**信息科技学院**

**《程序设计基础》课程设计报告**

2024/2025学年第 二 学期

课程名称 程序设计基础(C)课程设计

设计题目 　 外卖点餐服务系统

学生姓名 　　　 田典

学 号 202405315424

专业班级

指导教师 朱斌、洪峰、华冠萍

2025 年 3 月 3 日

|  |
| --- |
| 1.设计任务书  **1.1 题目与要求**  题目：外卖点餐服务系统  要求：   * 1. 明确设计任务，能口述并文字描述清楚自己的题目要求、期望结果；   2. 清楚设计过程中的主要知识运用，不清楚的及时与指导老师沟通；   3. 能进行总体设计，即清楚课题中的数据与文件设计、功能设计、函数模块设计；   4. 能对每个模块即函数进行详细设计编码；   5. 能对所有的函数模块进行联调；   6. 编码仔细，风格统一，尽量减少代码中的语法错误；   调试数据多样，注意特殊情况和边界数据处理，尽量减少代码的逻辑错误；  **1.2 本程序涉及的知识点**  【此处总结本次课程设计所做项目用到哪些程序设计的知识点】   1. 自定义结构体数组 2. 函数的使用 3. 文件的打开、存储和使用 4. For，while循环和if语句 5. 字符串的使用（strcpy，strcmp） 6. 指针的使用 |
| 2.总体设计  **2.1 总体设计思想**  【要求采用模块化的设计思想，画一个功能模块图，配以文字说明】  学籍管理系统  删除学籍信息  修改学籍信息  添加学籍信息  查询学籍信息  统计学籍信息  排序学籍信息  按学号删除  按姓名删除  查询全部  按姓名查询  按生日查询  按地址查询  统计同年出生的  统计同一专业的  按专业名称排名  按年龄大小排名    **2.2 结构体与文件说明**  【此处根据自己实际情况，写出定义的结构体、文件存储说明】  **结构体Dish（菜单）**  **Order（订单）**  **2.3 功能设计**  【此处根据自己实际设计的功能函数，用一个表格形式说明每个功能数的具体功能】   |  |  | | --- | --- | | **save\_menu\_to\_file** | 将菜单数据保存到文件 menu.txt 中 | | **load\_menu\_from\_file** | 从文件 menu.txt 加载菜单数据到 menu 数组中。 | | **save\_orders\_to\_file** | 将订单数据保存到文件 orders.txt 中 | | **load\_orders\_from\_file** | 从文件 orders.txt 加载订单数据到 orders 数组中。 | | **add\_dish** | 添加新菜品到菜单中。 | | **modify\_dish** | 修改菜单中指定菜品的详细信息。 | | **browse\_menu** | 浏览当前菜单中的所有菜品。 | | **add\_to\_cart** | 将指定菜品加入购物车。 | | **place\_order** | 将购物车中的订单提交。 | | **simulate\_payment** | 模拟支付，计算并显示总价。 | | **analyze\_revenue** | 分析总营业额。 | | **sort\_menu\_by\_price** | 按单价对菜单进行排序（升序）。 | |
| 3.部分主要代码及说明  【此处仅作参考，自己填写时删除内容，请写出定义了哪些函数，并对所定义的函数原型进行说明；并选择其中最重要的若干个函数源代码进行分析】  **add\_dish 函数**  void add\_dish() {  if (dish\_count >= MAX\_DISHES) {  printf("菜单已满，无法添加新菜品！\n");  return;  }  Dish new\_dish;  new\_dish.dish\_id = dish\_count + 1;  printf("请输入菜名: ");  scanf("%s", new\_dish.name);  printf("请输入类别: ");  scanf("%s", new\_dish.category);  printf("请输入单价: ");  scanf("%f", &new\_dish.price);  printf("请输入总数量: ");  scanf("%d", &new\_dish.total\_quantity);  menu[dish\_count++] = new\_dish;  save\_menu\_to\_file(); // 保存菜单数据到文件  printf("菜品添加成功！\n");  }   * **功能**：添加新菜品到菜单中。 * **实现**：   1. 检查菜单是否已满。   2. 输入菜品信息（菜名、类别、单价、总数量）。   3. 将新菜品添加到 menu 数组中，并更新 dish\_count。   4. 调用 save\_menu\_to\_file 保存数据。 * **关键点**：   1. 输入验证：菜单是否已满。   2. 数据存储：动态更新 menu 数组。   **add\_to\_cart 函数**  void add\_to\_cart(Order \*cart, int \*cart\_count) {  int id, quantity;  printf("请输入要加入购物车的菜品编号: ");  scanf("%d", &id);  printf("请输入要购买的份数: ");  scanf("%d", &quantity);  if (id < 1 || id > dish\_count) {  printf("无效的菜品编号！\n");  return;  }  Dish \*dish = &menu[id - 1];  if (dish->total\_quantity < quantity) {  printf("库存不足，无法购买！\n");  return;  }  cart[\*cart\_count].order\_id = \*cart\_count + 1;  printf("请输入客户电话: ");  scanf("%s", cart[\*cart\_count].phone);  printf("请输入客户地址: ");  scanf("%s", cart[\*cart\_count].address);  strcpy(cart[\*cart\_count].dish\_name, dish->name);  cart[\*cart\_count].quantity = quantity;  cart[\*cart\_count].total\_price = dish->price \* quantity;  dish->total\_quantity -= quantity;  (\*cart\_count)++;  printf("菜品已加入购物车！\n");  }   * **功能**：将指定菜品加入购物车。 * **实现**：   1. 输入菜品编号和购买数量。   2. 检查菜品编号和库存是否有效。   3. 输入客户电话和地址。   4. 将订单信息存储到 cart 数组中，并更新库存。   5. 更新 cart\_count。 * **关键点**：   1. 输入验证：菜品编号和库存是否有效。   2. 数据存储：动态更新 cart 数组。   **sort\_menu\_by\_price 函数**  c  复制  void sort\_menu\_by\_price() {  int i, j;  Dish temp;  for (i = 0; i < dish\_count - 1; i++) {  for (j = 0; j < dish\_count - 1 - i; j++) {  if (menu[j].price > menu[j + 1].price) {  temp = menu[j];  menu[j] = menu[j + 1];  menu[j + 1] = temp;  }  }  }  printf("菜单已按单价排序！\n");  }   * **功能**：按单价对菜单进行排序（升序）。 * **实现**：   1. 使用冒泡排序算法对 menu 数组按 price 排序。   2. 提示排序完成。 * **关键点**：   1. 排序算法：冒泡排序。   2. 数据更新：直接修改 menu 数组。   4.测试数据与结果截图  【此处请写出每一个主要操作的具体测试数据、预期结果以及实际操作结果的截图；】  **操作1. 添加菜品 (add\_dish)**  **测试数据**   * **菜名：红烧肉** * **类别：荤菜** * **单价：38.5** * **总数量：20**   **预期结果**   * **菜品成功添加到菜单中。** * **菜单文件 menu.txt 中新增一行：1 红烧肉 荤菜 38.50 20。** * **控制台输出：菜品添加成功！。**     操作二**修改菜品 (modify\_dish)**  **测试数据**   * 菜品编号：1（假设已存在） * 新菜名：糖醋排骨 * 新类别：荤菜 * 新单价：45.0 * 新总数量：15   **预期结果**   * 菜品信息更新为：   + 菜名：糖醋排骨   + 类别：荤菜   + 单价：45.0   + 总数量：15 * 菜单文件 menu.txt 中对应行更新为：1 糖醋排骨 荤菜 45.00 15。 * 控制台输出：菜品修改成功！。       **3. 浏览菜单 (browse\_menu)**  **测试数据**   * 假设菜单中有以下菜品：   1. 1 红烧肉 荤菜 38.50 20   2. 2 鱼香肉丝 荤菜 28.00 30   **预期结果**   * 控制台输出：   菜单列表:  菜品编号: 1, 菜名: 红烧肉, 类别: 荤菜, 单价: 38.50, 总数量: 20  菜品编号: 2, 菜名: 鱼香肉丝, 类别: 荤菜, 单价: 28.00, 总数量: 30    **4. 加入购物车 (add\_to\_cart)**  **测试数据**   * 菜品编号：1（假设菜品为 红烧肉，库存为 20） * 购买数量：3 * 客户电话：13812345678 * 客户地址：北京市朝阳区   **预期结果**   * 购物车中新增一条订单：   + 订单编号：1   + 客户电话：13812345678   + 客户地址：北京市朝阳区   + 菜名：红烧肉   + 数量：3   + 总价：115.50（38.50 \* 3） * 菜品库存更新为：17。 * 控制台输出：菜品已加入购物车！。   **5. 下单 (place\_order)**  **测试数据**   * 假设购物车中有以下订单：   1. 1 13812345678 北京市朝阳区 红烧肉 3 115.50   **预期结果**   * 订单成功提交，购物车清空。 * 订单文件 orders.txt 中新增一行：1 13812345678 北京市朝阳区 红烧肉 3 115.50。 * 控制台输出：下单成功！。    **6. 模拟支付 (simulate\_payment)**  **测试数据**   * 假设订单中有以下记录：   1. 1 13812345678 北京市朝阳区 红烧肉 3 115.50   2. 2 13987654321 上海市浦东新区 鱼香肉丝 2 56.00   **预期结果**   * 控制台输出：模拟支付完成，总价: 171.50。   **7. 营业额分析 (analyze\_revenue)**  **测试数据**   * 假设订单中有以下记录：   1. 1 13812345678 北京市朝阳区 红烧肉 3 115.50   2. 2 13987654321 上海市浦东新区 鱼香肉丝 2 56.00   **预期结果**   * 控制台输出：总营业额: 171.50。   **8. 按单价排序菜单 (sort\_menu\_by\_price)**  **测试数据**   * 假设菜单中有以下菜品：   1. 1 红烧肉 荤菜 38.50 20   2. 2 鱼香肉丝 荤菜 28.00 30   3. 3 清炒时蔬 素菜 18.00 50   **预期结果**   * 菜单按单价升序排序后：   1. 3 清炒时蔬 素菜 18.00 50   2. 2 鱼香肉丝 荤菜 28.00 30   3. 1 红烧肉 荤菜 38.50 20 * 控制台输出：菜单已按单价排序！。   **9. 退出系统**  **测试数据**   * 选择退出选项：9   **预期结果**   * 菜单和订单数据保存到文件。 * 控制台输出：退出系统。。 |
| 5.心得体会  【此处根据自己的实际情况写出本项目开发过程中遇到的问题，以及相应的解决方法，及其他心得体会】   * 1. **函数命名不准确，导致后期经常混杂；**   2. **指针掌握不熟练；**   3. **调试不熟练；**   **解决方法：**   1. **求助AI；** 2. **求助同学；** |
| 6.评分（内容完备性60%、格式规范性30%、创新性10%） |