

C++语言基础

迂者 - 贺利坚

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

<http://edu.csdn.net>





本节主题：

类的成员函数

类的成员函数

```
class Classname
{
public:
    void set_data(int n);
    void display( );
private:
    int num;
};

void Student::set_data(int n)
{... ..}

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

类的成员函数(简称类函数) 是一个类的成员，在类体中声明。

调用类成员函数的权限：

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

如果一个类中不包含成员函数，就等同于C语言中的结构体了，体现不出类在面向对象程序设计中的作用。

类的成员函数访问权限的设定

- 一般的做法：将需要被外界调用的成员函数指定为public，它们是类的对外接口。
- 并非要求把所有成员函数都指定为public：有的函数只被本类中的成员函数调用，以支持其他函数的操作，应该将它们指定为private。
 - 私有的成员函数只能被本类中的其他成员函数所调用，而不能被类外调用。
- 成员函数可以访问本类中任何成员(包括私有的和公用的)，可以引用在本作用域中有效的数据。



inline 成员函数

回顾内置(inline)函数

- 使用内置函数节省运行时间，却增加了目标程序的长度。
- 一般只将规模很小、使用频繁的函数声明为内置函数。
- 内置函数中不能包括复杂的控制语句。
- 类的成员函数也可以指定为内置函数。

```
inline int max(int,int,int);  
int main( )  
{  
    int i=10,j=20,k=30,m;  
    m=max(i,j,k);  
    cout<<"max="<<m<<endl;  
    return 0;  
}  
inline int max(int a,int b,int c)  
{  
    if(b>a) a=b;  
    if(c>a) a=c;  
    return a;  
}
```

类内定义的成员函数被隐含地指定为内置函数

```
class Student
{
public:
    void display( )
    {
        cout<<"num:"<<num<<endl;
        cout<<"name:"<<name<<endl;
        cout<<"sex:"<<sex<<endl;
    }
private:
    int num;
    string name;
    char sex;
};
```

也可以将成员函数显式地声明为内置函数

```
class Student
{
public:
    inline void display( )
    {
        cout<<"num:"<<num<<endl;
        cout<<"name:"<<name<<endl;
        cout<<"sex:"<<sex<<endl;
    }
private:
    int num;
    string name;
    char sex;
};
```



成员函数在类体外定义，应当用inline对内置函数作显式声明

```
class Student
{
public:
    inline void display( );
private:
    int num;
    string name;
    char sex;
};
inline void Student::display( )
{
    cout<<"num:"<<num<<endl;
    cout<<"name:"<<name<<endl;
    cout<<"sex:"<<sex<<endl;
}
```

只有在类外定义的成员函数规模很小而调用频率较高时，才将此成员函数指定为内置函数。

成员函数的存储方式

回顾：结构体的存储方式

```
struct Student
```

```
{  
    int num;  
    char name[20];  
    char sex;  
    int age;  
    double score;  
    char addr[30];  
}
```

```
Student student1;
```

struct和class有何关系？

各个域连续存储，需要的空间为各成员所需空间之和(考虑内存对齐)。

- ❑ 类 / 结构体类型是抽象的，在类型声明时，不涉及内存分配；
- ❑ 对象是具体的，在定义时，要分配内存。

num	name	sex	age	score	addr
10010	Li Fun	M	18	87.5	Beijing

对象的存储：怎么合理？

```
class Time
```

```
{
```

```
private:
```

```
    int hour;
```

```
    int minute;
```

```
    int sec;
```

```
    bool is_time( )
```

```
    { ... ... }
```

```
public:
```

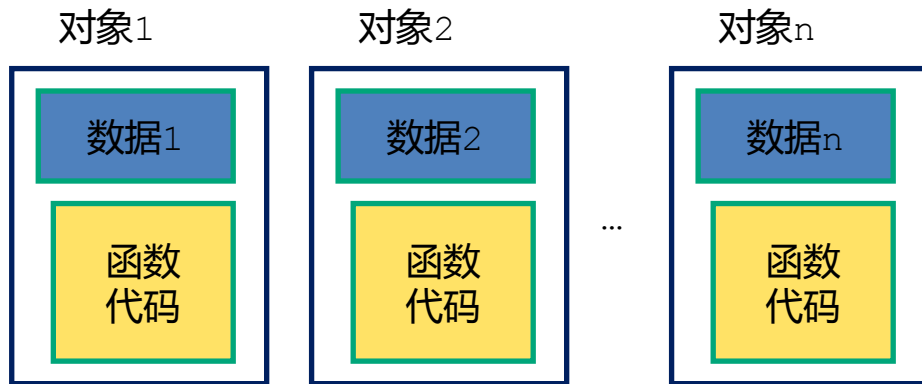
```
    void set( )
```

```
    {
```

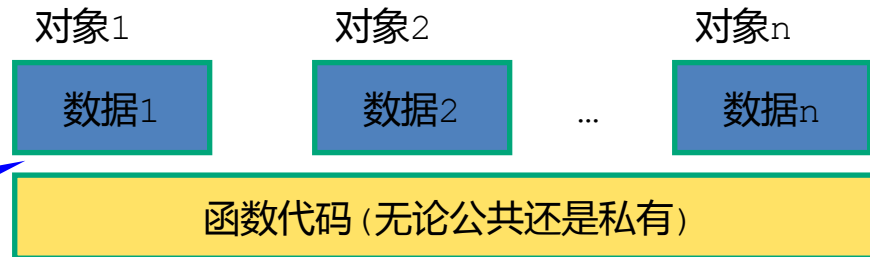
```
        cin>>a>>b>>c;
```

```
    }
```

```
};
```



or



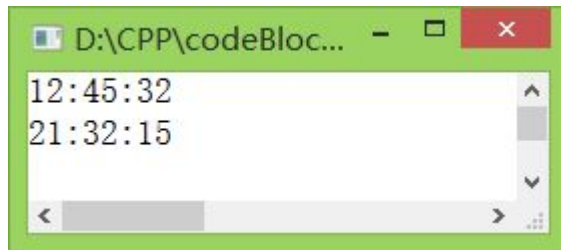
分开存储数据，
共享函数代码。

分清“你我”

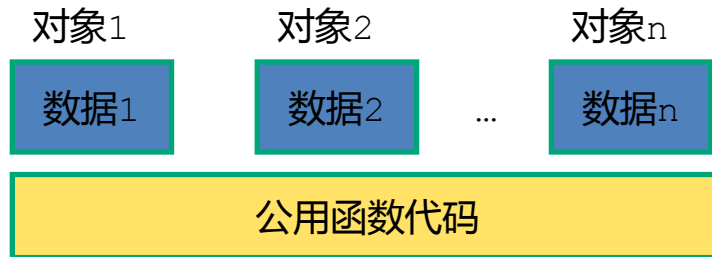
```
class Time
{
public:
    void set_time(int,int,int);
    void show_time();
private:
    int hour;
    int minute;
    int sec;
};
void Time::set_time(int h, int m,int s)
{
    hour=h;
    minute=m;
    sec=s;
}
```

```
void Time::show_time()
{
    cout<<hour<<":";
    cout<<minute<<":";
    cout<<sec<<endl;
}
```

```
int main( )
{
    Time t1,t2;
    t1.set_time(12,45,32);
    t2.set_time(21,32,15);
    t1.show_time( );
    t2.show_time( );
    return 0;
}
```



用同样形式的调用，
访问各自的数据成员。



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/>