中山大学移动信息工程学院本科生实验报告

（2017 年秋季学期）

课程名称：移动应用开发　　任课教师：

年级	15	专业（方向）	移动互联网
学号	15352155	姓名	赖贤城
电话	13727024851	Email	754578682@qq.com
开始日期	2017/11/4	完成日期	2017/11/4

一、 实验题目

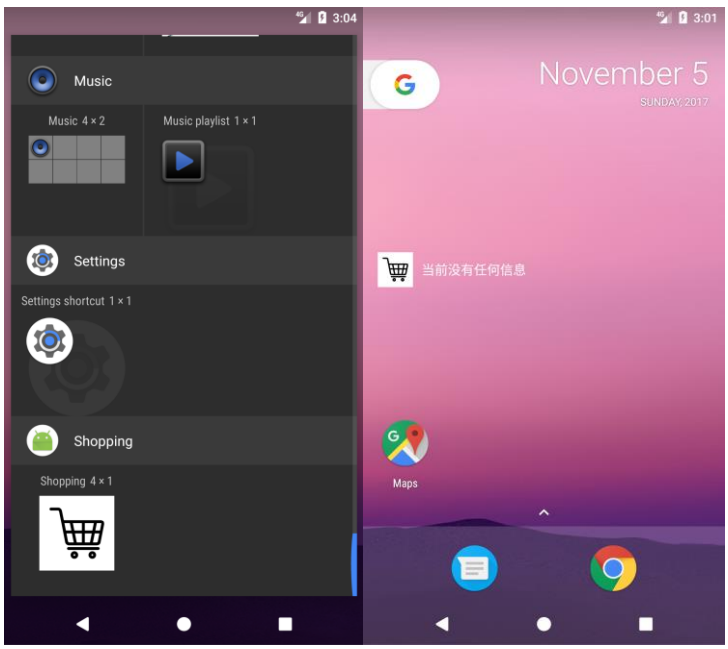
appwidget 及 broadcast 使用

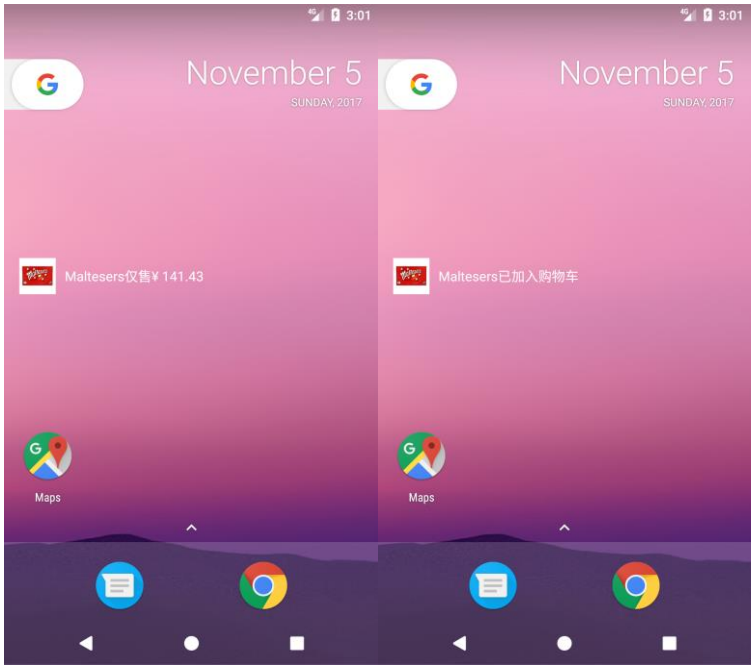
二、 实现内容

实现一个 Android 应用，实现静态广播、动态广播两种改变 widget 内容的方法。在上次实验的基础上进行修改，所以一些关于静态动态广播的内容会简略

三、 课堂实验结果

（1） 实验截图





（2） 实验步骤以及关键代码

1. 创建与注册 widget，定义 widget 布局

注册 widget

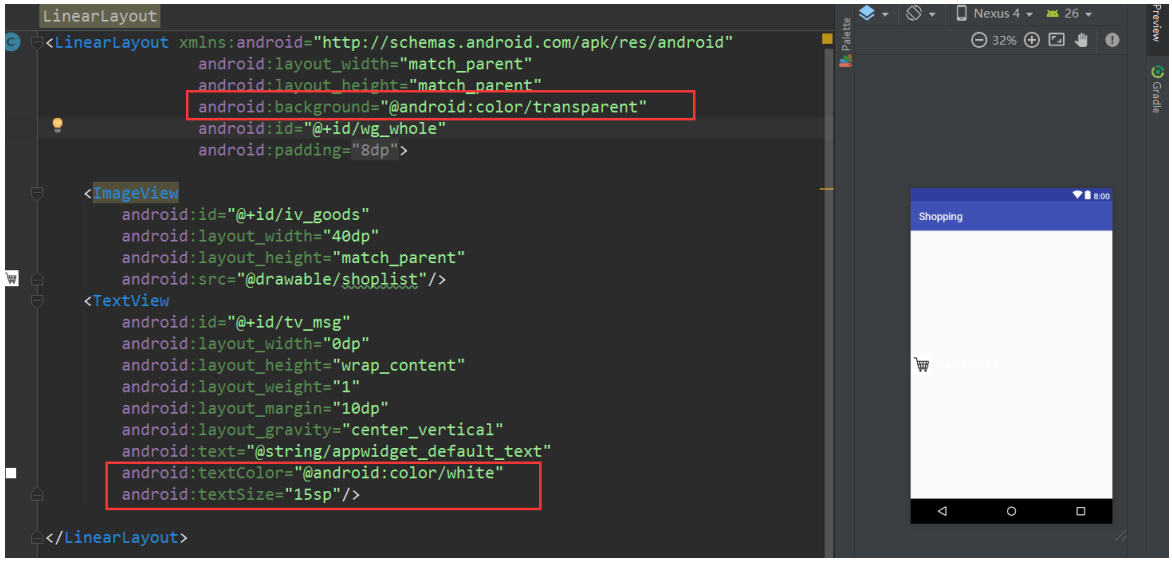
```
<receiver android:name=".widget.ShoppingWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE"/>
        <!-- action android:name="com.lxc.my.WIDGET_ADDITION"/ -->
        <action android:name="com.lxc.my.LAUNCH_WIDGET"/>
        <action android:name="com.lxc.action.widgetInitClick"/>
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/shopping_widget_info"/>
</receiver>
```

定义 widget 的宽高，预览图，及放置位置等属性

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
                    android:initialKeyguardLayout="@layout/shopping_widget"
                    android:initialLayout="@layout/shopping_widget"
                    android:minHeight="60dp"
                    android:minWidth="250dp"
                    android:previewImage="@drawable/shoplist_widget_preview"
                    android:resizeMode="horizontal|vertical"
                    android:updatePeriodMillis="86400000"
                    android:widgetCategory="home_screen">
</appwidget-provider>
```

定义 widget 的初始布局，其中背景是透明的，字体颜色，字号按照文档规定



2. 继承 AppWidgetProvider，重写 onUpdate

由于组件第一次被创建的时候就会回调这个方法，因此这里把界面更新为初始状态

```
@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
    super.onUpdate(context, appWidgetManager, appWidgetIds);
    for (int appWidgetId : appWidgetIds) {
        RemoteViews remoteViews = new RemoteViews(context.getPackageName(), R.layout.shopping_widget);
        updateRemoteView(context,  remoteViews, INIT_TYPE,  item: null, appWidgetId, appWidgetManager);
    }
}
```

其中的 updateRemoteView 是一个统一的更新 view 以及其对应的点击事件的函数，上面的代码传入的是 INIT_TYPE（初始状态），因此这时候设置 remoteView 的点击事件是打开 GoodActivity，也就是商品页

```
/**
 * 更新view
 */
public void updateRemoteView(Context context, RemoteViews remoteViews, int type,
                             GoodsItemBean item,
                             int appWidgetId, AppWidgetManager appWidgetManager) {
    Intent clickIntent = new Intent();
    if (type == INIT_TYPE){//初始状态
        clickIntent.setClass(context, GoodsActivity.class);
    }
    else if (type == RECOMMEND_TYPE){//推荐商品状态
        clickIntent.putExtra(GoodsActivity.JUMP_DETAIL_INFO, item);
        clickIntent.setClass(context, DetailActivity.class);
    }
    PendingIntent pendingIntent = PendingIntent.getActivity(context,  requestCode: 0, clickIntent, PendingIntent
    remoteViews.setOnClickPendingIntent(R.id.wg_whole, pendingIntent);
    appWidgetManager.updateAppWidget(appWidgetId, remoteViews);
}
```

3. 进入主界面用发广播更新 widget 为“推荐商品”

a. 主界面发送广播

在 onCreate 调用的时候发送广播，使用随机数获得一个索引，将该索引对应的商品对象包装在 intent 里面传递给广播。（注：下图函数在 onCreate 里面调用）

```
private void sendHotBroadcast() {
    Intent intent = new Intent(LAUNCH_WIDGET_ACTION);
    Random random = new Random();
    int randIndex = random.nextInt(goodsItems.size());
    intent.putExtra(LAUNCH_WIDGET_BROADCAST_INFO, goodsItems.get(randIndex));
    sendBroadcast(intent,  receiverPermission: null);
}
```

b. *AppWidgetProvider 中重写 onReceive，写入收到广播的逻辑*

接收到首页发来的广播的时候，对 remoteView 设置“推荐界面”对应的图片和文字，以及点击跳转事件（updateRemoteView 请看上述第二步）

```
ShoppingWidget  onReceive()
    @Override
    public void onReceive(Context context, Intent intent) {
        super.onReceive(context, intent);
        AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
        appWidgetIds = appWidgetManager.getAppWidgetIds(new ComponentName(context.getPackageName(),
                cls: "com.lxc.shopping.widget.ShoppingWidget"));
        RemoteViews remoteViews = new RemoteViews(context.getPackageName(), R.layout.shopping_widget);
        //添加到购物车
        if (intent.getAction().equals(DetailActivity.BROADCAST_WIDGET_ACTION)){
            GoodsItemBean itemBean = (GoodsItemBean) intent.getSerializableExtra(DetailActivity.ADDITION_WIDGET_B
            remoteViews.setImageViewResource(R.id.iv_goods, itemBean.imageRes);
            remoteViews.setTextViewText(R.id.tv_msg,  text: itemBean.name + "已加入购物车");
            for (int appWidgetId : appWidgetIds) {
                updateRemoteView(context,  remoteViews, ADD_TYPE, itemBean, appWidgetId, appWidgetManager);
            }
        }
        //推荐商品
        else if (intent.getAction().equals(GoodsActivity.LAUNCH_WIDGET_ACTION)){
            GoodsItemBean itemBean = (GoodsItemBean) intent.getSerializableExtra(GoodsActivity.LAUNCH_WIDGET_BROA
            remoteViews.setImageViewResource(R.id.iv_goods, itemBean.imageRes);
            remoteViews.setTextViewText(R.id.tv_msg,  text: itemBean.name + "仅售" + itemBean.price);
            for (int appWidgetId : appWidgetIds) {
                updateRemoteView(context,  remoteViews, RECOMMEND_TYPE, itemBean, appWidgetId, appWidgetManager);
            }
        }
```

c. *manifest 文件中静态注册广播*

为上一步的 receiver 注册，过滤器能接受的 action 就是 mainactivity 中发的广播的 action

```
<receiver android:name=".widget.ShoppingWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE"/>
        <!-- action android:name="com.lxc.my.WIDGET_ADDITION"/ -->
        <action android:name="com.lxc.my.LAUNCH_WIDGET"/>
        <action android:name="com.lxc.action.widgetInitClick"/>
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/shopping_widget_info"/>
</receiver>
```

4. 点击购物车发布广播，更新 widget 为“已加入购物车”

a. *点击购物车发送广播*

```
/**
 * 物品加入购物车
 */
private void addToShopList(){

    AddToShopListEvent event = new AddToShopListEvent(goodsItem);
    EventBus.getDefault().postSticky(event);

    Intent intent = new Intent(BROADCAST_WIDGET_ACTION);
    intent.putExtra(ADDITION_WIDGET_BROADCAST_INFO_ITEM, goodsItem);
    sendBroadcast(intent, receiverPermission: null);
```

b. 重写 BroadcastReceiver，收到“加入购物车”广播的时候通过 AppWidgetManager 更新 remoteView

AppWidgetManager 通过写入 widget 对应的 AppWidgetProvider 类名全称获得对应的 widget 的 id，然后根据取得的 item 设置图片文字以及点击跳转 activity 事件

```
@Override
public void onReceive(Context context, Intent intent) {
    AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(context);
    int [] appWidgetIds = appWidgetManager.getAppWidgetIds(new ComponentName(context.getPackageName(),
            cls: "com.lxc.shopping.widget.ShoppingWidget"));
    RemoteViews remoteViews = new RemoteViews(context.getPackageName(), R.layout.shopping_widget);
    //添加到购物车
    if (intent.getAction().equals(DetailActivity.BROADCAST_WIDGET_ACTION)){
        GoodsItemBean itemBean = (GoodsItemBean) intent.getSerializableExtra(DetailActivity.ADDITION_WIDGET_BROADC
        remoteViews.setImageViewResource(R.id.iv_goods, itemBean.imageRes);
        remoteViews.setTextViewText(R.id.tv_msg, text: itemBean.name + "已加入购物车");
        //设置点击跳转事件
        Intent clickIntent = new Intent();
        clickIntent.putExtra(DetailActivity.JUMP_GOODS_INFO, value: true);
        clickIntent.setClass(context, GoodsActivity.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, requestCode: 0, clickIntent, PendingIntent
        remoteViews.setOnClickPendingIntent(R.id.wg_whole, pendingIntent);

        appWidgetManager.updateAppWidget(appWidgetIds, remoteViews);
    }
```

c. 动态注册广播接收器

“加入购物车的广播接收器”的注册与取消

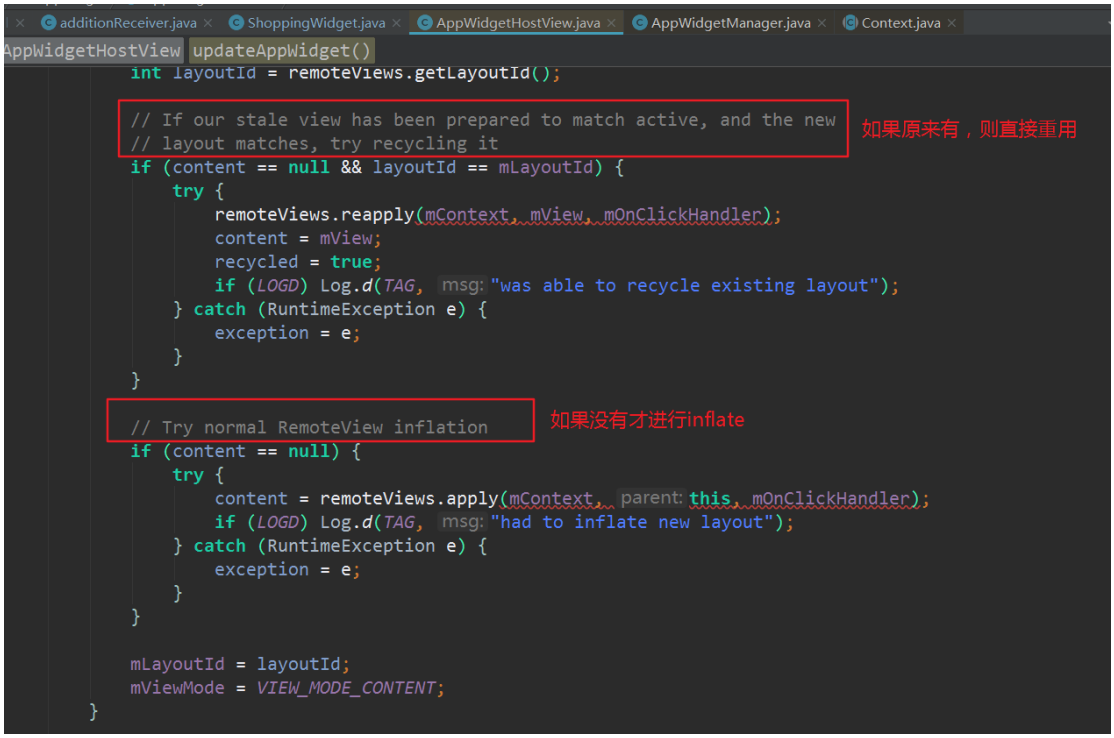
```
IntentFilter widgetIntentFilter = new IntentFilter(BROADCAST_WIDGET_ACTION);
widgetReceiver = new additionWidgetReceiver();
registerReceiver(widgetReceiver, widgetIntentFilter);

@Override
protected void onDestroy() {
    super.onDestroy();
    unregisterReceiver(receiver);
    unregisterReceiver(widgetReceiver);
}
```

（3） 实验遇到困难以及解决思路

1. 将 widget 从“已加入购物车”状态变成“没有新信息”状态的时候，需要用代码 setText 和 setImage 才能更新

解决：我以为 updateAppWidget 函数是重新设置了一遍 remoteView，因此觉得不 set 图片文字的话默认就是用的布局文件中定义好的图片文字。 但是**查看了源码**才知道原来如果 layoutId 与之前的相同的话，并不会重新 inflate，而只是执行 setText 等动作，所以就不会像我之前想的那样，以为不 set 就是布局文件中设置好的图片文字

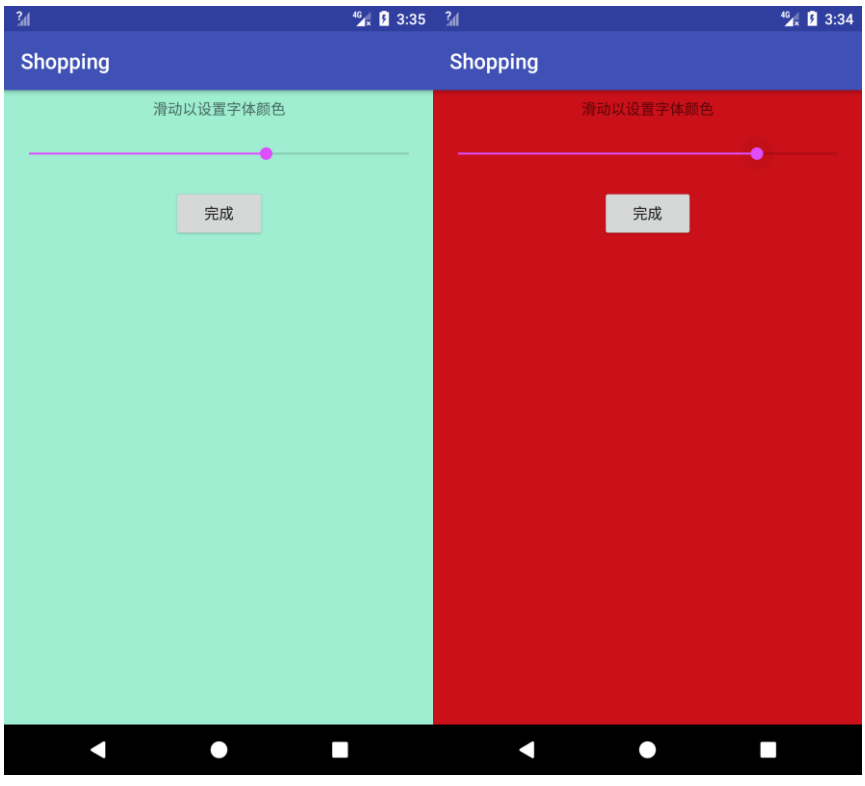


所以，要将 widget 的界面恢复到初始模样的话还是应该通过 set 方法才能实现

四、 课后实验结果

给 widget 提供初始的配置

1. 结果展示



2. 实验过程

1. 配置文件加上配置的 Activity

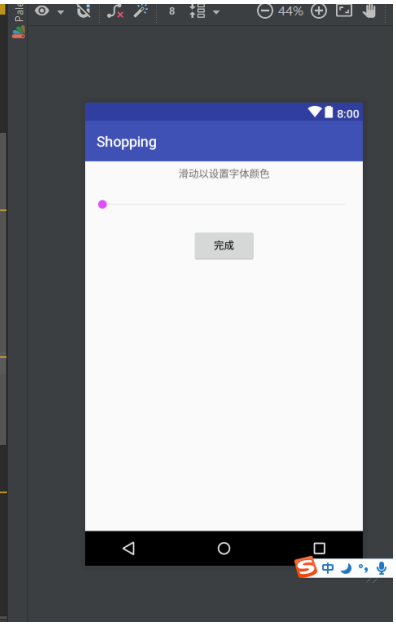
```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
                    android:initialKeyguardLayout="@layout/shopping_widget"
                    android:initialLayout="@layout/shopping_widget"
                    android:minHeight="60dp"
                    android:minWidth="250dp"
                    android:previewImage="@drawable/shoplist_widget_preview"
                    android:resizeMode="horizontal|vertical"
                    android:updatePeriodMillis="86400000"
                    android:widgetCategory="home_screen"
                    android:configure="com.lxc.shopping.ConfigurationActivity">
</appwidget-provider>
```

2. 配置的 Activity 的布局文件，主要是滑动条，用于选择颜色

```
<SeekBar
    android:id="@+id/sb_color"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="24dp"
    android:min="0"
    android:max="16777215"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"/>

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="滑动以设置字体颜色"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>

<Button
    android:id="@+id/bt_done"
    android:layout_width="wrap_content"
```



3. 用户通过滑动选择颜色

```
@Override
public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
    color = progress;
    color = color | 0xff000000;
}
```

4.点击完成的时候，更新 remoteView 的字体颜色，完成配置

```
@Override
public void onClick(View v) {
    if (mAppWidgetId != -1) {
        RemoteViews views = new RemoteViews(getPackageName(),
                R.layout.shopping_widget);
        Intent intent = new Intent( packageContext: this, GoodsActivity.class);
        PendingIntent pendingIntent = PendingIntent.getActivity( context: this,  requestCode: 0, intent, Pendin
        views.setOnClickPendingIntent(R.id.wg_whole, pendingIntent);
        views.setImageViewResource(R.id.iv_goods, R.drawable.shoplist);
        views.setTextViewText(R.id.tv_msg, getString(R.string.appwidget_default_text));
        views.setTextColor(R.id.tv_msg, color);
        appWidgetManager.updateAppWidget(mAppWidgetId, views);
        Intent result = new Intent();
        result.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, mAppWidgetId);
        setResult(RESULT_OK, result);
        finish();
    }
}
```

五、 实验思考及感想

1. 学新知识的时候，应该要从原理的层次去学，就像这次的 widget 的界面更新，只要理解了 updateAppWidget 函数它背后的原理，那么就可以不局限于实验文档的实现方式了，自己可以做很多灵活的处理
2. 当遇到问题的时候，应该尝试看看源码进行分析，一些无法相同的问题，一看代码几乎就能明白了，况且源码一般都有很多详细的注释

作业要求：

1. 命名要求: 学号_姓名_实验编号，例如 15330000_林 XX_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。