****

面向对象程序设计课程设计

（ 2023级 ）

题目：《简易选课系统》

院 系： 电气工程学院

专业班级： 智能科学与技术2301班

学 号： 2305040119

姓 名： 徐境

指导教师/职称： 李付学/副教授

指导教师/职称： 闫红/副教授

2023年6月24日

营口理工学院课程设计任务书

1. 设计题目

以下题目任选其一，具体说明详见《面向对象课程设计教学大纲》。

（1）学生信息管理系统设计

（2）学生综合测评系统

（3）图书信息及借阅管理系统

（4）学校运动会管理系统

（5）教师工作量管理系统

（6）学生的选课系统

（7）银行储蓄系统

（8）学生学籍管理程序

（9）通讯录程序设计

（10）班级成绩管理系统

（11）班级档案管理系统

（12）教师工资管理系统

（13）自拟题目

2.设计目的

本课程设计是智能科学与技术专业重要的实践性环节之一，是在学生学习完《程序设计基础》和《面向对象程序设计》课程后进行的一次全面的综合练习。本课程设计的目的和任务如下：

（1）巩固和加深学生对面向对象语言基本知识的理解和掌握；

（2）培养学生对管理信息系统分析、设计、开发的能力；

（3）培养学生综合运用多种技术，解决问题的能力；

（4）提高学生进行技术总结和撰写文档的能力。

3.设计主要内容及要求

根据任务书，选择其中一个题目进行软件设计，要求如下：

（1）系统设计合理，完成相应的功能；

（2）完成系统演示，保留系统的源代码。

1. 各阶段时间安排、应完成的主要工作（1周，共计5天）

第1天：下发任务书，查阅资料，选定设计题目。

第2天：制定任务分工，完成系统总体设计。

第3天：编码、测试和调试代码，实现基本功能。

第4天：撰写课程设计报告。

第5天：答辩、汇总成绩。

5.验收方式

（1）答辩；

（2）系统的源代码；

（3）课程设计报告。

营口理工学院课程设计成绩考核表

课程设计题目： 简易的选课系统

专业班级：智能科学与技术2301 学号：2305040119姓名： 徐境

|  |  |  |  |
| --- | --- | --- | --- |
| 考核内容 | | 所占比例 | 得分 |
| 课程设计报告 | | 30% |  |
| 平时表现 | | 35% |  |
| 答辩 | | 35% |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
| 同组人 | 无 | | |
| 指导  教师  评语 |  | | |
| 总评成绩(百分制) |  | | |
| 指导教师签字： | | | |

# 摘 要

随着现代教育信息化的发展，传统的手工选课方式已无法满足高校日益增长的学生人数和多样化的选课需求。本文设计并实现了一个简易的学生选课系统，旨在提高选课效率和准确性，减少教师和学生的工作负担。系统主要功能包括课程信息管理、用户管理、选课管理、查询与统计等。采用面向对象的设计方法和分层架构，确保系统的模块化和可维护性。通过实验与测试，验证了系统的功能和性能，达到了预期的设计目标。未来工作将重点在于增强系统的安全性和稳定性，优化用户体验，并增加智能化功能。

**关键词：** 学生选课系统、教育信息化、课程管理、用户管理、查询与统计、面向对象设计、分层架构

目 录

[**第1章 绪论** 5](#_Toc170243413)

[1.1 本课题研究背景 5](#_Toc170243414)

[1.2 对学生选课系统运用的概述 5](#_Toc170243415)

[**第2章 需求与系统概述** 6](#_Toc170243416)

[2.1 需求与功能设计 6](#_Toc170243417)

[2.2 系统框架 8](#_Toc170243418)

[2.3 学生选课操作流程 8](#_Toc170243419)

[**第3章 选课系统的详细说明** 9](#_Toc170243420)

[3.1 课程信息管理 9](#_Toc170243421)

[3.1.1 课程信息录入 9](#_Toc170243422)

[3.1.2 课程信息浏览 9](#_Toc170243423)

[3.1.4 课程信息删除 10](#_Toc170243424)

[3.2 用户管理 10](#_Toc170243425)

[3.2.1 用户注册与登录 10](#_Toc170243426)

[3.2.2 修改密码 11](#_Toc170243427)

[3.3 选课管理 11](#_Toc170243428)

[3.3.1 选课功能 11](#_Toc170243429)

[3.3.2 退课功能 11](#_Toc170243430)

[3.4 查询与统计 12](#_Toc170243431)

[3.4.1 按学分查询 12](#_Toc170243432)

[3.4.2 按课程性质查询 12](#_Toc170243433)

[3.4.3 综合查询与统计 12](#_Toc170243434)

[**第四章 实验与测试** 13](#_Toc170243435)

[4.1 实验环境 13](#_Toc170243436)

[4.2 测试策略 13](#_Toc170243437)

[4.2.2 集成测试 14](#_Toc170243438)

[4.2.3 系统测试 14](#_Toc170243439)

[4.3 测试案例 14](#_Toc170243440)

[4.3.1 用户注册与登录测试 14](#_Toc170243441)

[4.3.2 课程信息管理测试 15](#_Toc170243442)

[4.3.3 选课功能测试 15](#_Toc170243443)

[4.3.4 查询功能测试 16](#_Toc170243444)

[**第5章 总结与展望** 16](#_Toc170243445)

[5.1 总结 16](#_Toc170243446)

[5.1.1 系统功能总结 16](#_Toc170243447)

[5.1.2 技术实现总结 17](#_Toc170243448)

[5.1.3 实验与测试总结 17](#_Toc170243449)

[5.2 展望 17](#_Toc170243450)

[5.2.1 系统安全性与稳定性的增强 17](#_Toc170243451)

[5.2.2 用户界面的优化与用户体验的提升 18](#_Toc170243452)

[5.2.3 智能化功能的增加 18](#_Toc170243453)

[5.2.4 移动端选课功能的实现 18](#_Toc170243454)

[参考文献 I](#_Toc170243455)

[附 录 III](#_Toc170243456)

第1章 绪论

## 1.1 本课题研究背景

在现代教育的大背景下，信息化已经成为不可逆转的趋势。随着高校规模的不断扩大和学生人数的激增，传统的手工选课方式逐渐暴露出效率低下、易出错、无法满足多样化选课需求等诸多弊端。为了应对这些挑战，学生选课系统应运而生，并迅速成为高校教学管理的重要组成部分。

学生选课系统的引入，无疑是一场革命性的变革。它不仅大幅提高了选课的效率和准确性，更在根本上改变了教学管理的模式，使之更加符合现代教育的需求。对于学生而言，这样的系统意味着能够更加自主、灵活地规划自己的学习路径；对于教师和管理人员来说，则意味着能够更加高效、便捷地进行课程管理和学生选课情况的跟踪。

在这种背景下，设计并实现一个简易的学生选课系统不仅具有技术上的挑战性，更有着强烈的现实意义。它不仅能够解决当前高校选课过程中存在的问题，还能够为未来教育信息化的发展提供有益的参考和借鉴。

## 1.2 对学生选课系统运用的概述

学生选课系统不仅是一个技术产品，更是一个综合性的信息管理平台。其核心目标是整合和优化教学资源，提高教学管理的效率和学生的选课体验。

该系统首先为学生提供了一个方便快捷的选课渠道。通过系统，学生可以轻松地浏览到所有的课程信息，包括课程名称、授课教师、上课时间、地点等关键细节。这大大降低了学生获取信息的成本，使他们能够基于自身兴趣和学业规划，做出更加明智的选课决策。

同时，系统也为教师提供了强大的课程管理工具。教师可以轻松地录入、更新和删除课程信息，实时跟踪学生的选课情况，并根据需要进行调整和优化。这不仅提高了教师的教学效率，还使得课程管理变得更加科学和规范。

除了基本功能外，学生选课系统还具备一系列扩展功能，以满足不同用户群体的需求。例如，权限处理功能可以确保不同用户只能访问其被授权的信息，从而保护数据的隐私和安全；模糊查询和综合查询功能则使得用户能够根据自己的需求，灵活检索课程信息和学生选课情况；统计功能则帮助管理人员全面了解选课情况，为教学规划和资源配置提供有力支持。

值得一提的是，学生选课系统的用户界面设计也至关重要。一个直观、友好的界面能够大大提升用户的使用体验，使得选课过程更加顺畅和愉悦。因此，在设计和实现学生选课系统时，需要充分考虑用户的需求和习惯，确保系统的易用性和实用性。

综上所述，学生选课系统的运用不仅提高了高校教学管理的效率，还为学生和教师提供了更加便捷、高效的服务。随着技术的不断进步和教育信息化的深入发展，我们有理由相信，学生选课系统将在未来发挥更加重要的作用，推动高校教学管理向更高水平迈进。

此外，学生选课系统还可以作为高校信息化建设的一个重要组成部分，与其他管理系统（如学生信息管理系统、教务管理系统等）相互配合，共同构建一个完整、高效的高校信息化管理平台。这将为高校的教学、科研和管理提供更加全面、精准的数据支持，推动高校整体运营水平的提升。

第2章 需求与系统概述

## 2.1 需求与功能设计

在深入了解高校教学管理的实际需求后，我们对学生选课系统的功能和特性进行了详细规划。为了确保系统能够满足广大师生和管理人员的日常需求，本系统精心设计了以下几个核心功能：

1. 课程信息管理：

* 课程信息录入：为了确保课程信息的准确性和完整性，我们为教师提供了课程信息录入功能。教师可以方便地录入新课程信息，这些信息包括但不限于课程编号、课程名称、课程描述、授课教师、上课时间、上课地点、学分以及选课人数上限等。通过这一功能，教师可以轻松管理自己的课程信息，确保学生获取到最新、最准确的课程详情。
* 课程信息浏览：所有用户都可以通过此功能查看所有课程的详细信息，从而做出更加明智的选课决策。这一功能不仅提高了信息的透明度，还大大降低了学生获取课程信息的成本。
* 课程信息添加与删除：随着教学计划的调整，课程信息可能需要随时更新。因此，我们为教师提供了课程信息的添加和删除功能，确保课程信息的时效性和准确性。

1. 用户管理：

* 用户注册与登录：为了保障系统的安全性和用户数据的隐私，我们为用户提供了注册与登录功能。通过这一功能，教师可以管理自己的课程信息，学生可以查看和选择课程，管理人员可以进行系统维护和数据统计。
* 修改密码：为了保障用户账户的安全，我们提供了修改密码的功能。用户可以随时更改自己的登录密码，确保账户的安全。

1. 选课管理：

* 选课功能：学生可以通过此功能选择自己感兴趣的课程。系统会实时更新选课情况，确保学生选课的顺利进行。
* 退课功能：如果学生因故需要退课，可以通过此功能轻松实现。系统会及时更新课程人数和选课状态，确保教学资源的合理分配。

1. 查询与统计：

* 按学分查询：用户可以根据课程学分进行查询，方便学生根据自己的学业规划选择合适的课程。
* 按课程性质查询：为了满足不同用户的需求，我们提供了按课程性质查询的功能。用户可以轻松找到符合自己需求的课程。
* 综合查询与统计：为了提供更加全面的课程信息，我们开发了综合查询与统计功能。用户可以根据自己的需求进行多条件查询，并查看详细的统计数据。这一功能不仅为管理人员提供了便捷的数据分析工具，还为学生和教师提供了更加丰富的课程信息。

## 2.2 系统框架

本系统采用面向对象的设计方法，将系统划分为几个相对独立的模块，以提高系统的可维护性和可扩展性。主要模块包括：

1. 用户模块：此模块负责管理用户的注册、登录及权限分配。通过严格的身份验证和权限控制，确保系统的安全性和数据的隐私。
2. 课程模块：此模块负责课程的添加、删除、浏览和信息修改。通过这一模块，教师可以轻松管理自己的课程信息，确保课程数据的准确性和时效性。
3. 选课模块：此模块实现学生的选课和退课功能。通过友好的用户界面和简洁的操作流程，使学生能够轻松完成选课和退课操作。
4. 查询模块：此模块提供多种查询功能和统计报告生成功能。用户可以根据自己的需求进行灵活查询，并获取详细的统计数据和分析报告。

系统架构设计采用分层结构，包括数据层、业务逻辑层和表示层。这种分层设计使得系统更加清晰、易于维护，并提高了系统的可扩展性。数据层负责数据的存储与管理，确保数据的安全性和完整性；业务逻辑层负责实现具体的业务规则和逻辑处理，确保系统的正常运行；表示层负责与用户的交互，提供友好的用户界面和便捷的操作体验。

## 2.3 学生选课操作流程

为了确保学生能够顺利完成选课操作，我们设计了以下简洁明了的选课操作流程：

1. 学生首先通过登录界面输入自己的用户名和密码，验证身份后进入系统。
2. 在系统主页上，学生可以浏览到所有的课程信息，包括课程名称、授课教师、上课时间等关键细节。这些信息将帮助学生做出更加明智的选课决策。
3. 学生选择自己感兴趣的课程后，可以点击“选课”按钮提交选课申请。系统会实时验证选课条件，如课程是否已满员等。
4. 一旦选课申请提交成功，系统会立即确认选课结果，并更新课程信息。学生可以在系统主页上查看自己的选课情况，确保选课的准确性和及时性。

第3章 选课系统的详细说明

## 3.1 课程信息管理

课程信息管理是选课系统的核心功能之一，它涉及到课程的录入、浏览、添加和删除等操作。这些操作对于确保课程信息的准确性和完整性至关重要，同时也为学生提供了丰富、准确的课程资源。

## 3.1.1 课程信息录入

课程信息录入是构建选课系统的第一步，主要由教师或教学管理人员完成。在录入课程信息时，教师需要提供一系列关键数据，包括课程编号、课程名称、课程描述、授课教师信息、课程学分、课程人数上限等。这些信息将被精确地输入到系统中，并经过验证后存储到数据库中。

录入成功后，这些课程信息将立即对学生开放。学生可以通过系统浏览这些课程，了解课程的详细情况，从而做出更加明智的选课决策。此外，系统还支持对课程信息的实时更新，确保学生始终能够获取到最新、最准确的课程信息。

## 3.1.2 课程信息浏览

课程信息浏览功能为所有用户提供了一个直观、便捷的界面来查看系统中已有的课程信息。用户可以轻松地浏览到每门课程的详细信息，包括课程名称、授课教师、上课时间、地点以及课程描述等。这些信息有助于学生更好地了解课程内容和教学安排。

此外，系统还支持多种排序和筛选方式，如按课程编号、课程名称或课程学分进行排序。这种灵活的信息检索方式可以帮助学生快速找到自己感兴趣的课程，提高选课效率。

3.1.3 课程信息添加

随着教学计划的调整或新课程的开设，教师需要向系统中添加新的课程信息。课程信息添加功能与课程信息录入类似，允许教师在已有课程的基础上，轻松地添加新课程。这一功能确保了课程资源的持续更新和丰富，为学生提供了更多的选课选择。

在添加新课程时，教师需要提供与课程相关的基本信息，如课程名称、学分、授课教师等。系统会对这些信息进行验证，确保数据的准确性和完整性。添加成功后，新课程将立即对学生开放，供他们选择和浏览。

## 3.1.4 课程信息删除

当某些课程不再开设或需要更新时，教师可以利用课程信息删除功能从系统中移除这些课程。在删除课程之前，系统会进行一系列的检查操作，以确保没有学生已经选择了该课程。如果有学生已选该课程，系统会提醒教师先通知相关学生进行退课操作，以避免对学生选课造成影响。

删除操作完成后，该课程将从系统中彻底移除，不再对学生开放。这一功能有助于保持课程信息的时效性和准确性，确保学生只能选择到当前开设的课程。

## 3.2 用户管理

用户管理是选课系统的重要组成部分，它涉及到用户的注册、登录以及密码修改等操作。这些操作对于保护用户信息安全、确保系统正常运行具有重要意义。

## 3.2.1 用户注册与登录

为了确保系统的安全性和用户数据的隐私性，选课系统支持教师和学生的注册与登录功能。在注册过程中，用户需要提供唯一的用户名和密码作为身份验证的凭证。系统会对用户信息进行唯一性检查，避免重复注册的情况发生。

一旦注册成功，用户就可以使用自己的用户名和密码登录系统。登录时，系统会验证用户名和密码的正确性，并根据用户类型（教师、学生或管理员）提供相应的操作权限和菜单选项。这种身份验证机制确保了系统的安全性和数据的保密性。

## 3.2.2 修改密码

为了保护用户账户的安全，选课系统允许用户随时修改自己的登录密码。在修改密码时，用户需要输入旧密码以验证身份，然后输入并确认新密码。系统会对新密码进行强度检查，确保其符合安全标准后才会更新到数据库中。

通过修改密码功能，用户可以定期更新自己的密码，降低账户被盗用的风险。同时，如果用户怀疑自己的密码已经泄露或遗忘密码，也可以利用此功能进行重置。

## 3.3 选课管理

选课管理是选课系统的核心功能之一，它包括选课和退课两个主要操作。这些操作对于学生合理安排自己的学习计划和课程选择至关重要。

## 3.3.1 选课功能

选课功能是学生选课系统的核心功能之一，它允许学生根据自己的兴趣和学业规划选择合适的课程。在选课过程中，学生可以浏览所有可选课程的详细信息，并根据自己的需求进行选择。

系统会实时检查课程的选课条件，如课程是否已满员或是否存在时间冲突等。只有在满足选课条件的情况下，系统才允许学生进行选课操作。选课成功后，系统会立即更新课程的已选人数和学生的选课记录，确保数据的实时性和准确性。

## 3.3.2 退课功能

退课功能允许学生从已选课程中退出，以便他们根据自己的实际情况调整学习计划或选择其他更合适的课程。在退课过程中，学生需要选择要退的课程并提交退课申请。

系统会检查学生是否确实选了该课程，并在确认无误后进行退课操作。退课成功后，系统会立即更新课程的已选人数和学生的选课记录。这一功能为学生提供了灵活的选课调整空间，使他们能够更好地规划自己的学习路径。

## 3.4 查询与统计

查询与统计功能是选课系统的重要辅助工具，它帮助学生、教师和管理人员快速获取课程信息和选课情况的统计数据。

## 3.4.1 按学分查询

按学分查询功能允许用户根据课程学分进行筛选和查找。用户只需输入所需的学分数值或范围，系统就会列出所有符合条件的课程供用户选择和浏览。这一功能有助于学生根据自己的学业规划和学分要求选择合适的课程。

## 3.4.2 按课程性质查询

按课程性质查询功能使用户能够根据课程的性质（如必修课、选修课、实践课等）进行筛选。用户选择相应的课程性质后，系统将显示所有符合该性质的课程列表。这一功能有助于学生快速找到符合自己学习需求和兴趣的课程。

## 3.4.3 综合查询与统计

综合查询与统计功能提供了更为强大和灵活的信息检索和数据分析工具。用户可以根据多个条件进行组合查询，如按课程名称、学分、性质等多个维度进行筛选。同时，系统还提供了丰富的统计功能，如统计各类课程的选课人数、学分分布、教师工作量等。

这些统计数据不仅为学生提供了更加全面的课程信息和选课参考，还为教学管理人员提供了有力的数据支持，帮助他们更好地了解教学情况和进行资源优化配置。通过综合查询与统计功能，选课系统成为了一个强大的教学管理助手，为高校的教学管理工作带来了极大的便利和效益。

第四章 实验与测试

在软件开发过程中，实验与测试是至关重要的一环，它们确保了软件的质量和稳定性。在本章节中，我们将详细介绍针对选课系统的实验环境和测试策略，以及具体的测试案例。

## 4.1 实验环境

为了确保选课系统能够在各种环境中稳定运行，我们在一个具有代表性的环境中进行了实验和测试。实验环境包括硬件和软件两部分。

硬件环境

我们的测试主要在一台普通PC机上进行。这台机器的配置代表了大多数用户的硬件设备，因此，在这样的环境下进行测试能够确保我们的系统在广泛使用的硬件上表现良好。

软件环境

在软件方面，我们选择了广泛使用的Windows操作系统作为测试平台。Windows操作系统用户众多，具有广泛的兼容性，因此是测试的理想选择。此外，我们还使用了Visual Studio作为开发环境。Visual Studio提供了丰富的调试和测试工具，帮助我们有效地找出并修复系统中的问题。

## 4.2 测试策略

在软件开发中，测试策略的制定至关重要。为了确保选课系统的功能和性能达到预期，我们采用了多种测试方法，包括单元测试、集成测试和系统测试。

4.2.1 单元测试

单元测试是软件开发中最基础的测试之一。我们对选课系统的各个功能模块进行了独立的单元测试，以确保每个模块都能正确实现其预期功能。例如，我们测试了用户注册、登录、课程信息录入、选课、退课等核心功能。通过这些测试，我们能够确保每个功能模块都能独立地正常工作。

## 4.2.2 集成测试

在单元测试之后，我们进行了集成测试。这一阶段的目的是测试各个功能模块之间的接口和交互，确保它们能够协同工作。我们测试了用户登录后能否正确访问课程信息，选课后课程信息和用户选课记录能否正确更新等场景。集成测试帮助我们发现了模块间交互可能存在的问题，并进行了及时修复。

## 4.2.3 系统测试

最后，我们进行了全面的系统测试。这一阶段旨在测试整个系统在实际运行环境中的稳定性和可靠性。我们进行了功能测试，以确保所有功能都能按照预期工作；进行了性能测试，以评估系统在不同负载下的表现；还进行了压力测试，以检验系统在极端情况下的稳定性和恢复能力。

## 4.3 测试案例

为了更具体地说明我们的测试过程，以下是一些详细的测试案例。

## 4.3.1 用户注册与登录测试

在用户注册与登录功能的测试中，我们设计了以下步骤：

1. 访问注册页面：首先，我们尝试访问注册页面，检查页面是否能够正常加载，并验证所有必要的输入字段是否都显示出来。
2. 输入并提交注册信息：接着，我们在注册页面输入有效的用户名和密码，并提交注册信息。系统应该验证输入的有效性，并成功创建用户账户。
3. 访问登录页面并登录：注册成功后，我们访问登录页面，并输入刚才注册的用户名和密码进行登录。系统应该验证登录信息，并允许用户成功登录。
4. 验证登录状态：登录后，我们验证系统是否正确显示了用户的功能菜单，并确保用户能够访问其权限范围内的所有功能。

通过这个测试案例，我们确保了用户注册和登录功能的正确性和安全性。

## 4.3.2 课程信息管理测试

在课程信息管理功能的测试中，我们关注以下几个方面：

1. 教师登录并访问课程管理页面：教师用户登录系统后，应能够顺利访问课程管理页面。
2. 添加课程信息：教师在课程管理页面输入新课程的信息，并提交添加请求。系统应验证输入信息的完整性，并成功添加课程。
3. 浏览和验证课程信息：添加课程后，我们浏览课程列表，验证新增的课程是否正确显示，并检查课程信息的准确性。
4. 删除课程信息：我们选择一门课程进行删除操作，并验证系统是否成功删除了该课程。同时，我们检查课程列表，确保被删除的课程不再显示。

这个测试案例确保了教师能够有效地管理课程信息。

## 4.3.3 选课功能测试

选课功能是选课系统的核心功能之一。在测试中，我们关注以下几点：

1. 学生登录并浏览课程：学生登录系统后，应能够浏览到所有可选的课程信息。
2. 选择并添加课程：学生选择一门课程进行选课操作。系统应检查课程的选课条件（如课程容量、选课时间等），并在满足条件的情况下允许学生选课。选课成功后，我们验证课程的已选人数和学生的选课记录是否已正确更新。
3. 退课操作：学生选择已选的课程进行退课操作。系统应验证学生的退课请求，并在确认无误后更新课程的已选人数和学生的选课记录。

通过这个测试案例，我们确保了选课功能的正确性和可靠性。

## 4.3.4 查询功能测试

查询功能是帮助学生和教师快速找到所需课程的重要工具。在测试中，我们主要关注以下几点：

1. 访问查询页面并输入查询条件：用户访问查询页面，并输入查询条件，如学分要求、课程性质等。
2. 执行查询并验证结果：用户提交查询请求后，系统应根据输入的查询条件返回符合条件的课程列表。我们验证系统是否正确返回了满足条件的课程，并检查课程信息的准确性。

这个测试案例确保了查询功能的正确性和高效性，帮助用户快速找到所需的课程信息。

第5章 总结与展望

## 5.1 总结

经过一系列的设计与开发流程，简易学生选课系统已成功构建并经过严格的实验与测试。这一系统不仅仅是一个简单的选课工具，而是基于面向对象的设计思想，融入现代化软件开发理念，采用分层架构以确保系统的可扩展性和可维护性。以下是对本系统的全面总结。

## 5.1.1 系统功能总结

本系统主要实现了以下几大核心功能：

1. 课程信息管理：系统提供了完善的课程信息管理功能，允许管理员或教师用户轻松地添加、修改、删除和查询课程信息。这确保了课程数据的时效性和准确性。
2. 用户管理：系统支持多角色用户，包括学生、教师和管理员，并为每种角色提供了相应的权限管理。这保证了不同用户只能访问其权限范围内的功能，从而确保了数据的安全性。
3. 选课管理：学生可以通过系统进行选课和退课操作，而系统会实时更新选课状态，确保数据的实时性。同时，系统还会检查选课冲突、课程容量等问题，为学生提供流畅的选课体验。
4. 查询与统计：系统提供了强大的查询与统计功能，用户可以根据多种条件进行课程查询，同时系统还可以统计课程的选课人数、课程的热门度等信息，为教学管理提供有力的数据支持。

## 5.1.2 技术实现总结

在技术实现上，本系统采用了先进的软件架构和设计模式，确保系统的稳定性和可扩展性。通过分层架构，我们将系统分为数据访问层、业务逻辑层和表示层，这使得系统更加模块化，便于后续的维护和升级。

同时，我们还充分利用了面向对象的设计思想，通过类和对象的关系来模拟现实世界中的实体和关系，从而提高了代码的可读性和可重用性。

## 5.1.3 实验与测试总结

在实验与测试阶段，我们对系统进行了全面的单元测试、集成测试和系统测试，确保系统的每个功能模块都能正常工作，并且各模块之间的交互也是无误的。通过这些测试，我们成功地验证了系统的功能和性能，确保其达到了预期的设计目标。

## 5.2 展望

虽然简易学生选课系统已经取得了一定的成果，但随着教育信息化的不断深入和发展，我们还需要对系统进行持续的完善和优化。以下是对未来工作的展望：

## 5.2.1 系统安全性与稳定性的增强

随着网络攻击和数据泄露事件的频发，系统的安全性变得尤为重要。未来，我们将考虑增加更多的安全措施，如数据加密技术，以确保用户数据的安全性。同时，我们还会加强系统的权限控制功能，防止未经授权的访问和操作。

此外，我们还将致力于提高系统的稳定性。通过持续的监控和调优，我们将确保系统在高并发和大数据量的情况下仍能保持稳定运行。

## 5.2.2 用户界面的优化与用户体验的提升

用户界面是用户与系统交互的桥梁，其友好性和直观性直接影响到用户的使用体验。因此，我们将投入更多的资源来优化用户界面，使其更加简洁、美观和易用。

同时，我们还将通过用户反馈和数据分析来了解用户的需求和习惯，从而针对性地提升用户体验。例如，我们可以增加更多的个性化设置选项，让用户能够根据自己的喜好来定制系统界面和功能。

## 5.2.3 智能化功能的增加

随着人工智能和大数据技术的不断发展，为系统增加智能化功能已经成为可能。未来，我们将考虑引入推荐算法和机器学习技术，为用户提供更加个性化的课程推荐服务。这将有助于用户更快地找到适合自己的课程，提高选课的效率和满意度。

此外，我们还将探索智能排课的可能性。通过算法优化和数据分析，我们将尝试为用户生成更加合理和高效的课程表。

## 5.2.4 移动端选课功能的实现

随着移动互联网的普及和发展，越来越多的用户倾向于使用手机等移动设备来访问和使用各类应用。因此，实现移动端选课功能已经成为一个迫切的需求。

未来，我们将投入资源来开发移动端的选课应用，让用户能够随时随地进行选课操作。这将大大提高选课的便捷性和灵活性。同时，我们还将确保移动端应用与PC端系统之间的数据同步和一致性，为用户提供无缝的使用体验。

总之，简易学生选课系统虽然已经取得了一定的成果，但我们的工作还远远没有结束。通过不断的改进和创新，我们相信这个系统将能够更好地服务于广大师生，推动高校教学管理水平的提升。

# 参考文献

[1]Jutta M V,Gisela W. Data from the Panel Study ‘Refugees in the German Educational System (ReGES)’[J]. Journal of Open Psychology Data,2023,11(1).

[2]张海林,夏传良.C++面向对象程序设计课程教学研究[J].智能计算机与应用,2020,10(10):190-193.

[3]张小龙,王合霞.信息网络化技术促进普通高校体育课程模式改革的实践研究[J].山东师范大学学报(自然科学版),2017,32(03):151-154.

[4]滕金国,李贵安.高校学生选课系统的性能优化研究及实践——以陕西师范大学为例[J].中国教育信息化,2017(17):86-90.

[5]张俊,宋雪勦,曹慧敏等.高校教务管理系统选课算法研究综述[J].攀枝花学院学报,2017,34(02):49-52.DOI:10.13773/j.cnki.51-1637/z.2017.02.012.

[6]韦庆清,左军,崔如春.面向对象技术中数据访问方法研究[J].电脑知识与技术,2016,12(04):19-22+26.DOI:10.14004/j.cnki.ckt.2016.0442.

[7]曾佳平,杨秋辉,汪华龙等.基于动态插桩的C/C++内存泄漏检测工具的设计与实现[J].计算机应用研究,2015,32(06):1737-1741.

[8]谭艳. 高校教务管理系统优化方案研究[D].南华大学,2014.

[9]Mo W H. The Application Study on Choosing Course System Based on JBoss Cache of Distributed Cache Technology[J]. Applied Mechanics and Materials,2014,2948(496-500).

[10]徐书欣,顿贝贝.使用Visual C++开发大型应用软件系统的体会与探讨[J].数字技术与应用,2013(08):138+140.DOI:10.19695/j.cnki.cn12-1369.2013.08.095.

[11]王磊,蔡绍峰.C++课程在应用型本科院校中的新型教学模式[J].赤峰学院学报(自然科学版),2013,29(04):211-213.DOI:10.13398/j.cnki.issn1673-260x.2013.04.018.

[12]李凤琪. 校园教务管理信息系统设计与实现[D].北京工业大学,2012.

[13]王晓锋,李静.“C++语言程序设计”课程教学探讨[J].吉林工程技术师范学院学报,2011,27(03):76-77.

[14]李梦雨.文件I/O函数和标准I/O库函数的读写效率研究[J].软件导刊,2010,9(06):5-7.

[15]彭召意,朱艳辉,周玉.C++面向对象程序设计课程的实践教学研究[J].计算机教育,2010(05):151-154.DOI:10.16512/j.cnki.jsjjy.2010.05.026.

# 附 录

程序代码

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

#include <windows.h>

using namespace std;

// 课程信息数据结构

struct Course {

    string courseID;       // 课程编号

    string courseName;     // 课程名称

    int credits;           // 课程学分

    int enrolledStudents;  // 课程已选人数

    int maxStudents;       // 课程人数上限

    Course(string id, string name, int credit, int max)

        : courseID(id), courseName(name), credits(credit), enrolledStudents(0), maxStudents(max) {}

};

// 存储课程信息的向量

vector<Course> courses;

void displayMenu() {

    cout << "====== 学生选课系统 ======" << endl;

    cout << "1. 课程信息录入" << endl;

    cout << "2. 课程信息浏览" << endl;

    cout << "3. 课程信息添加" << endl;

    cout << "4. 课程信息删除" << endl;

    cout << "5. 学生登录" << endl;

    cout << "6. 学生浏览课程" << endl;

    cout << "7. 学生选课" << endl;

    cout << "8. 按学分查询课程" << endl;

    cout << "9. 按课程名称查询课程" << endl;

    cout << "0. 退出系统" << endl;

    cout << "请选择操作：";

}

void addCourse() {

    string id, name;

    int credit, max;

    cout << "输入课程编号: ";

    cin >> id;

    cout << "输入课程名称: ";

    cin >> name;

    cout << "输入课程学分: ";

    cin >> credit;

    cout << "输入课程人数上限: ";

    cin >> max;

    courses.push\_back(Course(id, name, credit, max));

    cout << "课程信息录入成功！" << endl;

}

void viewCourses() {

    cout << "当前课程信息如下：" << endl;

    for (const auto& course : courses) {

        cout << "课程编号: " << course.courseID

             << ", 课程名称: " << course.courseName

             << ", 学分: " << course.credits

             << ", 已选人数: " << course.enrolledStudents

             << ", 人数上限: " << course.maxStudents << endl;

    }

}

void deleteCourse() {

    string id;

    cout << "输入要删除的课程编号: ";

    cin >> id;

    auto it = remove\_if(courses.begin(), courses.end(), [id](Course& c) { return c.courseID == id; });

    if (it != courses.end()) {

        courses.erase(it, courses.end());

        cout << "课程删除成功！" << endl;

    } else {

        cout << "未找到该课程。" << endl;

    }

}

void searchCourseByCredits() {

    int credit;

    cout << "输入要查询的课程学分: ";

    cin >> credit;

    for (const auto& course : courses) {

        if (course.credits == credit) {

            cout << "课程编号: " << course.courseID

                 << ", 课程名称: " << course.courseName

                 << ", 学分: " << course.credits

                 << ", 已选人数: " << course.enrolledStudents

                 << ", 人数上限: " << course.maxStudents << endl;

        }

    }

}

void searchCourseByName() {

    string name;

    cout << "输入要查询的课程名称: ";

    cin >> name;

    for (const auto& course : courses) {

        if (course.courseName == name) {

            cout << "课程编号: " << course.courseID

                 << ", 课程名称: " << course.courseName

                 << ", 学分: " << course.credits

                 << ", 已选人数: " << course.enrolledStudents

                 << ", 人数上限: " << course.maxStudents << endl;

        }

    }

}

// 以下函数模拟学生的相关操作，例如登录和选课

void studentLogin() {

    cout << "学生登录成功。" << endl;

}

void studentViewCourses() {

    viewCourses();

}

void studentSelectCourse() {

    string id;

    cout << "输入要选修的课程编号: ";

    cin >> id;

    for (auto& course : courses) {

        if (course.courseID == id) {

            if (course.enrolledStudents < course.maxStudents) {

                course.enrolledStudents++;

                cout << "选课成功！" << endl;

            } else {

                cout << "该课程人数已满。" << endl;

            }

            return;

        }

    }

    cout << "未找到该课程。" << endl;

}

int main() {

    // 设置控制台输出为 UTF-8 编码

SetConsoleOutputCP(CP\_UTF8);

 while (true) {

        displayMenu();

        int choice;

        cin >> choice;

        switch (choice) {

            case 1:

                addCourse();

                break;

            case 2:

                viewCourses();

                break;

            case 3:

                addCourse();

                break;

            case 4:

                deleteCourse();

                break;

            case 5:

                studentLogin();

                break;

            case 6:

                studentViewCourses();

                break;

            case 7:

                studentSelectCourse();

                break;

            case 8:

                searchCourseByCredits();

                break;

            case 9:

                searchCourseByName();

                break;

            case 0:

                cout << "退出系统。" << endl;

                return 0;

            default:

                cout << "无效选择，请重新输入。" << endl;

        }

    }

    return 0;

}