

“数据结构”

课程设计报告

**设计题目**  小型公司人员信息管理系统

**姓 名**

**学 号**

**专 业**

**班 级**

**完成日期**

课程设计成绩评定表

|  |  |  |  |
| --- | --- | --- | --- |
| **设计题目** |  | **成绩** |  |
| **内容简介** |  | | |
| **评语** | **教师签名：** | | |

**（一） 需求和规格说明**

某小型公司，主要有四类人员：经理、技术人员、销售经理和推销员。要求存储这些人员的姓名、编号、级别、当月薪水，计算月薪总额并显示全部信息。

人员编号基数为1000，每输入一个人员的信息，编号顺序加1。

程序要对所有人员有提升级别的功能。为简单起见，所有人员的初始级别均为1级，然后进行升级，经理升为4级，技术人员和销售经理升为3级，推销员仍为1级。

月薪计算办法是：经理拿固定月薪8000元；技术人员按每小时100元领取月薪；推销员的月薪按该推销员当月销售额的4%提成；销售经理既拿固定月薪也领取销售提成，固定月薪为5000元，销售提成为所管辖部门当月销售总额的5‰。

**（二） 设计**

根据上述需求，设计一个基类employee,然后派生出technician（技术人员）类、manager（经理）类和salesman（推销员）类。由于销售经理（salesmanager）既是经理又是销售人员，兼具两类人员的特点，因此同时继承manager和salesman两个类。

在基类中，除了定义构造函数和析构函数以外，还应统一定义对各类人员信息都应有的操作，这样可以规范各派生类的基本行为。但是各类人员的月薪计算方法不同，不能在基类employee中统一定义计算方法。各类人员信息的显示内容也不同，同样不能在基类中统一定义显示方法。因此，在employee类中用纯虚函数的方式定义了计算月薪函数pay()和显示信息函数displayStatus()，然后在派生类中再根据各自的同名函数实现具体的功能。

由于salesmanager的两个基类又有公共基类employee，为避免二义性，这里将employee类设计为虚基类。

**系统类图**

salesmanager

virtual void pay();

vitual void displayStatus();

salesman

float CommRate;

float sales;

virtual void pay();

vitual void displayStatus();

manager

float monthlyPay

virtual void pay();

vitual void displayStatus();

technician

float hourlyRate

int workHours

virtual void pay();

vitual void displayStatus();

employee

char \*name

int individualEmpNo;

int grade;

float accumPay;

static int employeeNo;

virtual void pay();

void promote(int);

vitual void displayStatus();

**属性和方法定义**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类名** | **成员类别** | **类型** | **成员名** | **描述** |
| employee | 属性 | char \* | name | 雇员姓名 |
| int | individualEmpNo | 个人编号 |
| int | grade | 级别 |
| float | accumPay | 月薪总额 |
| int | employeeNo | 本公司雇员编号目前最大值 |
| 方法 | void | pay() | 计算月薪函数（为纯虚函数） |
| void | promote(int) | 升级函数 |
| void | DisplayStatus() | 显示人员信息（为纯虚函数） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类名** | **成员类别** | **类型** | **成员名** | **描述** |
| technician | 属性 | float | hourlyRate | 每小时酬金 |
| int | workHours | 当月工作时数 |
| 方法 | void | pay() | accumPay=hourlyRate\*workHours |
| void | DisplayStatus() | 显示技术人员信息 |
| manager | 属性 | float | monthlyPay | 固定月薪数 |
| 方法 | void | pay() | AccumPay=monthlyPay |
| void | DisplayStatus() | 显示经理信息 |
| salesman | 属性 | float | CommRate | 按销售额提取酬金百分比 |
| float | sales | 当月销售额 |
| 方法 | void | pay() | accumPay=sales\*CommRate |
| void | DisplayStatus() | 显示推销员信息 |
| salesmanager | 属性 |  |  |  |
| 方法 | void | pay() | accumPay=monthlyPay+CommRate\*sales |
| void | DisplayStatus() | 显示销售经理信息 |

**（三） 用户手册**

程序运行时，首先提示输入雇员姓名。

对于经理直接输出其工资及其它信息；

对于技术人员，程序提示输入其本月工作时数，然后输出其工资及其它信息；

对于推销员，程序提示输入其本月销售额，然后输出其工资及其它信息；

对于销售经理，程序提示输入其管辖部门本月销售总额，然后输出其工资及其它信息。

**（四） 调试及测试**

由于公司每增加一个雇员，无论他（她）是哪一类人员,其编号均是顺序加1，也就是employee类的所有派生类对象创建时，都要访问同一个employeeNo，因此将employeeNo定义为静态数据成员。

**（五） 运行实例：**

please input employee's name: zhang

please input employee's name: wang

please input employee's name: Li

please input employee's name: zhao

input zhang theworkHours of this month: 56

Technician: zhang No: 1001 month salary: 5600

Technician: zhang No: 1001 grade: 3 this month salary: 5600

Manager: wang No: 1002 month salary: 8000

Manager: wang No: 1002 grade: 4 this month salary: 8000

input Li the sales of this month: 47900

Salesman: Li No: 1003 month salary: 1916

Salesman: Li No: 1003 grade: 1 this month salary: 1916

input zhao the total sales of the department of this month: 123654

salesman: zhao No: 1004 month salary: 5618.27

salesmanager: zhao No: 1004 grade: 3 this month salary: 5618.27

**（六）进一步改进**

（1）目前程序中，经理月薪，技术人员的小时酬金和销售人员的销售额提成比例均是固定的，这不适应不同公司的需要，可考虑用带参数的构造函数来解决。

（2）销售经理月薪计算中，要输入其管辖部门当月销售总额。实际上，这可以通过将本部门所有推销员销售额相加而得到。可以考虑在推销员类中增加所属部门等属性来完成这方面的功能。

**（七）心得体会**

（可描述课程设计过程中出现的主要问题、原因及解决方法，课程设计的主要收获等）

**（八）对课程设计的建议**

**（九）附录⎯⎯源程序**