

# **继承与多态程序实例**

## **— 派生类定义和函数重定义**

【例3-8】定义一个雇员类Employee，包含char型数组成员变量m\_cName（姓名）和m\_cNo（编号），int型成员变量m\_nDepartment（部门号），成员函数SetName(char \*pcName)、SetNo(char \*pcNo)和SetDept(int nDept)分别用来设置姓名、编号和部门号，Display()函数用来输出雇员的信息，还包含一个构造函数Employee(char \*pcName, char \*pcNo, int nDept)。定义一个经理类Manager，派生于雇员类Employee，新增成员变量m\_nLevel表示级别和成员函数SetLvl(int nLvl)用来设置m\_nLevel，重定义Display()用来输出经理的信息。要求：

( 1 ) Manager类的成员函数可以直接访问Employee类中的成员变量，但类的外部不能直接访问。

( 2 ) 设计一个主函数如下：

```
int main()
{
    Employee e("张强", "001760", 12);
    e.Display();
    Manager m("李红", "981002", 12, 3);
    m.Display();
    m.SetName("王新");
    m.SetNo("991013");
    m.Display();
    return 0;
}
```

```
// Employee.h
class Employee
{
protected:
    char m_cName[20]; // 姓名
    char m_cNo[7];    // 编号
    int m_nDepartment; // 部门号
public:
    Employee(char *pcName, char *pcNo,
int nDept);
    void SetName(char *pcName);
    void SetNo(char *pcNo);
    void SetDept(int nDept);
    void Display();
};
```

```
// Employee.cpp
#include "Employee.h"
#include <iostream>
using namespace std;
```

```
Employee::Employee(char *pcName,
char *pcNo, int nDept)
```

```
{
    SetName(pcName);
    SetNo(pcNo);
    SetDept(nDept);
}
```

```
void Employee::SetName(char *pcName)
```

```
{
    strcpy(m_cName, pcName);
}
```

```
void Employee::SetNo(char *pcNo)
```

```
{
    strcpy(m_cNo, pcNo);
}
```

```
void Employee::SetDept(int nDept)
```

```
{
    m_nDepartment = nDept;
}
```

```
void Employee::Display()
```

```
{
    cout<<"姓名："<<m_cName<<endl
    <<"编号："<<m_cNo<<endl
    <<"部门号："<<m_nDepartment<<endl;
}
```

```
// Manager.h
#include "Employee.h"
class Manager : public Employee
{
private:
    int m_nLevel;        // 级别
public:
    Manager(char *pcName, char *pcNo,
            int nDept, int nLvl);
    void SetLvl(int nLvl);
    void Display();
};
```

```
// Manager.cpp
#include "Manager.h"
#include <iostream>
using namespace std;
Manager::Manager(char *pcName, char *pcNo,
                int nDept, int nLvl)
    : Employee(pcName, pcNo, nDept)
{
    SetLvl(nLvl);
}
void Manager::SetLvl(int nLvl)
{
    m_nLevel = nLvl;
}
void Manager::Display()
{
    cout<<"姓名："<<m_cName<<endl
         <<"编号："<<m_cNo<<endl
         <<"部门号："<<m_nDepartment<<endl
         <<"级别："<<m_nLevel<<endl;
}
```

```
// main.cpp
#include "Manager.h"
int main()
{
    Employee e("张强", "001760", 12);
    e.Display();
    Manager m("李红", "981002", 12, 3);
    m.Display();
    m.SetName("王新");
    m.SetNo("991013");
    m.Display();
    return 0;
}
```

运行结果：

姓名：张强

编号：001760

部门号：12

姓名：李红

编号：981002

部门号：12

级别：3

姓名：王新

编号：991013

部门号：12

级别：3