**第十届蓝桥杯大赛**

**互联网＋团队赛**



XXX设计报告

编号:xxx

西南财经大学天府学院

2019年3月18日

**智能学报系统**

**参赛学院：**西南财经大学天府学院

**参赛团队**：“TF”团队

**参赛队长**：何杭

**参赛队员**：何杭、李秀娟，贾谨荣

**项目简介：学术报告一直是我们学生聊天中的热门话题，下面简称学报，如何报名成功自己想听的学报，如何解决因为急事而浪费的学报机会，如何解决因为没有抢到学报而不能听取学报，如何从学报中获取有价值的意义，以及怎么解决报名学报之间的公平性，学术报告经历从人工排队到网上预约，因此解决了学生在排队上面浪费的时间，但由于计算机系统的原因，出现了程序化代抢学报，毫秒级提供大量参数，反而违反了学术报告个的公平公正，致使想听学报的同学听不到学报，不想听学报的同学，为了完成任务而去听学报，所以在网上预约的基础上，智能学报系统利用大数据精准推送学生感兴趣的学术报告，学生随时随地可以更改报名状态，查看自己听取学报记录，提供学术论坛直播等等，从而让学术报告更加智能化**

**项目特色：**

* 创新:学术报告预约新方式
* 公平性: 学报报名成功名额通过概率随机选中
* 及时性:智能推送学术报告报名情况
* 专业性:针对不同的专业推送不同类型的学术报告
* 合理性:用户的权重更改只和上一次学术报告情况有关
* 及时性: 用户可以直接在线编辑自己的学术论文观后感

**关键词**：学术报告，公平公正，智能推送，在线操作

目 录

[一、项目介绍 3](#_Toc533060186)

[1.1 项目背景 3](#_Toc533060187)

[1.2项目可行性分析与应用前景 4](#_Toc533060188)

[1.3项目特色 4](#_Toc533060189)

[1.4项目采用技术及平台 5](#_Toc533060190)

[1.5 SWOT分析 5](#_Toc533060191)

[二、功能设计 6](#_Toc533060192)

[2.1 总体功能结构图 6](#_Toc533060193)

[2.2 功能介绍 7](#_Toc533060194)

[2.2.1用户端 7](#_Toc533060195)

[2.2.2护工端 8](#_Toc533060196)

[三、开发设计 9](#_Toc533060197)

[3.1 数据接口设计 9](#_Toc533060198)

[3.1.1 外部数据接口 9](#_Toc533060199)

[3.1.2 内部数据接口 9](#_Toc533060200)

[3.2 数据库设计 44](#_Toc533060201)

[3.3 界面设计 51](#_Toc533060202)

[3.3.1 用户端手机界面设计 51](#_Toc533060203)

[3.3.2 护工端手机界面设计 57](#_Toc533060204)

[3.3.3 管理员端web界面设计 62](#_Toc533060205)

[3.4 架构设计 65](#_Toc533060206)

[3.4.1 后台服务器端 65](#_Toc533060207)

[3.4.2 安卓移动端 66](#_Toc533060208)

[3.4.3 管理员web端 70](#_Toc533060209)

# 一、项目介绍

## 项目背景

学术报告一直是我们学生聊天中的热门话题，下面简称学报，如何报名成功自己想听的学报，如何解决因为急事而耽搁的学报资源，如何从学报中获取有价值的意义，以及怎么解决报名学报之间的公平性，在早期，学术报告厅秉承着先到先得的思想，让同学们顶着炎炎夏日从中午开始排队到晚上六点半，既浪费了大量的时间，又没有从根源上解决实际的问题，再到后来的网上某班进行预约听取学报，虽然免弃了同学们不需要从早到晚的排队，但是又出现了新的问题，网上预约的本质是通过时间段让学生们登陆指定地址进行报名学报，对于一些非计算机专业的同学来说，作弊方法是使用鼠标宏和键盘宏等物理外挂抢学报，但由于抢学报的时间短暂，一次性只能抢学报一次 造成危害不是太大。而对于计算机专业的学术来说 恶意的向服务器提交大量的POST请求，而这种程序性的，批量式的 在毫秒内向服务器提交大量的数据造成巨大影响，甚至于形成了一条代听代抢代写的商业链，极大损害了学校，学生的利益，所以天府智能学报推送系统应运而生，一种更公平，更安全，更有利的新型抢学报方式

## 1.2项目可行性分析与应用前景

**设计思路**：

首先针对于上面学术报告的存在的问题，我们提出来以下几点思路，一是采取网上预约报名的方式，方便学生进行报名管理，二是学生可以在学术报告开展的前一天可以更改自己的报名状态，方便学生们不会因为事情而浪费学术报告资源，三是智能推送学术报告类型，供大家进行选择报名，四是和校园APP tfinfo进行合作，能查看自己学报听取状态，五是网上提交学术报告观后感，展示优秀的学术报告观后感，六是通过加权随机算法，影响下一次报名的因素只与上一次学术报告有关，七是打击代听，代抢，代写等风气

**效益分析**：

对于学校来说，每期请的名师教授所消耗的人力财力物力没有真正用到学生身上，对于学生来说，花费高额的代抢方式，没有听到自己想听的学术论文，代抢代写学生从长远看不仅仅危害了自己的利益，还危害更多想听学报的利益，所以需要让学校花费的财力用到实处，学生从学术报告中获取真正有价值的东西，这才是学校的初衷

，

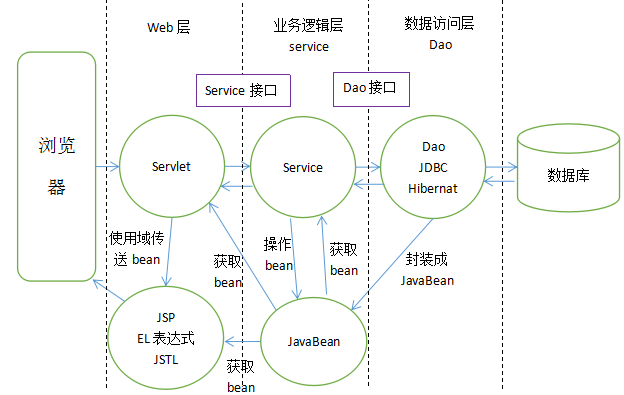
**实施计划及保障措施**：

1. 开发一个新的抢学报网站
2. 积极与学校进行互动，展示我们智能推送系统的价值
3. 从学生中积极寻求改进
4. 不断改进算法，促使算法利于大多数学生

## 1.3项目特色

* **创新**:网上预约，智能推送
* **及时性**:随时随地查看自己报名是否状态。
* **公平性**:通过加权随机算法，让学生都公平的听取学报
* **误差**:通过大数据拟合数据，保证数据稳定公平
* **纠正机制**:一旦学生连续三次报名都未成功将直接获得下轮报名成功机会

## 1.4项目采用技术及平台

* 后台数据库采用mysql数据库，服务器端和管理员web端以SSM+maven为架构，Spring+SpringMVC+MyBatis为开发模式，使用分层思想搭建。
* 服务器端开发结构图
* 
* 移动端安卓使用java在Android studio下开发，并用MVP(Modle+View

+Presenter)为开发模式，使用分层思想搭建。

针对此项目的Android版与IOS版APP，移动端安卓使用java在Android studio下开发，苹果端采用swift在xcode下开发。移动端界面设计图大致如下：

## 1.5 SWOT分析

|  |  |  |
| --- | --- | --- |
| 内部分析  外部分析 | 优势（S） | 劣势（W） |
| （1）智能推送学术报告类型  （2）学生支持。  （3）学校财力物力人力落到实处  （4）更优质的服务 | （1）技术上和x班相比技术不够  （2）学生已经习惯用x班抢学报  （3）网站设计简单。  （4）产品较为单一。 |
| 机会（O） | 策略（SO） | 策略（WO） |
| （1）学术报告不合理  （2）市场庞大，需求量大。  （3）学生心中迫切希望一种新的抢学术报告方式 | （1）提供网上咨询环节，针对学生进行宣传。  （2）与学校tfinfo进行合作，智能推送学报概况。 | （1）优化算法，在技术上创新  （2）改变学生习惯  （3）积极接受学生建议，提供更优质的服务 |
| 威胁（T） | 策略（ST） | 策略（WT） |
| （1）极易被模仿，可能会被后来者模仿甚至超越。  （ | 通过自己的服务态度以及服务质量给学生留下深刻印象 | 和学校tfinfo积极互动，让学生能一站式的获取最新学术报告推送 |

# 二、功能设计

## 2.1 总体功能结构图

****

* **会员中心：**主要包括了个人信息显示和修改，历史记录，查看系统公告；反馈及投诉。
* **咨询：**根据自己的需求，有两种不同的咨询方式，方便快捷地与客服人员联系。
* **查询：**提供两种搜索方式，根据自己想要的护理人员条件搜索；根据病人病情搜索出对应的护理方式。
* **订单：**在订单模块下有两种功能，我要护理（下单）和我的订单两个功能，下单页面中，每个用户的内容有所不同，主要包括地点的输入，可以从地图上进行选择，也可以自己输入；服务内容的描述；服务时间段的选择等。在我的订单，可以看到自己以前的订单，并且都可以根据类型进行显示，主要包括正在进行，已完成，已取消，全部订单，待评价和已评价的单。
* **交流支付：**客户与客户之间有一个讨论区，可以互相交流；在下单的过程中，需要提前将钱支付到第三方，在双方确认交易完成后方可选择付款。
* **消息推送：**消息推送功能，医疗生活小常识、所有下单成功、取消单成功，以及服务过程中需要服务方与被服务方进行相互评价的时候，都进行消息的推送。

## 2.2 功能介绍

### 2.2.1用户端

* **登录注册**
* **搜索护工**
* **查看相关护工信息**
* **向已选择护工发起订单**：为了引导护工准时守信的完成订单，每一步操作之后，双方都会收到目前订单状态的推送消息；
* **查看订单列表以及详细信息；**
* **支付功能**:在下单过程中，所花的费用需要从个人钱包进行支付，如果个人钱包余额不足，需要从（虚拟）银行卡中充值后进行支付；
* **评价护工**：用户对护工的评价可为之后用户选择护工提供参考；
* **向护工发起聊天**：为了双方更好的交流，避免需求不清楚的订单，【随护】提供即时聊天功能；
* **联系客服**：可能会出现双方对订单有不同的看法，该app提供申诉客服的功能，第一时间公平解决问题；
* **定位**：获取位置信息；
* **商城**：用户可通过商城购买生活用品和相关补品，展示商品信息、购物支付，也可为合格商家入驻提供条件。
* **消息推送**：所有下单成功，以及服务过程中需要下单方和护工接单方进行服务转态变化确认的时候，都进行消息的推送。

### 2.2.2护工端

* **登录**：通过认证核查的护工得到平台给的账号后进行登录；
* **接单**：护工将会收到即时的订单推送消息，以便第一时间获取到订单，完成服务；
* **查看订单列表以及详细信息；**
* **查看日程表**：避免日程冲突；
* **聊天**：为了双方更好的交流，避免需求不清楚的订单，【随护】提供即时聊天功能。及时通信模块
* **商城**：护工也可通过商城购买生活用品和相关补品，展示商品信息、购物支付，也可为合格商家入驻提供条件。

# 三、算法

## 3.1 算法描述

### 3.1.1 加权随机算法

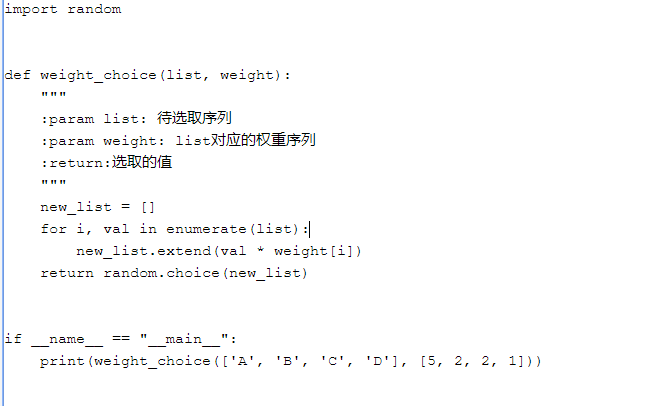
* **权重模型**

加权随机算法一般应用在以下场景：有一个集合S，里面比如有A,B,C,D这四项。这时我们想随机从中抽取一项，但是抽取的概率不同，比如我们希望抽到A的概率是50%,抽到B和C的概率是20%,D的概率是10%。一般来说，我们可以给各项附一个权重，抽取的概率正比于这个权重。那么上述集合就成了：

{A:5，B:2，C:2，D:1}

### 3.1.2 方法一

把序列按权重值扩展成:lists=[A,A,A,A,A,B,B,C,C,D],然后random.choice(lists)随机选一个就行。虽然这样选取的时间复杂度是O(1)，但是数据量一大，空间消耗就太大了。

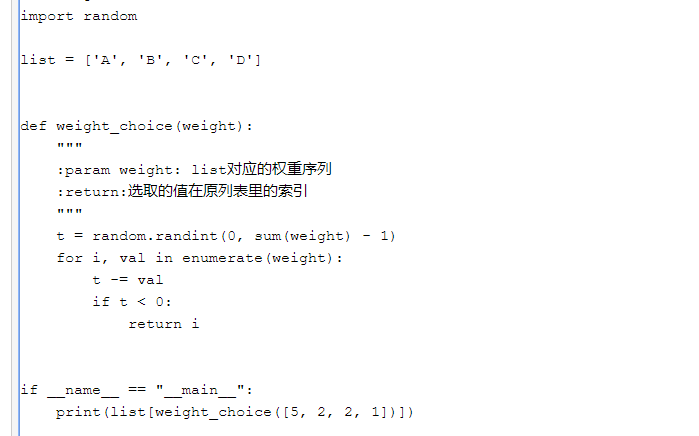


### 3.1.3 方法二

计算权重总和sum，然后在1到sum之间随机选择一个数R，之后遍历整个集合，统计遍历的项的权重之和，如果大于等于R，就停止遍历，选择遇到的项。

还是以上面的集合为例，sum等于10，如果随机到1-5，则会在遍历第一个数字的时候就退出遍历。符合所选取的概率。

选取的时候要遍历集合，它的时间复杂度是O（n）。



### 3.1.4 方法二

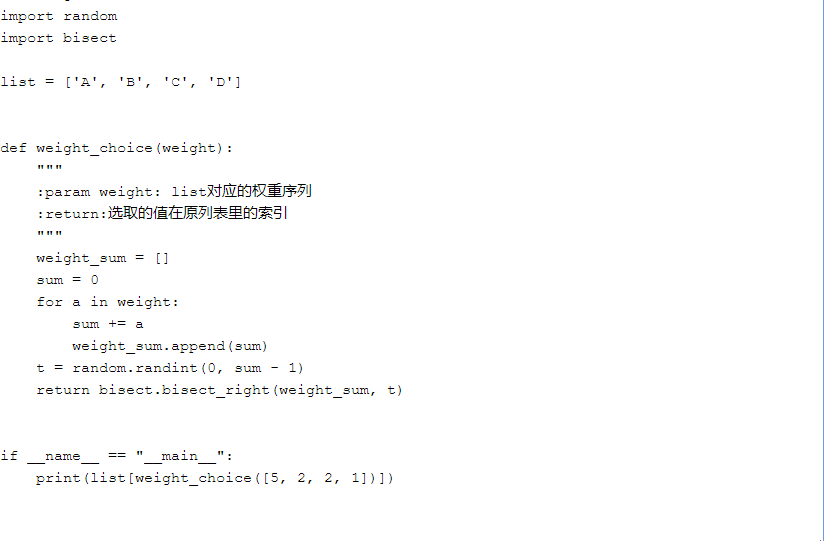
可以先对原始序列按照权重排序。这样遍历的时候，概率高的项可以很快遇到，减少遍历的项。（因为rnd递减的速度最快(先减去最大的数)）

比较{A:5，B:2，C:2，D:1}和{B:2，C:2，A:5，D:1}

前者遍历步数的期望是5/10\*1+2/10\*2+2/10\*3+1/10\*4=19/10而后者是2/10\*1+2/10\*2+5/10\*3+1/10\*4=25/10。

这样提高了平均选取速度，但是原序列排序也需要时间。

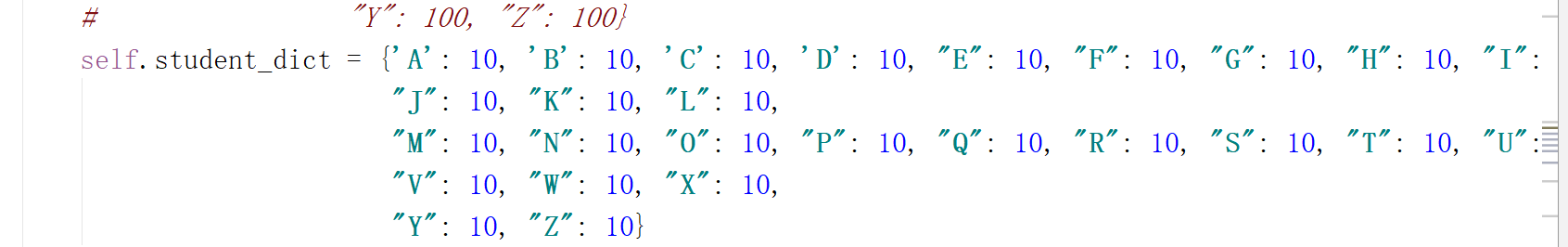
先搞一个权重值的前缀和序列，然后在生成一个随机数t后，可以用二分法来从这个前缀和序列里找，那么选取的时间复杂度就是O(logn)了。



## 3.2 算法解决

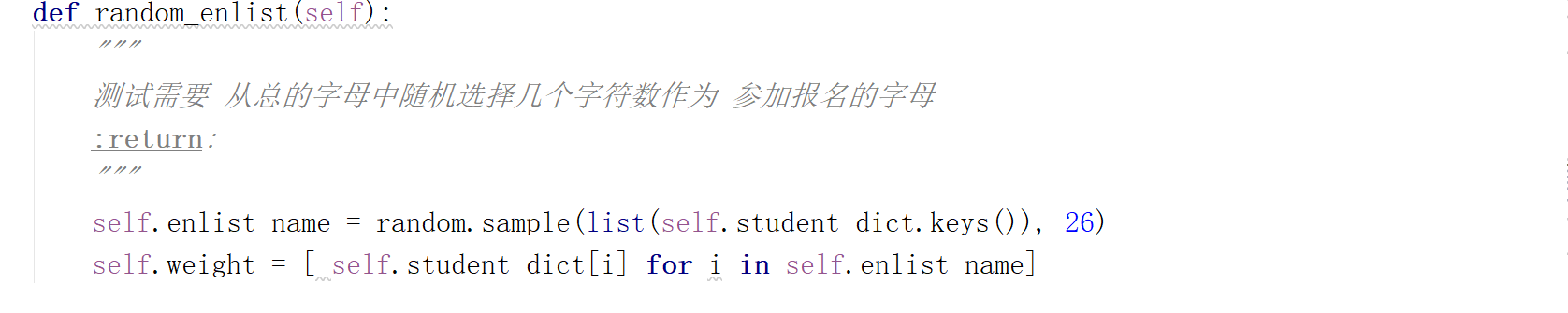
### 3.2.1 人数模拟

* **总人数模拟**



解释：我们用26个字母来进行模拟人数

* **报名人数模拟**



解释 ：我们从26个字母中随机选取字母作为我们报名的人数

* **重置学生权重函数**



解释：每个学期期末我们都将所有学生的权重全部初始化

* **权重异常**

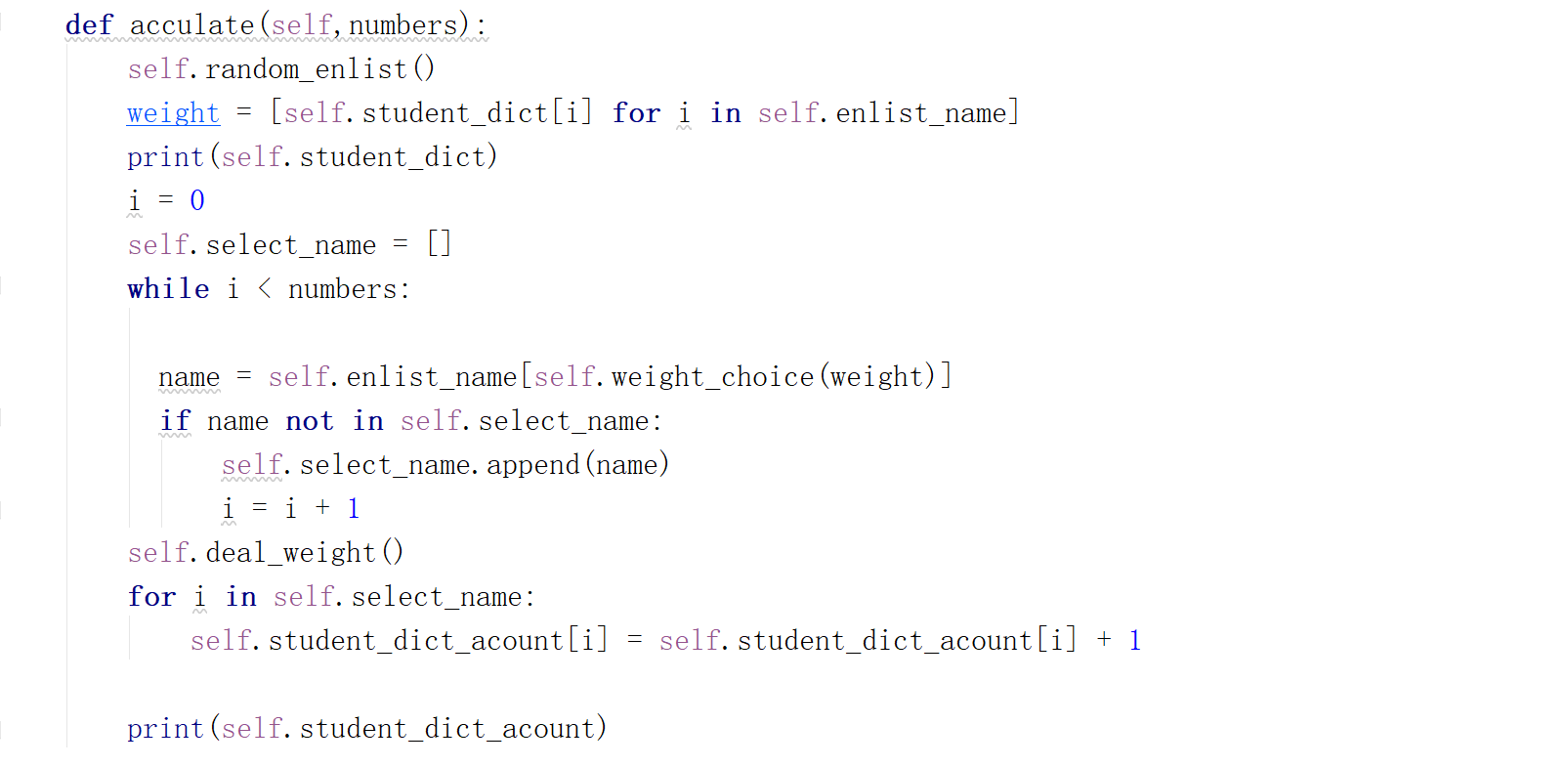


解释：当一个学生连续报名都没有报名成功时候。在第三次是将直接拥有报名成功的机会，同时权重初始化，权重降低，当一个学生连续报名且成功的时候，其权重将无限接近于0，将听取学报机会让给其他学生



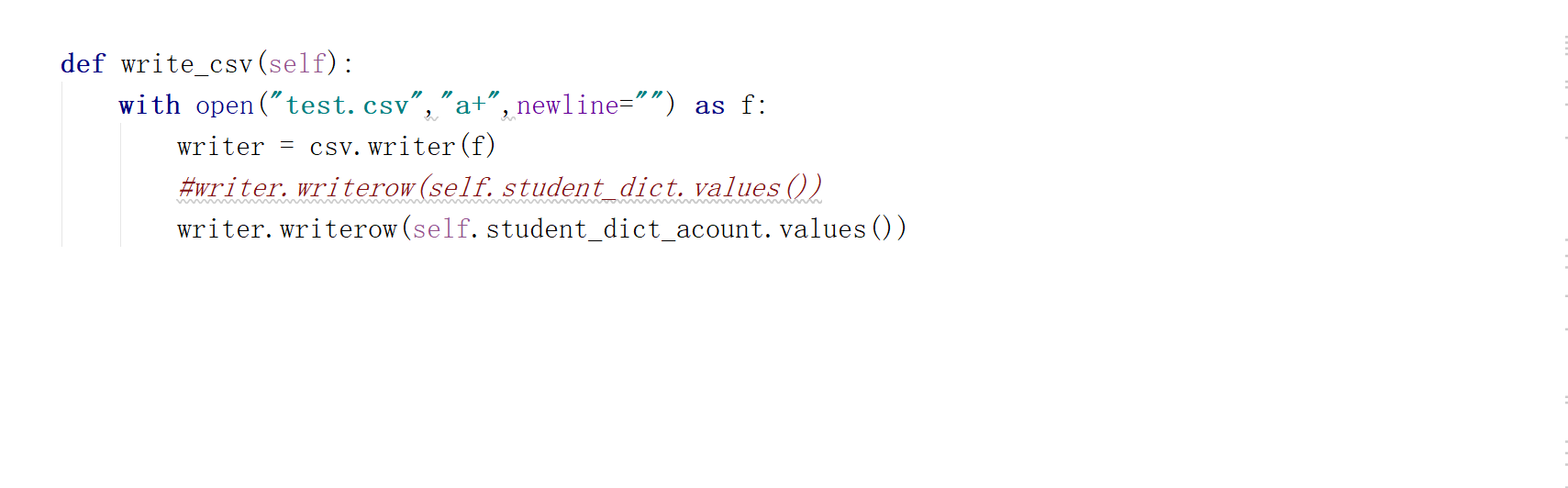


* **开始计算**



解释：返回学生的新权重值

* **写入后台**



解释最后将数据写入数据库，进行存储

### 3.2.1 权重系数

* **权重**

### 在数学上，为了显示若干量数在总量中所具有的重要程度，分别给予不同的比例系数，这就是加权，又称权重、权值。

### 在智能学术报告推送系统中，我们选择权重来计算学术报名且成功的概率大小，每个同学的权重初始值为1,权重的改变将直接影响下次听取学术报告的机会，

### 下面讲解权重变化模型

select\_student\_acount : 报名成功的人数。

enlist\_student \_acount : 报名的人数。

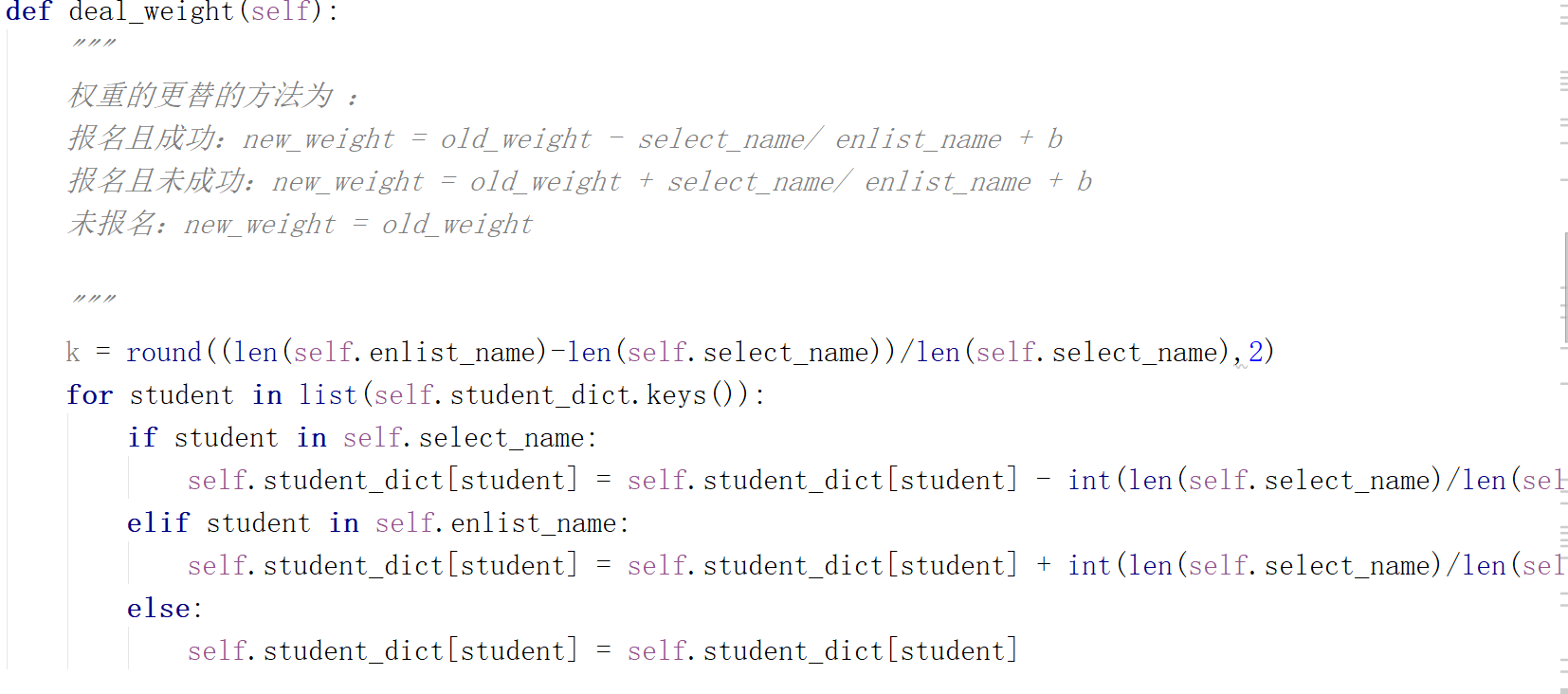
Init\_weights : 初始权重

new\_weights : 新权重值。

older\_weights :老权重值。

update\_weights : 权重变化值。

b : 误差纠正。



### update\_weights =

解释：权重变化值等于报名成功的学术人数除以报名学生的总人数

从上面公式中我们可以看出当enlist\_student \_acount的值越大update\_weights的值也就越小，当enlist\_student \_acount的值越小update\_weights的值也就相应的越大

在报名学报的过程中，报名人数越多，enlist\_student \_acount的值也就越大，对于权重的变化值也就越小，在这一点上，我们可以理解为，报名人数越多，权重变化值也就越小，下一次是否报名成功权重的影响就会降低，相对于报名未成功的，以及未报名成功的同学来说，新权重值变化不是特别影响下一次听学术报告的概率

而针对于报名人数越少时候enlist\_student \_acount的值也就越小，对于权重的变化值也就越大，我们映射到实际学术报告中，报名人数越少，权重的变化值也就越大，下一次是否报名成功的影响就会增大，相对于报名成功的同学来说，他的新权重值就会大幅度降低，而针对于报名未成功的同学来说，他的新权重值就会大幅度增加，而对于未报名的同学来说，权重就不会发生变化，

对于学术报告来说，学术报告的报名的越多，说明此次学报的热度也就越高，对于没有抢到的同学或者抢到成功的同学彼此的权重误差不是很大，有利于学报维持一个公平的环境，对于冷门的学术报告来说，报名成功的同学，权重大幅度降低，因为相对于热门的学术报告，抢到的概率也就越多，所以下次听报名且成功的概率也就越小，而对于报名但未成功的同学来说，权重大幅度上升，对于下一次选取中，权重越大，选中概率也就越大

### 3.2.2 权重计算

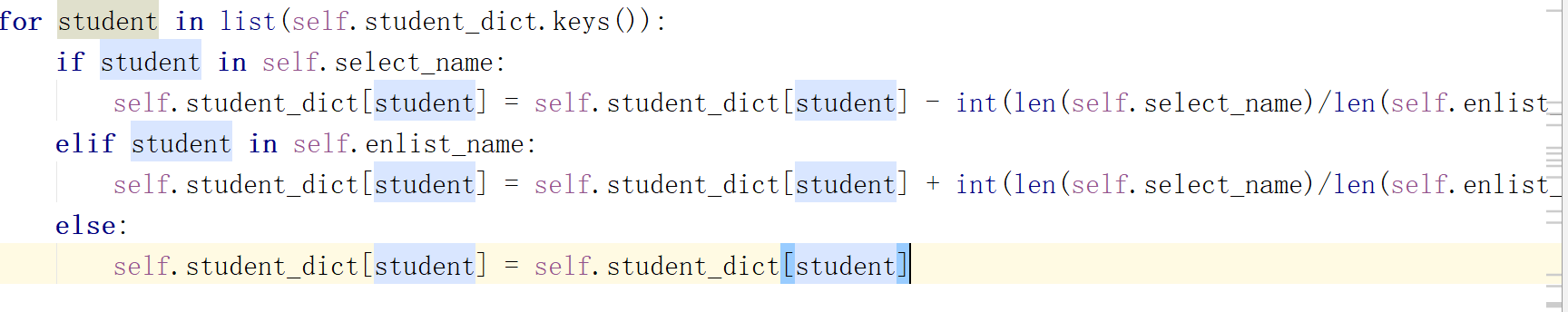
* **权重更新**

### new\_weights = old\_weights + update\_weights + b

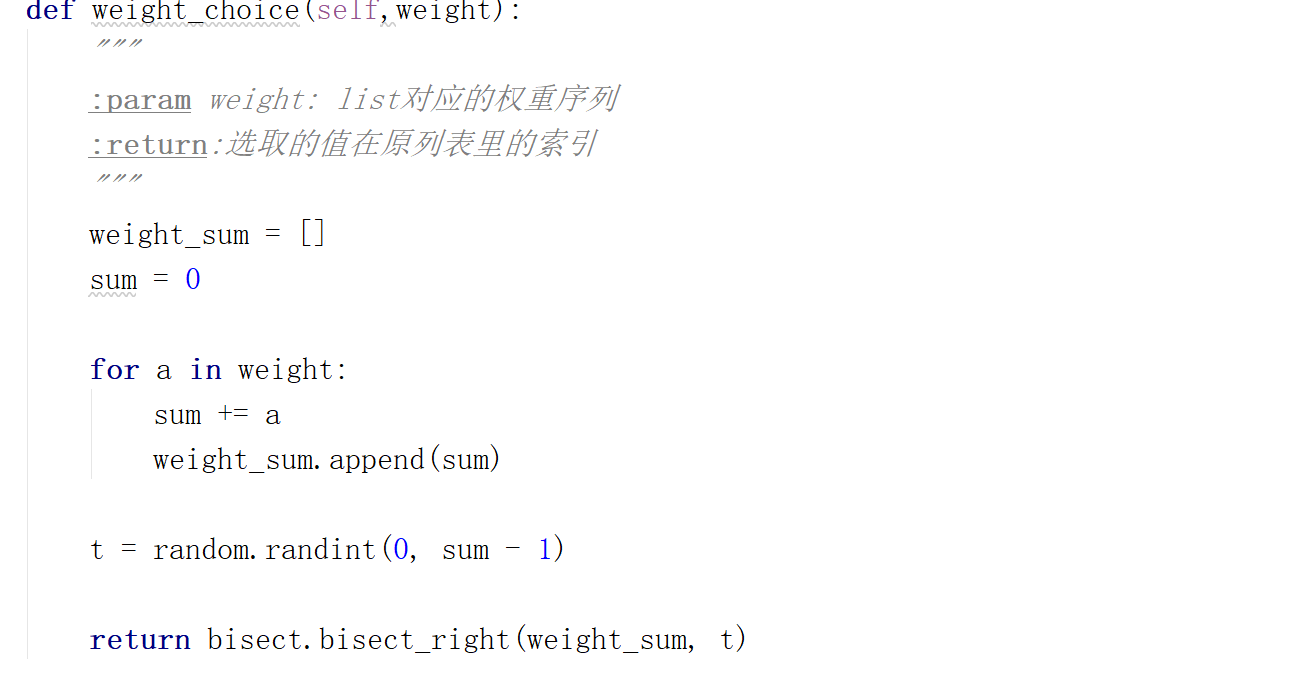
### 解释：新的权重值等于旧的权重值加上权重变化值再加上误差纠正值，

### 对于期初来说，所有学生的权重值为初始值，权重根据每期的学术报告活动就行变化，期末将重置权重，

### b：此数是一个误差纠正的值，用于纠正权重变化异常处理



### 3.2.3 报名成功人数在列表中的位置

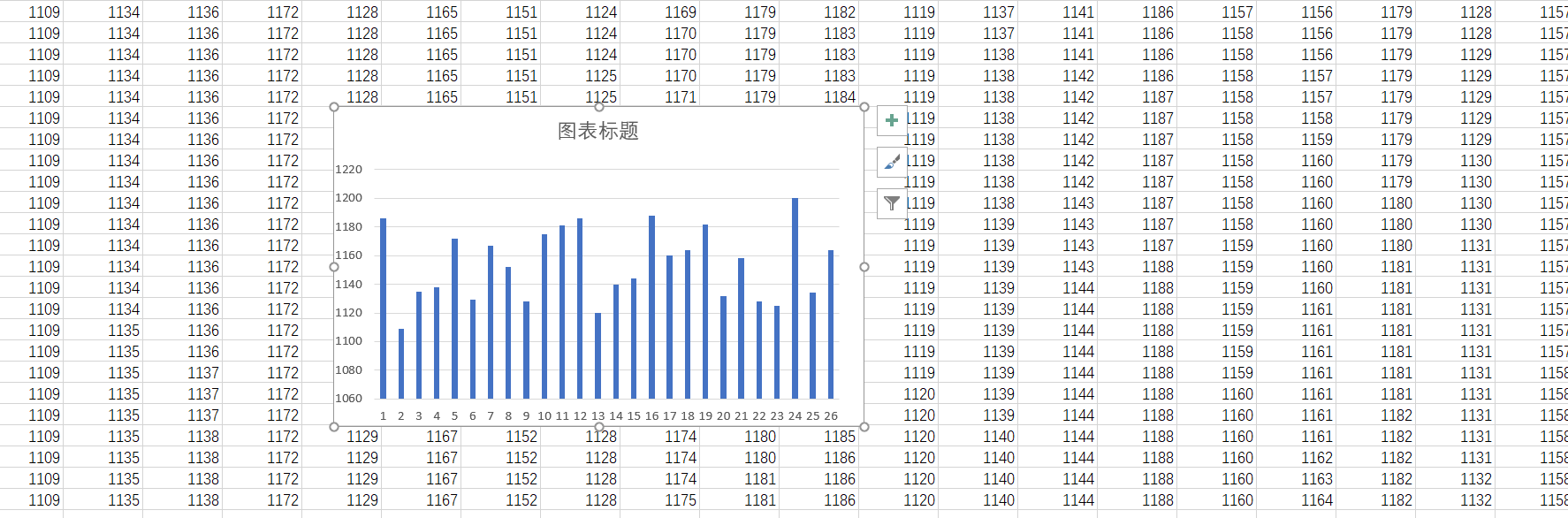


解释：计算权重总和sum，然后在1到sum之间随机选择一个数R，之后遍历整个集合，统计遍历的项的权重之和，如果大于等于R，就停止遍历，选择遇到的项。

还是以上面的集合为例，sum等于10，如果随机到1-5，则会在遍历第一个数字的时候就退出遍历。符合所选取的概率。

选取的时候要遍历集合，它的时间复杂度是O（n）。

### 3.2.4 返回每个学生报名成功的次数



解释：进行10000次模拟，26个字母返回的次数大概分布在1150左右，说明此算法合理

# 六、影响