二维码接力计算设计文档

**一．问题：**

N个人围成一圈(网上)，如何计算平均数？

**二．分析**

直接说出自己的数字（工资，评分等）显然是不行的，写在纸上也不太好，那样需要一个人收每个人的纸条然后由他算出平均分，最好的方法就是借助于软件，通过扫描二维码传递当前数字之和，每个人都要加上两个数字，一个数字是自己的信息，另一个数字是用来防止其他人推测出你的得分，一圈人加完之后再轮流把自己用于混淆的数字减去，最后得到所有人的和，算出平均分。

**三．设计思路：**

首先由第一个人将自己的数字和随便的一个数字相加，然后生成二维码，后面的人扫描二维码，得到前面的人的数字之和，然后将自己的数字和随便一个数字加上，得到新的数字之和然后生成新的二维码，后面的人再重复上述操作，直到所有人都将自己的数字加上。然后再每个人扫描二维码，将自己原来随便选的那个数字减去，生成新的二维码，直到所有人都将自己的随机数减去，最后得到总分，除以参加游戏的人数即可得到平均数。

**四．实现**

(1)在Android Studio中新建项目。

# 我做梦也没想到,新建项目之后我什么也没做，项目自己在build竟然build会报错，报了这个错：**unable to find valid certification path to requested target**，从字面意思上看是缺少证书，我从网上查到要添加网站的证书，于是我添加上了，但还是不行，我又按照有的博客中说的，清缓存-失败，添加了各种东西-失败，最后，我终于在一篇博客中找到了解决方法，<https://blog.csdn.net/da_caoyuan/article/details/104343058>，这篇博客给我们提供了一些阿里云的镜像，从这些镜像中选一个添加到工程下的 build.gradle中的两个repositories中，然后再build就成功了。

1. 接下来，我们需要添加一些依赖，首先，这是一个基于二维码的应用，我们要使用一个二维码的库-zxing,我在网上找了好几个zxing库，都不太可靠，然后我在github上找到了一个可靠的zxing库，这个库已经非常完善了，而且有详细的中文教程-><https://github.com/yuzhiqiang1993/zxing>。

要想使用依赖，必须在build.gradle中的allprojects下的repositories中添加jitpack支持->语句如下：**maven { url "https://jitpack.io" }** ，添加上jitpack支持，我们还需要添加依赖，将如下语句添加到依赖中：

implementation 'com.github.yuzhiqiang1993:zxing:2.2.5'

之后将项目build一下，就会把资源下载下来，现在我们就可以使用zxing库了。

然后，既然需要扫描二维码，那就需要开启摄像头权限，摄像头权限属于敏感权限，在Android 6.0以后不仅需要在配置文件中声明还需要动态申请，在这里我们使用一个当下非常流行的库-permissionsdispatcther,利用这个库我们可以很方便的动态申请权限，由于我们已经添加了jitpack支持，现在我们只需要将permissionsdispatcher库添加到依赖中，语句如下:

implementation 'com.github.hotchemi:permissionsdispatcher:2.3.1'  
annotationProcessor 'com.github.hotchemi:permissionsdispatcher-processor:2.3.1'

现在我们再build一下就可以使用permissionsdispatcher了，使用permissionsdispatcher是通过添加注释的方式进行，其中有两个注释是必须使用的，一个是@RuntimePermissions，放在类的前面，另一个是@NeedPermission,用来指明需要申请那些权限。将这两个注释写上，然后在make一下，就可以使用permissionsdispatcher的所有内容了。

(3)接下来我们就开始编写代码了，首先我们要设计一下前端界面，界面比较简单，最上面留一个空白，用来放置生成的二维码，下面有两行，第一行是生成二维码的输入框和生成按钮，第二行是扫描二维码的按钮和用于存放扫描结果的TextView,最下面放一个TextView，用于说明游戏规则，如图一所示:

 

图一 图二

(4)然后我们开始编写后端逻辑，我们给生成二维码的按钮和扫描二维码的按钮分别添加事件监听器，生成二维码比较简单，直接按照github上zxing库的使用说明，获取输入框内容后，一行语句就可以生成二维码，生成之后我们将二维码放在上面的空白位置，然后清空输入框和放置结果的TextView，防止别人偷看（生成数字1248如图二所示）。在扫描二维码按钮的事件监听器中，我们首先要判断当前Android 版本，如果小于6.0的话就不用动态申请权限了，直接开始扫描，否则的话就需要动态申请权限，申请权限成功之后才可以扫描。扫描我们通过Intent进行页面跳转实现。我们还需要写一个onActivityResult方法来接收扫描二维码的结果，并将该结果放在合适的位置，这样基本功能就完成了。

1. 改进-加密

现在我们生成二维码都是直接根据用户输入的字符串生成二维码，这样即使用浏览器扫二维码也能得到结果，所以我想要是自己加密一下会好一点，加密，就是在得到用户输入的字符串之后，先将字符串加密，然后根据加密之后的字符串生成二维码，这样，用浏览器或者是别的软件扫描就得不到真实值了。有加密就有解密，解密，就是在得到二维码扫描的结果之后，将得到的结果解密为真实结果。

由于我也没系统的学习过加密的知识，这里就做一下简单加密，这个软件的字符串都是数字，其实可以根据一个函数加密，但是这样加密后的结果仍然是一个数字，不太好，而且这样的话就不能输入汉字了，就少了一些乐趣，所以这里我们不使用这种方法，我使用的方法是首先获取待加密字符串的字节数组，然后将数组中每个元素减11（减11是为了防止数字加密之后仍然是数字），然后利用new String(字节数组)方法返回一个新的字符串,这样就完成了加密。加密之后要根据加密之后的结果生成二维码。

在扫描二维码之后，得到了一个结果，这个结果不是真实的值，而是加密之后的值，所以我们要先解密，解密的过程和加密的过程是相反的，首先我们获取待解密字符串的字节数组，然后我们将数组中的每个元素都加11，然后利用new String(字节数组)返回解密后的值，这样我们就获取了解密后的值。  

图三 图四 图五

图三为生成的1248的二维码，为了防止别人偷看已将输入框清空

图四为用本软件扫描的结果，扫描出了正确结果

图五为用浏览器扫描二维码的结果，可以看出，扫描得到的结果是我们加密之后的结果，这样我们就实现了简单的加密，这个软件也就完成了。