第二讲 C++类与对象

一、类与对象基本概念

对象(实例)：对象=数据类型（属性）+行为（成员函数）

类：对一组具有相同属性和相同行为的对象的抽象描述

由一个类可以创建同类的对象（struct）

消息传递（调用函数）

抽象：通过从特定实例中抽取共同性质以形成一般化概念的过程

“获取共性”是抽象过程的本质

面向对象操作的好处在于已经创建好的类可以在不同的场合被多次重复利用

二、面向对象语言基本原理

1封装性：

通过类来把数据与处理该数据的方法封装成一个整体，通过一定的外部接口与外界操作相连。即“用户并不关系数据是如何被处理的，用户只关心功能”

2继承性：

一个类直接继承其父类的全部描述，不需要重新进行定义，比如数据成员与成员函数

继承具有传递性，即更细分的类可以在父类的基础上进行，

比如“技术人员”可以在“员工”的基础上进行

多继承：一个类可以有多个父类 （容易发生二义性问题）

//比如“助教”可以既具有“学生”也可以具有“教师”的特性

单继承：一个类只有一个父类（更推荐）

3多态性： 静态多态性（在编译过程中完成）：比如同名函数通过不同的参数表来区别

//但是同名同参数表而只是返回值类型不同的函数是不可以共存的

动态多态性（在运行过程中完成）

1. 类与对象的运用

1、类与成员函数的声明

class product{

public: //对外接口

<成员函数或数据成员的声明>

Void input(); //变异函数

Double move()const; //访问函数（const只读取不修改）

//当没有声明任何权限时候，默认为private，即private可以省略不写

private: //内部处理，这样的处理能避免用户误改（黑匣子）

Double x;

Double y;

<数据成员或成员函数的声明>

1. 访问修饰符

//“+”表示可以访问，“-”表示不能访问

|  |  |  |  |
| --- | --- | --- | --- |
|  | public | private | protected |
| 同一类的成员函数 | + | + | + |
| 类外的成员函数 | + | - | - |
| 类的派生类的成员 | + | - | + |

Attention： 对于类的访问，只能使用public中的函数来访问（比如product.input()）

而不能直接访问private中的数据成员

类的头文件名与类的名字必须完全一致

1. 类的实现
2. 类外实现

通过#include” abc.h ”来包含自定义的类

然后通过类似定义函数的方式来定义类内的函数

如void abc::move()

{abcd;efgh;}

1. 内联函数实现

若在类内实现

若在类外实现则需要在函数前加关键字inline

//宏替换只替换代码而不进行语法检查，

但是内联函数会被编译器识别，替换时也会进行语法检查

1. 类的测试
2. 构造函数
3. 构造函数的定义
4. 语法形式：函数名就是类名，且没有返回值

Public：类名（ 参数表 ）

1. 调用：类外访问只能通过调用public的函数来实现，

而不能直接访问private的数据成员

且函数调用必须具有括号（），即使没有参数也需要写

如cout<<”size=”<<s1.getsize()<<endl;

1. 构造函数的类外实现必须声明作用域，否则会被识别为普通函数

如point::move(){Abcd;efgh;}

1. 未定义构造函数时由系统提供一个默认版本的构造函数，以随机值初始化。只要定义了构造函数，系统的默认版本就会自动被屏蔽
2. 初始化列表：与构造函数的初始化作用相同，但是可以提高初始化效率

如：Product::Product(String n,double p, int s):name(n), price(p), score(s){}

2、重载构造函数

重载构造函数举例：

class X

{ public:

X( ){};

X( int ) {};

X( int, char ) {};

X( float, char, int ) {};

};

void main( )

{

X a, b(1), c(1, ’c’), d(2.3, ’d’, 1);

}

1. 常见错误
2. 如果类中定义了含有const的访问函数，则在访问调用时也需要写上const，否则会被认为是重载函数。（函数签名要相同）
3. 数据成员是指针时一定需要安全地初始化，至少需要赋值为null，否则会成为危险的野指针。
4. 构造函数只在构造之前起作用，之后无法被调用而起效。

//如 lunchtime.timeset(); -->错误，构造函数无法被调用

Lunchtime=timeset(); -->正确，相当于使用构造函数新创建了一个类然后对应赋值

1. 重载构造函数之后系统自带的构造函数会被屏蔽，只有重载函数起效。

5、重载函数的参数表内形参名最好不要与数据成员相同。