**《程序设计综合实践》**

**课程设计报告**

题 目： 学生信息管理系统

班 级：

学生姓名：

学 号：

**一、系统功能介绍：**

1. 添加学生信息：系统管理员可以手动输入学生的学号、姓名、年龄、性别、出生年月、家庭住址、电话和电子邮件等信息，并将其保存到系统中。

2. 查询学生信息（按学号）：系统管理员和学生都可以根据学号查询学生的详细信息。

3. 查询学生信息（按姓名）：系统管理员和学生都可以根据姓名查询学生的详细信息。

4. 按学号排序：系统管理员可以对所有学生的信息按照学号进行排序。

5. 按姓名排序：系统管理员可以对所有学生的信息按照姓名进行排序。

6. 删除学生信息：系统管理员可以根据学号删除学生的信息。

7. 系统管理员菜单：系统管理员可以通过菜单选择相应的操作，包括添加学生信息、查询学生信息、排序学生信息和删除学生信息等。

8. 学生菜单：学生可以通过菜单选择查询学生信息的方式，包括按学号查询和按姓名查询。

**二、开发环境：**

1. 编译器：小熊猫DEVC

2. 开发环境：win11

**三、核心代码：**

以下是学生信息管理系统的核心代码：

bool compareStudentById(const Student& student1, const Student& student2) {

return student1.studentId < student2.studentId;

}

bool compareStudentByName(const Student& student1, const Student& student2) {

return student1.name < student2.name;

}

void sortStudentsById() {

sort(students.begin(), students.end(), compareStudentById);

cout << "已按学号排序。" << endl;

}

void sortStudentsByName() {

sort(students.begin(), students.end(), compareStudentByName);

cout << "已按姓名排序。" << endl;

}

void deleteStudent(const string& studentId) {

for (auto it = students.begin(); it != students.end(); ++it) {

if (it->studentId == studentId) {

students.erase(it);

cout << "学生信息删除成功！" << endl;

return;

}

}

cout << "没有找到该学号的学生。" << endl;

}

void menuAdmin() {

while (true) {

cout << "---------------------" << endl;

cout << "系统管理员菜单：" << endl;

cout << "1. 添加学生信息" << endl;

cout << "2. 查询学生信息（按学号）" << endl;

cout << "3. 查询学生信息（按姓名）" << endl;

cout << "4. 按学号排序" << endl;

cout << "5. 按姓名排序" << endl;

cout << "6. 删除学生信息" << endl;

cout << "0. 退出系统" << endl;

cout << "请输入操作编号: ";

int choice;

cin >> choice;

if (choice == 0) {

break;

}

switch (choice) {

case 1:

addStudent();

break;

case 2:

{

string studentId;

cout << "请输入要查询的学生学号: ";

cin >> studentId;

searchStudentById(studentId);

}

break;

case 3:

{

string name;

cout << "请输入要查询的学生姓名: ";

cin >> name;

searchStudentByName(name);

}

break;

case 4:

sortStudentsById();

break;

case 5:

sortStudentsByName();

break;

case 6:

{

string studentId;

cout << "请输入要删除的学生学号: ";

cin >> studentId;

deleteStudent(studentId);

}

break;

default:

cout << "无效的操作编号，请重新输入。" << endl;

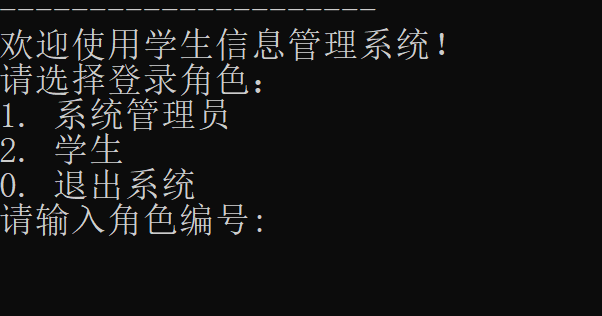
break;

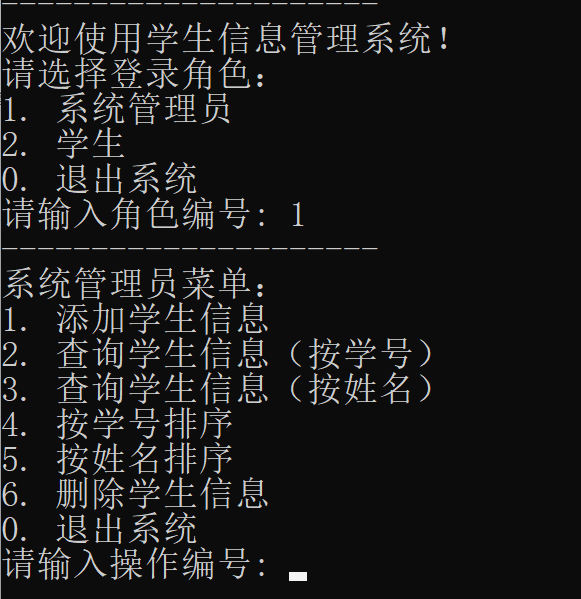
}

}

}

**四、界面展示：**





**五、心得体会：**

开发一个学生信息管理系统需要考虑到多个方面，包括数据结构设计、用户界面设计和功能实现等。在这个简单的实现中，我们使用了C++语言和一些基本的数据结构和算法，完成了基本的学生信息管理功能。通过这个项目，我们可以加深对C++语言和面向对象编程的理解，并学会如何在实际项目中设计和实现一个简单的系统。

在实际开发中，还可以根据需求进行扩展和完善，例如添加更多的学生信息字段、支持更多的查询和排序方式、增加权限控制等。同时，可以考虑使用数据库来存储学生信息，提高系统的性能和扩展性。