**分组:\_\_\_\_\_\_\_9\_\_\_\_\_**

****

信息科学与工程学院课程实验报告

《面向对象程序设计》

|  |  |
| --- | --- |
| 姓名： | 王振浩 |
| 学号： | 201711010213 |
| 班级： | 计工本1701 |
| 教师： | 张庆科 |
| 时间： | 2018-09-20 |

**面向对象程序设计实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 王振浩 | 班级 | 计工本1701 | 学号 | 201711010213 | 组号 | 9 |
| 时间 | 2018-09-20 | 地点 | ？？？ | 周次 | 4 | 页码 |  |
| 源码 | □ 无源码 □ 文档源码 □ 托管源码 | | | | | | |
| 报  告  内  容  报  告  内  容  报  告  内  容 | **实验报告要求**：请围绕实验目的、实验内容、实验过程及步骤(可添加文字、矢量图)、实验结论与分析进行撰写，凡涉及源代码内容可给出完整源码或附上源码托管网址。   1. 实验目的：熟悉vs编程环境，掌握多文件编程方法，熟悉github作业提交流程。 2. 实验内容：   设计一个简单的`学生成绩管理系统`来管理若干学生的信息，每个学生的信息包括学号、姓名、某门课的平时成绩、实验成绩、期末成绩、总成绩和排名，其中总成绩=0.2\\*平时成绩+0.2\\*实验成绩+0.6\\*期末成绩。系统的基本需求如下：  ```  1） 从文件data.txt中读取所有学生成绩信息。  2） 根据学生成绩信息计算出学生成绩排名。  3） 输入学号可查询某个学生的成绩信息。  4） 统计出该课程成绩的均值和方差，并对成绩分布简要分析。  3.实验过程及步骤：  1）读取txt文档  2）计算排名情况  3）输入查询学生学号，输出其各个信息  4）计算成绩均值和方差  **成绩管理系统.cpp：**  //# include <iostream>  //# include <string>  //# include <fstream>  //# include <stdlib>  #include <stdio.h>  # include "math.h"  # include <string.h>  #include <stdlib.h>  #include <windows.h>  //#include "score.h"  typedef struct  {  char id[10];  char name[20];  double fscore; //平时成绩  double escore; //实验成绩  double lscore; //期末成绩  double tscore; //总成绩  }SS;  //读入学生成绩信息  //从文件中读取学生信息  SS\*Readfile(int \*N)  {  SS \*stu; //定义SS类型的指针stu  FILE \*fp = NULL;//定义一个指向文件的指针  int count ;  int index = 0;  fp = fopen("data.txt", "r"); //用fopen以只读的方式打开文件data，并让指针fp指向文件  if (fp = NULL)  {  printf("error"); //如果指针fp指向空，那么文件打开失败  getchar();  }  else  {  fscanf(fp, "%d", &count); //文件成功打开，则将指针指向的第一个正数赋值给变量count  \*N = count;  }  printf("学生的人数为：%d\n", count);  getchar();  stu = (SS\*) malloc(count \* sizeof(SS));//动态分配空间  //读取每个同学的信息  while ((!feof(fp)) && (index < count))  {  //读入文件数据到内存  fscanf(fp, "%s%s%lf%lf%lf", (stu[index].id, stu[index].name, stu[index].fscore, stu[index].escore, stu[index].lscore));  //输出读入内存的学生信息  printf("学号：%s 姓名：%s 平时成绩：%lf 实验成绩：%lf 期末成绩：%lf", stu[index].id, stu[index].name, stu[index].fscore, stu[index].escore, stu[index].lscore);  index++;  }  fclose(fp);  return stu;  }  //计算学生的总成绩  void Lscore (SS stu[], int N)  {  int i = 0;  // float p = 0.2;  //float q = 0.6;  for (i = 0; i < N; i++)  {  stu[i].tscore = 0.2 \*stu[i].fscore + 0.2\*stu[i].escore + 0.6\*stu[i].lscore;  printf("学号：%s 姓名：%s 总成绩：%4.2lf", stu[i].id, stu[i].name, stu[i].tscore);  }  printf("学生总成绩计算完毕\n\n");  }  int Sort(SS stu[], int N)  {  int i = 0;  int j = 0;  int k = 0;  for (i = 0; i < N; i++)  for (j = i + 1; j < N; j++)  {  if (stu[i].tscore < stu[j].tscore)  {  k = i;    strcpy(stu[i].id, stu[j].id);  strcpy(stu[i].name , stu[j].name);  stu[i].fscore = stu[j].fscore;  stu[i].escore = stu[j].escore;  stu[i].lscore = stu[j].lscore;  stu[i].tscore = stu[j].tscore;  strcpy(stu[j].id, stu[k].id);  strcpy(stu[j].name, stu[k].name);  stu[j].fscore = stu[k].fscore;  stu[j].escore = stu[k].escore;  stu[j].lscore = stu[k].lscore;  stu[j].tscore = stu[k].tscore;  }  }  for (i = 0; i < N; i++)  {  printf("学号：%s 姓名：%s 平时成绩：%lf 实验成绩：%lf 期末成绩：%lf 总成绩：%4.2lf \n ", stu[i].id, stu[i].name, stu[i].fscore, stu[i].escore, stu[i].lscore, stu[i].tscore);  }  return 0;  }  int main()  {  int N = 0;  SS \*pstu = NULL;  pstu = Readfile(&N);  Lscore(pstu, N);  Sort(pstu, N);  free(pstu);  system("pause"); //停顿成绩作用  return 0;  }  **实验结论：**  **通过这次小实验初步了解熟悉了VS编程环境**，**掌握多文件编程方法。可以给编程带来极大的方便但这次实验对OOP了解还是比较初步，程序也比较依赖POP知识还需进一步学习。** | | | | | | |

：可根据内容自行拓展页面