# 职工管理系统代码文档

### main.cpp

```
1 #include <iostream>
 2
   #include <stdlib.h>
 3
 4 #include "workerManager.h"
   #include "worker.h"
 6 #include "employee.h"
 7
   #include "manager.h"
   #include "boss.h"
8
 9
10
    using namespace std;
11
12
   int main(){
13
       //实例化管理者对象
14
       workerManager wm;
15
       int choice = 0; //用来存储用户的选项
16
       while(true){
17
            //调用展示菜单成员函数
18
            wm.showMenu();
19
     if(choice==0||choice==1||choice==2||choice==3||choice==4||choice==5||choice==6||choic
    e = 7
20
                cout<<"请输入选择项:";
21
            else
                cout<<"当前选项不存在,请重新输入选择项:";
22
23
            cin>>choice;
24
            cout<<endl;</pre>
25
            switch(choice){
26
            case 0: //退出系统
27
               wm.exitSystem();
28
               break;
29
            case 1: //增加职工
30
               wm.addEmp();
31
               break;
32
            case 2: //显示职工
               wm.showEmp();
33
34
               break:
35
            case 3: //删除职工
36
               wm.delEmp();
37
               break;
38
            case 4: //修改职工
39
               wm.modEmp();
40
               break;
            case 5: //查找职工
41
42
               wm.findEmp();
```

```
43
               break;
44
           case 6: //排序职工
45
               wm.sortEmp();
46
               break;
47
           case 7: //清空文档
48
               wm.cleanFile();
49
               break;
50
           default:
               system("cls"); //清屏
51
52
               break;
53
           }
54
       }
55
        system("pause"); //包含在头文件<stdlib.h>中
        return 0;
56
57
    }
58
59
   //
         初始数据
60
   //
         000000 张三 3
         000001 李四 3
61
   //
         000021 赵希抟 2
62
    //
63
         000023 王五 1
   //
64
   //
         000030 Lisa 1
65
   //
         000450 赵六 2
         000620 李四 2
66
   //
67
    //
         003200 Marray 2
68
69
   //
70
         测试代码
71
   //
         worker * people = NULL;
72
    //
         people = new employee(1,"张三",1);
73
   //
         people->showInfo();
74
    //
         delete people;
75
    //
         people = new manager(2,"李四",2);
76
   //
         people->showInfo();
    //
         delete people;
77
78
         people = new boss(3,"王五",3);
   //
79
    //
         people->showInfo();
80
   //
         delete people;
81
   //
         return 0;
82
```

#### boss.cpp

```
#include "boss.h"
1
2
3
  //构造函数
  boss::boss(string id,string name,int deptid){
4
5
       this->m_id = id;
6
       this->m_name = name;
7
       this->m_deptid = deptid;
8
   }
   //显示个人信息
```

```
10 void boss::showInfo(){
   //cout<<"职工编号: "<<this->m_id<<"\tp工姓名: "<<this->m_name<<"\t 在职岗位: "<<this-
11
   >getDeptName()<<"\t职位编号: "<<this->m_deptid<<"\t岗位职责: 完成公司老板的任务"<<end1;
12
       std::ostringstream oss;
13
       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"
    <<end1;
       std::cout << oss.str();</pre>
14
   }
15
16
   //获取岗位名称
17
18
   string boss::getDeptName(){
       return string("公司老板");
19
20
   }
21
```

### employee.cpp

```
#include "employee.h"
 2
 3
 4
   //构造函数
 5
   employee::employee(string id,string name,int deptid){
       this->m_id = id;
 6
 7
       this->m_name = name;
 8
       this->m_deptid = deptid;
 9
   }
   //显示个人信息
10
   void employee::showInfo(){
11
       //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"
    <<end1;
       std::ostringstream oss;
13
       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"
    <<end1;
15
       std::cout << oss.str();</pre>
16
   }
   //获取岗位名称
17
18
   string employee::getDeptName(){
       return string("公司员工"); //相当于return "公司员工";
19
20
   }
```

#### boss.h

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

```
3
   #include "worker.h"
 4
 5 #pragma once
 6 #include <iostream>
 7
   using namespace std;
 8
 9
   //老板类
   class boss : public worker{
10
11
   public:
12
      //构造函数
13
      boss(string id,string name,int deptid);
14
      //显示个人信息
       virtual void showInfo();
      //获取岗位名称
16
17
       virtual string getDeptName();
18 };
19
20 #endif // BOSS_H_INCLUDED
```

### employee.h

```
1 #ifndef BOSS_H_INCLUDED
 2 #define BOSS_H_INCLUDED
 3
 4 #include "worker.h"
5 #pragma once
 6 | #include <iostream>
7
   using namespace std;
8
   //老板类
10 class boss : public worker{
   public:
11
12
      //构造函数
13
     //显示个人信息
     boss(string id,string name,int deptid);
14
15
      virtual void showInfo();
      //获取岗位名称
16
17
       virtual string getDeptName();
18 };
19
20 #endif // BOSS_H_INCLUDED
```

### manager.h

```
#ifndef MANAGER_H_INCLUDED

#define MANAGER_H_INCLUDED

#include "worker.h"

#pragma once
#include <iostream>
```

```
using namespace std;
8
9
   //经理类
   class manager : public worker{
10
11
   public:
       //构造函数
12
13
     manager(string id,string name,int deptid);
14
       //显示个人信息
15
      virtual void showInfo();
       //获取岗位名称
16
17
       virtual string getDeptName();
18 };
19
20 #endif // MANAGER_H_INCLUDED
```

#### worker.h

```
1 #ifndef WORKER_H_INCLUDED
 2
   #define WORKER_H_INCLUDED
 3
 4 #pragma once
 5 #include <iostream>
 6 #include <string>
   #include <iomanip>
   #include <sstream>
 9
   #include <ios>
10
11
   using namespace std;
12
13
   //职工抽象类
14
   class worker{
15
   public:
16
       //显示个人信息
17
      virtual void showInfo() = 0;
      //获取岗位名称
18
19
       virtual string getDeptName() = 0;
20
      //职工编号 默认六位 不足补0
21
       string m_id;
22
      //职工姓名
23
       string m_name;
24
       //部门编号
25
       int m_deptid;
26 };
27
28 #endif // WORKER_H_INCLUDED
```

## workerManager.h

```
#ifndef workermanager_H_included
#define workermanager_H_included
```

```
3
   #pragma once //防止头文件重复包含
 4
 5
   #include <iostream>
                         //包含输入输出流头文件
 6 #include <stdlib.h>
   #include "worker.h"
 7
8
   #include "manager.h"
9 #include "employee.h"
10 #include "boss.h"
   #include <fstream>
11
12
   #define FILENAME "empfile.txt"
13
14
   using namespace std; //使用标准命名空间
15
   class workerManager{
16
17
   public:
18
       //构造函数
19
       workerManager();
20
21
       //展示菜单
22
       void showMenu();
23
24
       //退出系统
25
       void exitSystem();
26
27
       //析构函数
28
       ~workerManager();
29
       //保存文件
30
31
       void saveFile();
32
33
       //添加职工
34
       void addEmp();
35
36
       //统计人数
37
       int getEmpnum();
38
39
       //初始化职工数据
40
       void initEmp();
41
42
       //判断文件是否为空的标志
43
       bool m_fileisempty;
44
45
       //记录职工人数
46
       int m_empnum;
47
48
       //打印职工信息
49
       void showEmp();
50
51
       //删除职工
52
       void delEmp();
53
54
       //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
55
       int isExist(string id);
```

```
56
57
       //修改职工
58
       void modEmp();
59
60
       //查找职工
61
       void findEmp();
62
       //编号排序
63
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
```