**分组:\_\_\_\_\_\_\_\_\_**

****

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： | 刘陶然 |
| 学号： | 201711010425 |
| 班级： | 计工本2班 |
| 教师： | 张庆科 |
| 时间： | 2018-9-19 |

**面向对象程序设计实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 刘陶然 | 班级 | 计工本2班 | 学号 | 201711010425 | 组号 | 15 |
| 时间 | 2018-9-19 | 地点 | 信息楼E312 | 周次 | 3 | 页码 | 共3页 |
| 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。    源代码：  //\*\*\*\*\*\* 第一次作业 \*\*\*\*\*\*//  /\*制作一个简单的成绩管理系统\*/  #include<iostream>  #include<cstdio>  #include<cmath>  #include<fstream>  #include"score\_administration.h"  #include<Windows.h>  using namespace std;  int main()  {  string filename;  filename = "data.txt"; //文件路径  read\_txt(filename); //把文件读取到结构体里面    calculate\_all\_students\_final\_score(); //计算所有学生的最终成绩...  all\_students\_rank(); //对所有学生进行排名  printf("输入0 ：推出成绩管理系统\n输入1 ：全部成绩预览\n输入2 ：查看整体均值/方差与成绩分析\n输入3 ：按学号查询学生信息\n");  while (1) {  printf("\n请您输入指令：\n\n");  int n = 0;  cin >> n;  if (n == 0) break;  switch (n)  {  case 1:{  information\_view(); //成绩预览  break;  }  case 2:{  calculate\_mean\_variance\_and\_analyze(); //计算均值和方差并做出分析...  break;  }  case 3: {  string n\_2;  while (1) {  printf("输入 000 ：结束查询\n输入学号进行查找\n");  cin >> n\_2;  if (n\_2 == string("000")) break;  else {  search\_student(n\_2);  }  }  break;  }  default: printf("您输入的指令无效，请重新输入 : \n");  }  }    printf("谢谢您的使用\n\n");  system("pause");  return 0;  }  头文件代码：  #include<iostream>  #include<cstdio>  #include<fstream>  #include<string>  #include<cmath>  #include<algorithm>  using namespace std;  int num; //学生人数  struct node\_info  {  char StudentID[20]; //学号  char Name[20]; //姓名  int common\_score; //平时成绩  int experiment\_score; //实验成绩  int final\_score; //期末成绩  double score; //最终成绩  int rank; //名次  }Student[100];  int str\_int(string str) { //将字符串类型转换成整形类型  int str\_num = str.length();  int sum = 0;  for (int i = 0; i < str\_num; i++) {  sum \*= 10;  sum += str[i] - '0';  }  return sum;  }  void read\_txt(string filename)  {  string temp;  ifstream readFile;  readFile.open(filename);  if (!readFile) { //判断文件是否可以正常打开  cout << "Cannot open the file\n" << endl;  return;  }  //小知识 ： 如果可以正常打开：：getline(str, 100, 0) 可以输出空格，遇到delim符号才截止。 最后一个参数0表示文本框遇到空字符（ASCLL码为32，文本框不可能有空字符）截止符。不加第三个参数0时，表示'\n'为截止符('\n'也是换行符)。  getline(readFile, temp);  num = temp[0] - '0'; //读取第一行，人数  for (int i = 1; i <= num; i++){ //将学生信息读入到结构体里面  for (int j = 1; j <= 5; j++) {  if (j == 1) readFile >> Student[i].StudentID;  if (j == 2) readFile >> Student[i].Name;  if (j == 3) {  readFile >> temp;  Student[i].common\_score = str\_int(temp);  }  if (j == 4) {  readFile >> temp;  Student[i].experiment\_score = str\_int(temp);  }  if (j == 5) {  readFile >> temp;  Student[i].final\_score = str\_int(temp);  }  }  }  readFile.close(); //关闭文件  }  void information\_view() //成绩预览情况预览  {  cout << "总人数 ： " << num << endl;  cout << " 学号 姓名 平时成绩 实验成绩 期末成绩 最终成绩 排名" << endl;  for (int i = 1; i <= num; i++) {  printf("%15s, %10s, %10d, %10d, %10d, %13.2f, %10d\n", Student[i].StudentID, Student[i].Name, Student[i].common\_score, Student[i].experiment\_score, Student[i].final\_score, Student[i].score, Student[i].rank);  //cout << Student[i].StudentID << " " << Student[i].Name << " " << Student[i].common\_score << " " << Student[i].experiment\_score << " " << Student[i].final\_score << endl;  }  }  double calculate\_final\_score(node\_info S) //计算每个学生的总成绩  {  double calculate\_score = 0;  calculate\_score += S.common\_score \* 0.2 + S.experiment\_score \* 0.2 + S.final\_score \* 0.6;  return calculate\_score;  }  void calculate\_all\_students\_final\_score() //计算所有学生的最终成绩  {  for (int i = 1; i <= num; i++) {  Student[i].score = calculate\_final\_score(Student[i]);  }  }  void calculate\_mean\_variance\_and\_analyze() //计算均值和方差并做出分析  {  printf("\n\n \*\*\*\*\*\* 所有学生的成绩分布情况 \*\*\*\*\*\*\n\n");  double common\_score\_mean = 0, common\_score\_variance = 0; //平时成绩 均值/方差  double experiment\_score\_mean = 0, experiment\_score\_variance = 0; //实验成绩 均值/方差  double final\_score\_mean = 0, final\_score\_variance = 0; //期末成绩 均值/方差  double score\_mean = 0, score\_variance = 0; //最终成绩 均值/方差  double sum = 0;  for (int i = 1; i <= 4; i++) { // | i == 1 时 计算平时成绩 均值/方差 | i == 2时 计算实验成绩 均值/方差 | ... 依次类推  for (int j = 1; j <= num; j++) {  if (i == 1) sum += Student[j].common\_score; //计算总和  if (i == 2) sum += Student[j].experiment\_score;  if (i == 3) sum += Student[j].final\_score;  if (i == 4) sum += Student[j].score;  }  if (i == 1) {  common\_score\_mean = sum / (1.0 \* num); //计算均值  for (int j = 1; j <= num; j++) {  common\_score\_variance += pow( fabs ( Student[j].common\_score - common\_score\_mean), 2); //计算方差  }  common\_score\_variance /= (1.0 \* num);  }  if (i == 2) {  experiment\_score\_mean = sum / (1.0 \* num);  for (int j = 1; j <= num; j++) {  experiment\_score\_variance += pow(fabs(Student[j].common\_score - experiment\_score\_mean), 2);  }  experiment\_score\_variance /= (1.0 \* num);  }  if (i == 3) {  final\_score\_mean = sum / (1.0 \* num);  for (int j = 1; j <= num; j++) {  final\_score\_variance += pow(fabs(Student[j].common\_score - final\_score\_mean), 2);  }  final\_score\_variance /= (1.0 \* num);  }  if (i == 4) {  score\_mean = sum / (1.0 \* num);  for (int j = 1; j <= num; j++) {  score\_variance += pow(fabs(Student[j].common\_score - score\_mean), 2);  }  score\_variance /= (1.0 \* num);  }  sum = 0; //再初始化为0；  }  cout << " 平时成绩 实验成绩 期末成绩 最终成绩" << endl;  printf("均值：%10.2f %10.2f %10.2f %10.2f\n", common\_score\_mean, experiment\_score\_mean, final\_score\_mean, score\_mean);  printf("方差：%10.2f %10.2f %10.2f %10.2f\n\n\n", common\_score\_variance, experiment\_score\_variance, final\_score\_variance, score\_variance);  //cout << "均值: " << common\_score\_mean << " " << experiment\_score\_mean << " " << final\_score\_mean << " " << score\_mean << endl;  //cout << "方差: " << common\_score\_variance << " " << experiment\_score\_variance << " " << final\_score\_variance << " " << score\_variance << endl;  // 平时成绩 实验成绩 期末成绩 最终成绩  // 均值： 86.50 83.67 92.83 89.73  // 方差： 30.92 38.94 71.03 41.37  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  //分析成绩：  string str;  ifstream readFile\_2; //定义一个读文件对象  readFile\_2.open("analyse.txt"); //打开需要读出的文件  readFile\_2 >> str; //读取分析结果  readFile\_2.close(); //关闭文件  //string str = "\n分析：平时成绩、实验成绩比较稳定，而期末成绩最不平稳，可以看出期末成绩更能反应平时的努力程度，更有区分度。\n 相应的期末成绩的均值很高而方差却很大说明成绩的两级分化比较严重"; //现有分析    int n = 0;  printf("输入0查看现有分析情况：\n输入1修改分析\n");  cin >> n;  if (n == 0) cout << str << endl;  else {  ofstream writeFile\_2; //定义一个写文件对象  writeFile\_2.open("analyse.txt"); //打开需要写入的文件  printf("请修改分析结果：\n");  cin >> str;  writeFile\_2 << str; //讲str的内容写进去  printf("\n修改成功\n");  writeFile\_2.close(); //关闭文件  }  }  int cmp(node\_info A, node\_info B) //定义排名规则  {  if (A.score >= B.score) return 1;  return 0;  }  void all\_students\_rank() //对所有学生进行排名  {  sort(Student + 1, Student + 1 + num, cmp);  for (int i = 1; i <= num; i++) {  Student[i].rank = i;  }  }  void search\_student(string str) //按学号查找同学  {  int flag = 0;  for (int i = 1; i <= num; i++) {  if (Student[i].StudentID == str) {  cout << " 学号 姓名 平时成绩 实验成绩 期末成绩 最终成绩 排名" << endl;  printf("%15s, %10s, %10d, %10d, %10d, %13.2f, %10d\n\n\n", Student[i].StudentID, Student[i].Name, Student[i].common\_score, Student[i].experiment\_score, Student[i].final\_score, Student[i].score, Student[i].rank);  flag = 1;  break;  }  }  if (flag == 0) printf("\n查询不到该同学，可能由于您输入的学号有误，请重新输入：\n");  } | | | | | | |

：可根据内容自行拓展页面