

东软秘密

**设计报告  
*（MySTS(Self-Test-System)算数小天才——小学算数自测系统）***

东软集团股份有限公司IT人才实训中心

版权所有，翻版必究

**更改履历**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 更改时间 | 更改人 | 更改章节 | 状态 | 更改描述 |
| 1 | V1.0 | 2019.7.21. | 刘宇科、姚远、兰鑫、梁堉 |  | 新建 |  |
| 2 | V2.0 | 2019.7.26 | 刘宇科、姚远、兰鑫、梁堉 | 整体修订 | 修改 |  |
| 3 |  |  |  |  |  |  |

状态：新建、增加、修改、删除。

目 录

[1 引言](#_Toc72) 4

[1.1 编制目的](#_Toc31214) 4

[1.2 词汇表](#_Toc30014) 4

[1.3 参考资料](#_Toc9081) 4

[2 系统开发环境](#_Toc28021) 4

[3 系统设计思路](#_Toc12777) 4

[4 系统公用代码设计](#_Toc23556) 6

[4.1 登录函数：](#_Toc6972) 6

[4.2 异常处理](#_Toc28503) 7

[5 功能模块设计](#_Toc1239) 7

[5.1 加法自测模块](#_Toc3292) 7

[5.2 减法自测模块](#_Toc3292) 10

[5.3 乘法自测模块](#_Toc3292) 12

[5.4 除法自测模块](#_Toc3292) 14

[5.5 学生菜单模块](#_Toc3292) 16

[5.6 老师菜单模块](#_Toc3292) 18

[5.7 成绩查询模块](#_Toc3292) 23

[5.8 学生评价模块](#_Toc3292) 25

[5.9 休息模块](#_Toc3292) 26

[5.10 错题集模块](#_Toc3292) 28

[5.11 综合自测模块](#_Toc3292) 29

[5.12 游戏模块](#_Toc3292) 31

[5.13 小测模块](#_Toc3292) 32

[6 数据结构设计](#_Toc29107) 34

[6.1 功能说明](#_Toc2343) 34

[6.2 数据表设计](#_Toc31439) 34

# 引言

## 编制目的

本报告详细完成对小学算术自测系统的整体设计，达到指导开发的目的，同时实现和测试人员及用户的沟通。

本报告面向详细设计人员、开发人员、测试人员及最终用户而编写，是了解系统的导航。

## 词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| comTest | 综合测试 |  |
| modifyPin | 修改密码 |  |
| manage | 管理学生信息 |  |
|  |  |  |
|  |  |  |

## 参考资料

《C语言大学教程（第八版）》

# 系统开发环境

操作系统： Windows XP

集成开发工具：Dev C++, CLion

编译环境：TDM-GCC 4.9.2 64-bit Release, MinGW

# 系统设计思路

为了实现学生的增删查改操作，我们将学生的信息定义为结构体并采用链表来进行储存，而对于老师这个结构体，由于我们知道老师的数目有限，所以我们采用结构体数组的方式来储存老师的数据。

为了实现登陆和注册功能，以此进一步实现对数据的保护，我们将学生和老师信息储存在文件当中，利用文件的存储和读取来更新和获得学生和老师的信息。

为了实现随机出题功能，我们调用了随机数来对这个部分进行实现。

为了更贴合小学生的算术现状，我们设置了错题本，利用每个学生的专属文件来对学生的错题进行存储，以便于学生复习和从小养成错题集习惯。

为了便于老师查看成绩，我们使用链表的排序实现了排行榜的功能，但考虑到对学生自身的保护，只有老师可以查看排行榜。

为了实现老师与学生的及时沟通和信息反馈，我们设置了评价模块，老师可以对学生进行评价，学生可在自己的端口查看老师对自己的评价。

为了最大化实现代码重用，我们前期进行了很多的讨论，将各个功能模块进行分析并将其拆分成小函数，以此来实现代码重用的功能。

为了获得更好的用户体验，我们自动处理了用户的不合法出入，防止出现大量的输入错误提示信息。

**系统设计具体思路：**

系统分为老师和学生两个端，均可实现登录，文件存取的功能，且均有一个主菜单。

学生主菜单可以选择基础测试，综合测试，游戏模式，小测，查看错题，查看成绩及评价和修改密码。每种测试类型对应专门的出题函数。测试错题或游戏记录保存到文件

老师菜单可以选择学生管理，排名查询和修改密码。在学生管理中可以选择增加，删除，查询和修改学生信息，给学生留评语。

如图3-1所示为系统设计思路图。



* 项目关键性技术：数组、指针、函数、结构体、单向链表

# 系统公用代码设计

## 登录函数：

### 功能描述

该功能目的在于保护用户的信息以及设置改系统的针对性，用户进入改系统以后，需要选择角色，如果老师选择了学生，或者学生选择了老师，会分别在输入了错误的账号密码和错误的密钥以后回到最初界面，学生只能进行登陆，老师可以进行登陆，也可以在输入密钥正确以后进行注册。为了避免老师学生角色选择，以及后期的选择错误，选择了在login函数里调用自己的方法，使其回到最初的界面。

流程图如图4-1所示：

**

### 函数方法描述login函数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| int | login | 实现用户登录和注册并用返回值说明是学生还是老师 | 无参数，返回值为1，表示学生，返回值为0，表示老师 |

#### password函数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| char\* | password | 实现密码的隐形输入，每输入一个字符都用\*代替，并实现了退格功能 | 无参数，返回表示密码的字符串 |

## 异常处理

### 功能说明

对用户输入的异常数据（包括数据类型错误，长度不符合，输入的数据范围不正确）进行处理

### 设计说明

如果用户输入的数据与系统要求的数据类型不符合，提示错误信息，并让用户重新输入。

如果用户输入的是字符串，会进行字符串长度的判断，如果字符串长度不在数据规范的范围内，将提示错误信息，并让用户重新输入。

如果输入的是数字，判断数字范围，如果数字不再规定的范围内，将会提示错误信息，并让用户重新输入。

# 功能模块设计

## *数据读取存储模块*

### 功能说明

该部分实现对于学生、老师信息的保存和读取。保存时，将现有的学生和老师的结构体内容写到文件当中，读取时，则从文件中将老师，和学生的信息读取到结构体当中。

### 函数、方法设计

#### readStudentFile函数 该函数的功能：实现创建链表，从文件中将学生信息读取到链表当中的功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | readStudentFile | 创建链表，从文件中将学生信息读取到链表当中 | 将链表头指针付给headStudent,以用于整个项目中的链表遍历 |

#### readTeacherFile函数 该函数的功能：实现从文件中将老师信息读取到文件当中的功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | readTeacherFile | 从文件中将老师信息读取到结构体数组当中 | 无 |

#### saveStudentFile函数 该函数的功能：实现从链表中的学生信息写到文件中的功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | saveStudentFile | 从链表中的学生信息写到文件中 | 用headStudent,进行链表遍历来输出学生信息 |

#### saveTeacherFile函数 该函数的功能：实现从结构体数组中的老师信息写到文件中的功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | saveTeacherFile | 从链表中的老师信息写到文件中 | 遍历结构体数组teachers来输出学生信息 |

### 相关数据表

|  |  |  |
| --- | --- | --- |
| 数据名称 | headStudent | teachers |
| 数据类型 | struct student\* | struct teacher[] |
| 数据说明 | 用来记录学生链表的头指针 | 用于储存老师信息 |

## *加法自测模块*

### 功能说明

用户首先选择自测模块，选择好了以后将进行难度选择，并进行题量的自定义，在做题过程中，如果正确，将会提示正确并进入下一题，如果错误会显示正确答案，并提醒用户是否要保存到错题集当中，在用户输入指令后进入下一题。

当答题结束时，会跳出结束界面，展示正确率和评价。

### 函数、方法设计

#### plus函数 该函数的功能：实现加法自测功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | plus | 实现难度选择、自定义题量，并调用plusDemo函数产生题目，对题目进行正误判断 | 无 |

plus、minus、multiply、divide函数流程图如图5-1所示：

#### 

#### plusDemo函数 该函数的功能：产生加法运算式并判断答案是否正确，返回一个整数值说明正误

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| int | plusDemo | 产生加法运算式并判断答案是否正确，返回一个整数值说明正误 | 由一个整形参数表示难易程度 |

### plusDemo函数流程图如图5-2所示：



#### solution函数 该函数的功能：对plusDemo函数的返回值进行处理，如果返回值表示正确，则输出提示答案正确，反之提示答案错误，输出正确答案，并提示用户将其保存到错题本中

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | solution | 对答案正误判断的结果进行处理 | 接受result参数，为1 表示正确，为0表示错误 |

solution函数流程如图5-3所示：



#### judge函数 该函数的功能：对测试的成绩进行评价，用参数正确率，总题数和正确题数来进行判断，对于不同的正确率，输出不同的评价。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | judge | 对测试的成绩输出评价 | 需要一个小数来行的数据表示正确率，两个整形数据分别表示题目总量和正确的题目数量 |

### 相关数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | difficulty | number | answer | difficulty |
| 数据类型 | int | int | int/double | int |
| 数据说明 | 用户输入整数：1表示选择易，2表示选择中，3表示选择难 | 用户输入正整数自定义题量 | 用户输入答案 | plusDemo参数：1表示易，2表示中，3表示难 |
| 数据名称 | correct | num | right | Result |
| 数据类型 | double | int | int | int |
| 数据说明 | 表示正确率 | 表示该次测试的题目总量 | 表示该次测试当中正确的题目数量 | 1表示正确，0表示错误 |

## 减*法自测模块*

### 功能说明

用户首先选择自测模块，选择好了以后将进行难度选择，并进行题量的自定义，在做题过程中，如果正确，将会提示正确并进入下一题，如果错误会显示正确答案，并提醒用户是否要保存到错题集当中，在用户输入指令后进入下一题。

当答题结束时，会跳出结束界面，展示正确率和评价。

### 函数、方法设计

#### minus函数 该函数的功能：实现减法自测功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | minus | 实现难度选择、自定义题量，并调用minusDemo函数产生题目，对题目进行正误判断 | 无 |

#### minusDemo函数 该函数的功能：产生减法运算式并判断答案是否正确，返回一个整数值说明正误

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| int | miusDemo | 产生减法运算式并判断答案是否正确，返回一个整数值说明正误 | 由一个整形参数表示难易程度 |

### minusDemo函数流程图如图5-2所示：

### 

#### solution函数 该函数的功能：对minusDemo函数的返回值进行处理，如果返回值表示正确，则输出提示答案正确，反之提示答案错误，输出正确答案，并提示用户将其保存到错题本中

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | solution | 对答案正误判断的结果进行处理 | 接受result参数，为1 表示正确，为0表示错误 |

solution函数流程如图5-3所示。

#### judge函数 该函数的功能：对测试的成绩进行评价，用参数正确率，总题数和正确题数来进行判断，对于不同的正确率，输出不同的评价。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | judge | 对测试的成绩输出评价 | 需要一个小数来行的数据表示正确率，两个整形数据分别表示题目总量和正确的题目数量 |

### 相关数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | difficulty | number | answer | difficulty |
| 数据类型 | int | int | int/double | int |
| 数据说明 | 用户输入整数：1表示选择易，2表示选择中，3表示选择难 | 用户输入正整数自定义题量 | 用户输入答案 | minusDemo参数：1表示易，2表示中，3表示难 |
| 数据名称 | correct | num | right | Result |
| 数据类型 | double | int | int | int |
| 数据说明 | 表示正确率 | 表示该次测试的题目总量 | 表示该次测试当中正确的题目数量 | 1表示正确，0表示错误 |

## 乘*法自测模块*

### 功能说明

用户首先选择自测模块，选择好了以后将进行难度选择，并进行题量的自定义，在做题过程中，如果正确，将会提示正确并进入下一题，如果错误会显示正确答案，并提醒用户是否要保存到错题集当中，在用户输入指令后进入下一题。

当答题结束时，会跳出结束界面，展示正确率和评价。

### 函数、方法设计

#### multiply函数 该函数的功能：实现乘法自测功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | multiply | 实现难度选择、自定义题量，并调用multiplyDemo函数产生题目，对题目进行正误判断 | 无 |

#### 5.4.2.2 multiplyDemo函数 该函数的功能：函数接受用户输入的困难度参数后，生成对应难度的乘法题目，打印给用 户，用户输入答案后，判断正误，并给出返回值传到divide函数中。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| int | multiplyDemo | 产生乘法运算式并判断答案是否正确，返回一个整数值说明正误 | 由一个整形参数表示难易程度 |

### multiplyDemo函数流程图如图5-2所示：

### 

### 相关数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据名称 | difficulty | number | answer | difficulty | multiplyDemo返回值 |
| 数据类型 | int | int | int/double | int | int |
| 数据说明 | 用户输入整数：1表示选择易，2表示选择中，3表示选择难 | 用户输入正整数自定义题量 | 用户输入答案 | multiplyDemo参数：1表示易，2表示中，3表示难 | 1表示正确，0表示错误 |

#### solution函数 该函数的功能：对multiplyDemo函数的返回值进行处理，如果返回值表示正确，则输出提示答案正确，反之提示答案错误，输出正确答案，并提示用户将其保存到错题本中

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | solution | 对答案正误判断的结果进行处理 | 接受result参数，为1 表示正确，为0表示错误 |

solution函数流程如图5-3所示。

#### judge函数 该函数的功能：对测试的成绩进行评价，用参数正确率，总题数和正确题数来进行判断，对于不同的正确率，输出不同的评价。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | judge | 对测试的成绩输出评价 | 需要一个小数来行的数据表示正确率，两个整形数据分别表示题目总量和正确的题目数量 |

### 相关数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | difficulty | number | answer | difficulty |
| 数据类型 | int | int | int/double | int |
| 数据说明 | 用户输入整数：1表示选择易，2表示选择中，3表示选择难 | 用户输入正整数自定义题量 | 用户输入答案 | multiplyDemo参数：1表示易，2表示中，3表示难(小数)，4表示难（分数） |
| 数据名称 | correct | num | right | Result |
| 数据类型 | double | int | int | int |
| 数据说明 | 表示正确率 | 表示该次测试的题目总量 | 表示该次测试当中正确的题目数量 | 1表示正确，0表示错误 |

## 除*法自测模块*

### 功能说明

用户首先选择自测模块，选择好了以后将进行难度选择，并进行题量的自定义，在做题过程中，如果正确，将会提示正确并进入下一题，如果错误会显示正确答案，并提醒用户是否要保存到错题集当中，在用户输入指令后进入下一题。

当答题结束时，会跳出结束界面，展示正确率和评价。

### 函数、方法设计

#### divide函数 该函数的功能：实现除法自测功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | divide | 实现难度选择、自定义题量，并调用divideDemo函数产生题目，对题目进行正误判断 | 无 |

#### divideDemo函数 该函数的功能：函数接受用户输入的困难度参数后，生成对应难度的乘法题目，打印给用户，用户输入答案后，判断正误，并给出返回值传到divide函数中。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| int | divideDemo | 产生除法运算式并判断答案是否正确，返回一个整数值说明正误 | 由一个整形参数表示难易程度 |

### divideDemo函数流程图如图5-2所示：



### 相关数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据名称 | difficulty | number | answer | difficulty | divideDemo返回值 |
| 数据类型 | int | int | int/double | int | int |
| 数据说明 | 用户输入整数：1表示选择易，2表示选择中，3表示选择难 | 用户输入正整数自定义题量 | 用户输入答案 | divideDemo参数：1表示易，2表示中，3表示难 | 1表示正确，0表示错误 |

#### solution函数 该函数的功能：对divideDemo函数的返回值进行处理，如果返回值表示正确，则输出提示答案正确，反之提示答案错误，输出正确答案，并提示用户将其保存到错题本中

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | solution | 对答案正误判断的结果进行处理 | 接受result参数，为1 表示正确，为0表示错误 |

solution函数流程如图5-3所示。

#### judge函数 该函数的功能：对测试的成绩进行评价，用参数正确率，总题数和正确题数来进行判断，对于不同的正确率，输出不同的评价。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | judge | 对测试的成绩输出评价 | 需要一个小数来行的数据表示正确率，两个整形数据分别表示题目总量和正确的题目数量 |

### 相关数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | difficulty | number | answer | difficulty |
| 数据类型 | int | int | int/double | int |
| 数据说明 | 用户输入整数：1表示选择易，2表示选择中，3表示选择难 | 用户输入正整数自定义题量 | 用户输入答案 | divideDemo参数：1表示易，2表示中，3表示难（小数）.4表示难（分数） |
| 数据名称 | correct | num | right | Result |
| 数据类型 | double | int | int | int |
| 数据说明 | 表示正确率 | 表示该次测试的题目总量 | 表示该次测试当中正确的题目数量 | 1表示正确，0表示错误 |

## 学生菜单模块

### 功能说明

用户以学生身份登录后可以通过此菜单模块使用各项功能操作。主菜单由menuS函数打印，读入1调用menuS\_base函数打印基础测试菜单，进行加减乘除或返回操作，读入2进行综合自测，读入3进入游戏模式，读入4调用 menuS\_inquiry函数，打印查询菜单，查看各种个人信息，。

### 函数、方法设计

menuS,menuS\_base,menuS\_inquiry函数的流程图5-6如图所示



#### menuS函数 该函数的功能：显示给学生选择各项功能的菜单。读入用户的选择序号，调用对应的函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | menuS | 显示学生功能菜单 | 无 |

#### 相关数据表

|  |  |
| --- | --- |
| 数据名称 | choice |
| 数据类型 | int |
| 数据说明 | 用户输入整数：1表示基础测试，2表示综合测试，3表示游戏模式，4表示小测，5表示查询，0表示退出系统。 |

#### menuS\_base函数 该函数的功能：显示给学生选择基础测试功能的菜单。读入用户的选择序号，调用对应的函数，读入1进行加法自测，读入2进行减法自测，读入3进行乘法自测，读入4进行除法自测，读入0返回主菜单。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | menuS\_base | 显示学生基础测试功能菜单 | 无 |

#### 相关数据表

|  |  |
| --- | --- |
| 数据名称 | choice |
| 数据类型 | int |
| 数据说明 | 用户输入整数：1表示加法自测，2表示减法测试，3表示乘法自测，4表示除法自测，0表示返回主菜单。 |

#### menuS\_inquiry函数 该函数的功能：显示给学生选择查询功能的菜单。读入用户的选择序号，调用对应的函数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | menuS\_inquiry | 显示学生查询功能菜单 | 无 |

### 相关数据表

|  |  |
| --- | --- |
| 数据名称 | choice |
| 数据类型 | int |
| 数据说明 | 用户输入整数：1表示查看错题集，2表示查看成绩记录，3表示查看老师评语0表示返回主菜单。 |

## 老师菜单模块

### 功能说明

用户以老师身份登录后可以通过此菜单模块使用各项功能操作

### 函数、方法设计

menuT,menuT\_manage函数的流程图5-7如图所示



#### menuT函数 该函数的功能：显示给老师选择功能的菜单。读入用户的选择序号，调用对应的函数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | menuT | 显示老师功能菜单 | 无 |

#### 相关数据表

|  |  |
| --- | --- |
| 数据名称 | choice |
| 数据类型 | int |
| 数据说明 | 用户输入整数：1表示查看学生排名，2表示管理学生信息，3表示添加评语，4表示修改密码， 0表示退出系统。 |

#### menuT\_manage函数 该函数的功能：显示给老师选择管理学生信息功能的菜单。读入用户的选择序号，调用对应的函数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | menuT\_manage | 显示老师管理学生信息功能菜单 | 无 |

#### 相关数据表

|  |  |
| --- | --- |
| 数据名称 | choice |
| 数据类型 | int |
| 数据说明 | 用户输入整数：1表示添加学生，2表示移除学生，3表示修改学生信息，4查询学生信息， 0表示返回主菜单。 |

#### addStudent函数 该函数的功能：老师选择添加注册学生到班级文件。

进入函数，老师输入想要添加的学生id,若该id对应的学生已经存在，则给出提示信息，等待用户

下一步指令，添加学生成功后，提醒用户继续添加或者返回上级菜单，最终退出函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| pHead | addStudent | 老师添加学生进入班级名单，该学生可以登录使用系统 | pHead |

addStudent函数流程图如图5-8所示



#### 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据名称 | id | fp | pHead |
| 数据类型 | char[] | FILE\* | Struct student\* |
| 数据说明 | 老师通过新增学生id，将学生加入班级名单，同时在链表中用来记录该学生信息的结构体链表节点 | 用于指向班级名单对应的文件 | 结构体头指针，用来遍历使用学生链表 |

#### deleteStudent函数 该函数的功能：老师选择移除注销学生。进入函数，老师输入想要移除的学生id,若该id对应的学生不存在，则给出提示信息，等待用户下一步指令，移除学生成功后，提醒用户继续添加或者返回上级菜单，最终退出函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| pHead | deleteStudent | 老师删除班级名单内的某个学生 | pHead |

deleteStudent函数流程图如图5-9所示



#### 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据名称 | id | fp | pHead |
| 数据类型 | char[] | FILE\* | Struct student\* |
| 数据说明 | 老师通过输入学生id，将该学生移除班级名单，同时在链表中该学生的结构体链表节点被删除，该学生不能再使用本系统 | 用于指向班级名单对应的文件 | 结构体头指针，用来遍历使用学生链表 |

#### modifyStudent函数 该函数的功能：老师选择修改某个学生的信息。进入函数，老师输入想要修改的学生id,若该id对应的学生不存在，则给出提示信息，等待用户下一步指令，让用户输入要修改的具体信息，确认是否修改，修改成功后，提醒用户继续修改或者返回上级菜单，最终退出函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| pHead | modifyStudent | 老师修改某个学生的基本信息 | pHead |

modifyStudent函数流程图如图5-10所示



#### 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据名称 | id | fp | pHead |
| 数据类型 | char[] | FILE\* | Struct student\* |
| 数据说明 | 老师通过输入学生id，从而修改该学生的基本信息。 | 用于指向班级名单对应的文件 | 结构体头指针，用来遍历使用学生链表 |

#### checkStudent函数 该函数的功能：老师选择查询具体学生的信息。进入函数，老师输入想要查看信息的学生id,若该id对应的学生不存在，则给出提示信息，等待用户下一步指令，查看学生成功后，提醒用户继续添加或者返回上级菜单，最终退出函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| pHead | checkStudent | 老师查询某个学生的基本信息 | pHead |

checkStudent函数流程图如图5-11所示



#### 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据名称 | id | fp | pHead |
| 数据类型 | char[] | FILE\* | Struct student\* |
| 数据说明 | 老师通过输入学生id，从而查看该学生的基本信息。 | 用于指向班级名单对应的文件 | 结构体头指针，用来遍历使用学生链表 |

## 成绩查询

### 功能说明

### 该部分分为学生查询自己成绩与老师查询排名后的学生成绩两个部分。

该系统主要面向学生与老师，而成绩对学生与老师都有着重要的意义，学生可以通过成绩调整自己的学习，老师可以通过查看学生成绩来调整教学，因此需要一个成绩查询的功能。

学生登录后，可以选择查看自己的成绩，系统将该学生的信息打印出来即可。

老师登录后，可以选择查看排名及成绩，系统先将学生成绩进行排序，然后按顺序输出学生的名次，姓名，学号及成绩

### 函数设计

实现该功能模块主要涉及学生查看自己成绩函数与老师查看学生成绩及排名函数，如图5-12所示

**

#### 学生查看自己成绩 功能：学生登录后，实现学生查看自己的成绩的功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | viewGrade | 学生查看自己的成绩 | 该函数不接收参数 |

#### 老师查看学生成绩及排名 功能：老师登录后，实现老师查看学生成绩及排名的功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void | rank | 老师查看学生成绩及排名 | 该函数不接收参数 |

#### 数组排序函数 功能：实现对数组排序

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void | sortForArray | 实现对数组排序 | double a[]待排序的数组,int n这个数组的元素个数 |

#### 统计学生人数函数 功能：实现统计链表中学生总人数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| Int型变量，代表学生总数 | sumOfStu | 统计链表中学生总人数 | 该函数不接收参数 |

#### 打印学生信息函数 功能：实现根据分数打印该分数学生的信息功能

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void | printNameAndId | 根据分数打印该分数学生的信息，如姓名，学号，成绩等 | double n，n代表分数 |

### 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据名称 | scores | testGrade | SumNumberForStu |
| 数据类型 | double[] | double | int |
| 数据说明 | 学生成绩数组 | 小测平均成绩 | 学生的总人数 |

|  |  |
| --- | --- |
| 数据名称 | chioce |
| 数据类型 | int |
| 数据说明 | 用户输入选择，为1时退出该功能 |

## 学生评价

### 功能说明

### 该部分分为学生查看老师对自己的评价与老师对学生评价两个部分。

该系统主要面向学生与老师，而评价对学生有较大的意义，学生可以通过老师对自己的评价调整自己的学习，因此需要一个评价的功能。

学生登录后，可以选择查看老师对自己的评价，系统将该学生的老师对他的评价打印出来即可。

老师登录后，可以选择对某位同学评价，评价完后系统将保存评价

### 函数、方法设计

实现该功能模块涉及老师评价函数与学生查看老师评价函数，如图5-13所示

**

#### 老师评价 功能：允许老师对指定学生进行评价。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void | evaluate | 允许老师对指定学生进行评价 | 该函数不接收参数 |

#### 学生查看评价 功能：允许学生查看老师对自己的评价。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void | evaluation | 允许学生查看评价 | 该函数不接收参数 |

### 相关数据表

|  |  |  |
| --- | --- | --- |
| 数据名称 | review | comment |
| 数据类型 | char[] | char[] |
| 数据说明 | 存储老师新的评价 | 学生结构体中老师的评价 |

## 休息模块

### 功能说明

该功能主要面向学生，劳逸结合有助于提高学生的效率，而学生一般不知道自己学习了的时间，因此需要一提醒学生休息的功能。

学生进入加减乘除练习模块后，加减乘除各模块分别都在开始时创建一个新的线程来计时，每隔一定的时间，提醒学生休息，并显示学生累计学习时间。

### 函数设计

实现该功能模块涉及计时函数与多线程的使用，计时函数完成时，系统提示休息，如图5-14所示

**

#### 计时函数 该函数的功能：根据参数值确定计时时间并每隔一定时间提醒学生休息。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void\* | timer | 根据参数值确定计时时间并每隔一定时间提醒学生休息 | void \*a，a所指的值为计时时间 |

### 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据名称 | startTime | endTime | n |
| 数据类型 | double | double | int\* |
| 数据说明 | 记录开始时挂钟时间 | 记录结束时挂钟时间 | n所指向的数为计时时间 |

## 错题集

### 功能说明

该功能主要面向学生，学生再次做自己做过得错题会很大程度提高学生的学习效率，因此需要保存学生错题的功能并允许学生查看自己的错题的功能。

当学生允许保存错题时，保存错题功能将学生的错题题干与答案均保存到该学生的错题本中去。

当学生登录后，选择查看错题功能，系统将读取该学生的错题本并将所有错题打印出来。

### 函数设计

实现该功能模块涉及错题保存函数，查看错题函数，如图5-15所示

**

#### 保存错题 该函数的功能：保存错题到该学生的错题本上。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void | addMistake | 保存错题到该学生的错题本上 | char x[20]该参数为学生做错的题目与答案 |

#### 查看错题 该函数的功能：学生可以查看他的错题。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 函数名 | 功能 | 参数说明 |
| void | printMistake | 学生可以查看他的所有错题 | 该函数不接收参数 |

### 相关数据表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据名称 | strOfMistake | mistakeFileName | fout | fin | strings | size |
| 数据类型 | char\* | char\* | FILE\* | FILE\* | char[][] | int |
| 数据说明 | 学生做错的题目 | 该学生错题本文件的地址 | 该学生文件指针 | 该学生文件指针 | 储存该学生的所有错题 | 统计错题量 |

## *5.12综合自测模块*

### 5.12.1 功能说明

学生首先选择综合自测模式，提示输入自定义题量，若输入为正整数，则调用出题函数comTestdemo按题量出题，否则提示重新输入。在做题过程中，如果正确，将会提示正确并进入下一题，如果错误会显示正确答案，并提醒用户是否要保存到错题集当中，在用户输入指令后进入下一题。

当答题结束时，会跳出结束界面，调用judge函数展示正确率和评价。

### 5.12.2函数、方法设计

#### comTest函数

#### 该函数的功能：学生进行综合自测，并选择是否保存错题，是将结果更新至学生错题文件中并返回菜单，否则直接返回菜单。。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | comTest | 学生进行综合自测，将结果更新至学生信息 | （void）无形参 |

**comTestDemo函数**

该函数的功能：综合测试的出题函数

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| int | comTestDemo | 综合自测的出题函数 | (int part , int difficulty，)part，diffculty为两个随机数，随机决定测试部分和难度。part为1-5，diffculty为1-3 |

comTest，comTestDemo函数流程图如图5-16所示：



#### 相关数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | flag | number | difficulty | part |
| 数据类型 | int | int | int | int |
| 数据说明 | 0代表错误，1代表正确 | 学生输入定义的题量 | 作为1-3的随机数标记出题难度 | 作为1-5的随机数编辑测试的部分 |

## *5.13游戏模块*

### 5.13.1功能说明

学生首先选择游戏模式，然后选择无尽模式或闯关模式，无尽模式系统循环出题，直到答错一道题后失败，打印游戏结果，错题信息并选择保存到错题本。游戏成绩若超过历史记录，则更新记录。

### 5.13.2函数、方法设计

### game函数

#### 该函数的功能：学生进行算数游戏，分为无尽模式和闯关模式。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | game | 学生进行算数游戏，分为无尽模式和闯关模式 | Void(无参数) |

game函数流程图如图5-17所示：



#### 5.13.2.1数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | gameNume | gameLevel | gameNum | flag |
| 数据类型 | int | int | int | int |
| 数据说明 | 学生在无尽模式中最高答题数 | 闯过的关卡数 | 出错为止正确的题目数 | 记录题目正误，0-错，1-对 |

## *5.14小测模块*

### 5.14.1 功能说明

学生选择的小测模块，调用comTestDemo函数出20题 ，全部作答后给出评价请选择是否保存错题，最后将小测成绩上传。

### 5.14.2函数、方法设计

#### test 函数

test函数流程图如图5-18所示：



#### 该函数的功能：学生进行加减乘除综合小测，小测成绩作为排名依据。

|  |  |  |  |
| --- | --- | --- | --- |
| 返回值 | 方法名 | 功能 | 参数说明 |
| void | test | 学生进行加减乘除综合小测，小测成绩作为排名依据。 | (Student\* p)为学生所在链表的头指针 |

### 5.14.3相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据名称 | starTime | endTime | flag |
| 数据类型 | double | double | int |
| 数据说明 | 小测开始时的时间 | 计算结束的时间 | 记录题目正误，0-错，1-对 |

# 数据结构设计

## 功能说明

学生结构体储存学生相关基本信息及成绩等，采用链表实现对学生的增删查改

老师结构体则主要储存老师基本信息。

## 数据表设计

### *表名：学生结构体数据表*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **函数型** | **可为空** | **默认** | **注释** |
| name | Char[] | 不可 | “ ” |  |
| id | Char[] | 不可 | “ ” |  |
| pin | Char[] | 不可 | “ ” |  |
| plusCorrect | double | 可 | 0 | 加法自测正确率 |
| plusNum | int | 可 | 0 | 加法自测做题数 |
| minusCorrect | double | 可 | 0 |  |
| minusNum | int | 可 | 0 |  |
| multiplyCorrect | double | 可 | 0 |  |
| multiplyNum | int | 可 | 0 |  |
| divideCorrect | double | 可 | 0 |  |
| divideNum | int | 可 | 0 |  |
| testGrade | double | 可 | 0 | 小测平均成绩 |
| testNum | int | 可 | 0 | 已做小测次数 |
| mistakeFileName | Char[] | 可 | “ ” | 错题本文件名 |
| mistakeNum | int | 可 | 0 | 错题数 |
| gameLevel | int | 不可 | 1 | 游戏闯关模式当前关卡 |
| gameNum | int | 可 | 0 | 游戏无尽模式当前最大题量 |
| grade | int[] | 可 | 0 | 历史成绩 |
| comment | char[] | 可 | “ ” | 老师评价 |
| next | struct student\* | 可 | NULL |  |

### 具体代码实现如下：

### typedef struct student {

### char name[20];

### char id[10];

### char pin[20];

### double plusCorrect;

### int plusNum;

### double minusCorrect;

### int minusNum;

### double multiplyCorrect;

### int multiplyNum;

### double divideCorrect;

### int divideNum;

### double testGrade;

### int testNum;

### char mistakeFileName[70];

### int mistakeNum;

### int gameLevel;

### int gameNum;

### int grade[20];

### char comment[70];

### struct student\* next;

### } St,\*PSt;

### *表名：老师结构体数据表*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **函数型** | **可为空** | **默认** | **注释** |
| name | Char[] | 不可 | “ ” |  |
| id | Char[] | 不可 | “ ” |  |
| pin | Char[] | 不可 | “ ” |  |

老师结构体代码实现如下：

typedef struct teacher {

char name[20];

char id[10];

char pin[20];

} Te;