

“数据结构”

课程设计报告

**设计题目**  学生成绩管理系统

**姓 名**  侯腾跃

**学 号**  202217477

**专 业**  计算机科学与技术

**班 级**  2022级4班

**完成日期**  7月1日

课程设计成绩评定表

|  |  |  |  |
| --- | --- | --- | --- |
| **设计题目** | 学生成绩管理系统 | **成绩** |  |
| **内容简介** | 主要功能是对批量学生的各门成绩进行录入、修改、查询、统计等，要求方便快速。记录学生的学号、姓名、班级、性别、联系电话以及课程和成绩；可以对学生的成绩按学号和姓名进行查寻；输出显示学生成绩；并实现排序、统计及格率和优秀率功能。 | | |
| **评语** | **教师签名：** | | |

**（一） 需求和规格说明**

(1)界面基本要求：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

学生成绩管理系统

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* F1 --[帮助](http://www.25175.com/help) \*\*

\*\* F2 --输入数据并存入文件 \*\*

\*\* F3 --根据学号查询成绩 \*\*

\*\* F4 --根据姓名查询成绩 \*\*

\*\* F5 --输出文件内容 \*\*

\*\* F6 –成绩排序 \*\*

\*\* F7 --统计及格和优秀人数 \*\*

\*\* ESC--退出系统 \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

另：提倡用MFC的对话框做简单的输入输出交互界面。

(2)功能要求：

1)[帮助](http://www.25175.com/help):系统使用方法的相关信息。

2)输入数据并存入文件：输入相关信息，并实现文件流的读写操作。

3)根据学号查询成绩：输入学号，查询学生的各门成绩

4)根据姓名查询成绩：输入姓名，查询学生的各门成绩

5)输出文件内容：屏幕输出显示所有学生的成绩

6) 成绩排序：对某门成绩或总分进行快速排序，显示、保存

7)统计及格和优秀人数：统计及格和优秀率。

8)退出

**（二） 设计**

1.printHelp(): 打印帮助信息。这个函数简单地输出一些操作选项供用户参考。时间复杂度为O(1)。

2.readScoresFromFile(): 从文件中读取成绩信息。这个函数打开文件并逐行读取数据，将每行数据解析为Score结构体对象，并将这些对象存储在一个向量中返回。时间复杂度取决于文件中的数据量，假设为N，那么时间复杂度为O(N)。

3.saveScoresToFile(const vector<Score>& scores): 将成绩信息保存到文件。这个函数打开文件并逐个写入每个Score结构体对象的数据。时间复杂度取决于成绩信息的数量，假设为N，那么时间复杂度为O(N)。

4.queryScoreById(const vector<Score>& scores): 根据学号查询成绩。这个函数会提示用户输入学号，然后在成绩信息向量中查找对应的学生，如果找到则输出学生的成绩信息，否则输出未找到提示。时间复杂度取决于成绩信息的数量，假设为N，那么时间复杂度为O(N)。

5.queryScoreByName(const vector<Score>& scores): 根据姓名查询成绩。这个函数会提示用户输入姓名，然后在成绩信息向量中查找对应的学生，如果找到则输出学生的成绩信息，否则输出未找到提示。时间复杂度取决于成绩信息的数量，假设为N，那么时间复杂度为O(N)。

6.printScores(const vector<Score>& scores): 输出文件内容。这个函数会遍历成绩信息向量，逐个输出每个学生的成绩信息。时间复杂度取决于成绩信息的数量，假设为N，那么时间复杂度为O(N)。

7.sortScores(vector<Score>& scores): 成绩排序。这个函数会根据用户选择的排序方式，使用C++的sort函数对成绩信息向量进行排序。排序的时间复杂度取决于排序算法的性能，可以认为是O(NlogN)。

8.countPassedAndExcellent(const vector<Score>& scores): 统计及格和优秀人数。这个函数会遍历成绩信息向量，逐个判断每个学生的成绩是否及格和优秀，并统计人数。时间复杂度取决于成绩信息的数量，假设为N，那么时间复杂度为O(N)。

9.main(): 程序入口。这个函数会不断循环，根据用户输入的选项调用相应的函数来执行操作。时间复杂度取决于用户的操作次数，假设为M，那么时间复杂度为O(M)。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **结构名** | **类型** | **函数名** | **形参** | **描述** |
| Score | Score | readScoresFromFile |  | 从文件中读取成绩信息，并返回一个Score结构体向量 |
| Void | printHelp |  | 打印帮助信息 |
| Void | saveScoresToFile | scores | 将成绩信息保存到文件中 |
| Void | queryScoreByName | scores | 根据学号查询成绩 |
| Void | queryScoreByName | scores | 根据姓名查询成绩 |
| Void | printScores | scores | 输出文件内容 |
| Void | sortScores | scores | 根据不同排序方式排序学生信息 |
| Void | countPassedAndExcellent | scores | 统计及格和优秀人数 |

**（三） 用户手册**

1)[帮助](http://www.25175.com/help):系统使用方法的相关信息。

2)输入数据并存入文件：输入相关信息，并实现文件流的读写操作。

3)根据学号查询成绩：输入学号，查询学生的各门成绩

4)根据姓名查询成绩：输入姓名，查询学生的各门成绩

5)输出文件内容：屏幕输出显示所有学生的成绩

6) 成绩排序：对某门成绩或总分进行快速排序，显示、保存

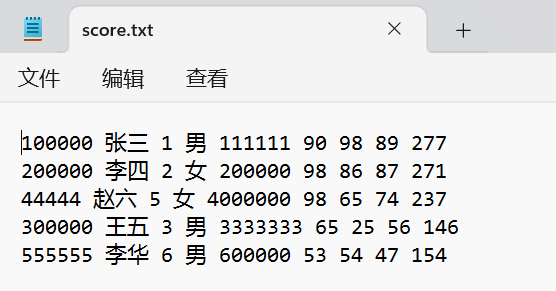
7)统计及格和优秀人数：统计及格和优秀率。

8)退出

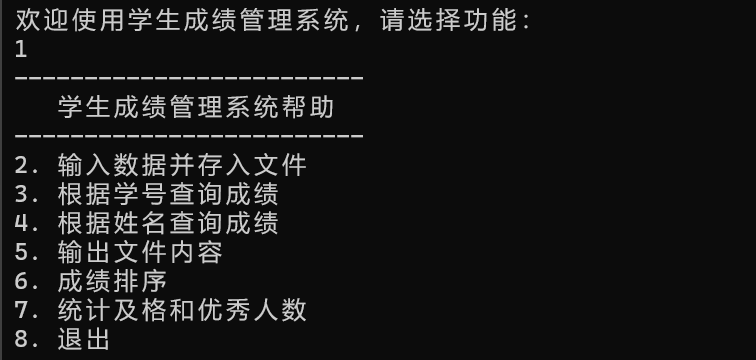
**（四）调试及测试**

每次输入一个学生的信息都被存入到文件score.txt中且不会发生改变。故以后每次调用sortScores()等函数时，都会输出原来的信息。

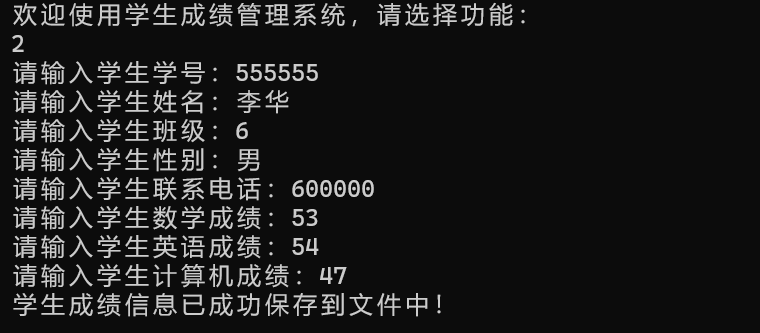
**（五） 运行实例：**

****

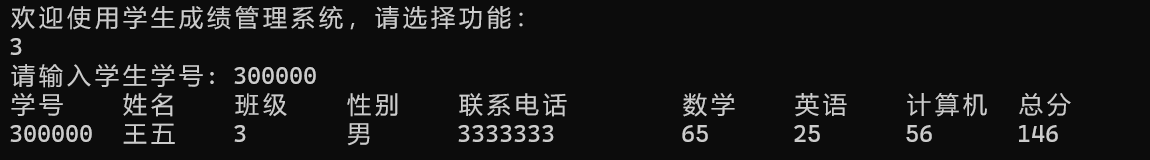
1)[帮助](http://www.25175.com/help):系统使用方法的相关信息。



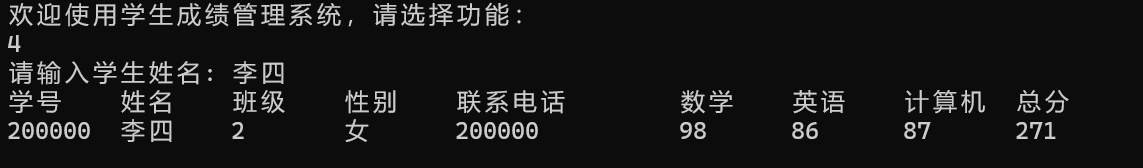
2)输入数据并存入文件：输入相关信息，并实现文件流的读写操作。



3)根据学号查询成绩：输入学号，查询学生的各门成绩



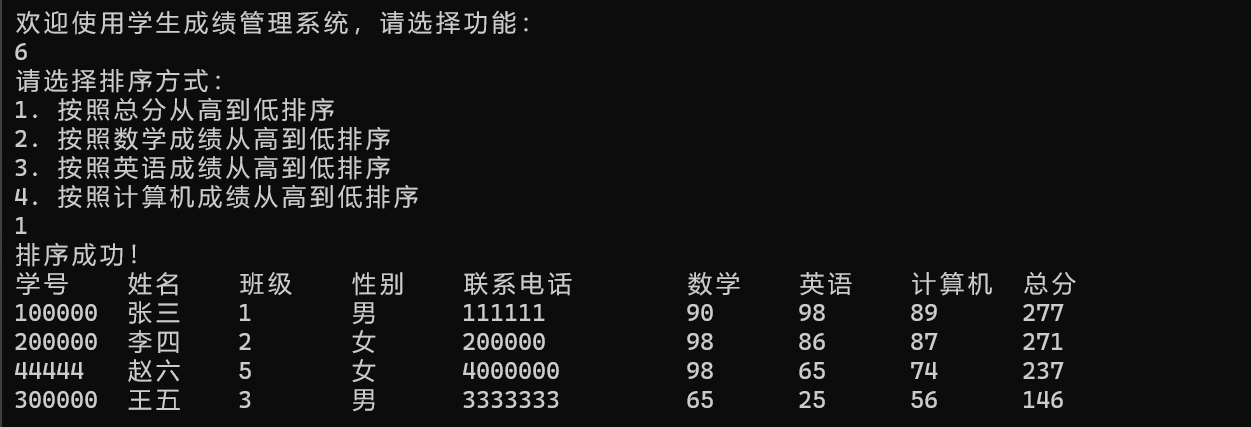
4)根据姓名查询成绩：输入姓名，查询学生的各门成绩



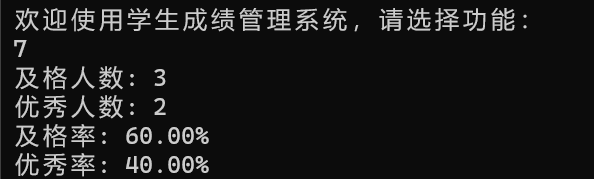
5)输出文件内容：屏幕输出显示所有学生的成绩



6) 成绩排序：对某门成绩或总分进行快速排序，显示、保存



7)统计及格和优秀人数：统计及格和优秀率。



**（六）进一步改进**

1.权限管理与用户验证：考虑到系统的安全性和数据的隐私性，可以引入用户验证和权限管理的功能。不同的用户拥有不同的权限，只能访问和修改相应权限范围内的数据。这样可以有效保护学生数据的安全和隐私。

2.用户反馈与改进机制：建立用户反馈与改进机制，在系统中添加反馈界面或意见收集渠道，获取用户的意见和建议。根据用户的反馈，进行适当的改进和优化，以提高系统的用户体验和功能完善程度。

**（七）心得体会**

学生成绩管理系统是一个典型的信息管理系统，涉及到数据的输入、查询、排序、统计和输出等多个功能。通过这个小项目的编写，我进一步巩固了对C++语言的使用和文件操作的理解。同时，我也意识到了良好的程序结构和代码风格对于项目的可扩展性和可维护性的重要性。在这个项目中，我将不同的功能划分为独立的函数，使得代码清晰、易读、易于扩展。

算法对于程序的效率和性能也有着重要影响。在这个项目中，我使用了简单的线性查找算法来根据学号和姓名查询成绩，这在小规模数据上效率尚可。但如果面对大规模数据时，更高效的算法如二分查找或哈希表等可以提高查询效率。而排序功能使用了C++标准库中的sort函数，这是一种高效的排序算法，但也可以根据实际需求选择其他排序算法。对于这里的统计功能，遍历一次数据即可完成，时间复杂度较低。

除了算法和数据结构的选择，代码的规范性和可读性也是非常重要的。良好的命名规范、注释说明和适当的代码缩进和格式化，都可以让代码更易于阅读和维护。对于以后的学习，我会更加注重算法书写能力，不断提升自己。

**（八）对课程设计的建议**

详细的需求分析：在开始课程设计之前，应该进行详细的需求分析。明确系统的功能和用户需求，包括输入数据、查询成绩、排序、统计、输出等功能，以及对数据的存储和读写操作。这样能够确保设计的系统符合实际需求，有助于后续的开发工作。

模块化设计：将整个系统划分为多个模块，每个模块负责不同的功能，如数据输入模块、查询模块、排序模块、统计模块、文件操作模块等。这样的设计能够提高代码的可复用性和可维护性，并且便于团队协作开发。

**（九）附录⎯⎯源程序**

