职工分三类：普通职工，经理，老板

成员数据：职工编号、姓名、岗位（三种岗位）、职责

初始化岗位之后自动确认职责

普通职工：完成经理下发的任务

经理：完成老板下发的任务，并下发任务给员工

老板：管理公司的所有事务

完成每一步操作之后，先暂停一下，按任意键继续，再返回菜单

需要功能：

退出系统

增加职工信息（可以批量添加，即先获取要添加的人数）

添加完之后自动写入文件

显示职工信息

删除离职职工

删除之后文件里面也得同步删除

修改职工信息（按照职工编号）

查找职工信息：选择通过姓名/编号

按照编号排序：升序还是降序由操作者决定（请选择排序方式）

文件也得同步排序

清空所有文档：清空文件中的所有信息（清空前需再次确认）

创建管理类：集合用户调用的多种函数

如：选择菜单+增删查改+文件读写交互+退出系统（ExitSystem）

创建三个文件

WorkerManager.h 定义类与成员函数

WorkerManager.cpp 具体实现成员函数

文本

AI 生成的内容可能不正确。

图片包含 图示

AI 生成的内容可能不正确。

Exit(0);//立即终止程序

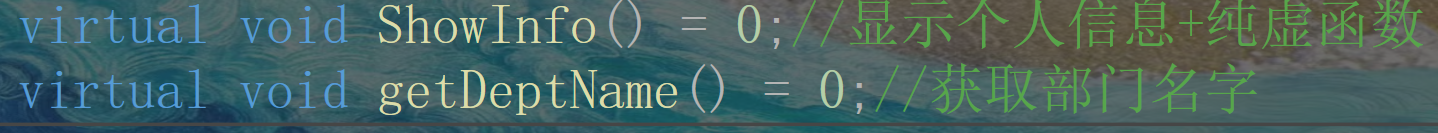
三个种类的职工抽象成Worker类

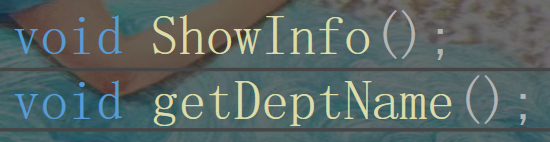
把Worker类定义在头文件Worker.h中，抽象类只需定义，无需实现

属性：员工编号、姓名、所在部门编号

由部门编号获得行为：职责和岗位名称

把普通员工类定义在employee.h中，实现在employee.cpp中



尽管这俩虚函数已经在Worker中定义过，但是Employee中依然要再次定义，可以省略virtual

文本

AI 生成的内容可能不正确。水平制表\t:实际上是在内容之间空几格

添加员工

管理类新增：int记录当前多少员工，Worker\*\*指向数组

数组应开辟Worker\*[5]而非Worker[5],因为抽象类指针指向子类对象实现多态，开辟在堆区，若在栈区，那么出了这个函数，内存会被回收

而new Worker\*[5]返回的是Worker\*\*

要求可以批量添加,故管理类需要一个成员属性：AddNum，根据AddNum来计算新的数组大小newsize(不能直接改变count+=AddNum，因为后续搬运原数组需要根据count确定搬运个数)

根据新的大小来开辟新的数组newspace，还需要判断：若原来数组不为空（或者count>0），需要把原来的数组拷贝到新的空间

由于涉及多态，所以需要先输入所有信息并用临时变量保存，再根据部门编号决定创建什么类型，然后再完成成员属性的初始化，注意是用数组元素接收new返回的地址

图片包含 文本

AI 生成的内容可能不正确。

输入新的职工：职工岗位ID三选一，用到switch语句

有一种更麻烦的写法：

图形用户界面, 应用程序

AI 生成的内容可能不正确。

创建Worker\*临时变量，输入完所有信息后再赋给数组的元素，别忘了delete掉指针

所有员工信息输入完成后，要释放掉原来的arr，再把newspace赋值给arr，newsize赋给count

再输出：成功添加了n名新职工

然后写入文件

文本

AI 生成的内容可能不正确。为了方便使用，我们可以把文件路径写成一个宏

文本

AI 生成的内容可能不正确。

每次打开程序要进行读文件

文本, 信件

AI 生成的内容可能不正确。图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

Ifstream对象有个成员函数：is\_open，文件不存在返回false

第一种情况：文件不存在

我们要在WorkerManager的构造函数中添加从文件读取数据的代码，并且还要判断文件是否存在。若不存在：输出文件不存在，人数初始化为0，FileIsEmpty赋为true，arr指针赋为NULL。此时不要再执行后续的代码，即直接return；

第二种情况：文件存在但是为空

创建一个字符ch，然后把文件第一个字符读进去，再用类的eof函数判断读到的字符是否为空，若为空，除了cout内容不一样，其他操作同上。

复习：读取数据的三种推荐方法（一种>>和两种getline）

注意eof函数不需要传参

FileIsEmpty：在每一次成功添加职工后，将其改为false

第三种情况：文件存在且有数据

我们读入数据也需要创建动态数组，但是数组的大小未知，故需要先实现一个函数来获取文件中成员信息的个数：getEmpNum

如果能同时读到编号、姓名、部门，则num++



文档中的数据是用空格分隔的，用>>来读取代码更简便

实现完之后，把得到的员工个数赋给count

读入数据：实现一个Init\_Emp函数

删除职工Del\_Emp

判断职工是否存在：IsExist，若存在则返回其在数组中的下标，不存在返回-1

这个函数在删除、查找、修改员工中都需要用到，我们单独封装

回到正文：用什么思路实现删除？

逐个前移完成覆盖，同时count-1。所需删除下标为i，则用i+1覆盖i，以此类推

注意：在考虑员工是否存在前，还需要考虑文件是否存在和是否为空

修改职工

找到要修改的职工后，先delete释放这块空间，再重新创建

为什么要重新创建而不能同个Worker\*指针修改呢？因为岗位可能会变化，需要创建新的类

查找职工：按编号/按姓名

需要考虑重名的情况，想想如何写

排序:顺序和倒序

这里用到选择排序算法

清空文件

原理：文本

AI 生成的内容可能不正确。

清空文件的同时，程序内部还要释放数组指针arr[i](堆区上的每个职工对象)以及count归0

Arr是Worker\*\*也得释放，文件是否为空的标准也得置为true

这时想起来：在析构函数里面，仅仅是释放了arr，而没有释放arr[i],我们优化一下

手机屏幕的截图

AI 生成的内容可能不正确。