**数据结构实验报告**

**（二）**

**学院 自动化学院**

**学号 08022311**

**姓名 陈鲲龙**

**日期 2023-11-22**

**实验内容**

本实验对公司组织架构管理进行简单模拟（参考：董事长-CXO-总监-经理-主管-员工），实现查看领导和下属个人信息（包括姓名、年龄、职位、联系方式）、添加员工、删除员工、查询员工等功能。

**实验要求**

1. 要求实现的公司组织架构管理功能：

* 请选择一种合适的数据结构表示公司组织架构，建立员工信息。
* 请用至少3种存储结构来实现，包含孩子兄弟表示法。
* 可实现员工的添加、删除。
* 可修改员工联系方式等信息。
* 可查询员工的领导、同事和下属等信息。
* 按职位高低输出所有员工。
* 输出所有超过35岁的员工。
* 统计公司里面的所有员工数量。

2) 调研该数据结构在实际生活中的其他应用，具体阐述并给出实现思路（可不用代码实现）。

1. **需求分析**

**设计任务：**本实验的需求是用树结构实现一个员工管理的程序，要求能添加、查找、删除、修改员工信息。

**输入规定：**使用者按打印出的提示输入对应项，数字或者字符串，例如年龄是数字，姓名是字符串，若输入不符合要求程序会提示重新输入直到符合格式。

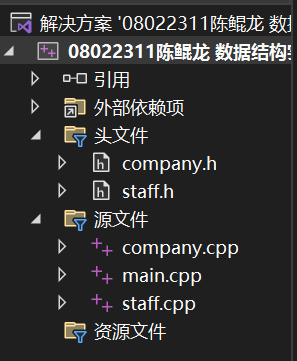
**输出形式：**全基于cout打印程序对应内容

**测试：**程序运行后先添加员工，从老板开始添加，即树结构的第一个根节点。

1. **概要设计**

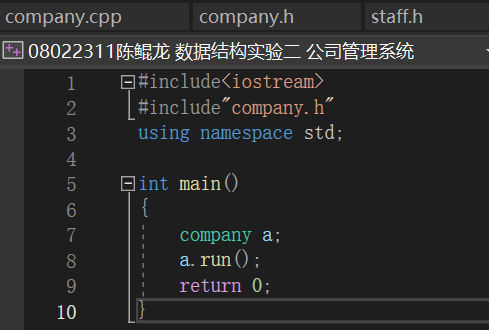
说明程序中用到的所有抽象数据类型定义、主程序流程以及各模块之间的层次（调用）关系。

**本实验架构：**



本项目涉及两个c++类，分别为company用来对公司（树）进行储存或其他操作，和staff用来对员工的信息（结点元素）进行储存或其他操作。

main中先构建company类的一个实例，然后执行其中的run()函数。

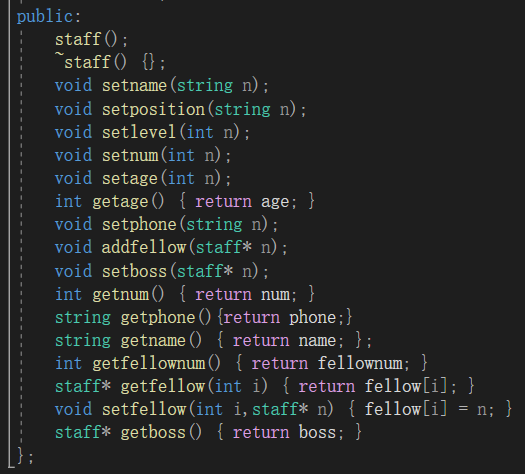


1. **详细设计**
2. 首先看staff类：



每位员工的私有变量都包含工号、级别、名字、年龄、职位、电话、下属名单（元素为staff\*的数组fellow）、下属人数、上司（指针staff\*）

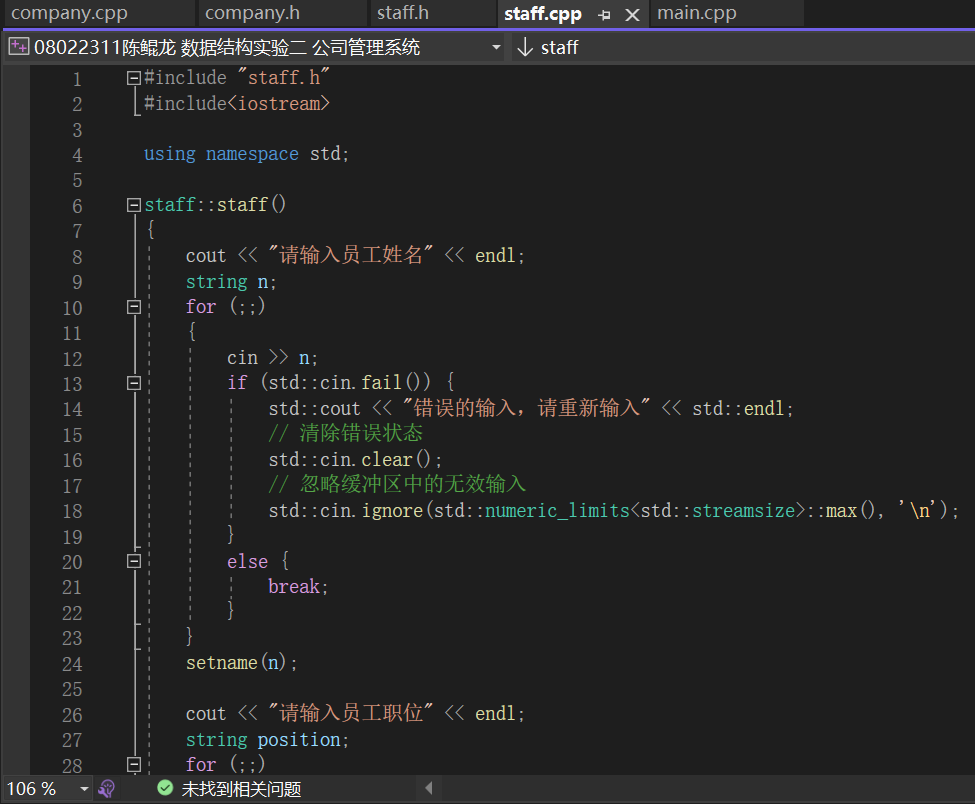
公有函数：



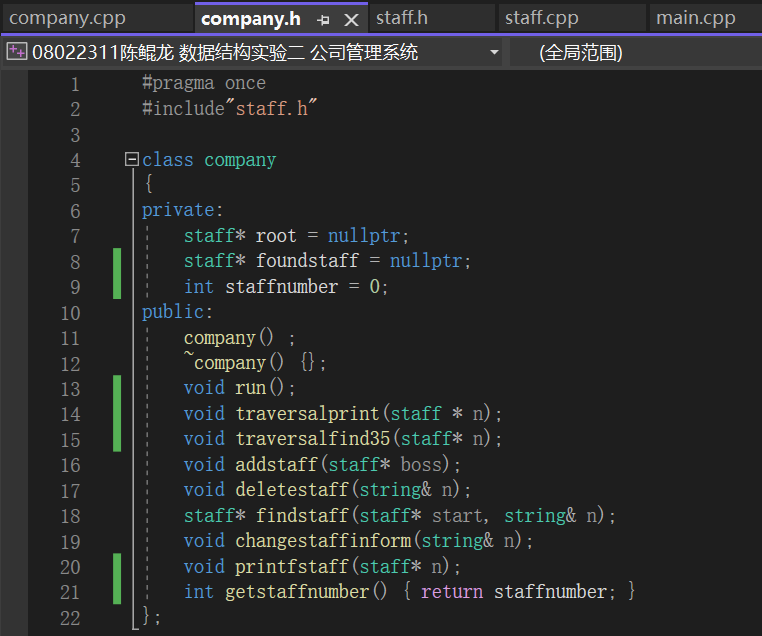
主要功能都是为私有变量的访问get...和修改set...

核心是构造函数

在构造时需要输入员工的各项信息，然后用set函数将变量修改，特别是每次输入都会检查是否符合格式，不符合则重新输入。



2.再看company类：



Private:

私有变量包含树的根节点root

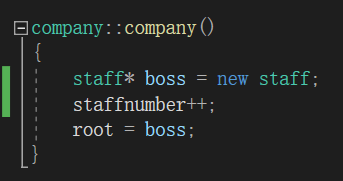
Foundstaff用于查找员工时涉及

以上两者的类型均为指向staff类的指针staff\*

Int变量Staffnumber记录公司一共有多少人

Public:

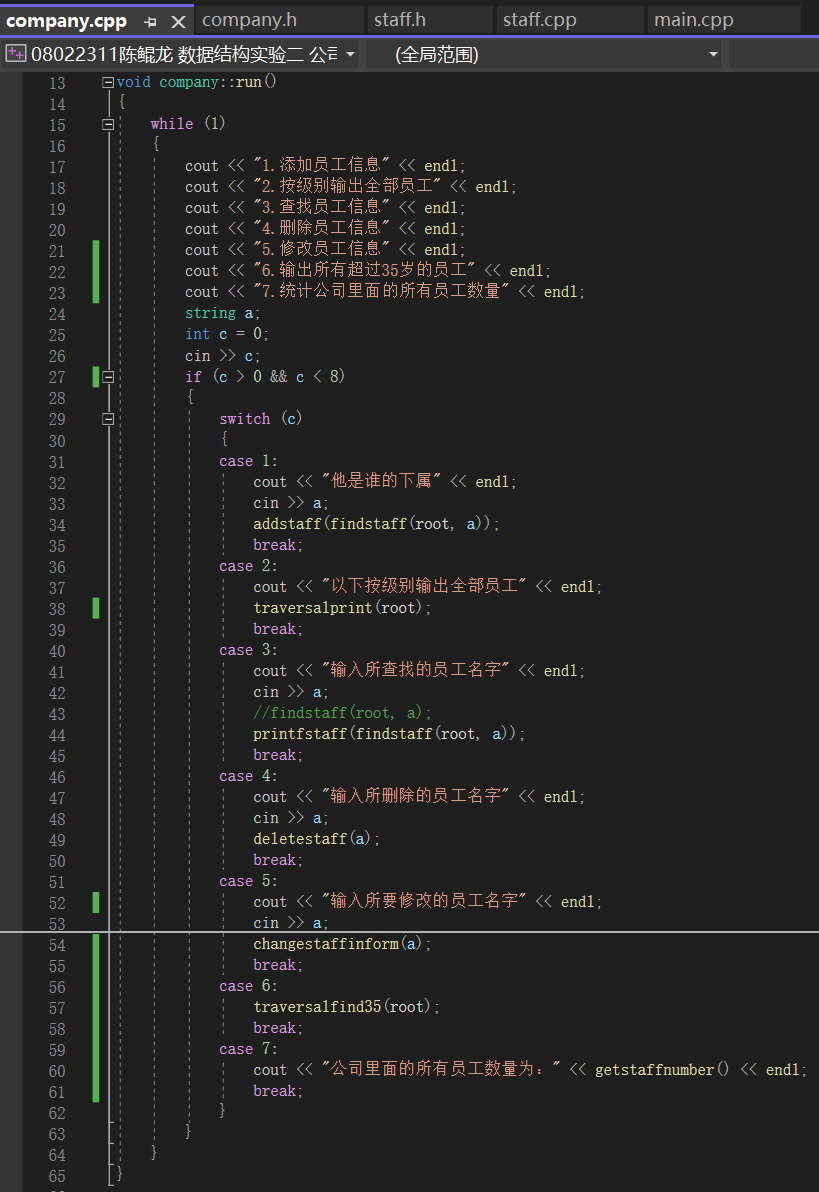
构造函数：



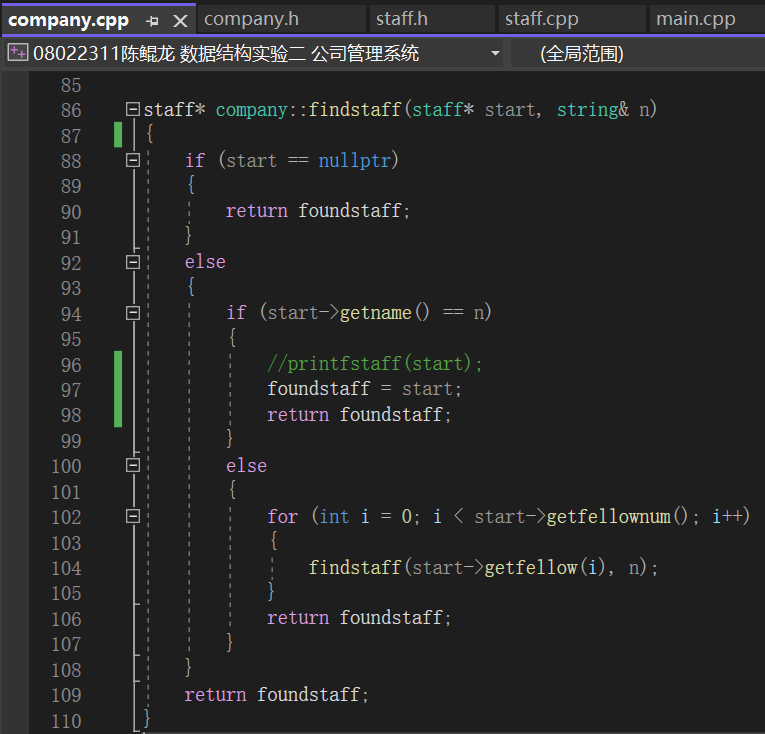
创建公司时首先要有一个老板添加为树的根节点。

Run函数：

使用switch case语句完成选项功能，每个选项下运行对应的函数



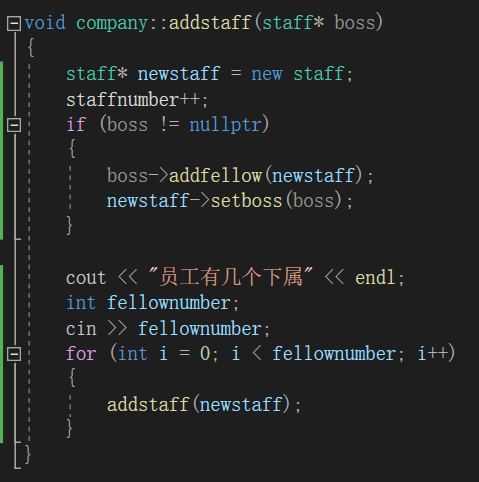
涉及最多最核心最考验指针、递归函数的findstaff函数：



参数Start就是从哪个地址开始找起，字符串n就是查找的判断标准，依据名字查找。

如果找到了就将地址赋值给founndstaff，没找到就是递归调用函数自己，去储存下属的数组里找，没找到就逐级return 并传回foundstaff。

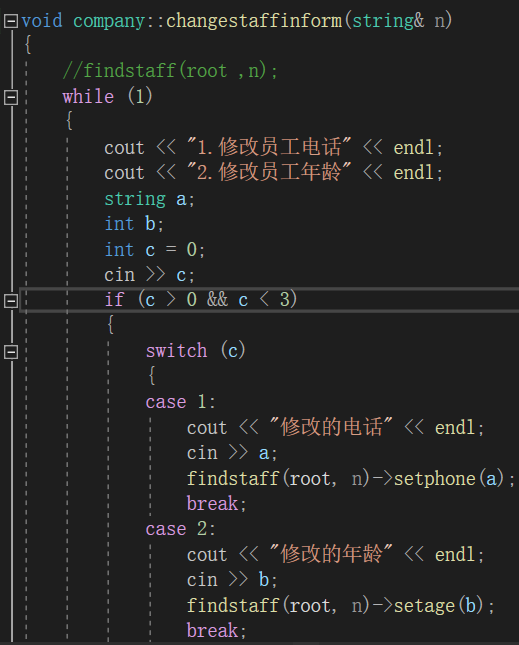
功能1 Case1：添加员工addstaff



在添加员工时，要先找到上司的地址，然后添加在旗下，那么如果这个员工本身也有下属的话，那么就递归再进行添加，即构造一个新的staff类的实例。

修改员工信息功能changeinform函数：

实现逻辑还是先用find函数找到被修改的员工的地址，再调用staff中的set系列函数来修改信息。



打印员工信息函数printstaff：

就是直接用staff的公有函数，获取各项信息，以及上司和下属。



遍历查找年龄大于35岁的员工和遍历树打印全体员工



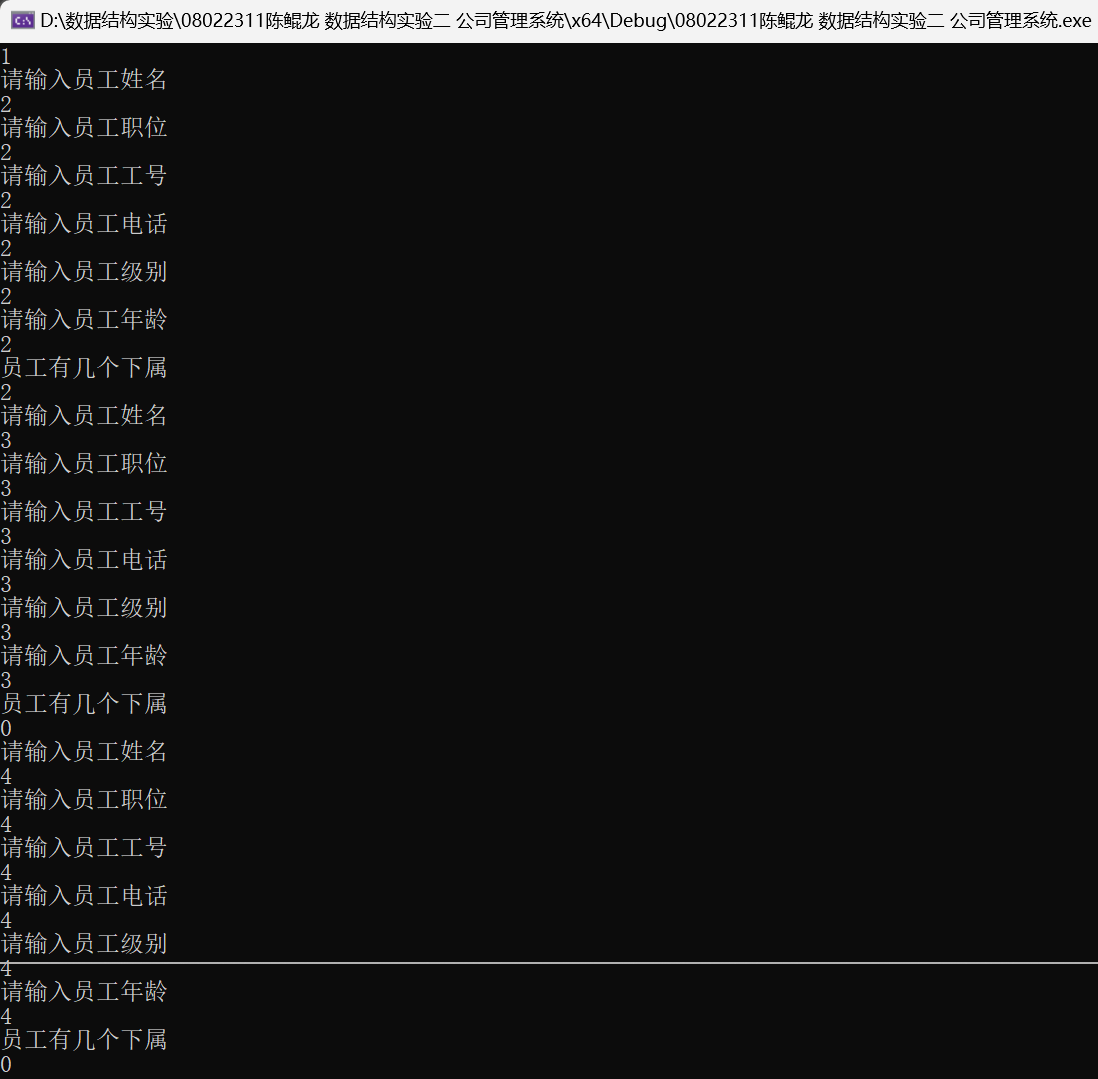
都是用递归实现，不断查找下属的下属……直到指针为空。

1. **调试分析&测试结果**

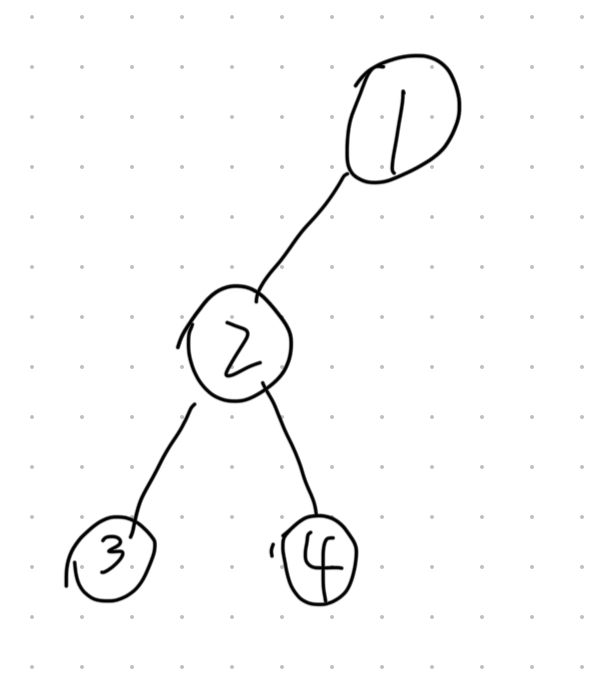
运行程序后首先添加根节点



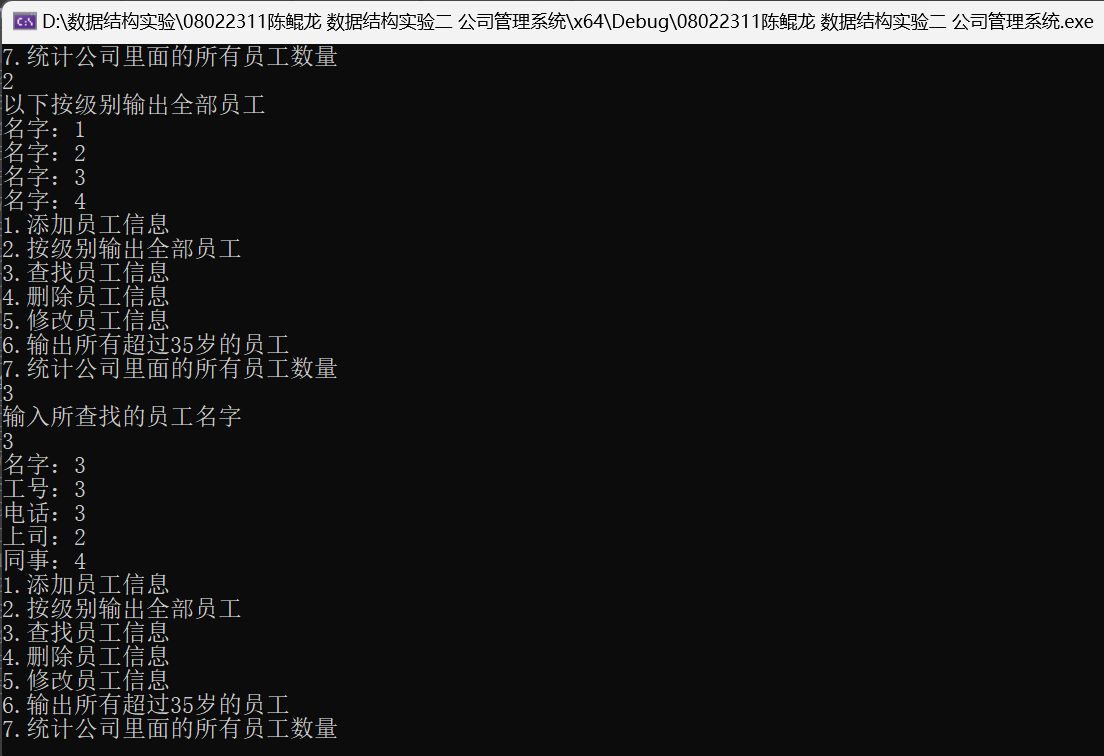
然后再添加几个员工的信息



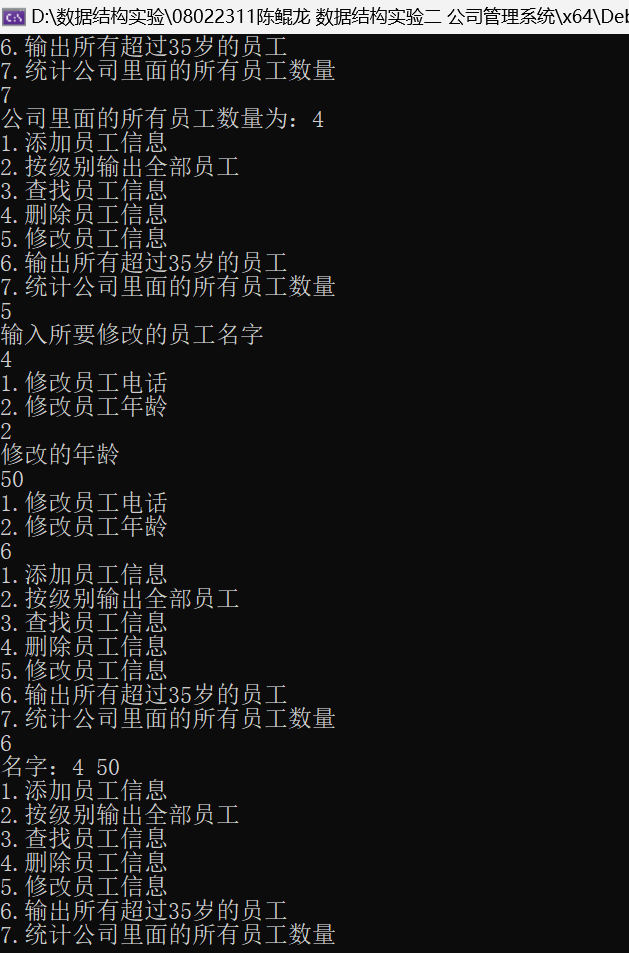
输入完这一串后，具象化树的结构就是如下图



Case2打印出了所有人员，case3也找到了员工3的相关信息：



然后先统计人数 4人没错；然后修改了员工4的年龄为50，最后查找年龄大于35的，正确找到了员工4。



1. **实验总结**

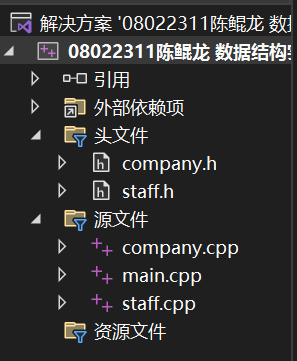
我能力太差，这个实验拖了太久才做出来，有一些问题不会处理，比如删除员工，实现逻辑就是把它上司指向它的指针删除，但是怎么处理被删除的员工的下属是个问题，还有就是只会写构造实例时实时输入信息，而不是在程序中事先构建树的一部分。不过findstaff函数的编写让我对树结构、函数递归、指针等概念更熟悉，理解更深了。

1. **附录**

列出源程序文件名清单。

（所列源程序文件打包作为实验报告的一个重要组成部分，作为实验评分依据）

包含c++类 staff & company ;以及main.cpp



1. **应用调研**

树结构是一种重要的数据结构，它在计算机科学和软件工程中有着广泛的应用。以下是一些树结构的常见应用：

1.文件系统： 操作系统中的文件系统通常以树的形式组织，每个目录是一个节点，包含文件和子目录。

2.数据库系统： 数据库索引通常使用树结构，如B树（平衡树）和B+树，以提高检索效率。

3.图形学： 场景图（Scene Graph）是一种用于组织和管理3D场景中对象关系的树结构。

4.编译器： 语法树（Syntax Tree）用于表示编程语言的语法结构，编译器可以通过遍历语法树来生成目标代码。

5.网络路由： 路由表通常使用树结构，例如Trie树，以便高效地查找最佳路径。

6.操作系统调度： 进程调度和资源管理可以通过树结构来组织，例如进程控制块树。

7.人工智能： 决策树用于机器学习中的分类和回归问题，表达了根据输入数据进行决策的规则。

8.XML和HTML文档： 文档对象模型（DOM）表示XML和HTML文档的结构，以树的形式组织。

9.无线通信： 用于组织无线传感器网络的层次结构，提高通信效率。

10.社交网络： 在社交网络分析中，可以使用树结构来表示社交关系。