### 辽师

### 《C++ STL》课程设计

学 院：计算机与信息技术学院

专 业：计算机科学与技术

班级序号：

学 号：

学生姓名：

完成时间：

## 实验目的

## 二.完成功能

## 三.设计思路及实验步骤

## 四.实验结果

## 实验总结

#include <iostream>

#include <fstream>

#include <sstream>

#include <vector>

#include <algorithm>

#include <map>

#include <iomanip>

using namespace std;

// 学生结构体

struct Student {

string id;

string name;

string className;

vector<int> scores;

int totalScore;

int classRank;

int schoolRank;

double percentile; // 百分位排名

string gender; // 性别（演示用，简单判断）

string highlight; // 用于Excel高亮标记

};

// 读取数据

vector<Student> readData(const string& filename) {

vector<Student> students;

ifstream infile(filename);

string line;

while (getline(infile, line)) {

stringstream ss(line);

Student stu;

ss >> stu.id >> stu.name >> stu.className;

int score;

stu.scores.clear();

stu.totalScore = 0;

// 假定后面 20 个成绩

for (int i = 0; i < 20; ++i) {

ss >> score;

stu.scores.push\_back(score);

stu.totalScore += score;

}

// 简单性别推测：名字最后一位（仅示例）

string last = stu.name.substr(stu.name.size() - 1, 1);

if (last == "男" || last == "刚" || last == "明") stu.gender = "男";

else if (last == "女" || last == "芳" || last == "娜" || last == "丽" || last == "婷") stu.gender = "女";

else stu.gender = "未知";

stu.highlight = ""; // 默认无标记

students.push\_back(stu);

}

return students;

}

// 班级排名

void classRanking(vector<Student>& students) {

map<string, vector<Student\*>> classes;

for (auto& stu : students) {

classes[stu.className].push\_back(&stu);

}

for (auto& cls : classes) {

auto& stuPtrs = cls.second;

sort(stuPtrs.begin(), stuPtrs.end(),

[](Student\* a, Student\* b) { return a->totalScore > b->totalScore; });

int rank = 1;

for (auto\* stu : stuPtrs) {

stu->classRank = rank++;

}

}

}

// 学校排名及百分位

void schoolRanking(vector<Student>& students) {

vector<Student\*> stuPtrs;

for (auto& stu : students) stuPtrs.push\_back(&stu);

sort(stuPtrs.begin(), stuPtrs.end(),

[](Student\* a, Student\* b) { return a->totalScore > b->totalScore; });

int n = stuPtrs.size();

for (int i = 0; i < n; ++i) {

stuPtrs[i]->schoolRank = i + 1;

stuPtrs[i]->percentile = 100.0 \* (n - i) / n;

}

}

// 生成高亮标记、优秀榜、警示榜

void markAndExportHighlights(vector<Student>& students, int topN = 10, int warningScore = 60) {

// 标记前N名和不及格

for (auto& stu : students) {

if (stu.schoolRank <= topN) stu.highlight = "优秀";

else if (stu.totalScore < warningScore) stu.highlight = "警示";

}

// 导出优秀榜

ofstream outGood("excellent\_list.csv");

outGood << "学号,姓名,班级,总分,学校排名\n";

for (auto& stu : students) {

if (stu.schoolRank <= topN)

outGood << stu.id << "," << stu.name << "," << stu.className << "," << stu.totalScore << "," << stu.schoolRank << "\n";

}

outGood.close();

// 导出警示榜

ofstream outWarn("warning\_list.csv");

outWarn << "学号,姓名,班级,总分,学校排名\n";

for (auto& stu : students) {

if (stu.totalScore < warningScore)

outWarn << stu.id << "," << stu.name << "," << stu.className << "," << stu.totalScore << "," << stu.schoolRank << "\n";

}

outWarn.close();

}

// 导出全体排名

void exportRanking(const vector<Student>& students, const string& filename) {

ofstream outfile(filename);

outfile << "学号,姓名,性别,班级,总分,班级排名,学校排名,百分位,高亮标记\n";

vector<Student> sorted = students;

sort(sorted.begin(), sorted.end(), [](const Student& a, const Student& b) {

return a.schoolRank < b.schoolRank;

});

for (const auto& stu : sorted) {

outfile << stu.id << "," << stu.name << "," << stu.gender << "," << stu.className << ","

<< stu.totalScore << "," << stu.classRank << "," << stu.schoolRank << ","

<< fixed << setprecision(1) << stu.percentile << "%," << stu.highlight << "\n";

}

outfile.close();

}

// 导出班级单独成绩单

void exportClassSheets(const vector<Student>& students) {

map<string, vector<const Student\*>> classMap;

for (const auto& stu : students) classMap[stu.className].push\_back(&stu);

for (const auto& kv : classMap) {

ofstream out("class\_" + kv.first + ".csv");

out << "学号,姓名,班级,总分,班级排名,学校排名,百分位,高亮标记\n";

vector<const Student\*> s = kv.second;

sort(s.begin(), s.end(), [](const Student\* a, const Student\* b) {

return a->classRank < b->classRank;

});

for (auto stu : s) {

out << stu->id << "," << stu->name << "," << stu->className << "," << stu->totalScore << ","

<< stu->classRank << "," << stu->schoolRank << "," << fixed << setprecision(1) << stu->percentile << "%," << stu->highlight << "\n";

}

out.close();

}

}

// 导出统计分析

void exportStats(const vector<Student>& students) {

map<string, vector<int>> classScores;

for (const auto& stu : students)

classScores[stu.className].push\_back(stu.totalScore);

ofstream out("class\_stats.csv");

out << "班级,人数,最高分,最低分,平均分\n";

for (auto& kv : classScores) {

int sum = 0, mx = -1, mn = 1e9;

for (int s : kv.second) {

sum += s;

mx = max(mx, s);

mn = min(mn, s);

}

double avg = sum \* 1.0 / kv.second.size();

out << kv.first << "," << kv.second.size() << "," << mx << "," << mn << "," << fixed << setprecision(2) << avg << "\n";

}

int sum = 0, mx = -1, mn = 1e9;

for (const auto& stu : students) {

sum += stu.totalScore;

mx = max(mx, stu.totalScore);

mn = min(mn, stu.totalScore);

}

double avg = sum \* 1.0 / students.size();

out << "全校," << students.size() << "," << mx << "," << mn << "," << fixed << setprecision(2) << avg << "\n";

out.close();

}

int main() {

vector<Student> students = readData("data.txt");

classRanking(students);

schoolRanking(students);

markAndExportHighlights(students, 10, 60);

exportRanking(students, "school\_ranking\_plus.csv");

exportClassSheets(students);

exportStats(students);

cout << "所有排名、榜单、统计分析已导出！\n"

<< "用Excel打开school\_ranking\_plus.csv等文件，配合条件格式可高亮查看。\n";

return 0;

}