程序设计实习

郭炜　微博 http://weibo.com/guoweiofpku
http://blog.sina.com.cn/u/3266490431
刘家瑛 微博 http://weibo.com/pkuliujiaying

继承和派生

(教材P215)

继承和派生的概念

- 继承：在定义一个新的类B时，如果该类与某个已有的类A相似(指的是B拥有A的全部特点)，那么就可以把A作为一个**基类**，而把B作为基类的一个**派生类(也称子类)**。

继承和派生的概念

- 派生类是通过对基类进行修改和扩充得到的。在派生类中，可以扩充新的成员变量和成员函数。
- 派生类一经定义后，可以独立使用，不依赖于基类。

继承和派生的概念

- 派生类拥有基类的全部成员函数和成员变量，不论是private、protected、public 。
  - 在派生类的各个成员函数中，不能访问基类中的private成员。

需要继承机制的例子

- 所有的学生都有的共同属性：
  - 姓名
  - 学号
  - 性别
  - 成绩

所有的学生都有的共同方法（成员函数）：

- 是否该留级
- 是否该奖励

需要继承机制的例子

而不同的学生，又有各自不同的属性和方法

研究生

导师

系

大学生

系

中学生

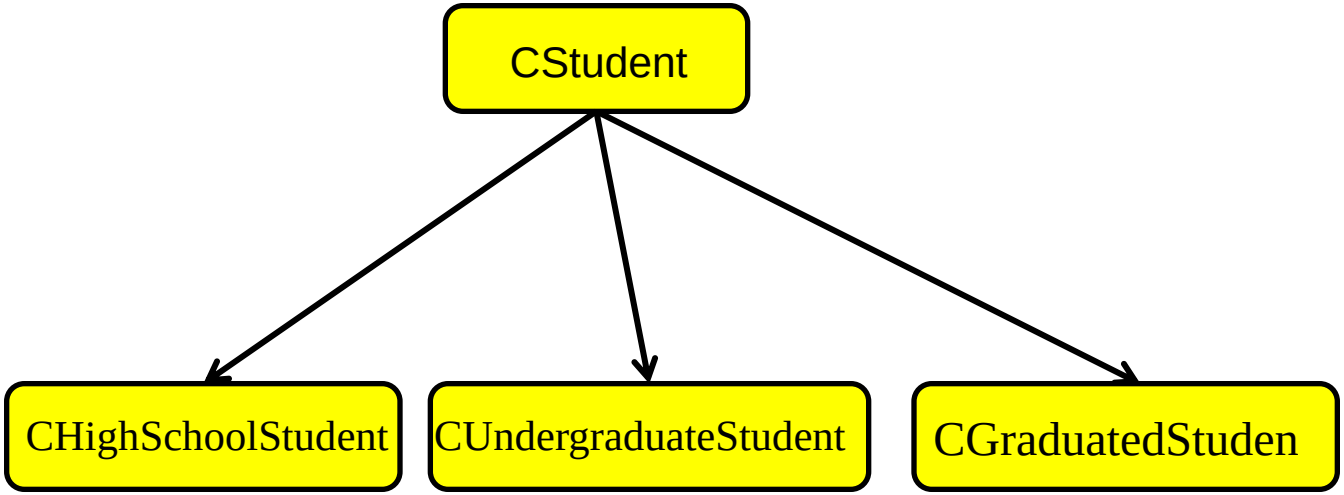
竞赛特长加分

需要继承机制的例子

- 如果为每类学生都从头编写一个类，显然会有不少重复的代码，浪费。

- 比较好的做法是编写一个“学生”类，概括了各种学生的共同特点，然后从“学生”类派生出“大学生”类，“中学生”类，“研究生类”。

需要继承机制的例子


CStudent
CHighSchoolStudent
CUndergraduateStudent
CGraduatedStuden

派生类的写法

```
class 派生类名：public 基类名
{

};
```

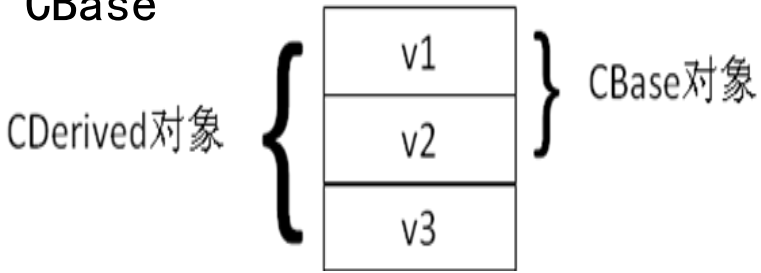
```
class CStudent {
        private:
                string sName;
                int nAge;
        public:
        bool IsThreeGood() { };
        void SetName( const string & name )
        { sName = name; }
        //.....
};
class CUndergraduateStudent: public CStudent {
        private:
                int nDepartment;
        public:
                bool IsThreeGood() {  .....  };  //覆盖
                bool CanBaoYan() { .... };
}; // 派生类的写法是：  类名: public 基类名
```

```
class CGraduatedStudent:public CStudent {
        private:
                int nDepartment;
                char szMentorName[20];
        public:
                int CountSalary() { ... };
};
```

派生类对象的内存空间

派生类对象的体积，等于基类对象的体积，再加上派生类对象自己的成员变量的体积。在派生类对象中，包含着基类对象，而且基类对象的存储位置位于派生类对象新增的成员变量之前。

```
class CBase
{
      int v1,v2;
};
class CDerived:public CBase
{
      int v3;
};
```



继承实例程序:学籍管理 (P228)

```
#include <iostream>
#include <string>
using namespace std;
class CStudent  {
    private:
        string name;
        string id;  //学号
        char gender; //性别,'F'代表女，'M'代表男
        int age;
    public:
        void PrintInfo();
        void SetInfo( const string & name_,const string & id_,
                int age_,       char gender_ );
        string GetName() {  return name;  }
};
```

```
class CUndergraduateStudent:public CStudent
{//本科生类，继承了CStudent类
    private:
        string department; //学生所属的系的名称
    public:
        void QualifiedForBaoyan() {   //给予保研资格
            cout << “qualified for baoyan” << endl;
        }
        void PrintInfo() {
            CStudent::PrintInfo(); //调用基类的PrintInfo
            cout << “Department:” << department <<endl;
        }
        void SetInfo( const string & name_,const string & id_,
            int age_,char gender_ ,const string &  department_) {
            CStudent::SetInfo(name_,id_,age_,gender_); //调用基类的SetInfo
            department = department_;
        }
};
```

```
void CStudent::PrintInfo()
{
        cout << "Name:" << name << endl;
        cout << "ID:" << id << endl;
        cout << "Age:" << age << endl;
        cout << "Gender:" << gender << endl;
}
void CStudent::SetInfo( const string & name_,const string & id_,
                        int age_,char gender_ )
{
        name = name_;
        id = id_;
        age = age_;
        gender = gender_;
}
```

```
int main()
{

    CUndergraduateStudent s2;
     s2.SetInfo(“Harry Potter ”,  “118829212”,19,‘M’,“Computer Science”);
     cout <<  s2.GetName() << “ ” ;
     s2.QualifiedForBaoyan ();
     s2.PrintInfo ();
     return 0;
}
```

输出结果：

Harry Potter qualified for baoyan
Name:Harry Potter
ID:118829212
Age:19
Gender:M
Department:Computer Science