

第7章 数据的抽象与封装

1、实体、对象与类的概念	2、类的定义	3、对象声明与引用	4、私有、公有与保护
5、日期类的设计	6、两种程序设计思想	7、汽车类的设计	8、几何图形圆类的设计
9、构造函数的定义	10、重载构造函数	11、析构函数的定义	12、整数翻译函数
13、实际意义的析构函数	14、Person类的设计	15、对象与指针	16、this指针

对象的声明

👉 用已定义的类去声明对象（即变量），类似于结构体

👉 声明对象的一般形式（类的实例化）：

〈类名〉 〈对象1〉, 〈对象2〉,;

例如：**Clock XJTU_Big_Ben, Omega, Rossini;**

这条语句声明了三个对象，对象名分别是

XJTU_Big_Ben

Omega

Rossini

也可以称声明了三个变量，变量名分别是

XJTU_Big_Ben

Omega

Rossini

对象的引用

👉 对象的引用类似于结构体

👉 对类中成员的访问，采用. 运算符

例如：**Rossini.Set(8,32,53,1000);**

Rossini.Show_Time();

👉 一般在类体外，只有公有成员可以采用. 运算符访问

👉 私有、保护成员都不能采用. 运算符访问

👉 可以直接进行对象赋值

👉 例如：**Omiga=Rossini;**

对钟表类的测试主函数

```
int main()
{
    Clock XJTU_Big_Ben;
    XJTU_Big_Ben.Set(0,0,0,1000);
    cout<<"钟表设置的时间：\n";
    XJTU_Big_Ben.Show_Time();
    XJTU_Big_Ben.Report_Time();
    XJTU_Big_Ben.Run();
    XJTU_Big_Ben.Set(9,59,50,9000);
    cout<<"\n钟表设置的时间：\n";
    XJTU_Big_Ben.Run();
    XJTU_Big_Ben.Report_Time();
    cout<<endl;
    XJTU_Big_Ben.Show_Time();
    cout<<endl;
    return 0;
}
```

对象的存储空间

👉 若一个类声明了五个对象，存储空间如何分配？

👉 是否对象中的数据和函数成员都占用存储空间？

👉 C++程序的存储分配结构图？

👉 数据成员分配存储空间

👉 函数成员共用一个存储空间

👉 在例7-1中增加下面语句：

👉 **cout << sizeof(XJTU_Big_Ben) << endl;**

👉 **cout << sizeof(Clock) << endl;**

👉 输出结果都是16

引用对象应该注意

- 👉 成员函数直接访问同类中的数据成员（公有和私有）
 - 👉 对象访问其数据成员或成员函数需使用运算符。
 - 👉 禁止直接访问一个对象中的私有成员
 - 👉 同类对象之间可以整体赋值
 - 👉 对象用作函数的参数时属于赋值调用
 - 👉 对象可以用作函数值类型，函数可以返回一个对象
-

类的定义、声明、引用小结

- 👉 公有成员：外界能够直接访问，起着接口的作用
 - 👉 私有成员：外界不能直接访问，使得成员被隐藏起来
 - 👉 一个实体的属性往往被定义为私有成员，方法被定义为公有成员
 - 👉 通过类中提供的函数对类中的数据进行操作，达到数据的安全性
 - 👉 类名、各成员名称要符合标识符规范
 - 👉 成员函数可在类中定义，也可在类外定义
 - 👉 各成员的访问权限应根据实际需要确定
 - 👉 书写顺序任意，有时public放在前边，有时写几个public或几个private
 - 👉 类中的数据成员不能使用表达式进行初始化
-

**钟表类的对象声明与引用介绍完了
下一个微片段介绍私有、公有与保护**

感谢收看！