Ch-08

类和对象的特性

主要内容

- 面向对象程序设计方法概述
- 类的声明和对象的定义
- 类的成员函数
- 对象成员的引用
- 类的封装性和信息隐蔽

8.1 面向对象程序设计方法概述

8.1.1 什么是面向对象的程序设计

口 对象

客观世界中任何一个事物都可以看成一个对象。任何一个对象都具有静态和动态的特征。静态特征称为属性，动态特征称为行为，外界给对象发出的信息一般称作消息。

面向对象的程序设计首先要确定系统中包括哪些对象，要分别设计这些对象。在C++ 中，每个对象由数据和行为（函数）两部分组成。数据代表了属性；行为是对数据操作的代码，在程序设计中又称方法。触发对象的某个行为（即调用对象中的函数）就是向对象传递一个消息，要求对象执行某个操作。

□ 封装与信息隐蔽

对象的封装是把它的部分属性和功能对外界屏蔽，使外界无法看到和使用这些属性和功能。在设计对象时，要周密地考虑如何进行封装，把对象的内部实现和外部行为分隔开来。

封装性是面向对象程序设计的一个重要特点，在此有两个含义：

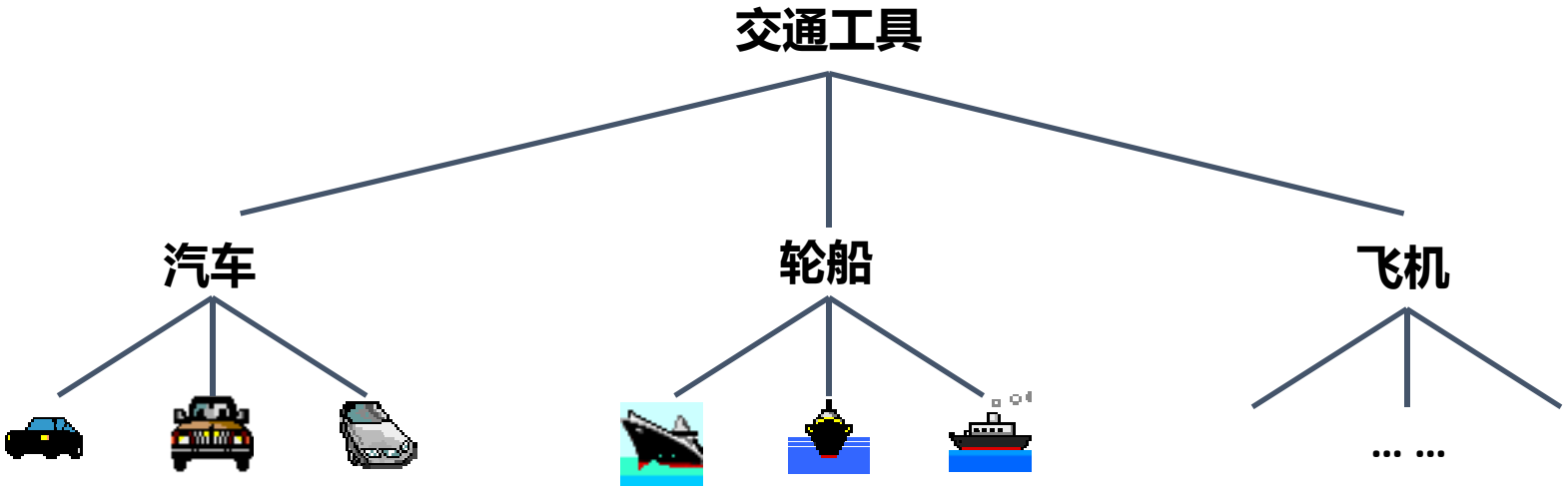
1、把有关的数据和行为封装在一个对象中，形成一个基本单位，各个对象之间相互独立，互不干扰。

2、把对象中的某些部分对外隐蔽，只留下与外界联系的接口接收外界的消息，这种对外界隐蔽的做法称为信息屏蔽。信息隐蔽还有利于数据安全。

抽象

- 抽象是对一个对象的本质特征的描述，这个特征将这个对象与其他种类的对象区别开来；
- 抽象是通过从特定的实例中抽取共同的性质以形成一般化的概念的过程；
- 抽象具有层次；

在C++中，类是对象的抽象，而对象是类的具体实例。



□ 继承与重用

“白马”不仅具有“马”的所有属性和行为，而且在“马”的基础上也增加了一些新的属性和行为，因此可以基于“马”去创建“白马”，这种就是C++中的继承机制。其中“马”称为父类或基类，“白马”称为子类或派生类。

C++中，利用继承机制可以在一个已有的类的基础上建立一个新类，这就是“软件重用”思想。这对于大型软件的开发具有重要的意义。

□ 多态

多个相似而不完全相同的对象，当它们收到外界传递过来的同一个消息时，各自执行不同的操作，这种现象叫多态。

在C++中所谓多态性是指：由继承而产生的不同的派生类，它们的对象对同一个消息会作出不同的响应。

8.1.2 面向对象程序设计的特点

面向对象程序设计包括两个方面：

1、设计所需的各种类，即决定把哪些数据和操作封装在一起。

2、考虑怎样向对象发送消息（调用对象的成员函数），实现所需的操作。这时设计程序象一个总调度，不断地向各个对象发送消息（命令），让这些对象活动起来（激活这些对象），完成各自的职责范围工作，各个对象的操作完成，整体任务也就完成了。

8.1.3 面向对象的软件开发

1. 面向对象分析
2. 面向对象设计
3. 面向对象编程
4. 面向对象测试
5. 面向对象维护

8.2 类的声明和对象的定义

8.2.1 类和对象的关系

- 类是面向对象程序设计的核心；
- 类是一种复杂数据类型，它是将不同类型的数据和与这些数据相关的操作封装在一起的集合体；
- 类是对某一类对象的抽象，对象是某一种类的实例；
- 类是抽象的，不占用内存空间；对象是具体的，要占用内存空间。

8.2.2 类的声明

```
class   类名   {
        private:
          私有成员声明
        public:
          公有成员声明
        protected:
          保护成员声明
};
```

例如：

```
class Student {
private:
        int   num;
        string  name;
        char sex;
public:
        void   setdata();
        void  display();
};
```

说明：

1、class是关键字，声明类类型；类名遵循标识符命名规则，且一般首字母大写（非必须）。

2、private、public、protected也是关键字，是成员访问限定符，其后必须跟冒号。

- **private：私有成员限定在该类的内部使用，即只允许该类中的成员函数使用（调用）私有成员数据（函数），类外（除友元外）不能访问。**
- **public：公有成员不受类的限制，可以在类内或类外自由使用，是提供给用户的接口。**
- **protected：保护成员只允许在类内及该类的派生类中使用，即保护成员的作用域是该类及该类的派生类。。**

在定义类时，这三类成员不分前后顺序，也可以重复出现。一般推荐最多出现一次。

3、每一个限定符（private等）在类中可使用多次。一旦使用了限定符，该限定符一直有效，直到下一个限定符开始为止。

4、若类中没有指定任何访问属性，则默认是私有的。

5、在类中不允许对所定义的数据成员进行初始化。

6、在类声明结尾的 } 后面如不直接定义对象就必须跟分号。

◆ 类和结构体的区别

C++允许用struct定义一个类，但两者还是有区别的：

- 用class声明的类中，所有成员的默认访问属性是private；
- 用struct声明的类中，所有成员的默认访问属性是public。

8.2.3 对象的定义

对象定义的三种方法：

- 先声明类，然后再定义对象

（1）class 类名 对象名表　　例如：class Student std1, std2;

（2）类名　对象名表　　例如：Student std1, std2;

- 在声明类的同时定义对象

例如：class Student { … } std1, std2;

- 不出现类名，直接定义对象

例如：class { … } std1, std2;

8.3 类的成员函数

8.3.1 成员函数的性质

- 成员函数是函数中的一种，具有函数所有的特性和使用方法；
- 成员函数可以访问本类中的所有成员；
- 一般方法是把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。

8.3.2 成员函数的声明和定义

成员函数既可以在类中进行定义；也可以只在类中声明，而在类外进行定义（推荐）。

□ 在类外定义成员函数的格式

函数类型　类名::函数名(形参表) { 函数体 }

```
void  Student :: setdata()
{
        cin >> num;
        cin >> name;
        cin >> sex;
}
```

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

□说明：

- 在类外定义成员函数时，必须在函数名前增加类名，用于限定它属于哪个类，而“::” 就是作用域限定符或称作用域运算符。
- 上面的例子中如果在::前不带类名，或函数名前既无类名又无作用域限定符::，则表示该函数是一个普通的全局函数。

8.3.3 内联成员函数

C++在类内定义的成员函数默认是inline函数；但在类外定义成员函数时，则不会被默认为inline，此时必须在声明函数时在函数前加inline关键字。

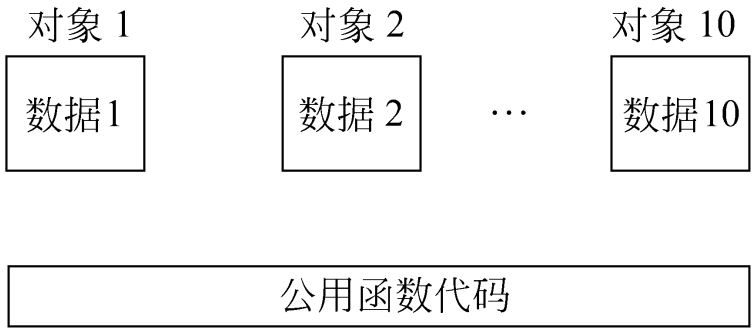
```
class Student {
private:
        int   num;
        …
public:
        void  display() {
              cout << num << endl;
              cout << name << endl;
              cout << sex << endl;
        }
};
```

```
class Student {
private:
        int   num;
        …
public:
        inline void  display();
};

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

8.3.4 成员函数的存储方式

C++用类去定义对象时，只为对象的数据成员分配内存空间，而该类的所有对象共享一个成员函数空间，即成员函数没有存储在对象的内存空间中。



说明：虽然调用不同对象的成员函数时执行的是同一段代码，但是执行结果一般是不同的，这是因为这段代码操纵的数据是不同的，为此C++设置了一个隐含的对象指针this，在调用时this指向不同的对象。

8.4 对象公有成员（数据或函数）的访问

- 通过对象名和成员运算符访问公有成员

  对象名.公有成员名

  例如：std1.display(); // 调用成员函数

- 通过指向对象的指针访问公有成员

  对象指针->公有成员名

  例如：Student *pStd = &std1; pStd->display();

- 通过对象的引用来访问公有成员

  对象引用.公有成员名

  例如：Student &rStd = std1; rStd.display();

8.5 类的封装性和信息隐蔽

口 外部接口与私有实现的分离

- 在声明类时，一般将成员数据指定为私有，而把成员函数指定为公有，外界通过公有成员函数实现对私有成员数据的访问，这样公有成员函数就是类的外部接口。
- 但公有成员函数操作数据的细节（即函数的实现）对用户是隐藏的，这种实现就是私有实现。

这样做会使得程序（尤其是大程序）的设计、修改和调试变得方便和简单。

□ 类声明和成员函数定义的分离

C++中通常将类的声明放在一个头文件中，而将相关成员函数的定义放在一个同名的cpp文件中。

在实现中，C++将若干常用的功能相近的类声明集中在一起，形成类库。类库包括C++编译系统提供的标准类库和用户类库。

类库有两个组成部分：（1）类声明头文件；（2）经过编译的成员函数的定义，它是目标文件。

用户只要把类库装入C++编译系统所在的子目录中，并在程序中用include命令把类声明的头文件包括到程序中，就能使用这些类和其中的成员函数。

□ 面向对象程序设计中的几个名词

- 类的成员函数在面向对象程序理论中又称为方法，方法是对数据的操作。
- 外界通过发消息使用对象的公有方法。
- 所谓消息其实就是一条命令，由程序语句实现。

例如：std1.display(); 是向对象std1发送一个消息，让它执行display方法。这里，std1是对象，display()是方法，语句std1.display();是消息。