阅读C++ prime plus 第14 章并总结关键内容.

1. C++的代码重用

类模板——另一种重用代码的方法。其作用是：我们能够使用通用术语定义类，然后使用模板来创建针对特定类型定义的特殊类。

1. 包含对象成员的类
2. valarray类简介

是一个模板类，用于处理数值或具有类似特性的类，它支持诸如将数组中所有元素的值相加以及在数组中找出最大和最小的值等操作。声明对象时，必须指定具体的数据类型。例如：

valarray<int>q\_values;

valarray类的一些方法:

operator ：用于访问各个元素。

size( )：返回包含的元素数。

sum( )：返回所有元素的总和。

max( )：返回最大的元素。

min( )：返回最小的元素。

1. 私有继承

当使用私有继承，基类的公有成员和保护成员都将成为派生类的私有成员。

区分：使用公有继承，基类的公有方法将成为派生类的公有方法；

使用私有继承，基类的公有方法将成为派生类的私有方法。

1.使用包含时将使用对象名来调用方法，而使用私有继承时将

使用类名和作用域解析运算符来调用方法。

1. 使用私有继承时，例中string对象没有名称，则访问内部的string对象的方法是：使用强制类型转换。例如：

const string & Student::Name() const

{

return(const string &)\*this;

}

1. 区分：包含和私有继承

包含：易于理解；继承会引起很多问题，如包含同名方法的独立的基类或共享祖先的独立基类，而使用包含不太可能遇到这样的麻烦；能够包括多个同类的子对象。

私有继承：提供的特性比包含多；派生类可以重新定义虚函数，但包含类不能。

1. 保护继承

保护继承是私有继承的变体，在列出基类时使用关键protected，例如：class Student :protected std::string,

protected std::valarray<double>

{...};

1. 多重继承（MI）

MI描述的是有多个直接基类的类。与单继承一样，公有MI表示的也是is-a关系。

1. 虚基类

虚基类使得从多个类（它们的基类相同）派生出的对象只继承一个基类对象。例如：

class Singer :virtual public Worker{...};

class Waiter :public virtual Worker{...};//通过在类声明中使用关键字virtual，可以使Worker被用作Singer和Waiter的虚基类。

1. 区分：虚基类和非虚基类

（1）如果基类是虚基类，派生类将包含基类的一个子对象；如果基类不是虚基类，派生类将包含多个子对象。

（2）如果类从不同的类那里继承了两个或更多的同名成员，则使用该成员名时，如果没有用类名进行限定，将导致二义性。但如果使用的是虚基类，则这样做不一定会导致二义性。

四、类模板

C++的类模板为生成通用的类声明提供了一种更好的方法。

1. 定义类模板

模板类以如下代码开头：

template<class Type>

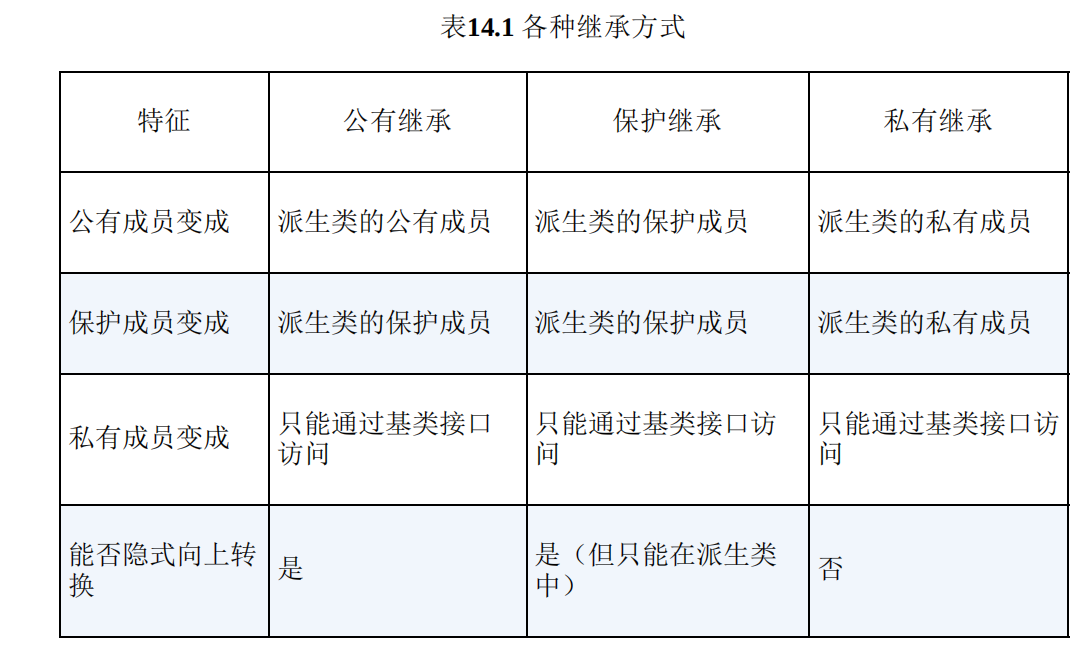
或者template<typename Type>

需要将所有模板信息放在一个头文件中，并在要使用这些模板的文件中包含该头文件。

1. 使用模板类

需要声明一个类型为模板类的对象，方法是使用所

需的具体类型替换泛型名。



1. 正确使用指针栈

让调用程序提供一个指针数组，其中每个指针都指向不同的字符串。

1. 模板多功能性

模板类可用作基类，也可用作组件类，还可用作其他模板的类型参数。还可以递归使用模板。

1. 模板的具体化

具体化：类模板与函数模板很相似，都可以有隐式实例化、显式实例化和显式具体化。即使用具体的类型生成类声明。

1. 隐式实例化：声明一个或多个对象，指出所需的类型，而编译器使用通用模板提供的处方生成具体的类定义。
2. 显式实例化

当使用关键字template并指出所需类型来声明类时，编译器将生成

类声明的显式实例化。

1. 显式具体化：是特定类型的定义。需要在为特殊类型实例化时，对模板进行修改，可以创建显式具体化。

（4）部分具体化：部分限制模板的通用性。

6.模板类和友元

模板的友元分3类：

非模板友元；约束模板友元，即友元的类型取决于类被实例化时的类型；非约束模板友元，即友元的所有具体化都是类的每一

个具体化的友元。

7.模板别名

可使用typedef为模板具体化指定别名，例如：

typedef std::array<double,12>arrd;

arrd gallons;