****

**《程序设计基础》课程设计**

设计题目： 课题3：职工工资管理系统

院 系： 计算机学院

班 级： 人工智能22-H1班

设 计 者： 秦思佳，王双双

学 号： 202202911359133，202202911359137

指导教师： 于业斌，邢雪霞

设计时间： 2023/8/20

目录

[1. 目的和要求 1](#_Toc30569)

[1.1. 设计内容目的 1](#_Toc30576)

[1.2. 设计要求 1](#_Toc18279)

[2. 需求分析 2](#_Toc29389)

[3. 方案设计 3](#_Toc15464)

[4. 详细设计 4](#_Toc25378)

[5. 实现（包含系统运行截图及完整代码） 5](#_Toc28552)

[5.1. 程序代码： 5](#_Toc26374)

[5.2. 程序截图 9](#_Toc3882)

[6. 总结（心得体会及收获） 9](#_Toc31962)

# 目的和要求

## 设计内容目的

1. 项目内容

在计算机中建立相应的数据结构，利用程序实现工资的输入、查询、增、删、改、统计、输出等操作。设计一个能够记录公司员工工资的数据结构（比如包括：员工号，姓名，基本工资，补贴金额，奖励金额，扣除金额、实发工资等）、文件（文件作为工资数据的保存介质）。其中：实发工资=基本工资＋补贴金额＋奖励金额-扣除金额

1. 项目目的：

熟悉VScode的开发环境，掌握基本的编程方法，熟悉用C语言完成一个应用程序的设计过程，掌握有关编辑和调试程序的方法和技巧；

主要掌握C语言中的结构体、指针及文件操作等内容；

了解开发简单应用程序的全过程，包括：分析、设计、编程、调试和编写设计报告。

## 设计要求

1. 应用文件保存工资管理数据。
2. 新增：能将所有输入的员工工资数据保存到文件中，并可选择继续输入或退出。
3. 查找：输入员工的员工号，并在文件中查找。若找到，则显示该员工的工资详情，否则提示未 找到。无论查找成功否，应还可选择继续查找或退出。
4. 删除：输入员工的员工号，并在文件中查找。若找到，提示是否确定删除，若选择是，则删除 文件中该员工信息，若选择否，则取消，否则提示无此员工信息，无论查找成功否，应还可选择 继续查找或退出。
5. 修改：输入员工的员工号，并在文件中查找。若找到，显示该员工的所有信息，并可进行输入。 完成后将现输入的内容覆盖原内容保存到文件中。否则提示无此员工信息，无论查找成功否，应 还可选择继续查找或退出。
6. 统计：分类汇总并输出所有员工的基本工资、补贴金额、奖励金额、扣除金额、实发工资的总 金额和平均金额。
7. 查看：能够看到所有录入的员工信息
8. 退出：退出程序

# 需求分析

对系统进行功能性分析、技术分析、可行性分析等

## 功能性分析

新建一个文本文件，文件名为：“职工信息”将每一个职工的姓名、职工号以及基本工资、补贴金额、奖励金额、扣除金额、实发工资，作为一个记录。该软件能建立一个新的数据文件或给已建立好的数据文件增加记录。

**数据示例：**

**员工号: 1**

**姓名: 王双双**

**基本工资: 1000.00**

**补贴金额: 1000.00**

**奖励金额: 10000.00**

**扣除金额: 10.00**

**实发工资: 11990.00**

**员工号: 2**

**姓名: 秦思佳**

**基本工资: 2000.00**

**补贴金额: 200.00**

**奖励金额: 2000.00**

**扣除金额: 20.00**

**实发工资: 4180.00**

**（注：请大家将该文本文件后保存到自己的存储设备，运行程序前请保存相应的副本）**

在主函数中读入原始数据文件的内容，保存在对应的结构体数组中，然后显示菜单，利用菜单调用各个函数，分别完成如下功能：(建议数据文件和程序文件放在同一目录中，使用fopen函数以只读的方式打开文件)。

1. 对职工信息实现查询和显示功能：计算个人工实发工资=基本工资＋补贴金额＋奖励金额-扣除金额并按工资条的形式输出（输出格式自行设计）。
2. 修改记录：可以对数据文件的任意记录的数据进行修改并在修改前后对记录内容进行显示。
3. 添加记录：录入新的职工信息数据。在录入新的信息前，首先判断该职工号是否存在，如果不存在，则可进行数据录入并依次输入职工的相关信息；如果存在，则提示不能进行添加操作。
4. 删除记录：可删除数据文件中的任一记录。

## 技术分析

1. 文件读写操作：使用C语言中的文件操作函数，如fopen、fread、fwrite等，来读取和写入职工信息到数据文件中。
2. 结构体：定义一个结构体来表示每个职工的信息，包括员工号、姓名、基本工资、补贴金额、奖励金额、扣除金额等字段。
3. 菜单界面：使用printf和scanf函数创建一个用户友好的命令行菜单界面，以便用户选择不同的操作。

## 可行性分析

1. 数据量估算：根据数据示例，每个职工的信息都包括若干字段，但并不多，适合存储在文本文件中。系统应该能够处理大量的职工信息。
2. 易用性：命令行界面对于这种小型的管理系统是足够的，用户可以通过简单的命令进行各种操作。
3. 稳定性：使用文件来存储数据，数据持久性较好，即使程序退出，数据也不会丢失。
4. 可维护性：代码模块化，易于维护和扩展。

# 方案设计

对每个功能模块进行介绍（包含系统功能组成框图）

图1 系统功能模块框图

主函数main()

菜单函数menu（）

void saveDataToFile()

录入

void addEmployee()

新增

void findEmployee()查找

void deleteEmployee()

删除

void modifyEmployee()

修改

void calculateStatistics()

统计

void viewAllEmployees()

查看

void Eight()

删除

职工工资管理

职工信息录入

职工信息查询

计算工资

添加信息

修改信息

删除信息

统计

按姓名查询

按职工号查询

实发工资、应发工资

图2 系统框图

# 详细设计

1. 首先是定义结构体

// 定义员工结构体

struct Employee {

    char emp\_id[20];

    char name[50];

    float basic\_salary;

    float allowance;

    float bonus;

    float deduction;

};

1. 数据的保存

// 保存数据到文本文件

void saveDataToFile() {

    FILE\* file = fopen("C:\\Users\\HP\\Desktop\\class 3\\职员信息.txt", "a");

    if (file == NULL) {

        printf("无法打开文件。\n");

        return;

    }

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

        fprintf(file, "员工号: %s\n", employees[i].emp\_id);

        fprintf(file, "姓名: %s\n", employees[i].name);

        fprintf(file, "基本工资: %.2f\n", employees[i].basic\_salary);

        fprintf(file, "补贴金额: %.2f\n", employees[i].allowance);

        fprintf(file, "奖励金额: %.2f\n", employees[i].bonus);

        fprintf(file, "扣除金额: %.2f\n", employees[i].deduction);

        fprintf(file, "实发工资: %.2f\n", employees[i].basic\_salary + employees[i].allowance + employees[i].bonus - employees[i].deduction);

        fprintf(file, "\n");

    }

    fclose(file);

}

1. 其他功能的模块化实现

# 实现（包含系统运行截图及完整代码）

## 程序代码：

C语言代码：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// 定义员工结构体

struct Employee {

    char emp\_id[20];

    char name[50];

    float basic\_salary;

    float allowance;

    float bonus;

    float deduction;

};

// 员工数组和员工数量

struct Employee employees[100];

int employeeCount = 0;

// 保存数据到文本文件

void saveDataToFile() {

    FILE\* file = fopen("C:\\Users\\HP\\Desktop\\class 3\\职员信息.txt", "a");

    if (file == NULL) {

        printf("无法打开文件。\n");

        return;

}

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

        fprintf(file, "员工号: %s\n", employees[i].emp\_id);

        fprintf(file, "姓名: %s\n", employees[i].name);

        fprintf(file, "基本工资: %.2f\n", employees[i].basic\_salary);

        fprintf(file, "补贴金额: %.2f\n", employees[i].allowance);

        fprintf(file, "奖励金额: %.2f\n", employees[i].bonus);

        fprintf(file, "扣除金额: %.2f\n", employees[i].deduction);

        fprintf(file, "实发工资: %.2f\n", employees[i].basic\_salary + employees[i].allowance + employees[i].bonus - employees[i].deduction);

        fprintf(file, "\n");

}

    fclose(file);

}

// 新增员工

void addEmployee() {

struct Employee newEmployee;

    printf("请输入员工号: ");

    scanf("%s", newEmployee.emp\_id);

    printf("请输入姓名: ");

    scanf("%s", newEmployee.name);

    printf("请输入基本工资: ");

    scanf("%f", &newEmployee.basic\_salary);

    printf("请输入补贴金额: ");

    scanf("%f", &newEmployee.allowance);

    printf("请输入奖励金额: ");

    scanf("%f", &newEmployee.bonus);

    printf("请输入扣除金额: ");

scanf("%f", &newEmployee.deduction);

employees[employeeCount++] = newEmployee;

    printf("员工信息已保存。\n");

}

// 查找员工

void findEmployee() {

    char empIdToFind[20];

    printf("请输入要查找的员工号: ");

scanf("%s", empIdToFind);

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

        if (strcmp(empIdToFind, employees[i].emp\_id) == 0) {

            printf("找到员工信息：\n");

            printf("员工号: %s\n", employees[i].emp\_id);

            printf("姓名: %s\n", employees[i].name);

            printf("基本工资: %.2f\n", employees[i].basic\_salary);

            printf("补贴金额: %.2f\n", employees[i].allowance);

            printf("奖励金额: %.2f\n", employees[i].bonus);

            printf("扣除金额: %.2f\n", employees[i].deduction);

            printf("实发工资: %.2f\n", employees[i].basic\_salary + employees[i].allowance + employees[i].bonus - employees[i].deduction);

            return;

        }

}

    printf("未找到该员工信息。\n");

}

// 删除员工

void deleteEmployee() {

    char empIdToDelete[20];

    printf("请输入要删除的员工号: ");

scanf("%s", empIdToDelete);

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

        if (strcmp(empIdToDelete, employees[i].emp\_id) == 0) {

            // 将该员工从数组中移除

            for (int j = i; j < employeeCount - 1; j++) {

                employees[j] = employees[j + 1];

            }

            employeeCount--;

            printf("员工信息已删除。\n");

            return;

        }

}

    printf("未找到要删除的员工信息。\n");

}

// 修改员工信息

void modifyEmployee() {

    char empIdToModify[20];

    printf("请输入要修改的员工号: ");

scanf("%s", empIdToModify);

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

        if (strcmp(empIdToModify, employees[i].emp\_id) == 0) {

            printf("员工原信息如下：\n");

            printf("员工号: %s\n", employees[i].emp\_id);

            printf("姓名: %s\n", employees[i].name);

            printf("基本工资: %.2f\n", employees[i].basic\_salary);

            printf("补贴金额: %.2f\n", employees[i].allowance);

            printf("奖励金额: %.2f\n", employees[i].bonus);

            printf("扣除金额: %.2f\n", employees[i].deduction);

            printf("实发工资: %.2f\n", employees[i].basic\_salary + employees[i].allowance + employees[i].bonus - employees[i].deduction);

            printf("请输入新的员工信息：\n");

            printf("请输入姓名: ");

            scanf("%s", employees[i].name);

            printf("请输入基本工资: ");

            scanf("%f", &employees[i].basic\_salary);

            printf("请输入补贴金额: ");

            scanf("%f", &employees[i].allowance);

            printf("请输入奖励金额: ");

            scanf("%f", &employees[i].bonus);

            printf("请输入扣除金额: ");

            scanf("%f", &employees[i].deduction);

            printf("员工信息已修改。\n");

            return;

        }

}

    printf("未找到要修改的员工信息。\n");

}

// 统计员工工资信息

void calculateStatistics() {

    float totalBasicSalary = 0.0;

    float totalAllowance = 0.0;

    float totalBonus = 0.0;

    float totalDeduction = 0.0;

    float totalNetSalary = 0.0;

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

        totalBasicSalary += employees[i].basic\_salary;

        totalAllowance += employees[i].allowance;

        totalBonus += employees[i].bonus;

        totalDeduction += employees[i].deduction;

        totalNetSalary += employees[i].basic\_salary + employees[i].allowance + employees[i].bonus - employees[i].deduction;

}

    printf("所有员工的工资统计信息如下：\n");

    printf("基本工资总额: %.2f\n", totalBasicSalary);

    printf("补贴总额: %.2f\n", totalAllowance);

    printf("奖励总额: %.2f\n", totalBonus);

    printf("扣除总额: %.2f\n", totalDeduction);

    printf("实发工资总额: %.2f\n", totalNetSalary);

    printf("平均实发工资: %.2f\n", totalNetSalary / employeeCount);

}

// 查看所有职员信息

void viewAllEmployees() {

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

        printf("员工号: %s\n", employees[i].emp\_id);

        printf("姓名: %s\n", employees[i].name);

        printf("基本工资: %.2f\n", employees[i].basic\_salary);

        printf("补贴金额: %.2f\n", employees[i].allowance);

        printf("奖励金额: %.2f\n", employees[i].bonus);

        printf("扣除金额: %.2f\n", employees[i].deduction);

        printf("实发工资: %.2f\n", employees[i].basic\_salary + employees[i].allowance + employees[i].bonus - employees[i].deduction);

        printf("\n");

    }

}

int main() {

int choice;

    do {

        printf("职工工资管理系统\n");

        printf("1. 新增员工\n");

        printf("2. 查找员工\n");

        printf("3. 删除员工\n");

        printf("4. 修改员工信息\n");

        printf("5. 统计员工工资信息\n");

        printf("6. 查看所有职员信息\n");

        printf("7. 退出\n");

        printf("请选择操作: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                addEmployee();

                break;

            case 2:

                findEmployee();

                break;

            case 3:

                deleteEmployee();

                break;

            case 4:

                modifyEmployee();

                break;

            case 5:

                calculateStatistics();

                break;

            case 6:

                viewAllEmployees();

                break;

            case 7:

                // 在程序结束前保存数据到文本文件

                saveDataToFile();

                printf("程序已退出。\n");

                break;

            default:

                printf("无效的选项，请重新选择。\n");

                break;

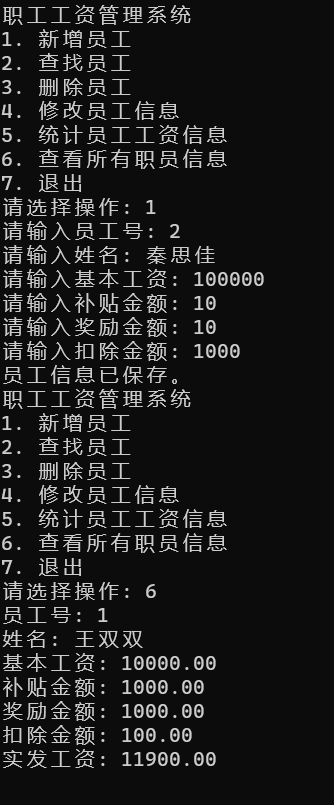
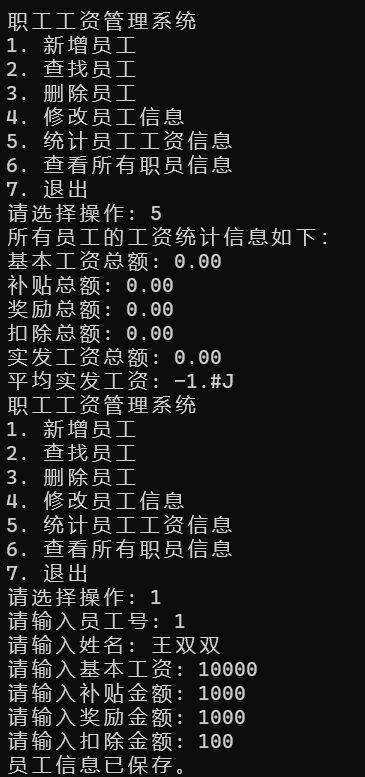
        }

} while (choice != 7);

    return 0;

}

## 程序截图



# 总结（心得体会及收获）

经过两个星期的上机实践学习，使我对c语言有了更进一步的认识和了解，要想学好它要重在实践，要透过不断的上机操作才能更好地学习它，透过实践，我也发现我的好多不足之处，首先是自己在指法上还不行，经常按错字母，透过学习也有所改善;再有对c语言的一些标准库函数不太了解，还有对函数调用的正确使用不够熟悉，还有对c语言中经常出现的错误也不了解，透过实践，使我在这几个方面的认识有所提高。透过实践的学习，我认到学好计算机要重视实践操作，不仅仅仅是学习c语言，还是其它的语言，以及其它的计算机方面的知识都要重在实践，所以后在学习过程中，我会更加注视实践操作，使自己便好地学好计算机。

# 参考文献

《计算机操作系统（第四版）》教材

《数据结构》教材

《C语言程序设计》教材

百度文库