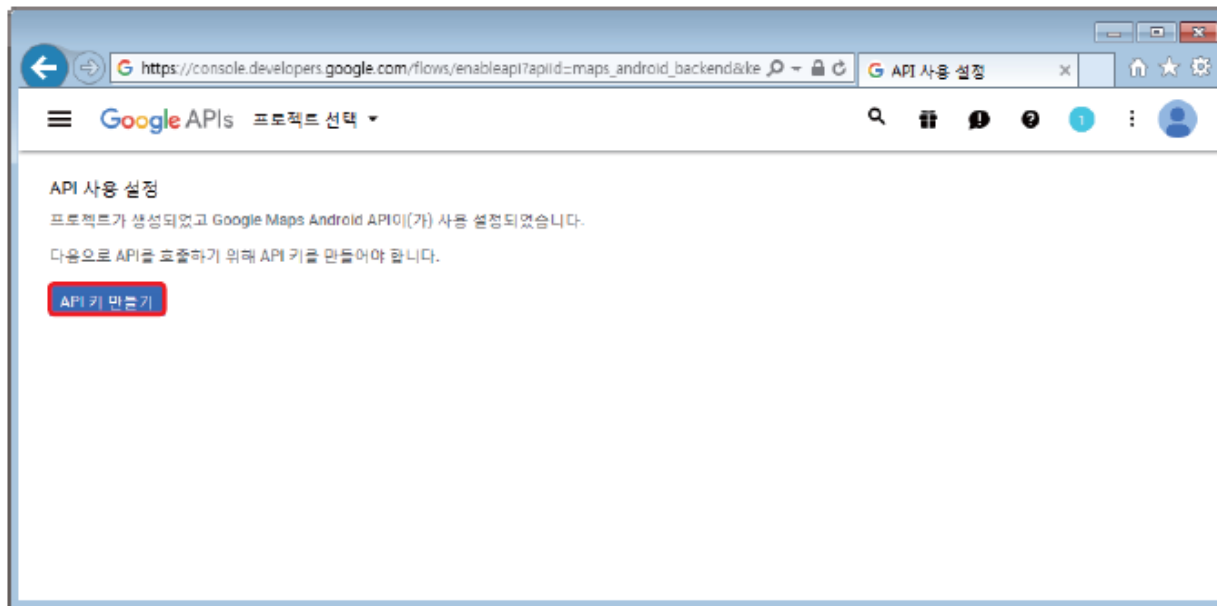
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 구글 웹 사이트에서 구글 지도 API 키 발급, 프로젝트 수정



(d) [API 키 만들기] 클릭

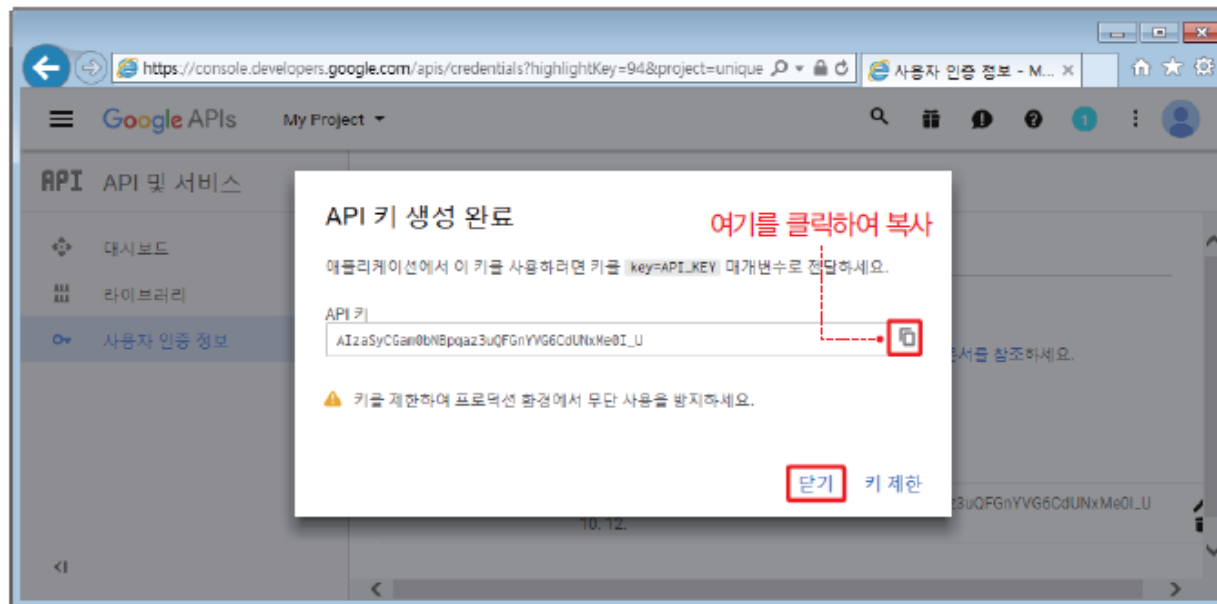
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 구글 웹 사이트에서 구글 지도 API 키 발급, 프로젝트 수정



(e) API 키 복사하고 [닫기] 클릭

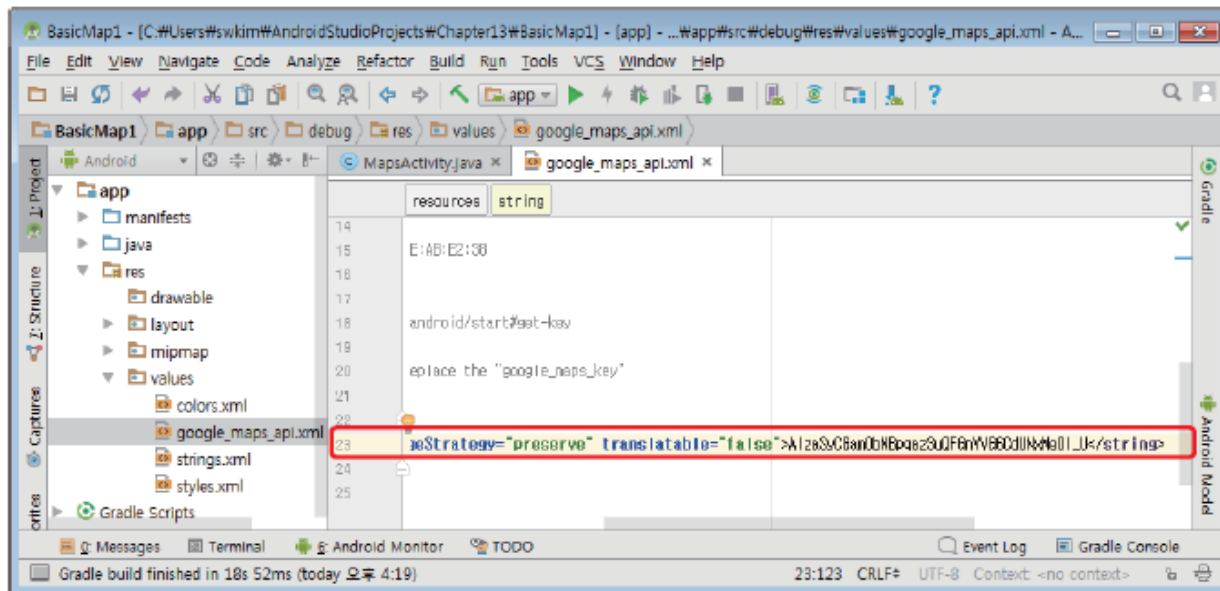
03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 구글 웹 사이트에서 구글 지도 API 키 발급, 프로젝트 수정



(f) 문자열 리소스 값 변경

그림 13-13 구글 지도 API 키 발급과 프로젝트 수정

03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ 실행 화면



(a) 초기 화면



(b) 마커 클릭 후

그림 13-14 실행 화면

03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ BasicMap1 예제: 매니페스트

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="andbook.example.basicmap1">
4      ...
5
6
7
8
9      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
10     <application
11         ...
12
13
14
15
16
17
18
19
20
21
22
23
24
25         <meta-data
26             android:name="com.google.android.geo.API_KEY"
27             android:value="@string/google_maps_key"/>
28         <activity
29             ...
30
31
32
33
34         </activity>
35     </application>
36 </manifest>
```

03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ BasicMap1 예제: 레이아웃

activity_main.xml

```
1 <fragment xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:map="http://schemas.android.com/apk/res-auto"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/map"
5     android:name="com.google.android.gms.maps.SupportMapFragment"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="andbook.example.basicmap1.MapsActivity"/>
```

03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ BasicMap1 예제: 자바 코드

MainActivity.java

```
1  public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
2
3      private GoogleMap mMap;
4
5      @Override
6      protected void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_maps);
9
10         SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
11             .findFragmentById(R.id.map);
12         mapFragment.getMapAsync(this);
```

03 구글 지도 ▶ 구글 지도 기초



실습 13-4

BasicMap1

■ BasicMap1 예제: 자바 코드

```
13     }  
14  
15     @Override  
16     public void onMapReady(GoogleMap googleMap) {  
17         mMap = googleMap;  
18         LatLng sydney = new LatLng(-34, 151);  
19         mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));  
20         mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));  
21     }  
22 }
```

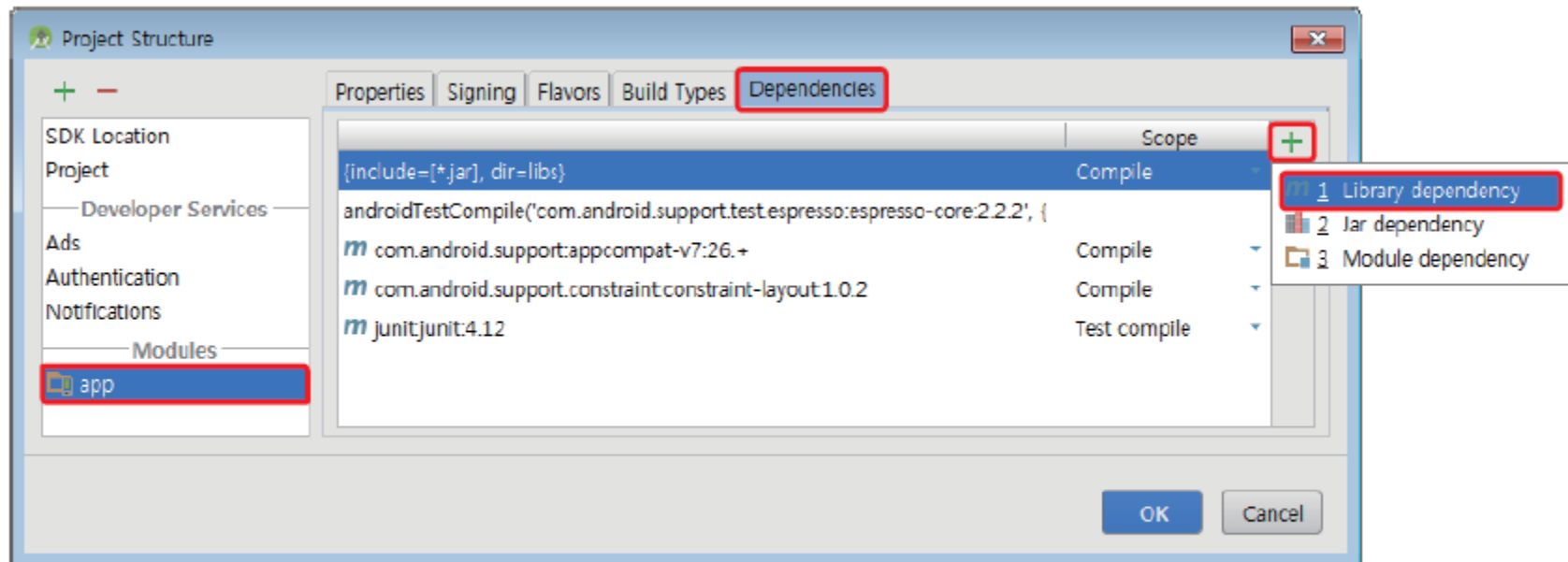
03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스에 필요한 라이브러리 추가



(a) 'Library dependency' 선택

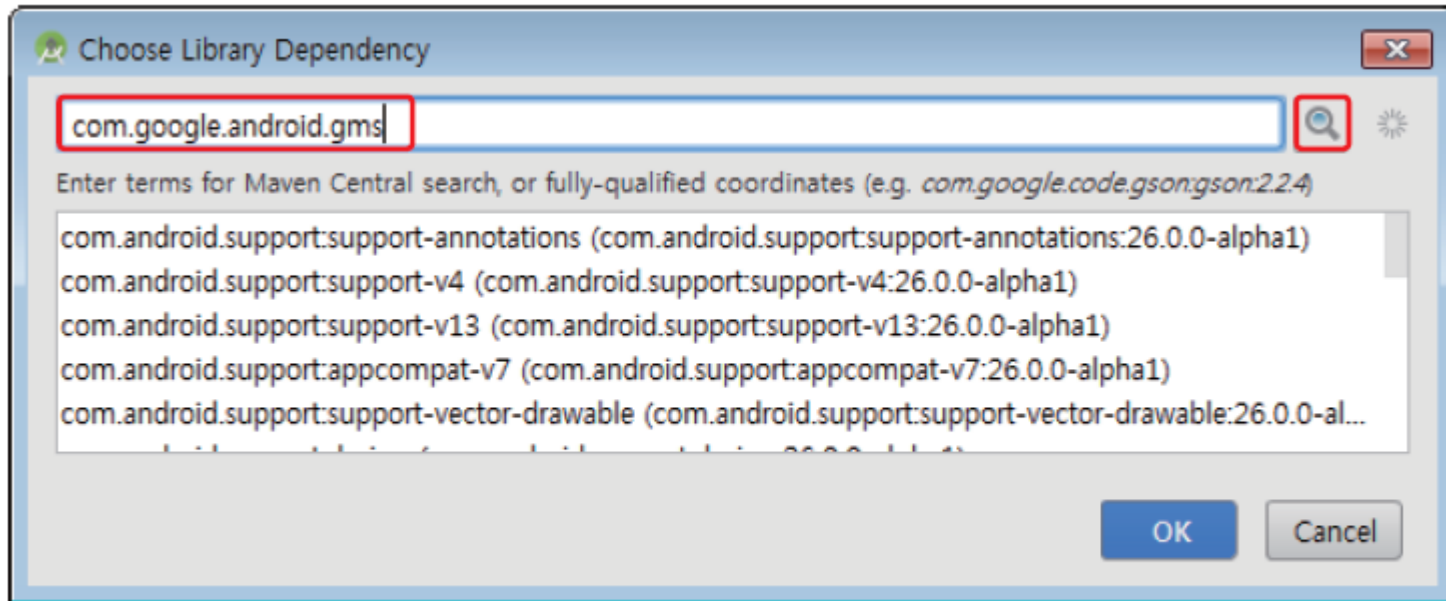
03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스 사용에 필요한 라이브러리 추가



(b) "com.google.android.gms"로 검색

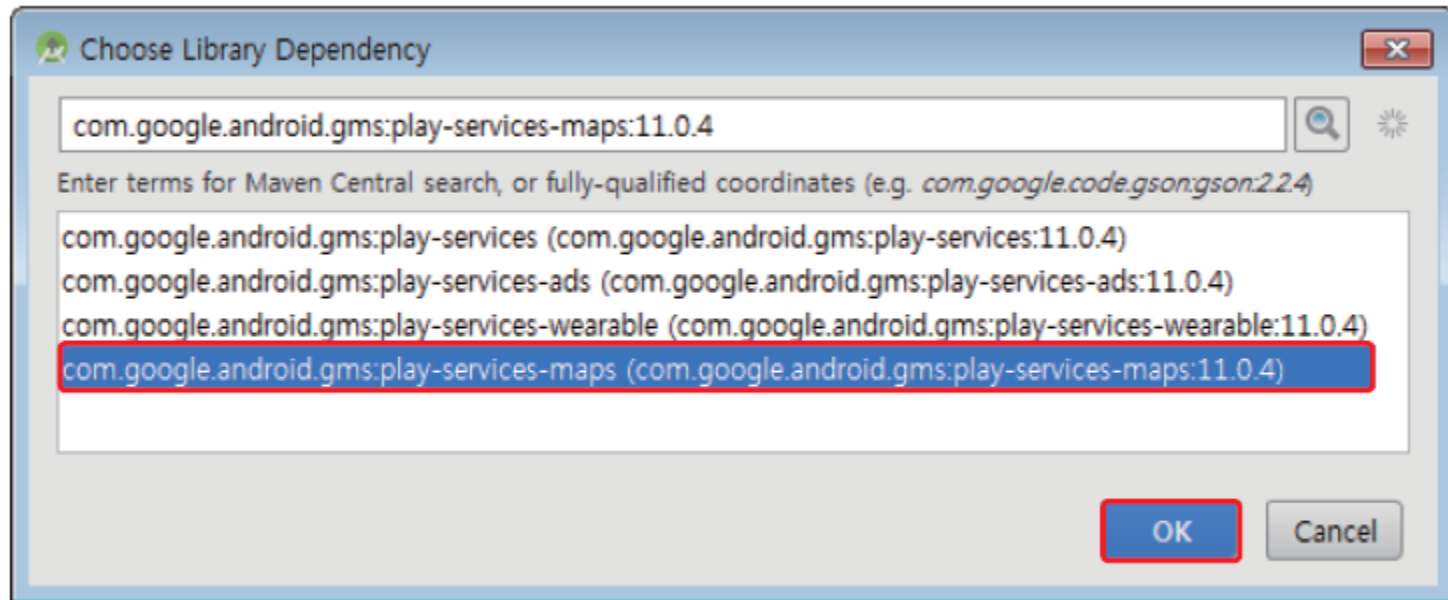
03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스에 필요한 라이브러리 추가



(c) 'com.google.android.gms:play-services-maps:숫자'를 선택하고 [OK] 클릭

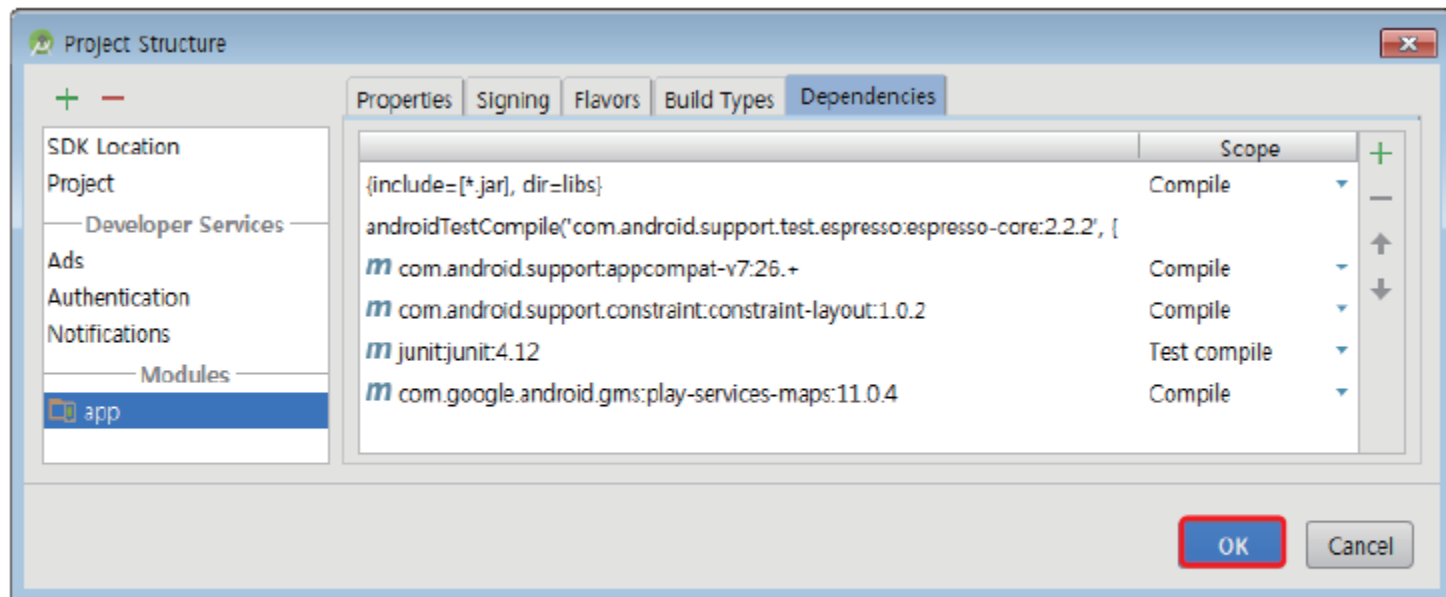
03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스에 필요한 라이브러리 추가



(d) [OK] 클릭

그림 13-15 구글 지도 서비스 라이브러리 추가

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

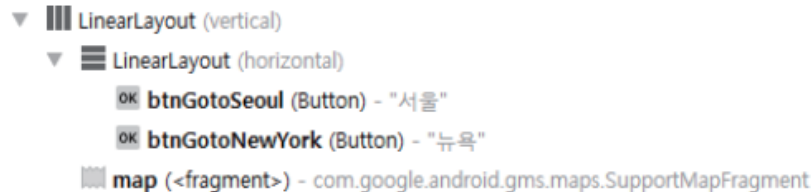


그림 13-16 컴포넌트 트리

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <LinearLayout
7          android:layout_width="match_parent"
8          android:layout_height="wrap_content">
9          <Button
```

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

```
10         android:id="@+id/btnGotoSeoul"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:onClick="mOnClick"
14         android:text="서울"/>
15     <Button
16         android:id="@+id/btnGotoNewYork"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:onClick="mOnClick"
20         android:text="뉴욕"/>
21 </LinearLayout>
22 <fragment
23     android:id="@+id/map"
24     android:name="com.google.android.gms.maps.SupportMapFragment"
25     android:layout_width="match_parent"
26     android:layout_height="match_parent"/>
27 </LinearLayout>
```

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

- 컴포넌트 트리를 참고하여 res/layout/activity_main.xml 수정

menu_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3      <item
4          android:id="@+id/normal"
5          android:title="일반"/>
6      <item
7          android:id="@+id/hybrid"
8          android:title="하이브리드"/>
9      <item
10         android:id="@+id/satellite"
11         android:title="위성"/>
12     <item
13         android:id="@+id/terrain"
14         android:title="지형"/>
15 </menu>
```

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스 사용을 위해 앱을 등록하고 API 키 얻기



(a) console.developers.google.com 사이트 방문

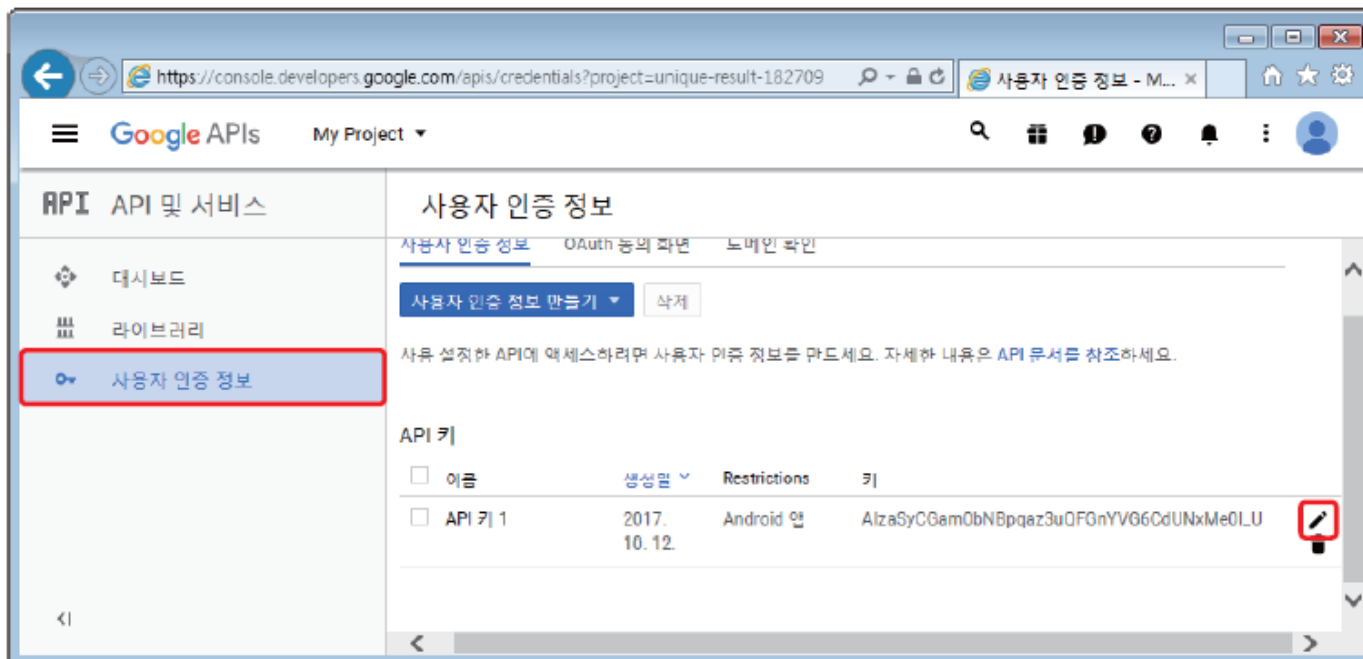
03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스를 위해 앱을 등록하고 API 키 얻기



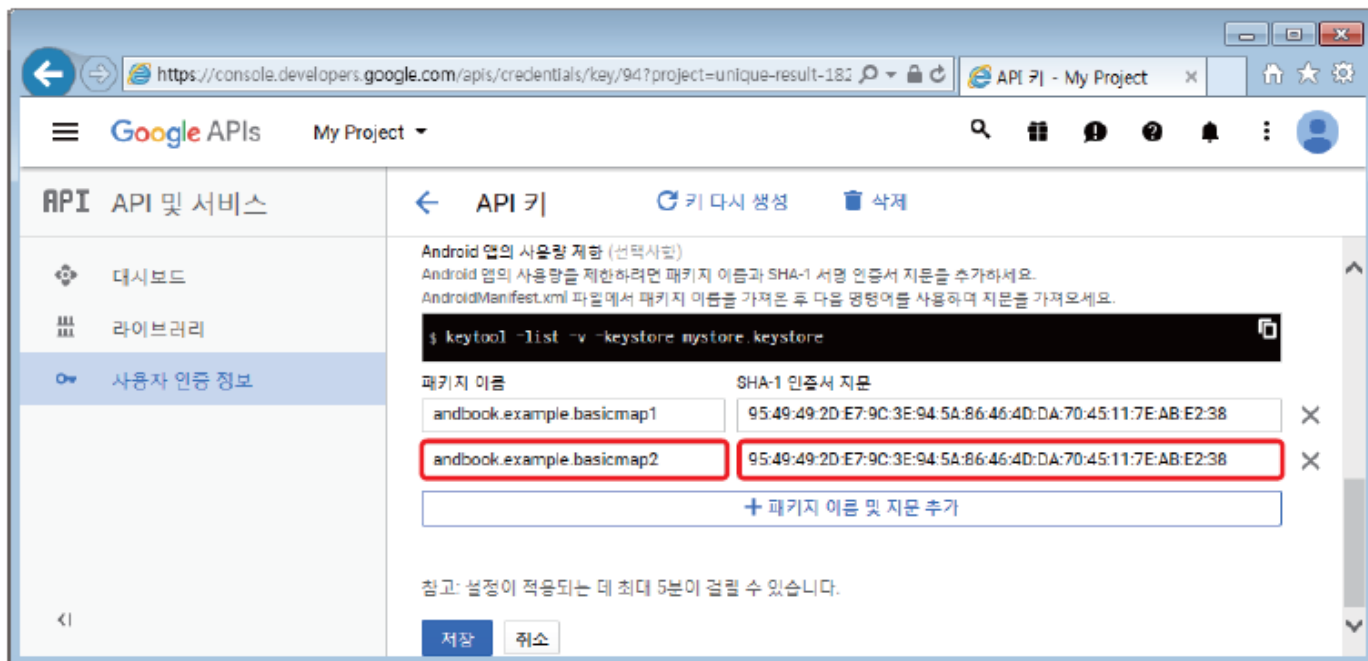
(b) 사용자 인증 정보에서 API 키 편집 시작

03 구글 지도 ▶ 구글 지도 활용

실습 13-5

BasicMap2

■ 구글 지도 서비스 사용을 위해 앱을 등록하고 API 키 얻기



(c) 패키지 이름과 SHA-1 인증서 지문 등록 후 저장

그림 13-17 구글 지도 서비스 사용을 위한 앱 등록과 API 키 발급

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

- 매니페스트에 퍼미션 추가
- 'build.gradle (Module: app)' 열기 -> targetSdkVersion을 22로 수정 -> 우측 상단의 'Sync Now' 클릭

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="andbook.example.basicmap2">
4      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
5      <application
6          ...
7          ...
12         <meta-data
13             android:name="com.google.android.geo.API_KEY"
14             android:value="AIzaSyCGam0bNBpqaz3uQFGnYVG6CdUNxMe0I_U"/>
15         <activity android:name=".MainActivity">
16             ...
17             ...
20         </activity>
21     </application>
22 </manifest>
```

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스 사용을 위해 앱을 등록하고 API 키를 얻는다

MainActivity.java

```
1  public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {
2
3      private GoogleMap mMap;
4      private static final int DEFAULT_ZOOM = 13;
5
6      @Override
7      protected void onCreate(Bundle savedInstanceState) {
8          super.onCreate(savedInstanceState);
9          setContentView(R.layout.activity_main);
10
11          SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
12              .findFragmentById(map);
13          mapFragment.getMapAsync(this);
14      }
15
16      @Override
17      public void onMapReady(GoogleMap googleMap) {
18          mMap = googleMap;
19          mMap.setMyLocationEnabled(true);
20          mMap.getUiSettings().setMyLocationButtonEnabled(true);
21          mMap.getUiSettings().setZoomControlsEnabled(true);
22      }
```

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스 사용을 위해 앱을 등록하고 API 키를 얻는다

```
23
24     public void mOnClick(View v) {
25         if (mMap == null) return;
26         switch (v.getId()) {
27             case R.id.btnGotoSeoul:
28                 mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
29                     new LatLng(37.548974, 126.994013), DEFAULT_ZOOM));
30                 break;
31             case R.id.btnGotoNewYork:
32                 mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
33                     new LatLng(40.784001, -73.971980), DEFAULT_ZOOM));
34                 break;
35         }
36     }
37
38     @Override
39     public boolean onCreateOptionsMenu(Menu menu) {
40         MenuInflater inflater = getMenuInflater();
41         inflater.inflate(R.menu.menu_main, menu);
```

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 구글 지도 서비스 사용을 위해 앱을 등록하고 API 키를 얻는다

```
42     return super.onCreateOptionsMenu(menu);
43 }
44
45 @Override
46 public boolean onOptionsItemSelected(MenuItem item) {
47     switch (item.getItemId()) {
48         case R.id.normal:
49             mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
50             break;
51         case R.id.hybrid:
52             mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
53             break;
54         case R.id.satellite:
55             mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
56             break;
57         case R.id.terrain:
58             mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
59             break;
60     }
61     return super.onOptionsItemSelected(item);
62 }
63 }
```

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 실행 화면



(a) 초기 화면



(b) [서울] 클릭



(c) [뉴욕] 클릭

03 구글 지도 ▶ 구글 지도 활용



실습 13-5

BasicMap2

■ 실행 화면



(d) [하이브리드] 선택



(e) [위성] 선택



(f) [지형] 선택

그림 13-18 실행 화면

