

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

类的声明和对象的定义

类的声明和对象的定义-形式1

```
class Student
```

类名

```
{
```

```
private:
```

```
int num;
```

```
char name[20];
```

```
char sex;
```

```
public:
```

```
void set_data(int n, char *p, char s)
```

```
{
```

```
num=n;
```

```
strcpy(name,p);
```

```
sex=s;
```

```
}
```

```
void display( )
```

```
{
```

```
cout<<"num: "<<num<<endl;
```

```
cout<<"name: " <<name<<endl;
```

```
cout<<"sex: " <<sex<<endl<<endl;
```

```
}
```

```
};
```

数据成员

成员函数

```
int main()
```

```
{
```

```
Student stud1,stud2;
```

```
stud1.set_data(1,"John",'f');
```

```
stud2.set_data(2,"Mary",'m');
```

```
stud1.display();
```

```
stud2.display();
```

```
return 0;
```

```
}
```

对象

用对象调用成员函数

类外访问公有成员成员

成员的访问权限：

- private(私有的)
- public(公用的)
- protected(受保护的)

成员函数当中操纵私有数据成员

访问权限——公共的还是私有的

```
class Student
```

```
{
```

```
private:
```

```
    int num;
```

```
    char name[20];
```

```
    char sex;
```

```
public:
```

```
    void set_data(int n, char *p, char s)
```

```
    {
```

```
        num=n;
```

```
        strcpy(name,p);
```

```
        sex=s;
```

```
    }
```

```
    void display( )
```

```
    {
```

```
        cout<<"num: "<<num<<endl;
```

```
        cout<<"name: " <<name<<endl;
```

```
        cout<<"sex: " <<sex<<endl<<endl;
```

```
    }
```

```
};
```

访问权限选择的一般原则

☞ 数据成员：private

☞ 成员函数：public

```
struct Student
```

```
{
```

```
    int num;
```

```
    char name[20];
```

```
    char sex;
```

```
};
```

C++中的class是C中struct的延伸

☞ class: 访问权限默认private

☞ struct: 访问权限默认public

定义对象的方法

```
class Student //声明类类型
{
    ... ..
}
.....
Student stud1 , stud2;
```

先声明类类型，然后再定义对象

```
class Student //声明类类型
{
    ... ..
} stud1 , stud2;
```

在声明类类型的同时定义对象

命名规范

- 📁 类名：首字母大写
- 📁 对象名：首字母小写

```
class //声明类类型
{
    ... ..
} stud1 , stud2;
```

不出现类名，直接定义对象

类声明的第2种形式——更实用和常见

```
class Student
{
public:
    void set_data(int n, char *p,char s);
    void display( );
private:
    int num;
    char name[20];
    char sex;
};

void Student::set_data(int n, char *p,char s)
{... ..}

void Student::display( )
{... ..}

演示.....
```

```
class 类名
{
public:
    公用的数据和成员函数;
private:
    私有的数据和成员函数;
};

... .. 成员函数的定义
```

在类外定义成员函数——将惯例成为习惯

//类体中写成员函数的声明，类外面进行函数定义

```
class Student
```

```
{
```

```
public:
```

```
    void display( );
```

```
private:
```

```
    int num; .....
```

```
};
```

```
void Student::display( )
```

```
{
```

```
    cout<<"num:"<<num<<endl;
```

```
    .....
```

```
}
```

```
Student stud1,stud2;
```

先在类体中作原型声明，然后在类外定义

::—作用域限定符(field qualifier)或称作用域运算符，用它声明函数是属于哪个类的。

若无作用域限定，意味着？

一个完整的程序

```
#include <iostream>
#include <cstring>
using namespace std;
class Student
{
public:
    void set_data(int n, char *p,char s);
    void display( );
private:
    int num;
    char name[20];
    char sex;
};
void Student::set_data(int n, char *p,char s)
{
    num=n;
    strcpy(name,p);
    sex=s;
}
```

```
void Student::display( )
{
    cout<<"num: "<<num<<endl;
    cout<<"name: " <<name<<endl;
    cout<<"sex: " <<sex<<endl<<endl;
}
```

```
int main()
{
    Student stud1,stud2;
    stud1.set_data(1,"He",'f');
    stud2.set_data(2,"She",'m');
    stud1.display();
    stud2.display();
    return 0;
}
```


并非所有成员函数必须公有

```
class Time
{
public:
    void set_time( );
    void show_time( );
private:
    bool is_time(int, int, int);
    int hour;
    int minute;
    int sec;
};
```

```
bool Time::is_time(int h,int m, int s){
    if (h<0 || h>24 || m<0 || m>60 || s<0 || s>60)
        return false;
    return true;
}
```

```
void Time::set_time( )
{
    char c1,c2;
    cout<<"请输入时间(格式hh:mm:ss)";
    while(1)
    {
        cin>>hour>>c1>>minute>>c2>>sec;
        if(c1!=':' || c2!=':')
            cout<<"格式不正确，请重新输入"<<endl;
        else if (!is_time(hour,minute,sec))
            cout<<"时间非法，请重新输入"<<endl;
        else
            break;
    }
}
```

将只被成员函数访问的成员函数，定义为私有是正确的选择。

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>