职工管理系统代码文档

main.cpp

```cpp
#include <iostream>
#include <stdlib.h>

#include "workerManager.h"
#include "worker.h"
#include "employee.h"
#include "manager.h"
#include "boss.h"

using namespace std;

int main(){
    //实例化管理者对象
    workerManager wm;
    int choice = 0; //用来存储用户的选项
    while(true){
        //调用展示菜单成员函数
        wm.showMenu();

 if(choice==0||choice==1||choice==2||choice==3||choice==4||choice==5||choice==6||choice==7)
            cout<<"请输入选择项：";
        else
            cout<<"当前选项不存在,请重新输入选择项：";
        cin>>choice;
        cout<<endl;
        switch(choice){
        case 0: //退出系统
            wm.exitSystem();
            break;
        case 1: //增加职工
            wm.addEmp();
            break;
        case 2: //显示职工
            wm.showEmp();
            break;
        case 3: //删除职工
            wm.delEmp();
            break;
        case 4: //修改职工
            wm.modEmp();
            break;
        case 5: //查找职工
            wm.findEmp();
```

```
            break;
        case 6: //排序职工
            wm.sortEmp();
            break;
        case 7: //清空文档
            wm.cleanFile();
            break;
        default:
            system("cls");  //清屏
            break;
        }
    }
    system("pause");    //包含在头文件<stdlib.h>中
    return 0;
}

//    初始数据
//    000000  张三  3
//    000001  李四  3
//    000021  赵希抟  2
//    000023  王五  1
//    000030  Lisa  1
//    000450  赵六  2
//    000620  李四  2
//    003200  Marray  2


//    测试代码
//    worker * people = NULL;
//    people = new employee(1,"张三",1);
//    people->showInfo();
//    delete people;
//    people = new manager(2,"李四",2);
//    people->showInfo();
//    delete people;
//    people = new boss(3,"王五",3);
//    people->showInfo();
//    delete people;
//    return 0;

```

boss.cpp

```
#include "boss.h"

//构造函数
boss::boss(string id,string name,int deptid){
    this->m_id = id;
    this->m_name = name;
    this->m_deptid = deptid;
}
//显示个人信息
```

```cpp
void boss::showInfo(){
//cout<<"职工编号: "<<this->m_id<<"\t职工姓名: "<<this->m_name<<"\t  在职岗位: "<<this->getDeptName()<<"\t职位编号: "<<this->m_deptid<<"\t岗位职责: 完成公司老板的任务"<<endl;
    std::ostringstream oss;
    oss <<"职工编号: "<<std::left<<std::setw(6) <<std::setfill('0')<<this->m_id<<"\t职工姓名: "<<std::left<<std::setw(8)<<std::setfill('#')<<this->m_name<<"\t  在职岗位: "<<this->getDeptName()<<"\t职位编号: "<<this->m_deptid<<"\t岗位职责: 完成普通职工的任务"<<endl;
    std::cout << oss.str();
}

//获取岗位名称
string boss::getDeptName(){
    return string("公司老板");
}

```

employee.cpp

```cpp
#include "employee.h"


//构造函数
employee::employee(string id,string name,int deptid){
    this->m_id = id;
    this->m_name = name;
    this->m_deptid = deptid;
}
//显示个人信息
void employee::showInfo(){
    //cout<<"职工编号: "<<setw(4)<<this->m_id<<"\t职工姓名: "<<this->m_name<<"\t  在职岗位: "<<this->getDeptName()<<"\t职位编号: "<<this->m_deptid<<"\t岗位职责: 完成普通职工的任务"<<endl;
    std::ostringstream oss;
    oss <<"职工编号: "<<std::left<<std::setw(6) <<std::setfill('0')<<this->m_id<<"\t职工姓名: "<<std::left<<std::setw(8)<<std::setfill('#')<<this->m_name<<"\t  在职岗位: "<<this->getDeptName()<<"\t职位编号: "<<this->m_deptid<<"\t岗位职责: 完成普通职工的任务"<<endl;
    std::cout << oss.str();
}
//获取岗位名称
string employee::getDeptName(){
    return string("公司员工");  //相当于return "公司员工";
}
```

boss.h

```cpp
#ifndef BOSS_H_INCLUDED
#define BOSS_H_INCLUDED
```

```

#include "worker.h"
#pragma once
#include <iostream>
using namespace std;

//老板类
class boss : public worker{
public:
    //构造函数
    boss(string id,string name,int deptid);
    //显示个人信息
    virtual void showInfo();
    //获取岗位名称
    virtual string getDeptName();
};

#endif // BOSS_H_INCLUDED
```

employee.h

```
#ifndef BOSS_H_INCLUDED
#define BOSS_H_INCLUDED

#include "worker.h"
#pragma once
#include <iostream>
using namespace std;

//老板类
class boss : public worker{
public:
    //构造函数
    boss(string id,string name,int deptid);
    //显示个人信息
    virtual void showInfo();
    //获取岗位名称
    virtual string getDeptName();
};

#endif // BOSS_H_INCLUDED
```

manager.h

```
#ifndef MANAGER_H_INCLUDED
#define MANAGER_H_INCLUDED

#include "worker.h"
#pragma once
#include <iostream>
```

```cpp
using namespace std;

//经理类
class manager : public worker{
public:
    //构造函数
    manager(string id,string name,int deptid);
    //显示个人信息
    virtual void showInfo();
    //获取岗位名称
    virtual string getDeptName();
};

#endif // MANAGER_H_INCLUDED
```

worker.h

```cpp
#ifndef WORKER_H_INCLUDED
#define WORKER_H_INCLUDED

#pragma once
#include <iostream>
#include <string>
#include <iomanip>
#include <sstream>
#include <ios>

using namespace std;

//职工抽象类
class worker{
public:
    //显示个人信息
    virtual void showInfo() = 0;
    //获取岗位名称
    virtual string getDeptName() = 0;
    //职工编号 默认六位 不足补0
    string m_id;
    //职工姓名
    string m_name;
    //部门编号
    int m_deptid;
};

#endif // WORKER_H_INCLUDED
```

workerManager.h

```cpp
#ifndef WORKERMANAGER_H_INCLUDED
#define WORKERMANAGER_H_INCLUDED
```

```cpp

#pragma once    //防止头文件重复包含
#include <iostream>     //包含输入输出流头文件
#include <stdlib.h>
#include "worker.h"
#include "manager.h"
#include "employee.h"
#include "boss.h"
#include <fstream>
#define FILENAME "empfile.txt"

using namespace std;    //使用标准命名空间

class workerManager{
public:
    //构造函数
    workerManager();

    //展示菜单
    void showMenu();

    //退出系统
    void exitSystem();

    //析构函数
    ~workerManager();

    //保存文件
    void saveFile();

    //添加职工
    void addEmp();

    //统计人数
    int getEmpnum();

    //初始化职工数据
    void initEmp();

    //判断文件是否为空的标志
    bool m_fileisempty;

    //记录职工人数
    int m_empnum;

    //打印职工信息
    void showEmp();

    //删除职工
    void delEmp();

    //判断职工编号是否存在  存在返回职工所在数组位置  不存在返回-1
    int isExist(string id);
```

```
56
57      //修改职工
58      void modEmp();
59
60      //查找职工
61      void findEmp();
62
63      //编号排序
64      void sortEmp();
65
66      //清空文件
67      void cleanFile();
68
69      //职工数组指针
70      worker ** m_emparray;   //存储指向worker类指针的数组
71  };
72
73  #endif // WORKERMANAGER_H_INCLUDED
```