

第9章 位置服务与地图应用



学习目标

AIMS

- 01 了解位置服务的概念
- 02 了解地图密钥的申请方法
- 03 掌握获取位置信息的方法
- 04 掌握MapView和BaiduMap的使用方法
- 05 掌握百度地图覆盖层的使用方法

9.1

位置服务

• 位置服务

- 位置服务（Location-Based Services, LBS），又称定位服务或基于位置的服务，融合了GPS定位、移动通信、导航等多种技术，提供与空间位置相关的综合应用服务
- 位置服务首先在日本得到商业化的应用
- 2001年7月，DoCoMo发布了第一款具有三角定位功能的手持设备
- 2001年12月，KDDI发布第一款具有GPS功能的手机
- 基于位置的服务发展更加迅速，涉及商务、医疗、工作和生活的各个方面，为用户提供定位、追踪和敏感区域警告等一系列服务

9.1

位置服务

• 位置服务

- Android平台支持提供位置服务的API，在开发过程中主要用到LocationManager和LocationProviders对象
- LocationManager可以用来获取当前的位置，追踪设备的移动路线，或设定敏感区域，在进入或离开敏感区域时设备会发出特定警报
- LocationProviders则是提供定位功能的组件集合，集合中的每种组件以不同的技术提供设备的当前位置，区别在于定位的精度、速度和成本等方面

9.1 位置服务

- 位置服务

- 为了使开发的程序能够提供位置服务，首先的问题是如何获取LocationManager
- 获取LocationManager可以通过调用android.app.Activity.getSystemService()函数获取，代码如下：

```
1 String serviceString = Context.LOCATION_SERVICE;
```

```
2 LocationManager locationManager = (LocationManager) getSystemService(serviceString);
```

- 代码第1行的Context.LOCATION_SERVICE指明获取的是位置服务
- 代码第2行的getSystemService()函数，可以根据服务名称获取Android提供的系统级服务

9.1 位置服务

•位置服务

•Android支持的系统级服务表

Context类的静态常量	值	返回对象	说明
LOCATION_SERVICE	location	LocationManager	控制位置等设备的更新
WINDOW_SERVICE	window	WindowManager	最顶层的窗口管理器
LAYOUT_INFLATER_SERVICE	layout_inflater	LayoutInflater	将XML资源实例化为View
POWER_SERVICE	power	PowerManager	电源管理
ALARM_SERVICE	alarm	AlarmManager	在指定时间接受Intent
NOTIFICATION_SERVICE	notification	NotificationManager	后台事件通知
KEYGUARD_SERVICE	keyguard	KeyguardManager	锁定或解锁键盘
SEARCH_SERVICE	search	SearchManager	访问系统的搜索服务
VIBRATOR_SERVICE	vibrator	Vibrator	访问支持振动的硬件
CONNECTIVITY_SERVICE	connection	ConnectivityManager	网络连接管理
WIFI_SERVICE	wifi	WifiManager	Wi-Fi连接管理
INPUT_METHOD_SERVICE	input_method	InputMethodManager	输入法管理

9.1 位置服务

- 位置服务

- 在获取到LocationManager后，还需要指定LocationManager的定位方法，然后才能够调用
LocationManager.getLastKnownLocation()方法获取当前位置
- 目前LocationManager中主要有两种定位方法
 - **GPS定位**：可以提供更加精确的位置信息，但定位速度和质量受到卫星数量和环境情况的影响
 - **网络定位**：提供的位置信息精度差，但速度较GPS定位要迅速

9.1 位置服务

- 位置服务

- LocationManager支持定位方法

LocationManager类的静态常量	值	说明
GPS_PROVIDER	gps	使用GPS定位，利用卫星提供精确的位置信息，需要android.permissions.ACCESS_FINE_LOCATION用户权限
NETWORK_PROVIDER	network	使用网络定位，利用基站或WiFi访问的提供近似的位置信息，需要具有如下权限： android.permission.ACCESS_COARSE_LOCATION 或android.permission.ACCESS_FINE_LOCATION.

9.1 位置服务

• 位置服务

- 在指定LocationManager的定位方法后，则可以调用getLastKnownLocation()方法获取当前的位置信息。以使用GPS定位为例，获取位置信息的代码如下：

```
1 String provider = LocationManager.GPS_PROVIDER;  
2 Location location = locationManager.getLastKnownLocation(provider);
```

- 代码第2行返回的Location对象中，包含了可以确定位置的信息，如经度、纬度和速度等
- 通过调用Location中的getLatitude()和getLongitude()方法可以分别获取位置信息中的纬度和经度，示例代码如下：

```
1 double lat = location.getLatitude();  
2 double lng = location.getLongitude();
```

9.1 位置服务

• 位置服务

- 在很多提供定位服务的应用程序中，不仅需要获取当前的位置信息，还需要监视位置的变化，在位置改变时调用特定的处理方法
- LocationManager提供了一种便捷、高效的位置监视方法 `requestLocationUpdates()`，可以根据位置的距离变化和时间间隔设定，产生位置改变事件的条件，这样可以避免因微小的距离变化而产生大量的位置改变事件
- LocationManager中设定监听位置变化的代码如下：
`locationManager.requestLocationUpdates(provider, 2000, 10, locationListener);`
 - 第1个参数是定位的方法，GPS定位或网络定位
 - 第2个参数是产生位置改变事件的时间间隔，单位为毫秒
 - 第3个参数是距离条件，单位是米
 - 第4个参数是回调函数，用于处理位置改变事件

9.1 位置服务

• 位置服务

- 代码将产生位置改变事件的条件设定为距离改变为10米，时间间隔为2秒
- 实现locationListener的代码如下：

```
1  LocationListener locationListener = new LocationListener(){
2      public void onLocationChanged(Location location) {
3      }
4      public void onProviderDisabled(String provider) {
5      }
6      public void onProviderEnabled(String provider) {
7      }
8      public void onStatusChanged(String provider, int status, Bundle extras) {
9      }
10 };
```

9.1

位置服务

- 位置服务

- 第2行代码onLocationChanged()在位置改变时被调用
- 第4行的onProviderDisabled()在用户禁用具有定位功能的硬件时被调用
- 第6行的onProviderEnabled()在用户启用具有定位功能的硬件时被调用
- 第8行的onStatusChanged()在定位功能硬件状态改变时被调用，例如，从不可获取位置信息状态到可以获取位置信息的状态，反之亦然

9.1

位置服务

- 位置服务

- 为了使GPS定位功能生效，还需要在AndroidManifest.xml文件中加入用户许可
- 实现代码如下：

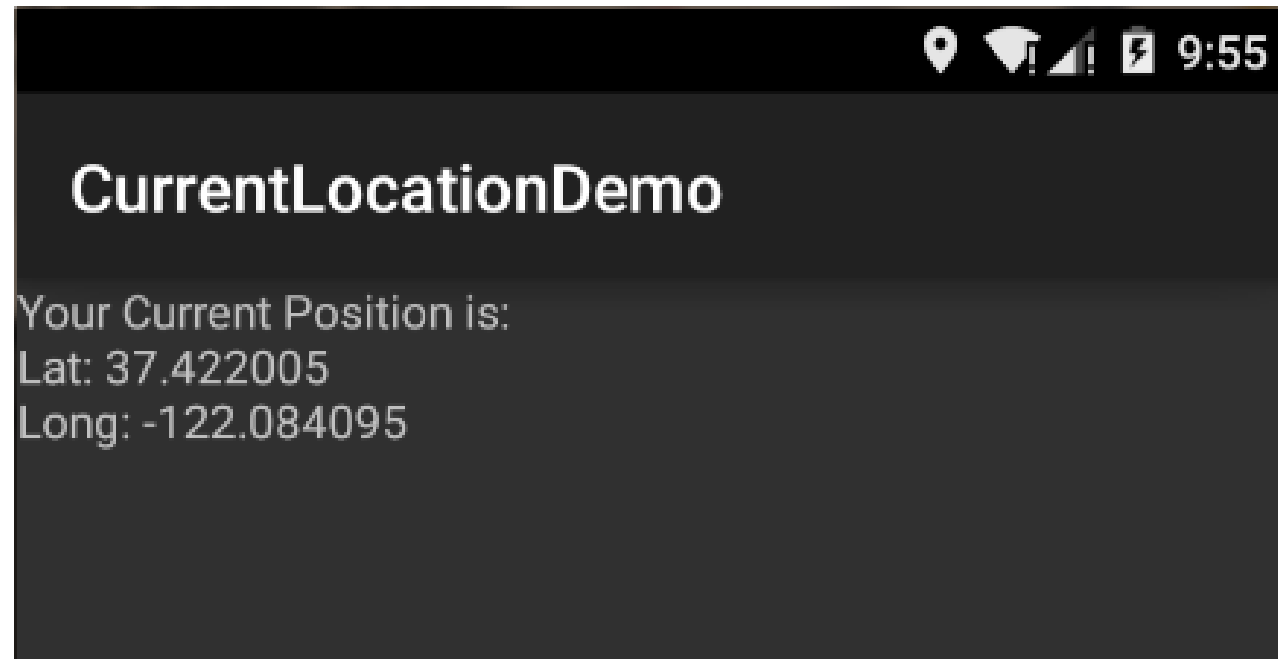
```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>  
<uses-permission android:name="android.permission.INTERNET" />
```

9.1

位置服务

- 位置服务

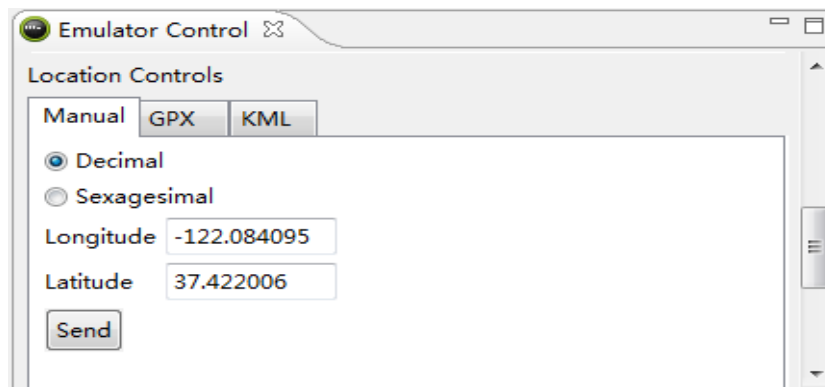
- CurrentLocationDemo是一个提供基本位置服务的示例，可以显示当前位置信息，并能够监视设备的位置变化
- CurrentLocationDemo的用户界面如下图所示：



9.1 位置服务

• 位置服务

- 位置服务一般都需要使用设备上的硬件，最理想的调试方式是将程序上传到物理设备上运行，但在没有物理设备的情况下，也可以使用Android模拟器提供的虚拟方式模拟设备的位置变化，调试具有位置服务的应用程序
- 首先打开DDMS中的模拟器控制，在Location Controls中的Longitude和Latitude部分输入设备当前的经度和纬度，然后点击Send按钮，就将虚拟的位置信息发送到Android模拟器中，如下图所示：



9.1

位置服务

•位置服务

- 在程序运行过程中，可以在模拟器控制器中改变经度和纬度坐标值，程序在检测到位置的变化后，会将最新的位置信息显示在界面上
- 下面给出CurrentLocationDemo示例中CurrentLocationDemoActivity.java文件的完整代码

```
1 package edu.hrbeu.CurrentLocationDemo;  
2  
3 import android.Manifest;  
4 import android.app.Activity;  
5 import android.content.Context;  
6 import android.content.pm.PackageManager;  
7 import android.location.Location;  
8 import android.location.LocationListener;
```

9.1

位置服务

•位置服务

```
9 import android.location.LocationManager;
10 import android.os.Bundle;
11 import android.widget.TextView;
12 import androidx.core.app.ActivityCompat;
13
14 public class CurrentLocationDemoActivity extends Activity {
15     private static final int REQUEST_EXTERNAL_STORAGE = 1;
16     private static String[] PERMISSIONS_STORAGE = {
17         Manifest.permission.ACCESS_COARSE_LOCATION,
18         Manifest.permission.ACCESS_FINE_LOCATION,
19         Manifest.permission.INTERNET};
20     @Override
21     public void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.main);
```

9.1

位置服务

•位置服务

```
24     String serviceString = Context.LOCATION_SERVICE;
25     LocationManager locationManager = (LocationManager) getSystemService(serviceString);
26
27     String provider = LocationManager.GPS_PROVIDER;
28     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
    {
29         ActivityCompat.requestPermissions(this, PERMISSIONS_STORAGE,
30             REQUEST_EXTERNAL_STORAGE);
31     }
32
33     Location location = locationManager.getLastKnownLocation(provider);
34
35     getLocationInfo(location);
36
```

9.1

位置服务

•位置服务

```
37     locationManager.requestLocationUpdates(provider, 2000, 0, locationManager);
38 }
39
40 private void getLocationInfo(Location location){
41     String latLongInfo;
42     TextView locationText = (TextView)findViewById(R.id.label);
43
44     if (location != null){
45         double lat = location.getLatitude();
46         double lng = location.getLongitude();
47         latLongInfo = "Lat: " + lat + "\nLong: " + lng;
48     }
49     else{
50         latLongInfo = "No location found";
51     }
```

9.1

位置服务

•位置服务

```
52     locationText.setText("Your Current Position is:\n" + latLongInfo);
53 }
54
55 private final LocationListener locationListener = new LocationListener(){
56
57     @Override
58     public void onLocationChanged(Location location) {
59         getLocationInfo(location);
60     }
61
62     @Override
63     public void onProviderDisabled(String provider) {
64         getLocationInfo(null);
65     }
66 }
```

9.1 位置服务

• 位置服务

```
67     @Override
68     public void onProviderEnabled(String provider) {
69         getLocationInfo(null);
70     }
71
72     @Override
73     public void onStatusChanged(String provider, int status, Bundle extras) {
74
75     }
76 };
77 }
```

9.2 百度地图应用

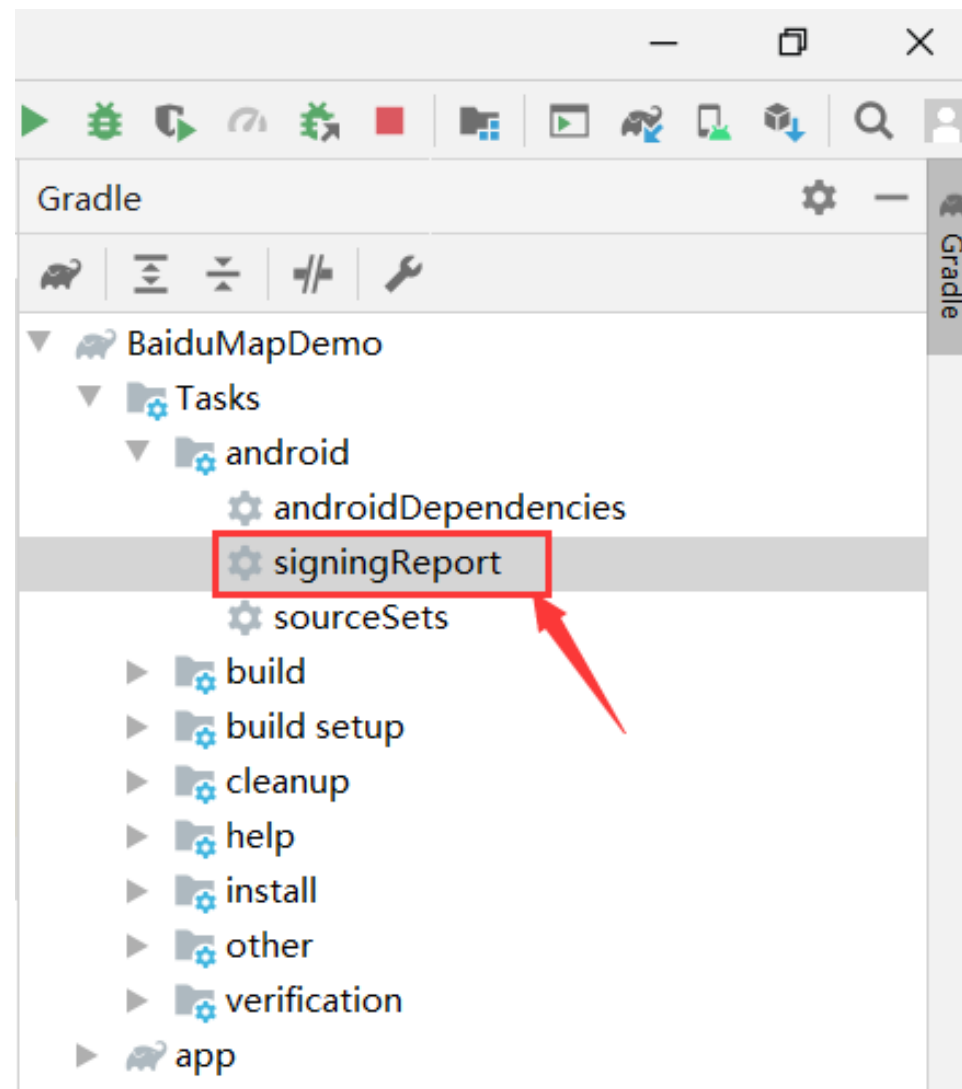
• 9.2.1 申请地图密钥

- 为了在手机中更直观地显示地理信息，程序开发人员可以直接使用百度提供的地图服务，实现地理信息的可视化开发
- 只要使用MapView（`com.baidu.mapapi.map.MapView`）就可以将百度地图嵌入到Android应用程序中
- 但在使用MapView进行开发前，必须向百度申请经过验证的“地图密钥”（Map API Key），这样才能正常使用百度的地图服务
- “地图密钥”是访问百度地图数据的密钥，无论是模拟器还是在真实设备中需要使用这个密钥

9.2 百度地图应用

•9.2.1 申请地图密钥

- 获取证书的保存地址的方法：首先打开Android Studio，在Android Studio右侧边栏，点击Gradle工具栏，找到Tasks目录下的signingReport文件，然后双击运行

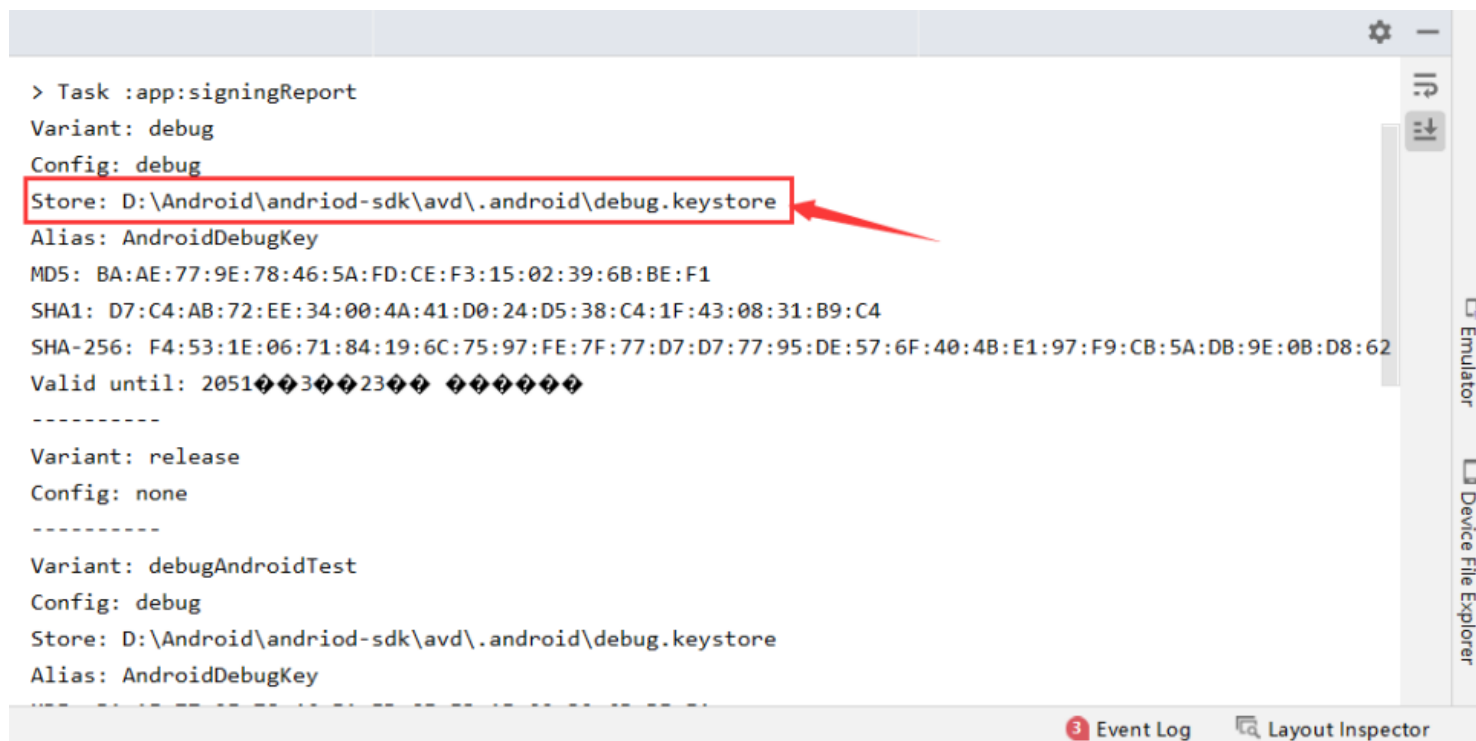


9.2

百度地图应用

•9.2.1 申请地图密钥

- 运行成功后，Android Studio下方控制台会输出对应的文件内容，如下图所示Store后面的路径就是Debug证书的保存位置。



```
> Task :app:signingReport
Variant: debug
Config: debug
Store: D:\Android\andriod-sdk\avd\.android\debug.keystore
Alias: AndroidDebugKey
MD5: BA:AE:77:9E:78:46:5A:FD:CE:F3:15:02:39:6B:BE:F1
SHA1: D7:C4:AB:72:EE:34:00:4A:41:D0:24:D5:38:C4:1F:43:08:31:B9:C4
SHA-256: F4:53:1E:06:71:84:19:6C:75:97:FE:7F:77:D7:D7:77:95:DE:57:6F:40:4B:E1:97:F9:CB:5A:DB:9E:0B:D8:62
Valid until: 2051-03-23 00:00:00

-----
Variant: release
Config: none
-----

Variant: debugAndroidTest
Config: debug
Store: D:\Android\andriod-sdk\avd\.android\debug.keystore
Alias: AndroidDebugKey
-----
```

9.2 百度地图应用

• 9.2.1 申请地图密钥

- 为了获取Debug证书MD5散列值的，需要打开命令行工具CMD，然后切换到keystore的目录，即D:\Android\andriod-sdk\avd\.android，输入如下命令：

keytool -list -v -keystore debug.keystore

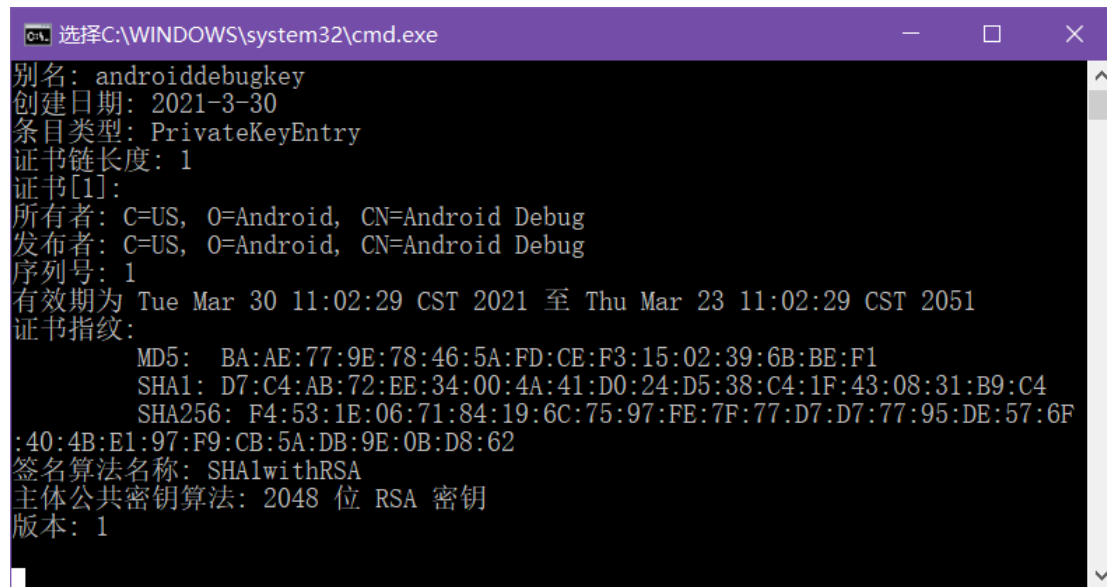
- Keytool是JDK提供的工具，如果提示无法找到keytool，可以将<Java SDK>/bin的路径添加到系统的PATH变量中

9.2

百度地图应用

•9.2.1 申请地图密钥

- 提示输入keystore密码时，输入密码android，或直接输入回车，MD5、SHA1和SHA256散列值将都显示出来。如下图所示，笔者的SHA1散列值为
D7:C4:AB:72:EE:34:00:4A:41:D0:24:D5:38:C4:1F:43:08:31:B9:C4



```
C:\WINDOWS\system32\cmd.exe
别名: androiddebugkey
创建日期: 2021-3-30
条目类型: PrivateKeyEntry
证书链长度: 1
证书[1]:
所有者: C=US, O=Android, CN=Android Debug
发布者: C=US, O=Android, CN=Android Debug
序列号: 1
有效期为 Tue Mar 30 11:02:29 CST 2021 至 Thu Mar 23 11:02:29 CST 2051
证书指纹:
    MD5: BA:AE:77:9E:78:46:5A:FD:CE:F3:15:02:39:6B:BE:F1
    SHA1: D7:C4:AB:72:EE:34:00:4A:41:D0:24:D5:38:C4:1F:43:08:31:B9:C4
    SHA256: F4:53:1E:06:71:84:19:6C:75:97:FE:7F:77:D7:D7:77:95:DE:57:6F:
:40:4B:E1:97:F9:CB:5A:DB:9E:0B:D8:62
签名算法名称: SHA1withRSA
主体公共密钥算法: 2048 位 RSA 密钥
版本: 1
```

9.2

百度地图应用

•9.2.1 申请地图密钥

- 申请“地图密钥”的最后一步是打开申请页面，输入应用名称、SHA1散列值、应用程序的包名以及选择应用类型，比如本次使用示例程序包名为edu.hrbeu.BaiduMapDemo，应用名称输入：BaiduMapDemo，应用类型选择Android SDK，发布版SHA1和开发版SHA1输入：
D7:C4:AB:72:EE:34:00:4A:41:D0:24:D5:38:C4:1F:43:08:31:B9:C4，
PackageName输入：edu.hrbeu.BaiduMapDemo。
- 申请页面的地址是<http://lbsyun.baidu.com/apiconsole/key>

9.2 百度地图应用

- 9.2.1 申请地图密钥
- 获取Map API Key页面

[返回](#) 创建应用

应用名称: BaiduMapDemo ✔ 输入正确

应用类型: Android SDK

启用服务:

☒ Android地图SDK	☒ Android定位SDK	☒ Android导航离线SDK
☒ Android导航SDK	☒ 静态图	☒ 全景静态图
☒ 坐标转换	☒ 鹰眼轨迹	☒ 全景URL API
☒ Android导航 HUD SDK	☒ 推荐上车点	☒ 地图SDK境外底图
☒ 地理编码	☒ 逆地理编码	☒ 全景SDK
☒ Android AR识别SDK	☒ 国内天气查询	☒ 海外天气查询

参考文档:

Android地图SDK	Android定位SDK	Android导航	静态图
全景静态图	坐标转换	鹰眼轨迹	推荐上车点
地图SDK境外底图	地理编码	逆地理编码	

*发布版SHA1: 如何获取 D7:C4:AB:72:EE:34:00:4A:41:D0:24:D5:38:C4:1F:43:08:31:B9: ✔ 输入正确

开发版SHA1: D7:C4:AB:72:EE:34:00:4A:41:D0:24:D5:38:C4:1F:43:08:31:B9: ✔ 输入正确

*PackageName: 如何获取 edu.hrbeu.BaiduMapDemo ✔ 输入正确

安全码: D7:C4:AB:72:EE:34:00:4A:41:D0:24:D5:38:C4:1F:43:08:31:B9:C4;edu.hrbeu.BaiduMapDemo

Android SDK安全码组成: SHA1+包名。(查看详细配置方法) 新申请的Mobile与Browser类型的ak不再支持云存储接口的访问, 如要使用云存储, 请申请Server类型ak。

提交

9.2 百度地图应用

•9.2.1 申请地图密钥

- 输入后，点击提交按钮，将生成申请“地图密钥”的获取结果，如下图所示：

百度地图 开放平台

为什么选择百度地图 NEW! 功能与服务 NEW! 解决方案 NEW! 开发文档 反馈与帮助 服务升级 控制台

控制台看板

应用管理

我的应用

回收站

额度管理

商用授权

特色服务平台

个人中心

工单

站内信

我的应用

请输入AK

创建应用 回收站

每页显示30条

应用编号	应用名称	访问应用 (AK)	应用类别	备注信息 (双击更改)	应用配置	用量查询
24078377	BaiduMap...	pYCqPRfLGcNm6qY	android应用		设置 删除	

您当前创建了 1 个应用

上一页 1 下一页

9.2 百度地图应用

•9.2.1 申请地图密钥

- 笔者获取的“地图密钥”是pYCqPRfLGcNm6qYht4rECKwNCKL4URY，在以后使用到MapView的时候都需要输入这个密钥
- 但需要注意的是，读者必须根据Debug证书的SHA1值，自己到百度地图开放平台网站上申请一个用于调试程序的“地图密钥”，而不能使用上面笔者申请到的“地图密钥”，并且密钥与应用名称和应用程序包名要一一对应

9.2 百度地图应用

• 9.2.2 使用百度地图

- MapView是地图的显示控件，可以设置不同的显示模式，例如普通地图（2D，3D）、卫星图、实时交通图和开放高清4K地图
- BaiduMap则是MapView的控制器，可以控制MapView的显示中心和缩放级别等功能
- 下面的内容以BaiduMapDemo为例，说明如何在Android系统中开发百度地图程序
- 这个示例将在程序内部设置一个坐标点，然后在程序启动时，使用MapView控件在地图上显示这个坐标点的位置

9.2 百度地图应用

•9.2.2 使用百度地图

- 因为普通版本的Android SDK并不包含百度地图的开发扩展库，因此建立工程时需将百度地图开发包添加到工程中，这样就可以使用百度地图的所有功能
- 下面介绍如何下载、导入百度地图开发包以及开发包的使用

9.2

百度地图应用

• 9.2.2 使用百度地图

• 1. 下载开发包

- 从百度地图开发平台网站上可以下载最新的百度地图开发包，在这里笔者只选择了基础定位和基础地图，如果后期想使用其他功能，可以下载更高级别的开发包
- 本次选择开发包格式为JAR，应用发布平台为标准开发包
- 下载地址为
<http://lbsyun.baidu.com/index.php?title=sdk>

Android SDK下载

选择功能

定位SDK (二选一)

☒ 基础定位

开发包体积最小，但只包含基础定位能力(卫星定位/WiFi/基站)、基础位置描述能力

☐ 全量定位

包含基础定位、离线定位、室内高精度定位能力，以及更多辅助功能(如地理围栏等)

地图SDK

☒ 基础地图 (含室内图)

包括基础地图、室内图、卫星图、路况图展示和各种覆盖物绘制，还有与地图相关的操作、事件监听

☐ 步骑行导航 (含基础地图)

包含基础地图全部功能，并提供步行导航和骑行导航的能力

☐ 检索

包括POI检索、路线规划、地理编码查询等功能

导航SDK

☐ 驾车导航 (含TTS)

包含语音播报、路线规划和导航等功能

全景SDK

☐ 全景图

包含检索、展示全景图、与全景图交互以及添加覆盖物等功能

选择配置

开发包格式

AAR

☒ JAR

应用发布平台 ⑦

☒ 标准开发包

GooglePlay

开发包

示例代码

类参考

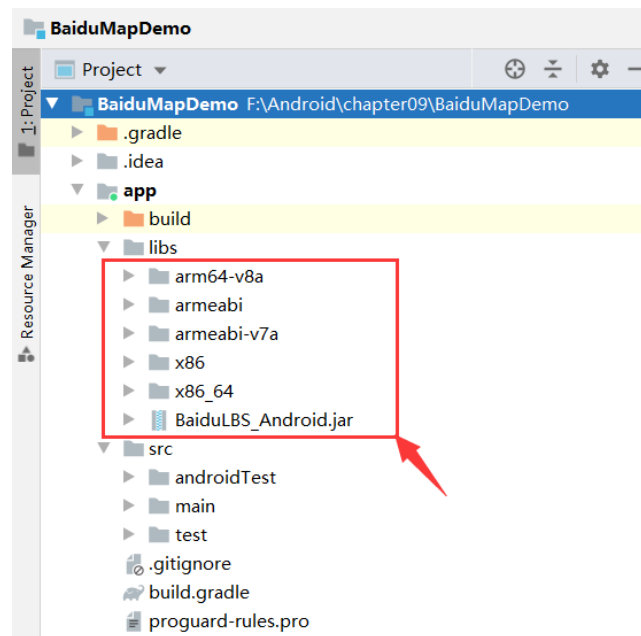
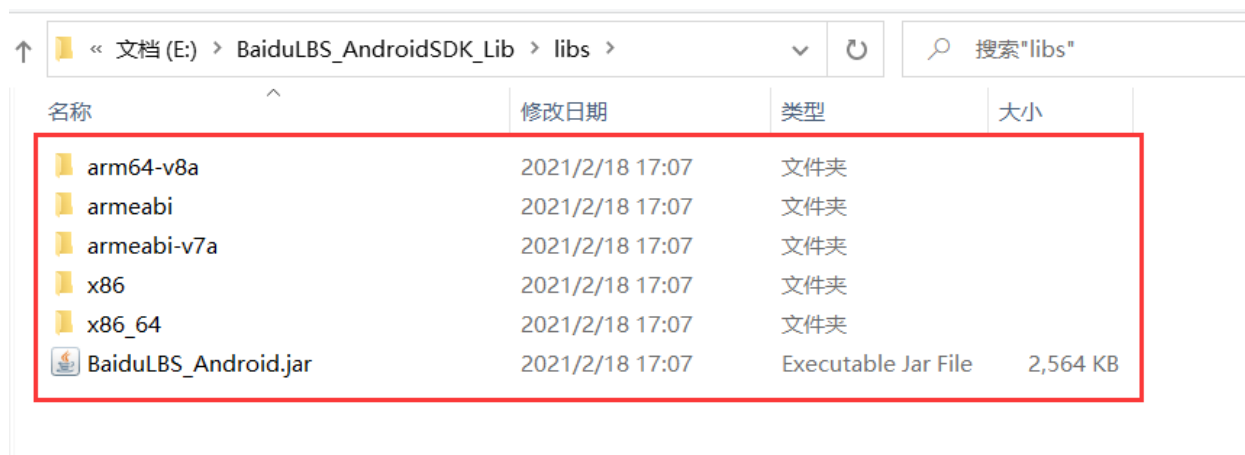
9.2

百度地图应用

•9.2.2 使用百度地图

•2.导入开发包

- 将下载好的压缩包文件进行解压如下图所示，再把解压后的libs文件夹下的所有文件复制到Android Studio项目工程的app/libs目录下



9.2 百度地图应用

• 9.2.2 使用百度地图

• 2. 导入开发包

- 百度地图开发包导入Android Studio工程之后还需要进行配置
- 首先在app目录下的build.gradle文件中配置android块的sourceSets标签并在dependencies块中新增implementation files('libs\\BaiduLBS_Android.jar')语句
- 然后点击右上角的“Sync Now”按钮进行同步

9.2 百度地图应用

• 9.2.2 使用百度地图

• 2. 导入开发包

• build.gradle 文件的部分配置代码如下

```
1. ....省略其他代码
2. android {
3. ....
4.     sourceSets {
5.         main {
6.             jniLibs.srcDir 'libs'
7.         }
8.     }
9. }
10. dependencies {
11.     implementation files('libs\\BaiduLBS_Android.jar')
12. ....
13. }
```

9.2

百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 配置完工程后，修改/res/layout/activity_main.xml文件，布局文件中加入一个MapView控件，用于在布局中放置地图
- activity_main.xml文件的完整代码如下

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.      android:orientation="vertical"
4.      android:layout_width="match_parent"
5.      android:layout_height="match_parent">
6.      <com.baidu.mapapi.map.MapView
7.          android:id="@+id/bmapView"
8.          android:layout_width="match_parent"
9.          android:layout_height="match_parent"
10.         android:enabled="true"
11.         android:clickable="true" />
12. </LinearLayout>
```

9.2 百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 使用百度地图的各功能组件之前需要调用

“`SDKInitializer.initialize(getApplicationContext())`”对百度地图SDK进行初始化操作。因此在`BaiduMapDemoApplication`类的`onCreate()`方法中完成对百度地图SDK的初始化

9.2

百度地图应用

•9.2.2 使用百度地图

•3.显示地图

- 下面给出整个BaiduMapDemoApplication.java文件的完整代码

```
1. package edu.hrbeu.BaiduMapDemo;
2.
3. import android.app.Application;
4. import com.baidu.mapapi.CoordType;
5. import com.baidu.mapapi.SDKInitializer;
6. public class BaiduMapDemoApplication extends Application {
7.
8.     @Override
9.     public void onCreate() {
10.         super.onCreate();
11.         SDKInitializer.initialize(getApplicationContext());
12.         SDKInitializer.setCoordType(CoordType.BD09LL);
13.     }
14. }
```

9.2 百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 代码第11行传入ApplicationContext初始化SDK。百度地图SDK的所有接口均支持百度坐标和国测局坐标，即BD09LL和GCJ02两种坐标类型，代码第12行默认设置为BD09LL坐标类型
- 仅在布局中添加MapView控件，还不能够直接在程序中调用这个控件，还需要创建地图Activity，负责管理MapView的生命周期

9.2 百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 下面给出整个BaiduMapDemoActivity.java文件的完整代码

```
1 package edu.hrbeu.BaiduMapDemo;  
2 import android.app.Activity;  
3 import android.os.Bundle;  
4  
5 import com.baidu.mapapi.map.BaiduMap;  
6 import com.baidu.mapapi.map.MapStatus;  
7 import com.baidu.mapapi.map.MapStatusUpdate;  
8 import com.baidu.mapapi.map.MapStatusUpdateFactory;  
9 import com.baidu.mapapi.map.MapView;  
10 import com.baidu.mapapi.model.LatLng;  
11
```

9.2 百度地图应用

- 9.2.2 使用百度地图
 - 3.显示地图

```
12 public class BaiduMapDemoActivity extends Activity {  
13     private BaiduMap baiduMap;  
14     private MapView mapView;  
15  
16     @Override  
17     protected void onCreate(Bundle savedInstanceState) {  
18         super.onCreate(savedInstanceState);  
19         setContentView(R.layout.activity_main);  
20  
21         mapView = (MapView) findViewById(R.id.bmapView);  
22         baiduMap = mapView.getMap();  
23     }
```

9.2 百度地图应用

- 9.2.2 使用百度地图
 - 3. 显示地图

```
24     Double lng = 126.676530486;  
25     Double lat = 45.7698895661;  
26     LatLng point = new LatLng(lat,lng);  
27     MapStatus mMapStatus = new MapStatus.Builder()  
28         .target(point)  
29         .zoom(11)  
30         .build();  
31     MapStatusUpdate mMapStatusUpdate = MapStatusUpdateFactory.newMapStatus(mMapStatus);  
32     baiduMap.setMapStatus(mMapStatusUpdate);  
33     baiduMap.setMapType(BaiduMap.MAP_TYPE_NORMAL);  
34 }  
35
```

9.2 百度地图应用

- 9.2.2 使用百度地图
 - 3.显示地图

```
36  @Override
37  protected void onResume() {
38      super.onResume();
39      mapView.onResume();
40  }
41  @Override
42  protected void onPause() {
43      super.onPause();
44      mapView.onPause();
45  }
46
```

9.2 百度地图应用

- 9.2.2 使用百度地图
 - 3. 显示地图

```
47  @Override
48  protected void onDestroy() {
49      super.onDestroy();
50      mapView.onDestroy();
51  }
52 }
```

9.2 百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 第22行获取了BaiduMap，用于在第32行和第33行设置MapView的地图状态和地图类型
- 第27行到第30行代码获取一个地图状态构造器来定义地图状态，
- 第28行设置MapView的“预订显示中心点”
- 第29行设置地图缩放级别
- 第30行创建地图状态对象
- 第24行和第25行设置地理坐标点的经度为126.676530486、纬度为45.7698895661
- 但在代码第28行，没有直接使用这个坐标的，而是将其转化为LatLng再使用

9.2

百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 第31行定义MapStatusUpdate对象，以便描述地图状态将要发生的变化
- 第33行是设置地图显示类型，设置BaiduMap.MAP_TYPE_NORMAL则为普通地图，设置BaiduMap.MAP_TYPE_SATELLITE则为卫星地图，设置BaiduMap.MAP_TYPE_NONE则为空白地图
- 第36行到第51行实现了在Activity中对地图生命周期管理

9.2 百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 运行前还需要在AndroidManifest.xml文件中添加配置刚获取的“地图密钥”（下方代码第17行到第19行）和权限许可（下方代码第8行到第11行），原因是获取百度地图是需要使用互联网和读写存储权限
- 代码第14行还需要声明对百度地图SDK进行初始化的Application

9.2 百度地图应用

•9.2.2 使用百度地图

•3.显示地图

- AndroidManifest.xml文件的完整代码如下：

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3  package="edu.hrbeu.BaiduMapDemo"
4  android:versionCode="1"
5  android:versionName="1.0" >
6  <uses-sdk android:minSdkVersion="14" />
7
8  <uses-permission android:name="android.permission.INTERNET" />
9  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
10 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
11 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
12
```

9.2 百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- AndroidManifest.xml文件的完整代码如下：

```
13 <application
14     android:name=".BaiduMapDemoApplication"
15     android:icon="@mipmap/ic_launcher"
16     android:label="@string/app_name">
17 <meta-data
18     android:name="com.baidu.lbsapi.API_KEY"
19     android:value="pYCqPRfLGcNm6qYht4rECKwNCKL4URY" />
20 <activity
21     android:label="@string/app_name"
22     android:name=".BaiduMapDemoActivity">
```

9.2 百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- AndroidManifest.xml文件的完整代码如下：

```
23     <intent-filter>
24         <action android:name="android.intent.action.MAIN" />
25         <category android:name="android.intent.category.LAUNCHER" />
26     </intent-filter>
27 </activity>
28 </application>
29 </manifest>
```

9.2

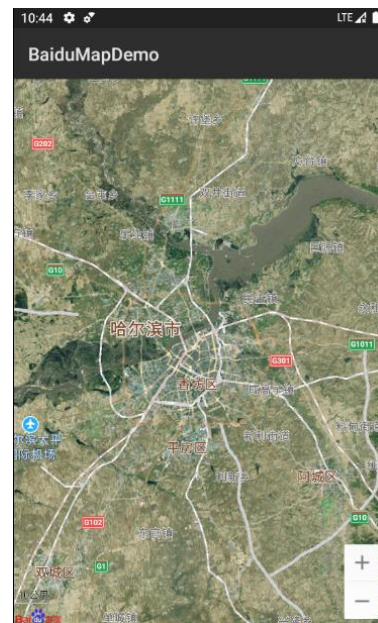
百度地图应用

• 9.2.2 使用百度地图

• 3. 显示地图

- 程序运行时需要连接互联网，运行结果如下图所示
- 地图模式

卫星模式



9.2 百度地图应用

•9.2.3 地图上使用覆盖层

- 在很多的地图应用中都需要在地图上显示信息或绘制图形
- 通过在MapView上添加覆盖层，可以在指定的位置添加注解、绘制图像或处理进行鼠标事件等
- 百度地图上可以加入多个覆盖层，所有覆盖层均都在地图图层之上，每个覆盖层均可以对用户的点击事件做出响应

9.2 百度地图应用

•9.2.3 地图上使用覆盖层

- 下面内容以MapOverlayDemo示例，说明如何在百度地图上添加点标记（Marker）覆盖层和文字（Text）覆盖层，并在预订的物理坐标上显示图标和提示信息
- 右图是MapOverlayDemo示例的运行结果



9.2 百度地图应用

• 9.2.3 地图上使用覆盖层

- 下面给出整个MapOverlayDemo.java文件的完整代码

```
1  package edu.hrbeu.MapOverlayDemo;
2
3  import android.app.Activity;
4  import android.os.Bundle;
5
6  import com.baidu.mapapi.map.BaiduMap;
7  import com.baidu.mapapi.map.BitmapDescriptor;
8  import com.baidu.mapapi.map.BitmapDescriptorFactory;
9  import com.baidu.mapapi.map.MapStatus;
10 import com.baidu.mapapi.map.MapStatusUpdate;
11 import com.baidu.mapapi.map.MapStatusUpdateFactory;
12 import com.baidu.mapapi.map.MapView;
13 import com.baidu.mapapi.map.MarkerOptions;
```

9.2 百度地图应用

•9.2.3 地图上使用覆盖层

```
14 import com.baidu.mapapi.map.OverlayOptions;
15 import com.baidu.mapapi.map.TextOptions;
16 import com.baidu.mapapi.model.LatLng;
17
18 public class MapOverlayDemoActivity extends Activity {
19     private BaiduMap baiduMap;
20     private MapView mapView;
21
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_main);
26
27         mapView = (MapView) findViewById(R.id.bmapView);
28         baiduMap = mapView.getMap();
```

9.2 百度地图应用

•9.2.3 地图上使用覆盖层

```
30 Double lng = 126.676530486;
31 Double lat = 45.7698895661;
32 LatLng point = new LatLng(lat,lng);
33 MapStatus mMapStatus = new MapStatus.Builder()
34     .target(point)
35     .zoom(11)
36     .build();
37 MapStatusUpdate mMapStatusUpdate = MapStatusUpdateFactory.newMapStatus(mMapStatus);
38 baiduMap.setMapStatus(mMapStatusUpdate);
39 baiduMap.setMapType(BaiduMap.MAP_TYPE_NORMAL);
40
41 BitmapDescriptor bitmap = BitmapDescriptorFactory
42     .fromResource(R.drawable.marker);
43
44 OverlayOptions markerOptions = new MarkerOptions()
```

9.2 百度地图应用

•9.2.3 地图上使用覆盖层

```
45  .position(point)
46  .icon(bitmap);
47
48  OverlayOptions textOptions = new TextOptions()
49  .text("标记点")
50  .bgColor(0xAAFFFF00)
51  .fontSize(24)
52  .fontColor(0xFFFF00FF)
53  .rotate(0)
54  .position(point);
55
56  baiduMap.addOverlay(markerOptions);
57  baiduMap.addOverlay(textOptions);
58  }
59 }
```

9.2 百度地图应用

• 9.2.3 地图上使用覆盖层

- 第41行到第42行的工厂类BitmapDescriptorFactory的formResource方法构建了一个Marker图标对象bitmap
- 第44行到第46行的MarkerOptions类是用来设置Marker覆盖物的属性
- 第45行的position()方法是用来设置Marker的位置坐标
- 第46行的icon()方法是用来设置Marker的图标
- 第48行到第54行的TextOptions类是用来设置文字覆盖层的属性
- 第49行设置文字内容

9.2 百度地图应用

• 9.2.3 地图上使用覆盖层

- 第50行设置文字背景颜色
- 第51行设置文字大小
- 第52行设置文字颜色
- 第53行设置文字旋转角度
- 第54行设置文字的位置坐标
- 在第56行和第57行使用addOverlay()方法将MarkerOptions和TextOptions对象添加到MapView中



习题：

- 1. 讨论位置服务和地图应用的发展前景。
- 2. 编程实现轨迹追踪软件。每间隔60秒，同时距离移动大于1米的情况下，记录一次位置信息，在百度地图上绘制600秒的行动轨迹。