**数据结构课程设计报告**

小组成员：2017025323 殷建平

2017025325 李细军

课设时间：2019年2月18日～2019年2月24日

2019年2月27日

|  |  |
| --- | --- |
| **题目** | 学生成绩管理系统的设计与实现 |
| **编译环境** | Microsoft Visual Studio 2019 Preview |
| **设**  **计**  **内**  **容**  **和**  **设**  **计**  **要**  **求** | 1. 每一条记录包括一个学生的学号、姓名、3门课成绩、平均成绩（结构体数组或链表）。 2. 输入功能：可以一次完成若干条记录的输入。建议数据采用一次性导入。 3. 显示功能：完成全部学生记录的显示。 4. 查找功能：完成按姓名或学号查找学生记录，并显示该学生基本信息。 5. 排序功能：按学生平均成绩进行排序。 6. 插入功能按平均成绩高低插入一条学生记录（有序表的插入）。 7. 将学生记录存在文件score中。 8. 提供一个界面来调动功能，界面和各个功能的操作界面尽可能清晰美观。 |
| **程 序 流 程 图** | |
|  | |

|  |
| --- |
| **子程序（函数）和变量说明** |
| Main\_menu():主菜单函数；  Stu\_info\_input():学生信息的录入；  结构体数组定义：typedef struct  {  char id[20];  char name[2000];  int score[3];  double ave;  }STUDENT;  Stu\_info\_show():学生信息全部显示；  Stu\_info\_sort():学生信息排序；  STUDENT t，插入排序中为了交换学生信息所引入的中间变量。  Stu\_info\_search\_name():通过姓名查找某学生信息；  Int match=0：查找遍历过程中用来指示位置的变量。  Stu\_info\_search\_number():通过学生学号查找某学生信息；  Int match=0：查找遍历过程中用来指示位置的变量。  Stu\_info\_save():学生信息保存；  Stu\_input\_ordered():学生信息的顺序表插入。  STUDENT temp：插入排序过程中为了交换学生信息所引入的中间变量。 |
| **源代码中存在的不足** |
| 在本程序中，由于时间紧迫，设计仓促，学生的信息只能通过键盘一条一条的输入，没有实现通过csv格式的文件直接导入，造成在实际使用过程中效率较低，输入过程较为繁琐。此外，各功能的菜单也较为简陋，仅有最基本的文字信息。以上就是本程序中存在不足，在未来值得改进的地方。 |
| **程 序 源 代 码** |
| #include <stdio.h>  #include <string.h>  #include <stdlib.h>  void main\_menu(void); //菜单函数  void stu\_info\_input(void); //信息输入函数  void stu\_info\_show(void); //信息显示函数  void stu\_info\_sort(void); //平均分排序  void stu\_info\_search\_name(void); //根据姓名查找  void stu\_info\_search\_number(void); //根据学号查找  void stu\_info\_save(void); //信息保存  void stu\_info\_input\_ordered(void); //顺序表插入  void input\_done(void);  //主函数  int main()  {  printf("Welcome!\n");  main\_menu();  return 0;  }  //结构体定义  typedef struct  {  char id[20];  char name[2000];  int score[3];  double ave;  }STUDENT;  STUDENT stu[5000];  int count = 0;  //菜单函数  void main\_menu()  {  printf("Please input your choose:\n");  printf("1\ Add stu\_info.\n");  printf("2\ Search stu\_info by stu\_number.\n");  printf("3\ Search stu\_info by stu\_name.\n");  printf("4\ Show all stu\_info.\n");  printf("5\ Save all stu\_info.\n");  printf("6\ Sort stu\_info.\n");  printf("7\ Input stu\_info by order.\n");  printf("Input your choose:\n");  int chs;  scanf\_s("%d", &chs);  switch (chs)  {  case 1:stu\_info\_input(); main\_menu(); break;  case 2:stu\_info\_search\_number(); main\_menu(); break;  case 3:stu\_info\_search\_name(); main\_menu(); break;  case 4:stu\_info\_show(); printf("\n"); main\_menu(); break;  case 5:stu\_info\_save(); main\_menu(); break;  case 6:stu\_info\_sort(); main\_menu(); break;  case 7:stu\_info\_input\_ordered(); main\_menu(); break;  default:printf("Please input the correct number!"); main\_menu();  }  }  void input\_done()  {  stu\_info\_sort();  }  //学生信息输入函数  void stu\_info\_input(void)  {  printf("Please input stu\_ID:");  scanf\_s("%s", &stu[count].id, sizeof(&stu[count].id));  printf("Please input stu\_name:");  scanf\_s("%s", &stu[count].name, sizeof(stu[count].name));  printf("Please input stu\_C++ score:");  scanf\_s("%d", &stu[count].score[0]);  printf("Please input stu\_PYTHON score:");  scanf\_s("%d", &stu[count].score[1]);  printf("Please input stu\_JAVA score:");  scanf\_s("%d", &stu[count].score[2]);  stu[count].ave = (stu[count].score[0] + stu[count].score[1] + stu[count].score[2])/3;  count++;  printf("Sucessfully input!");  printf("\n");  }  //学生信息打印  void stu\_info\_show(void)  {  printf("ID\tName\t\tScore\_CPP\tScore\_PYTHON\tScore\_JAVA\t\n");  for (int m = 0; m < count; m++)  {  printf("%s\t%s\t\t%d\t%d\t\t%d\t%f\n", stu[m].id, stu[m].name, stu[m].score[0], stu[m].score[1], stu[m].score[2],stu[m].ave);  }  printf("Successfully display!");  printf("\n");  }  //学生平均分排序  void stu\_info\_sort(void)  {  int i, j;  STUDENT t;  for (j = 0; j < count - 1; j++)  for (i = j + 1; i < count; i++)  if (stu[i].ave < stu[j].ave)  {  t = stu[i];  stu[i] = stu[j];  stu[j] = t;  }  }  //按姓名查找  void stu\_info\_search\_name(void)  {  printf("Please input the name you want:");  char name\_want[20];  scanf\_s("%s", &name\_want,sizeof(name\_want));  int match = 0;  for (int i = 0; i <= count; i++)  {  if (strcmp(stu[i].name, name\_want)==0)  {  match++;  printf("ID\tName\t\tScore\_CPP\tScore\_PYTHON\tScore\_JAVA\t\n");  printf("%s\t%s\t\t%d\t%d\t\t%d\t%f\n", stu[i].id, stu[i].name, stu[i].score[0], stu[i].score[1], stu[i].score[2], stu[i].ave);  }  }  if (match == 0)  {  printf("Not found!");  }  printf("\n");  }  //按学号查找  void stu\_info\_search\_number(void)  {  printf("Please input the number you want:");  char number\_want[20];  scanf\_s("%s", &number\_want, sizeof(number\_want));  int match = 0;  for (int i = 0; i <= count; i++)  {  if (strcmp(stu[i].id, number\_want) == 0)  {  match++;  printf("ID\tName\t\tScore\_CPP\tScore\_PYTHON\tScore\_JAVA\t\n");  printf("%s\t%s\t\t%d\t%d\t\t%d\t%f\n", stu[i].id, stu[i].name, stu[i].score[0], stu[i].score[1], stu[i].score[2], stu[i].ave);  }  }  if (match == 0)  {  printf("Not found!");  }  printf("\n");  }  //文件保存  void stu\_info\_save(void)  {  FILE\* fp;  int i, d;  fp = fopen ("D://test.txt", "w");  if (fp == NULL)  {  printf("File cannot open!");  exit(0);  }  else  {  for (i = 0; i <= count; i++)  {  fprintf\_s(fp, "%s\t%s\t\t%d\t%d\t\t%d\t%f\n", stu[i].id, stu[i].name, stu[i].score[0], stu[i].score[1], stu[i].score[2], stu[i].ave);  }  printf("Successfully saved!");  printf("\n");  }  }  //顺序表插入学生信息  void stu\_info\_input\_ordered(void)  {  stu\_info\_sort();  printf("Please input stu\_ID:");  scanf\_s("%s", &stu[count].id, sizeof(&stu[count].id));  printf("Please input stu\_name:");  scanf\_s("%s", &stu[count].name, sizeof(stu[count].name));  printf("Please input stu\_C++ score:");  scanf\_s("%d", &stu[count].score[0]);  printf("Please input stu\_PYTHON score:");  scanf\_s("%d", &stu[count].score[1]);  printf("Please input stu\_JAVA score:");  scanf\_s("%d", &stu[count].score[2]);  stu[count].ave = (stu[count].score[0] + stu[count].score[1] + stu[count].score[2]) / 3;  count++;  STUDENT temp;  temp = stu[count-1];  //printf("%s\t%s\t\t%d\t%d\t\t%d\t%f\n", stu[count-1].id, stu[count-1].name, stu[count-1].score[0], stu[count-1].score[1], stu[count-1].score[2], stu[count-1].ave);  int m;  for (m = 1; m <= count - 2; m++)  {  if (stu[count-1].ave >= stu[m - 1].ave && stu[count-1].ave < stu[m].ave)  {  break;  }  }  int n;  for (n = count-1; n >=m+1; n--)  {  stu[n] = stu[n - 1];  }  stu[m] = temp;  } |