

## 实验十一

### 传感器及地图相关应用

## 【实验目的】

1. 学会使用 android 手机不同的传感器，包括磁强计、加速度传感器等。
2. 学会使用 android 位置服务获得经纬度信息。
3. 学会使用百度地图基础API进行地图应用的开发。

## 【实验内容】

1. 实现最终结果实现效果如图 1所示：

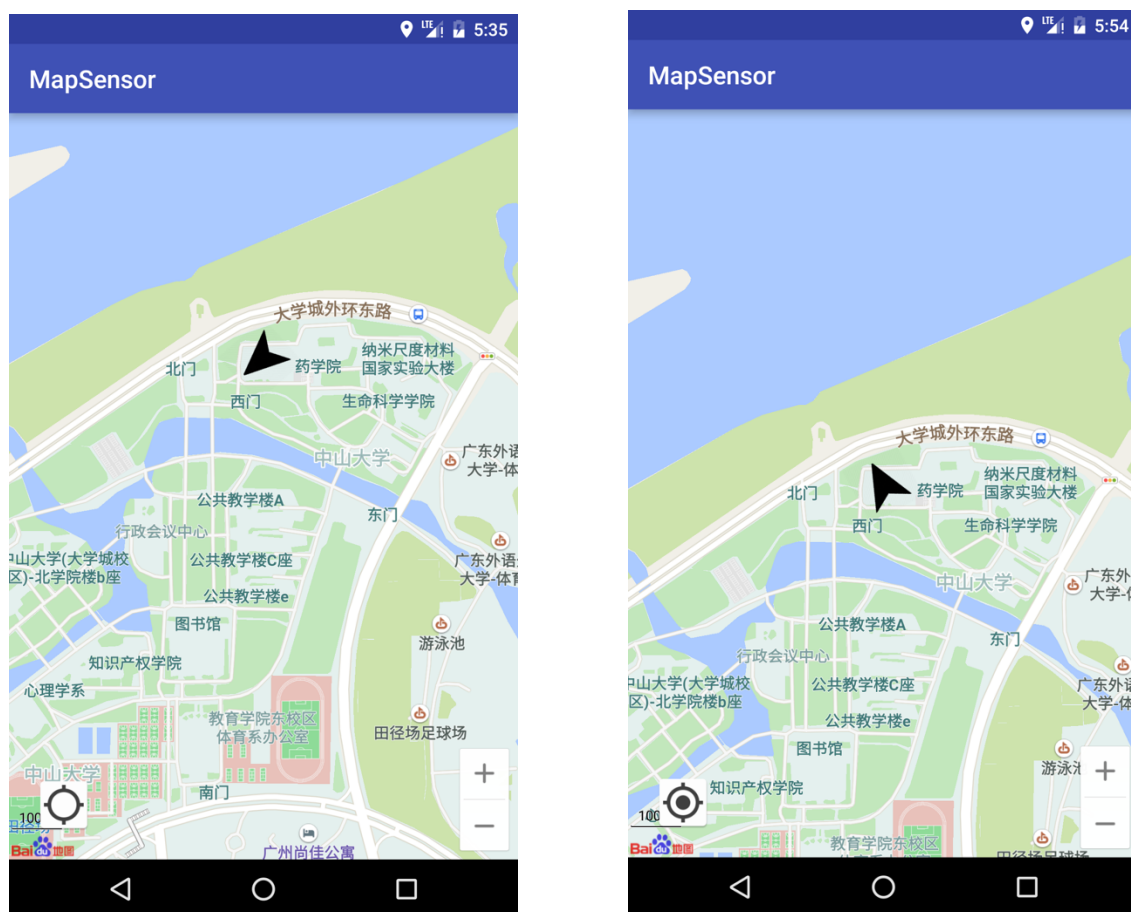







图 1 实验效果图

## 2. 主要分为以下三部分

- a) 调用百度地图SDK实现地图的展示以及交互，其中包括左下角的按钮要求实现的功能如下：
  - i. 当按钮处于  状态时：如果GPS位置有更新，那么更新用户当前所在位置，并使得当前位置的箭头图标一直保持在屏幕中心。
  - ii. 按钮处于  状态时，更新当前位置的箭头的时候不移动地图。
  - iii. 点击按钮可以切换这两种状态，当按钮进入  状态的时候，使得表示当前位置的箭头位于屏幕中央。
  - iv. 如果按钮处于  的状态手动平移百度地图的时候，地图进行平移并且按钮变为  状态。
- b) 获取当前所在位置与方向，并在百度地图上显示出来，如图中黑色箭头所示，箭头中心位置为定位结果，箭头方向为手机的朝向。
- c) 实现简单摇一摇功能，当晃动手机的时候作出相应的响应，其中，摇一摇之后会发生什么事情，可以发挥创造力自己实现，这也占本次实验的分数的一部分。

## 【参考内容】

1. 首先要申请百度的开发者密钥<sup>1</sup>，申请地址：

<http://lbsyun.baidu.com/apiconsole/key>

首先，在申请AK的表单中选择AndroidSDK，发现还要输入发布版SHA1以及报名这两个信息。

---

<sup>1</sup>申请密钥 Android SDK <http://lbsyun.baidu.com/index.php?title=androidsdk/guide/key>

应用名称: MapSensor

应用类型: Android SDK

启用服务:

<input checked="" type="checkbox"/> 云检索API	<input checked="" type="checkbox"/> Javascript API	<input checked="" type="checkbox"/> Place API v2
<input checked="" type="checkbox"/> Geocoding API v2	<input checked="" type="checkbox"/> IP定位API	<input checked="" type="checkbox"/> 路线交通API
<input checked="" type="checkbox"/> Android地图SDK	<input checked="" type="checkbox"/> Android导航离线SDK	<input checked="" type="checkbox"/> Android导航SDK
<input checked="" type="checkbox"/> 静态图API	<input checked="" type="checkbox"/> 全景静态图API	<input checked="" type="checkbox"/> 坐标转换API
<input checked="" type="checkbox"/> 鹰眼API	<input checked="" type="checkbox"/> 全景URL API	<input checked="" type="checkbox"/> Android导航 HUD SDK
<input checked="" type="checkbox"/> 云逆地理编码API	<input checked="" type="checkbox"/> Routematrix API	<input checked="" type="checkbox"/> 云地理编码API

\* 发布版SHA1: 请输入发布版SHA1

开发版SHA1: 请输入开发版SHA1

\* 包名: 请输入包名

安全码: 输入sha1和包名后自动生成

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。

其中, 应用包名为build.gradle里的ApplicationId, 并且保证这个ApplicationId与Manifest里的package字段名是一样的。

```
manifest
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="sysu.mobile.mapsensor">
```

图 2 Manifest 里的应用包名

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "24.0.0"

    defaultConfig {
        applicationId "sysu.mobile.mapsensor"
        minSdkVersion 19
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
```

图 3 build.gradle 里的应用包名

发布版SHA1为了方便直接使用Android默认的debug.keystore里的SHA1，进入用户文件夹（Windows为C:\Users\username，Linux为/home/username，Mac为/Users/username）后，输入如下命令即可看到SHA1值（输入密钥库口令里直接回车）

```

~$ cd .android && keytool -list -v -keystore debug.keystore
输入密钥库口令：

***** WARNING WARNING WARNING *****
* 存储在您的密钥库中的信息的完整性 *
* 尚未经过验证！ 为了验证其完整性， *
* 必须提供密钥库口令。 *
***** WARNING WARNING WARNING *****

密钥库类型：JKS
密钥库提供方：SUN

您的密钥库包含 1 个条目

别名：androiddebugkey
创建日期：2015-6-7
条目类型：PrivateKeyEntry
证书链长度：1
证书[1]:
所有者：CN=Android Debug, O=Android, C=US
发布者：CN=Android Debug, O=Android, C=US
序列号：
有效期开始日期：Sun Jun 07 13:11:27 CST 2015，截止日期：Tue May 30 13:11:27 CST 2045
证书指纹：
MD5:
SHA1:
SHA256:
签名算法名称：SHA1withECDSA
版本：3

```

图 4 获取 SHA1 值

证书指纹里的SHA1值为在申请应用AK的时候需要填写的。

点击确定后即可得到一个Android SDK的应用AK

8537342 MapSensor Android端 设置 删除

图 5 成功申请应用 AK

2. 下载百度地图SDK的依赖库<BaiduLBS\_AndroidSDK\_Lib.zip>，按照百度地图SDK配置环境与发布<sup>2</sup>把依赖库放到相应的文件夹并添加到工程里，并按照HelloBaiduMap<sup>3</sup>中的步骤，在AndroidManifest.xml文件里添加权限以及应用AK。

<sup>2</sup>配置环境及发布 Android SDK：<http://lbsyun.baidu.com/index.php?title=androidsdk/guide/buildproject>

<sup>3</sup> Hello BaiduMap：<http://lbsyun.baidu.com/index.php?title=androidsdk/guide/hellobaidumap>

**注意：**

这次实验如果使用 API 23 以上系统运行，需要完善动态申请权限的步骤，不能手动在应用信息里给权限来实现。

3. 特别的，由于Android API 23以上需要动态申请权限，这里使用一个Github上的开源工具RxAndroid<sup>4</sup>进行处理，也可以自己根据Android官方的API进行或者其他的工具。首先需要在build.gradle中加入需要的依赖项：

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.4.0'
    compile 'com.tbruyelle.rxpermissions:rxpermissions:0.9.0@aar'
    compile 'io.reactivex:rxandroid:1.2.1'
    compile 'io.reactivex:rxjava:1.1.6'
}
```

然后新建一个启动的Activity，在这个单独的Activity的onCreate方法里检查是否获取需要的权限，如果没有则直接退出程序，实现代码参考如下：

```
RxPermissions rxPermissions = new RxPermissions(this);
rxPermissions
    .request(Manifest.permission.ACCESS_FINE_LOCATION,
            Manifest.permission.WRITE_EXTERNAL_STORAGE)
    .subscribe((Action1) (granted) -> {
        if (granted) { // Always true pre-M
            Toast.makeText(CheckPermissionActivity.this, "Granted", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(CheckPermissionActivity.this, MainActivity.class);
            startActivity(intent);
            finish();
        } else {
            Toast.makeText(CheckPermissionActivity.this, "App will finish in 3 secs...", Toast.LENGTH_SHORT).show();

            mHandler.postDelayed(() -> {
                finish();
            }, 3000);
        }
    });
```

4. 手机作为多媒体平台，其中传感器发挥了非常重要的作用，调用手机大多数传感器首先需要获取传感器的管理器，传感器都需要由这个传感器管理器进行管理，包括注册传感器注销传感器等操作。

```
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
```

5. 手机的方向信息是通过地磁传感器和加速度传感器共同计算出来的，那么要获取地磁传感器和加速度传感器的信息：

```
mMagneticSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
mAccelerometerSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```

<sup>4</sup> <https://github.com/ReactiveX/RxAndroid>



6. 在Activity在前台运行（`onResume`）的时候注册传感器，在离开前台（`onPause`）的时候取消注册。同时在注册传感器的时候，需要指定传感器的监听器，当传感器数据更新的时候，系统会回调监听器里的`onSensorChange`函数，我们便可以在这里对传感器数据进行相应处理。

```
// register magnetic and accelerometer sensor into sensor manager (onResume
mSensorManager.registerListener(mSensorEventListener, mMagneticSensor,
    SensorManager.SENSOR_DELAY_GAME);
mSensorManager.registerListener(mSensorEventListener, mAccelerometerSensor,
    SensorManager.SENSOR_DELAY_GAME);

// unregister sensors (onPause
mSensorManager.unregisterListener(mSensorEventListener);

// sensor event listener
private SensorEventListener mSensorEventListener = new SensorEventListener() {

    float[] accValues = null;
    float[] magValues = null;
    long lastShakeTime = 0;

    @Override
    public void onSensorChanged(SensorEvent event) {
        switch (event.sensor.getType()) {
            case Sensor.TYPE_ACCELEROMETER:
                // do something about values of accelerometer
                break;
            case Sensor.TYPE_MAGNETIC_FIELD:
                // do something about values of magnetic field
                break;
            default:
                break;
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}
};
```

7. 手机的朝向是通过地磁传感器和加速度传感器共同计算出来了，Android 提供了计算这个数值的API，首先通过`getRotationMatrix`得到一个旋转矩阵，然后使用`getOrientation`得到手机的朝向：

```
float[] R = new float[9];
float[] values = new float[3];

SensorManager.getRotationMatrix(R, null, accValues, magValues);
SensorManager.getOrientation(R, values);

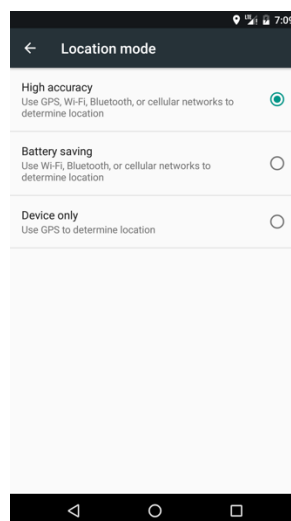
float newRotationDegree = (float) Math.toDegrees(values[0]);
```

- 同时，使用加速度传感器可以实现检测手机摇一摇的功能，实现有很多种方法可以自己思考一下。
- 获取用户的经纬度信息需要使用位置管理器`LocationManager`，同时需要确认手机打开了位置服务。

```
mLocationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);
```

- 位置服务有一个位置的数据来源称为`provider`，可以分为`NETWORK_PROVIDER`和`GPS_PROVIDER`，其中`NETWORK_PROVIDER`使用基站和Wi-Fi信号来决定用户的位置，更新速度更快耗电量更少，但可能精确度较低，而`GPS_PROVIDER`使用GPS卫星进行定位，精度高但只能在室外使用，耗电量也更高。在使用过程中可以选择使用其中一个`provider`，也可以两个`provider`同时使用。

在Setting->Location->Mode里，有如下三个选项，High accuracy表示`GPS_PROVIDER` 以及`NETWORK_PROVIDER` 同时启用，而Battery saving只启用了`NETWORK_PROVIDER`， Device only只启用了`GPS_PROVIDER`。



因此，在判断使用哪个`Provider`的时候，首先使用`isProviderEnabled`函数判断该`Provider`是否已经启用，从而选取一个合适的`Provider`。

- 与传感器数据获取方式类似，可以注册`LocationListener`监听位置信息的更新并作出相应的处理，特别的，使用`getLastKnownLocation`可以立刻获得一个位置坐标，如果之前有进行过定位操作，那么这个位置就是之前获取到的位置缓存，如果没有获得或位置，那么返回一个空对象。



```
// get location immediately
Location location = mLocationManager.getLastKnownLocation(provider);

// register location update listener
mLocationManager.removeUpdates(mLocationListener);

// unregister update listener
mLocationManager.requestLocationUpdates(provider, 0, 0, mLocationListener);

private LocationListener mLocationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        // processing new location
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};
```

另外，调用getLastKnownLocation(**provider**) 获取定位缓存的时候，可以参考如下的方式选择较好的provider（不过第一次运行程序得到的结果都为空）

```
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE);
criteria.setAltitudeRequired(false);
criteria.setBearingRequired(false);
criteria.setCostAllowed(true);

String provider = mLocationManager.getBestProvider(criteria, true);
mLocationManager.getLastKnownLocation(provider);

mCurrentLocation = mLocationManager.getLastKnownLocation(provider);
```

实现LocationListener时可以参考博客<http://blog.csdn.net/limb99/article/details/8765584>中的策略，确定使用GPS还是Network。

12. 到此，我们已经实现了获取用户位置以及获取用户方向这两部分的功能，接下来主要是地图相关功能的实现。
13. 首先在布局文件里添加百度地图：

```
<com.baidu.mapapi.map.MapView
    android:id="@+id/bmapview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:clickable="true" />
```

14. 在Activity中初始化百度地图，使用百度地图里MyLocationConfiguration 添加当前所在位置的箭头，并根据用户方向以及位置构造MyLocationData，相应设置箭头的方向与位置。

```

Bitmap bitmap = Bitmap.createScaledBitmap(BitmapFactory.decodeResource(getResources(),
    R.mipmap.pointer), 100, 100, true);
BitmapDescriptor bitmapD = BitmapDescriptorFactory
    .fromBitmap(bitmap);

mBMapView.getMap().setMyLocationEnabled(true);
MyLocationConfiguration config = new MyLocationConfiguration(
    MyLocationConfiguration.LocationMode.NORMAL, true, bitmapD);
mBMapView.getMap().setMyLocationConfiguration(config);

MyLocationData.Builder data = new MyLocationData.Builder();
data.latitude(desLatLng.latitude);
data.longitude(desLatLng.longitude);
data.direction(mSensorUtils.getCurrentRotation());

mBMapView.getMap().setMyLocationData(data.build());

```

15. 要注意的是，LocationManager返回的Location中的坐标系与百度地图的坐标系并不一样，需要进行坐标系的转换，可以参考百度地图SDK文档计算工具部分<sup>5</sup>，转换代码参考如下，最后LatLng得到的经纬度为百度地图坐标系下的经纬度。

```

mConverter.from(CoordinateConverter.CoordType.GPS);
mConverter.coord(new LatLng(location.getLatitude(), location.getLongitude()));
LatLng desLatLng = mConverter.convert();




```

16. 在地图中使某个坐标居中是使用MapStatus 实现，具体代码参考如下：

```

MapStatus mMapStatus = new MapStatus.Builder().target(desLatLng).build();
MapStatusUpdate mMapStatusUpdate = MapStatusUpdateFactory.newMapStatus(mMapStatus);
mBMapView.getMap().setMapStatus(mMapStatusUpdate);

```

17. 关于左下角的按钮，本身是一个开关ToggleButton，然后自定义开关的背景为如下xml，就可以实现选中是而再点击一下取消选中是。

```


<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@mipmap/center_on" android:state_checked="true" />
    <item android:drawable="@mipmap/center_off" />
</selector>

```

<sup>5</sup> 计算工具：<http://lbsyun.baidu.com/index.php?title=androidsdk/guide/tool>

18. 布局文件中ToggleButton参考布局如下:

```
<ToggleButton
    android:id="@+id/tb_center"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="20dp"
    android:layout_marginStart="20dp"
    android:background="@drawable/toggle_center"
    android:checked="true"
    android:textOff=""
    android:textOn="" />
```

19. 手动拖动地图平移之后按钮变为的实现方法是实现BaiduMap中的setOnMapTouchListener 函数, 参考代码如下:

```
mBMapView.getMap().setOnMapTouchListener(new BaiduMap.OnMapTouchListener() {
    @Override
    public void onTouch(MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_MOVE:
                mToggleButton.setChecked(false);
                break;
            default:
                // TODO Process other motion
                break;
        }
    }
});
```

20. Android传感器还有更多有趣的应用还等大家以后慢慢发现。

## 【参考资料】

1. 参考项目文件列表如下:

```
.
├── app.iml
├── build.gradle
├── libs
│   ├── BaiduLBS_Android.jar
│   └── arm64-v8a
│       ├── libBaiduMapSDK_base_v4_0_0.so
│       ├── libBaiduMapSDK_cloud_v4_0_0.so
│       ├── libBaiduMapSDK_map_v4_0_0.so
│       ├── libBaiduMapSDK_search_v4_0_0.so
│       ├── libBaiduMapSDK_util_v4_0_0.so
│       └── liblocSDK7.so
```

```
├── armeabi
│   ├── libBaiduMapSDK_base_v4_0_0.so
│   ├── libBaiduMapSDK_cloud_v4_0_0.so
│   ├── libBaiduMapSDK_map_v4_0_0.so
│   ├── libBaiduMapSDK_search_v4_0_0.so
│   ├── libBaiduMapSDK_util_v4_0_0.so
│   └── liblocSDK7.so
├── armeabi-v7a
│   ├── libBaiduMapSDK_base_v4_0_0.so
│   ├── libBaiduMapSDK_cloud_v4_0_0.so
│   ├── libBaiduMapSDK_map_v4_0_0.so
│   ├── libBaiduMapSDK_search_v4_0_0.so
│   ├── libBaiduMapSDK_util_v4_0_0.so
│   └── liblocSDK7.so
├── x86
│   ├── libBaiduMapSDK_base_v4_0_0.so
│   ├── libBaiduMapSDK_cloud_v4_0_0.so
│   ├── libBaiduMapSDK_map_v4_0_0.so
│   ├── libBaiduMapSDK_search_v4_0_0.so
│   ├── libBaiduMapSDK_util_v4_0_0.so
│   └── liblocSDK7.so
├── x86_64
│   ├── libBaiduMapSDK_base_v4_0_0.so
│   ├── libBaiduMapSDK_cloud_v4_0_0.so
│   ├── libBaiduMapSDK_map_v4_0_0.so
│   ├── libBaiduMapSDK_search_v4_0_0.so
│   ├── libBaiduMapSDK_util_v4_0_0.so
│   └── liblocSDK7.so
├── proguard-rules.pro
└── src
    ├── androidTest
    ├── main
    │   ├── AndroidManifest.xml
    │   ├── java
    │   │   ├── sysu
    │   │   │   ├── mobile
    │   │   │   │   ├── mapsensor
    │   │   │   │   │   ├── BaseApplication.java
    │   │   │   │   │   ├── CheckPermissionActivity.java
    │   │   │   │   │   ├── MainActivity.java
    │   │   │   │   │   └── SensorUtils.java
    │   └── res
    │       ├── drawable
    │       │   └── toggle_center.xml
    │       ├── layout
    │       │   ├── activity_check_permission.xml
    │       │   └── activity_main.xml
    │       ├── mipmap
    │       │   └── pointer.png
    │       ├── mipmap-mdpi
    │       │   ├── center_off.png
    │       │   ├── center_on.png
    │       │   └── ic_launcher.png
    │       └── values
    │           ├── colors.xml
    │           ├── dims.xml
    │           ├── strings.xml
    │           └── styles.xml
    └── test
```

## 2. 相关文档资料

- a) Android官方开发文档-Sensors Overview  
[https://developer.android.com/guide/topics/sensors/sensors\\_overview.html](https://developer.android.com/guide/topics/sensors/sensors_overview.html)
- b) Android官方开发文档-Location Strategies  
<https://developer.android.com/guide/topics/location/strategies.html>
- c) 百度地图ADK开发文档  
<http://lbsyun.baidu.com/index.php?title=androidsdk/guide/introduction>
- d) Android官方培训课程中文版-显示位置地址  
<http://hukai.me/android-training-course-in-chinese/location/index.html>

### 【提交说明】

- 1、deadline: 下一次实验课前一天晚上 12 点
- 2、提交 ftp: <ftp://222.200.185.18:1890/>, 作业提交文件夹中对应的文件夹下
- 3、命名与目录结构要求:

附件命名及格式要求: 学号\_姓名\_labX.zip (姓名中文拼音均可)

重复提交命名格式要求: 学号\_姓名\_labX\_Vn.zip

目录结构:

```
14331111_huashen_lab1 --  
|  
-- lab1实验报告.pdf  
|  
-- lab1_code (包含项目代码文件)
```

其中项目代码文件为项目文件夹, 提交之前先 clean