****

**演讲比赛流程**

**管理系统**

学系： 信息科学学院

专业： 计算机科学与技术

项目成员： 梁憬乐17052016

刘国金17052104

刘国超17052041

李林杰17052077

指导教师： 江嘉治

二〇二〇 年 六 月

目录

[一、演讲比赛流程管理系统需求分析 3](#_Toc44485020)

[1.1演讲比赛流程管理系统及其问题陈述 3](#_Toc44485021)

[1.2演讲比赛流程管理系统用例析取 3](#_Toc44485022)

[二、用例建模 5](#_Toc44485023)

[2.1用例设计 5](#_Toc44485024)

[2.2演讲比赛流程管理系统用例规约 5](#_Toc44485025)

[2.2.1展示菜单用例的用例规约 5](#_Toc44485026)

[2.2.2进行比赛用例的用例规约 6](#_Toc44485027)

[2.2.3查看往届比赛成绩用例的用例规约 7](#_Toc44485028)

[2.2.4清空比赛记录用例的用例规约 7](#_Toc44485029)

[2.2.5退出系统用例的用例规约 8](#_Toc44485030)

[三、系统整体架构设计 9](#_Toc44485031)

[3.1架构描述 9](#_Toc44485032)

[3.2架构图 9](#_Toc44485033)

[四、类图建模 9](#_Toc44485034)

[4.1选手类的设计 9](#_Toc44485035)

[4.2演讲比赛管理类的设计 10](#_Toc44485036)

[五、行为建模 10](#_Toc44485037)

[5.1顺序图 10](#_Toc44485038)

[5.2状态图 10](#_Toc44485039)

[六、设计模式 10](#_Toc44485040)

[七、系统测试 11](#_Toc44485041)

|  |  |  |  |
| --- | --- | --- | --- |
| 小组成员 | 姓名 | 学号 | 任务分工 |
| 组长 | 梁憬乐 | 17052016 | 编写程序，uml建模 |
| 组员 | 刘国金 | 17052104 | Uml建模，编写程序 |
| 组员 | 刘国超 | 17052041 | 查找资料，编写文档 |
| 组员 | 李林杰 | 17052077 | 查找资料，编写文档 |

# 一、演讲比赛流程管理系统需求分析

## 1.1问题陈述

本系统设计为基于STL的演讲比赛管理系统，采用C++语言设计程序，用于管理演讲比赛流程。分为两个类，选手类和演讲比赛管理类，选手类存放选手信息，演讲比赛管理类存放不同功能的操作及容器。系统主要分为五个功能：展示菜单、进行比赛、查看往届比赛记录、清空记录、退出系统。

#### 资源

本演讲比赛流程管理系统提供了选手信息与选手排名

#### 活动列表

1. 用户通过抽签对选手比赛进行排序及分组。
2. 用户可以查看评委的打分。
3. 用户可以通过查看选手的成绩，可查看晋级结果。
4. 用户可以查看每一届比赛胜出者成绩。

#### 信息资源列表

·基本选手信息（姓名，编号）

·选手成绩信息（编号，每轮成绩）

·评委打分信息

·获胜选手信息（冠亚季军编号，最终得分）

## 1.2用例图

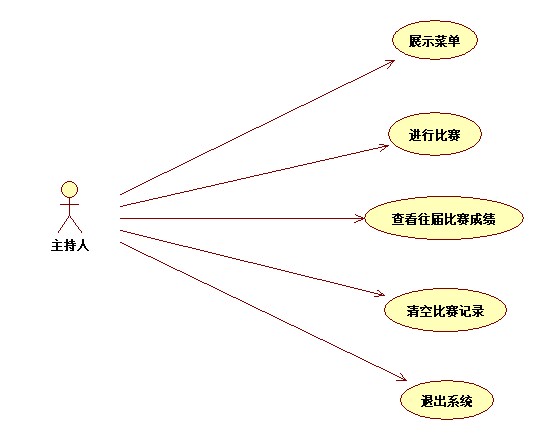


图1 演讲比赛流程管理系统用例图

## 1.3用例规约

### 1.3.1展示菜单用例

1．简要说明

本用例允许用户查看菜单并选择菜单上的功能使用户和系统进行交互。

2．事件流

2.1基本事件流

（1）查看菜单

用户进入系统时菜单自动弹出来。

（2）选择功能

用户输入菜单上的功能编号，系统执行相应功能：

若用户输入“1”，则执行“开始比赛”功能；

若用户输入“2”，则执行“查看往届成绩”功能；

若用户输入“3”，则执行“清空比赛记录”功能；

若用户输入“0”，则执行“退出系统”功能。

2.2备选事件流

用户输入非菜单上的编号，则清屏所输入的内容，用户继续输入菜单编号。

### 1.3.2进行比赛用例

1.简要说明

本用例允许用户用系统对选手进行抽签、评分，并显示每位选手得分、显示晋级信息，保存成绩到文件中。

2.事件流

2.1基本事件流

用例开始于用户选择菜单上的“1”键功能。

1. 第一轮比赛选手抽签

a.用户输入“1”，进入开始比赛功能。

b.第一轮选手进行抽签，抽签后给出12位选手编号排序，分为两组，一组6名选手。

c.选手进行比赛。

1. 第一轮比赛评委打分

系统给选手评分，去掉最高分和最低分算出平均分，得出每个选手的第一轮比赛成绩。

1. 显示第一轮晋级信息
2. 第一轮选手中每组最高分的三名选手成功晋级到下一轮比赛。
3. 系统显示晋级的六名选手的姓名、编号、成绩。
4. 第二轮选手抽签

a.用户输入“1”，开始第二轮比赛。

b.第二轮选手进行抽签，六名选手按照抽签顺序进行比赛。

1. 第二轮评委打分

系统给选手评分，去掉最高分和最低分算出平均分，得出每个选手的第二轮比赛成绩。

1. 显示胜出者信息

系统显示最后胜出三名选手的姓名、编号、成绩。

1. 保存记录

系统将本届比赛胜出三名选手的编号及成绩保存到文件speech.csv中。

2.2备选事件流

无

### 1.3.3查看往届比赛成绩用例

1.简要说明

本用例允许用户查看往届比赛成绩。成功查找到成绩时显示往届比赛成绩。

2.事件流

用例开始于用户选择菜单上的“2”键功能。

2.1基本事件流

（1）用户在菜单跳出后输入菜单上的编号“2”，系统读取往届所有比赛的成绩。

（2）系统将往届比赛的前三名选手编号、姓名、得分输出到显示器上。

2.2备选事件流

（1）当往届比赛记录为空时（未进行比赛或记录被清空），系统将显示“文件不存在”或“文件为空”。

（2）若系统内有比赛记录，但文件被删除了，系统依旧输出往届比赛成绩。

### 1.3.4清空比赛记录用例

1.简要说明

本用例允许用户对系统内的比赛记录进行清空操作。

2.事件流

用例开始于用户选择菜单上的“3”键功能。

2.1基本事件流

（1）用户在菜单跳出后，输入“3”，系统输出“是否确认清空文件？ 1、是 2、否”。

（2）若用户输入“1”，系统执行比赛记录清空功能；若用户输入“2”或其他，则清屏并跳到输入菜单编号。

（3）用户输入“1”后，若系统内有比赛记录，则系统会将系统内及文件上的比赛记录清空；若系统内无比赛记录，则系统依旧执行清空功能。

（4）完成清空记录后，系统输出“清空成功！”。

2.2备选事件流

（1）在系统输出“是否清空文件1、是 2、否”后，用户若输入“2”或其它,

系统则清屏并跳到输入菜单编号。

（2）当执行清空功能时，若系统内及文件中没有比赛记录或没有文件时，依旧执行清空功能。

### 2.2.5退出系统用例

1.简要说明

本用例允许用户让系统执行退出系统操作。

2.事件流

用例开始于用户选择菜单上的“0”键功能。

2.1基本事件流

用户在菜单跳出后输入“0”，执行退出系统。

2.2备选事件流

无

# 三、系统整体架构设计

## 3.1架构描述

本系统采用MVC（Model View Controller）模式进行架构设计。MVC是软件工程中的一种软件架构模式，将软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）。

1) 视图（View）：

即视图层，为用户提供交互界面，负责接收用户的命令，交给控制层处理，显示处理结果，并且调用控制层提供的接口进行数据传递。

2) 控制器（Controller）：

即业务逻辑层是系统的核心组成部分。管理所有系统功能有关的数据流和控制流，从而实现系统的逻辑功能。向上提供服务接口给表示层调用。在控制层中，不同的子系统提供独立的服务接口，各自独立实现，互不干预影响。控制层依赖于持久化提供的持久化服务。

3) 模型（Model）：

即数据访问层，封装了与应用程序的业务逻辑相关的数据，它也是持久化层包括数据实体和持久化逻辑实现，并向上提供接口。属于不同子系统的数据实体放置在不同的包中。

## 3.2架构图



图 2 演讲比赛流程管理系统架构图

# 四、类图建模

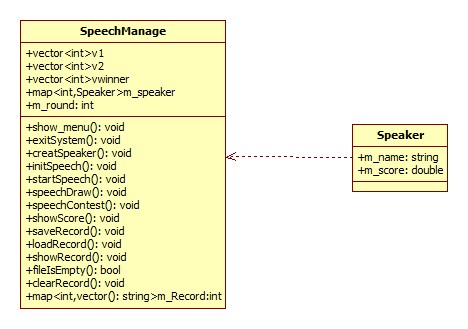


图 3 类图

# 五、用例析取

## 5.1进行比赛用例

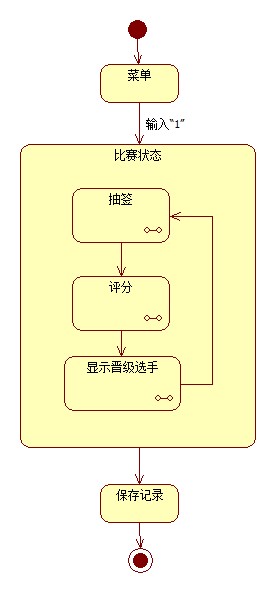


图 4进行比赛用例状态图

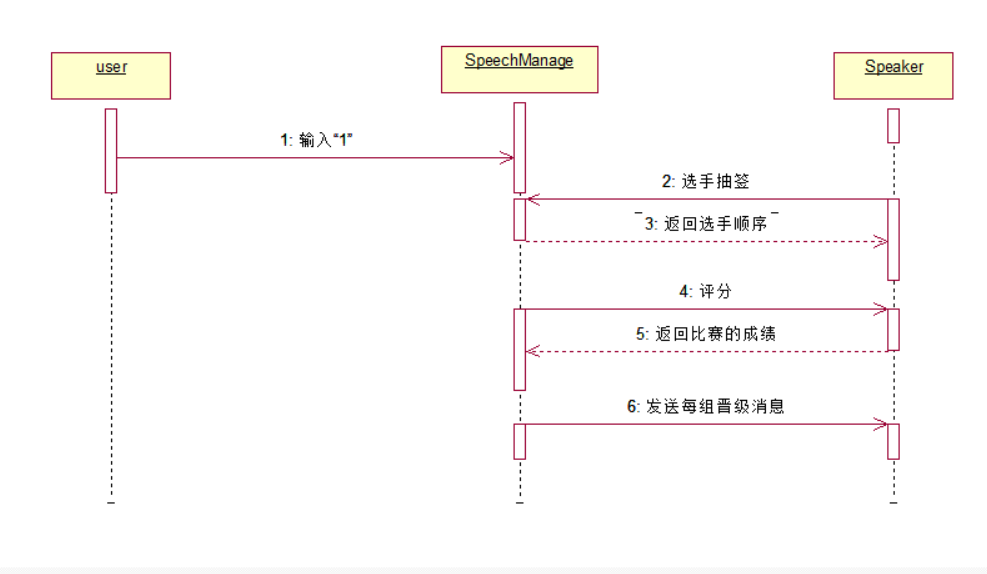


图 5进行比赛用例时序图

## 5.2查看往届成绩用例

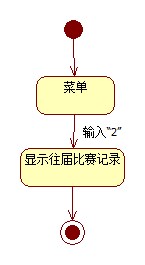


图 6查看往届比赛用例状态图

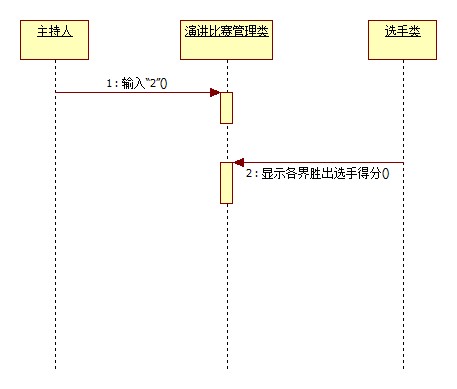


图 7查看往届比赛用例时序图

## 5.3清空比赛记录用例

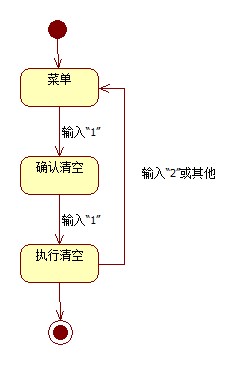


图 8 清空比赛记录用例状态图

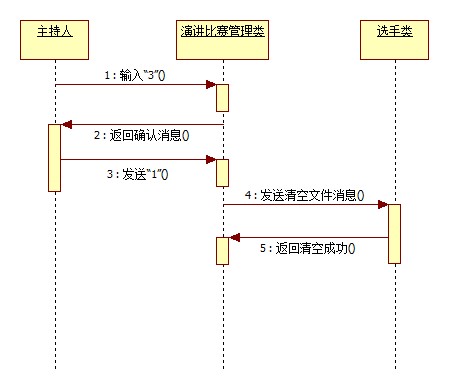


图 9 清空比赛记录用例时序图

## 5.4退出系统用例

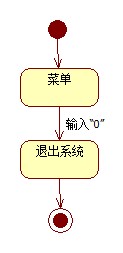


图 10 退出系统用例状态图

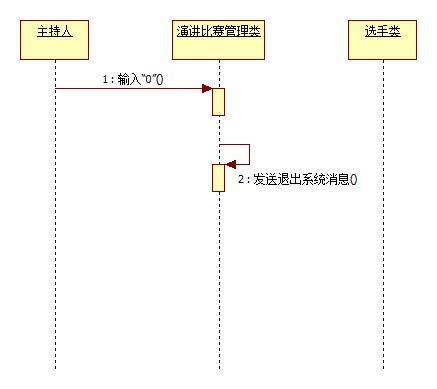


图 11 退出系统用例时序图

# 六、设计模式

本项目运用了观察者模式。

**1、抽象主题（Subject）：**

它把所有观察者对象的引用保存到一个聚集里，每个主题都可以有任何数量的观察者。抽象主题提供一个接口，可以增加和删除观察者对象。

**2、具体主题（Concrete Subject）：**

将有关状态存入具体观察者对象；在具体主题内部状态改变时，给所有登记过的观察者发出通知。

**3、抽象观察者（Observer）：**

为所有的具体观察者定义一个接口，在得到主题通知时更新自己。

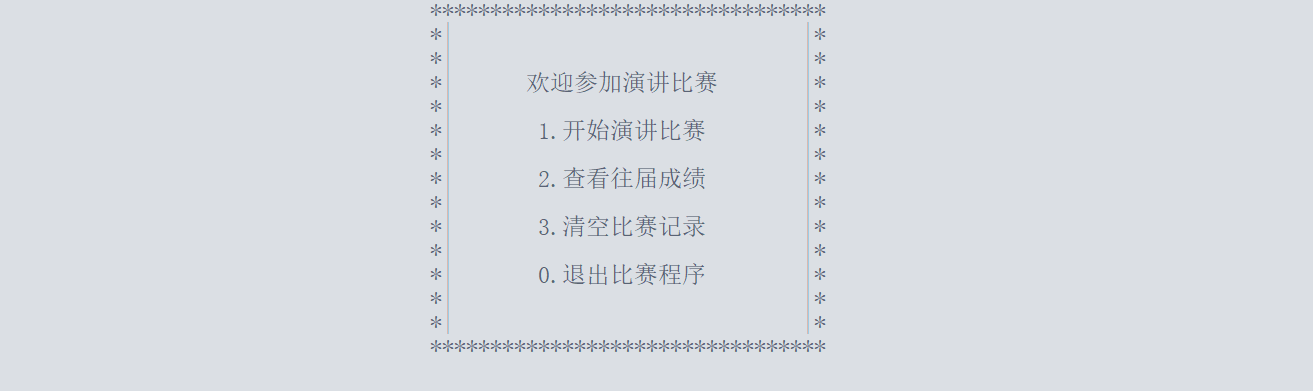
**4、具体观察者（Concrete Observer）：**

实现抽象观察者角色所要求的更新接口，以便使本身的状态与主题状态协调。

# 七、系统测试

## 7.1展示菜单

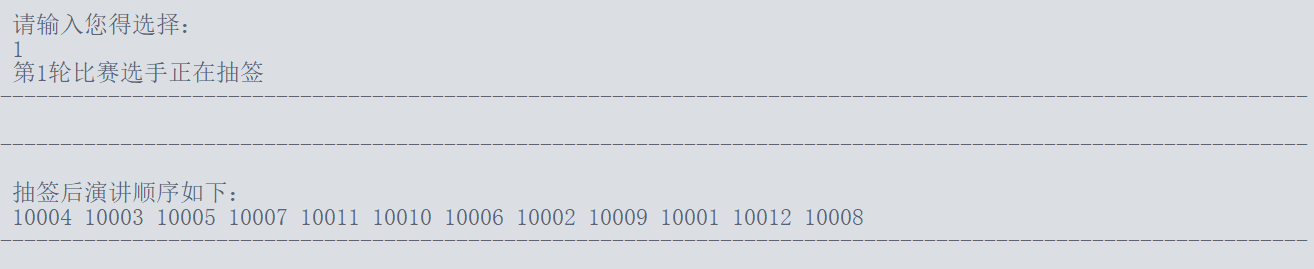
运行系统自动弹出菜单



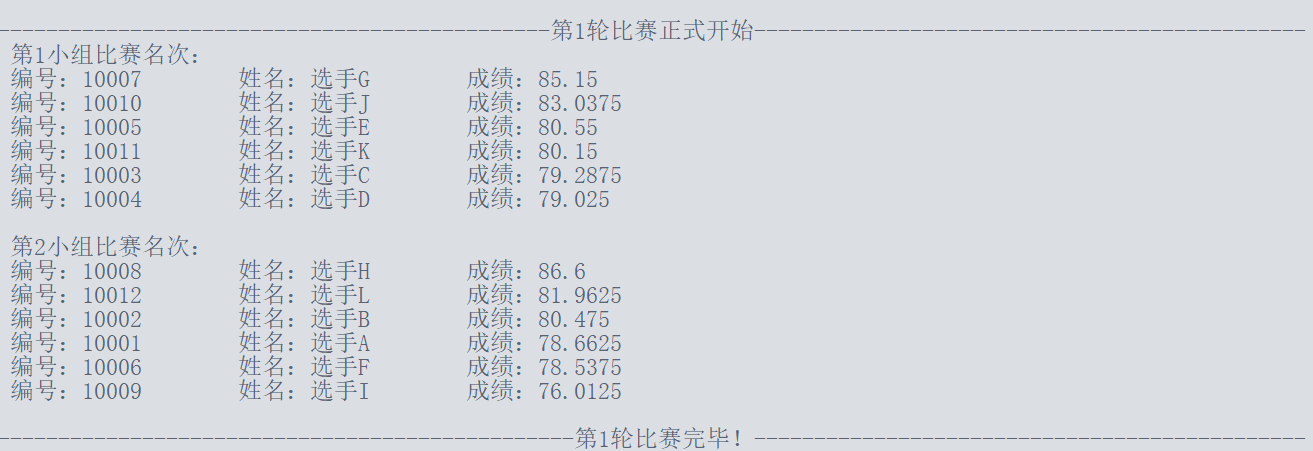
## 7.2开始比赛

### 第一轮比赛

输入“1”后开始抽签



评分



显示晋级信息



### 第二轮比赛

抽签



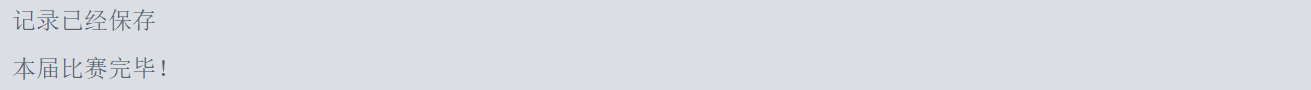
评分



显示最终胜出选手信息



保存比赛记录

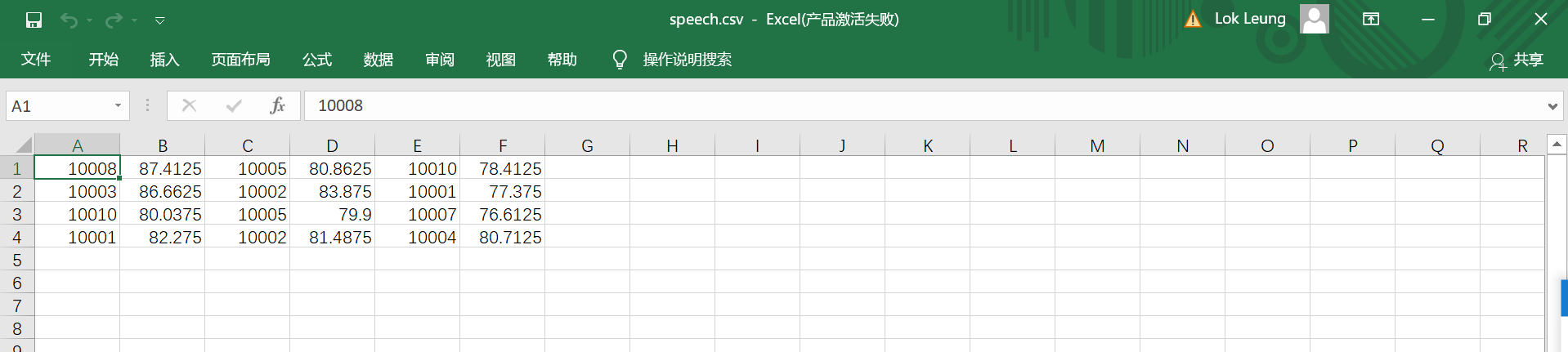


## 7.3查看往届比赛成绩

输入“2”

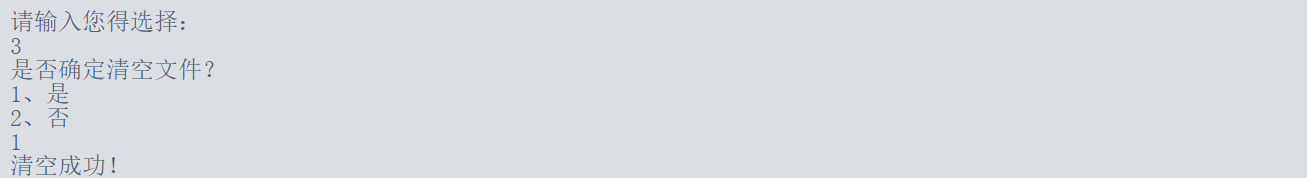


打开文件查看比赛记录

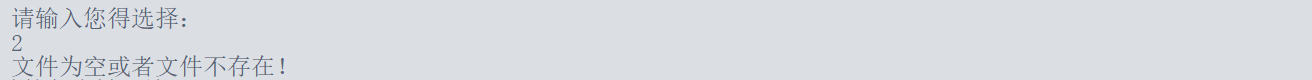


## 7.4清空比赛记录

输入“3”,再输入“1”



清空后输入“2”查看往届比赛记录



清空后文件中比赛记录也清空



## 7.5退出系统

输入“0”

