**学生课程设计报告**

**2023 —— 2024 学年 第 2 学期**

|  |
| --- |
| **课程名称 面向对象程序设计** |
| **学 院 计算机科学技术** |
| **专 业 计算机科学与技术** |
| **学 号** |
| **姓 名** |

|  |  |  |  |
| --- | --- | --- | --- |
| **起 止 周** | 18-19周 | **周数** | 2 |
| **实习地点** | 实验教学中心 | | |
| **目的：**通过实际问题情境，综合应用面向对象思维与分析设计方法，查阅文 献，分析并解决问题，完成正确的类和成员的职责划分，运用集成化开发工具进行面向对象 设计、编码，并且撰写项目文档。 请大家根据题目要求完成程序的分析、设计和功能代码编写工作，并完成课程设计报告。 实验进行过程中，每个学生独立动手完成项目，并要求学生充分发挥自己的创意，设计 出优秀的课程设计作品。 | | | |
| **要求：**  .课程题目：学生学籍管理程序 内容要求：设计学生学籍管理程序。要求能够对学生的学籍信息进行存储、删除、检索 和显示。能够统计全部学生的总成绩，及其在班上的排名，能够统计各科的平均成绩及 及格率，并具有错误提示功能  1.必须完成题目要求的全部功能。  2.根据题目，恰当定义一个或多个类，注意每个类功能划分的合理性。  3.整个程序应该是一个功能完整的、可运行的程序。 | | | |

|  |
| --- |
| 进度安排及主要内容：  1. 需求分析  1.1 程序功能需求  本程序旨在实现一个学生信息管理系统，具体功能如下：  - 学生信息管理：  - 添加学生信息  - 删除学生信息  - 查询学生信息  - 打印所有学生信息  - 班级信息管理：  - 添加班级信息  - 删除班级信息  - 查询班级信息  - 打印所有班级信息  - 成绩管理：  - 查询所有学生的成绩  - 查询班级成绩  - 查询各科成绩统计  2. 程序设计  2.1 类和方法的主要功能  Student 类  功能：用于存储学生的基本信息和成绩。  - 属性：  - id：学生ID  - name：学生姓名  - gender：学生性别  - age：学生年龄  - origin：学生出生地  - enrollmentYear：入学年份  - className：班级名称  - phoneNumber：联系电话  - grades：学生各科成绩  StudentManagement 类  功能：用于管理学生信息，包括添加、删除、查询和打印学生信息。  - 属性：  - students：存储学生信息的映射表  - 方法：  - addStudent(const Student& student)：添加学生信息  - removeStudent(const string& id)：删除学生信息  - queryStudent(const string& id) const：查询学生信息  - printInfo(const Student& student) const：打印学生信息  - validateGender(const string& gender)：验证性别是否合法  - validateId(const string& id)：验证ID是否合法  ClassManagement 类  功能：用于管理班级信息，包括添加、删除、查询班级信息，以及班级与学生的关联。  - 属性：  - classes：存储班级信息的映射表（班级ID -> 专业）  - classStudents：存储班级和学生关联的映射表（班级ID -> 学生ID列表）  - 方法：  - addClass(const string& classID, const string& major)：添加班级信息  - removeClass(const string& classID)：删除班级信息  - classExists(const string& classID) const：判断班级是否存在  - getClassMajor(const string& classID) const：获取班级的专业  - addStudentToClass(const string& classID, const string& studentID)：将学生加入班级  - removeStudentFromClass(const string& classID, const string& studentID)：将学生从班级中移除  - queryClass(const string& classID, const StudentManagement& studentManager) const：查询班级信息  - getClassStudents(const string& classID) const：获取班级的所有学生  - getStudentClass(const string& studentID) const：获取学生所属的班级  - getStudentMajor(const string& studentID) const：获取学生的专业  - getMajorClasses(const string& major) const：获取某专业的所有班级  ScoreManagement 类  功能：用于管理学生成绩，提供查询所有学生成绩、查询班级成绩、查询各科成绩统计的功能。  - 方法：  - queryAllScores(const StudentManagement& studentManager, const ClassManagement& classManager) const：查询所有学生的成绩  - queryClassScores(const string& classID, const ClassManagement& classManager, const StudentManagement& studentManager) const：查询班级成绩  - querySubjectStats(const StudentManagement& studentManager) const：查询各科成绩统计  ManagementSystem 类  功能：主控制类，管理学生、班级和成绩模块的交互，提供用户界面。  - 属性：  - studentManager：学生管理对象  - classManager：班级管理对象  - scoreManager：成绩管理对象  - 方法：  - initializeData()：初始化数据  - printMenu()：打印主菜单  - menuStudent()：打印学生管理菜单  - menuClass()：打印班级管理菜单  - menuScore()：打印成绩管理菜单  - handleStudentManagement()：处理学生管理操作  - handleClassManagement()：处理班级管理操作  - handleScoreManagement()：处理成绩管理操作  - run()：运行系统  2.2 类间关系  类间关系的描述：  - ManagementSystem 类依赖于 StudentManagement、ClassManagement 和 ScoreManagement 类。  - StudentManagement 类与 Student 类关联，负责管理 Student 对象。  - ClassManagement 类与 StudentManagement 类关联，负责管理班级与学生的关系。  - ScoreManagement 类依赖于 StudentManagement 和 ClassManagement 类，负责查询和统计学生成绩。  2.3 重要功能的程序流程图  以下为重要功能的程序流程图示例：  添加学生信息  1. 用户选择添加学生信息。  2. 系统提示用户输入学生ID。  3. 系统调用 validateId 方法验证ID是否合法。  - 如果ID不合法，提示用户重新输入。  - 如果ID合法，继续下一步。  4. 系统提示用户输入学生其他信息（姓名、性别、年龄、出生地、入学年份、班级、联系电话、各科成绩）。  5. 系统调用 validateGender 方法验证性别是否合法。  - 如果性别不合法，提示用户重新输入。  - 如果性别合法，继续下一步。  6. 系统调用 addStudent 方法添加学生信息。  7. 系统调用 addStudentToClass 方法将学生加入对应班级。  8. 系统提示用户学生信息添加成功。  查询班级成绩  1. 用户选择查询班级成绩。  2. 系统提示用户输入班级ID。  3. 系统调用 classExists 方法判断班级是否存在。  - 如果班级不存在，提示用户重新输入。  - 如果班级存在，继续下一步。  4. 系统调用 getClassStudents 方法获取班级的所有学生。  5. 系统遍历班级学生列表，调用 getStudents 方法获取每个学生的成绩。  6. 系统计算班级的总平均分、各科平均分、各科合格率、优秀率。  7. 系统打印班级成绩统计表。  程序流程图示例（文字描述）：  - 添加学生信息：  - 输入学生ID  - 验证ID是否合法  - 输入其他学生信息  - 验证性别是否合法  - 添加学生信息  - 将学生加入对应班级  - 提示用户学生信息添加成功  - 查询班级成绩：  - 输入班级ID  - 判断班级是否存在  - 获取班级的所有学生  - 获取每个学生的成绩  - 计算班级的总平均分、各科平均分、各科合格率、优秀率  - 打印班级成绩统计表  3. 程序调试及结果  3.1 实验环境  操作系统：Windows 10  开发工具：Visual Studio Code  编译器：G++ (MinGW64)  编程语言：C++ 3.2  遇到的问题及解决方案 在开发过程中，我们遇到了一些问题，并通过以下方式解决：  - 多文件编译问题：在使用多文件编译时，遇到了头文件找不到的问题。通过在编译命令中添加所有源文件，以及正确设置头文件的相对路径，解决了这个问题。  - 数据一致性问题：在删除学生信息时，需要先从班级中移除该学生，才能保证数据的完整性。  - 输出格式问题：在输出成绩统计表时，使用了循环计算每一列的最大字符长度，并使用 setw 设置输出格式，确保表格对齐。  3.3 调试结果 所有功能模块均通过了测试，程序能够正确地添加、删除、查询学生和班级信息，并能正确统计和查询学生成绩。  3.4 收获与总结 通过本次课程设计，我收获了以下几点经验：  1. 学会了如何合理划分功能模块，使用类和对象管理数据，使程序结构清晰，易于维护和扩展。  2. 通过查找资料，学习了如何解决多文件编译和数据一致性等问题，提高了调试和解决问题的能力。  3. 在输出格式方面，学会了使用循环计算每一列的最大字符长度，并使用 setw 设置输出格式，确保输出的表格对齐美观 |
| **成绩：** |
| **指导教师评语(从学生出勤、学习态度、纪律、实习效果、实习报告数据和图样完整性、系统性、书写工整等方面给出定性和定量的评价)：**  指导教师签字：  2024 年 7 月 7 日 |

实习报告

|  |
| --- |
| （本页可加页） |