Const的作用总结

1. 常对象（将对象声名为const）
2. 在定义对象时可以加关键字const，用来指定常对象，常对象有定义过的初值，在对象的使用周期中，对象的值都不会被改变，常见形式有：

类名 + const + 对象名 or const + 类名 + 对象名

例如 const Time t1(10,15,36);之后任何试图改变对象t1值的语句都为非法。

1. 同时调用了常函数的成员函数也需要const声明，置于函数声明末，我们也称这样的成员函数为常成员函数，常成员函数可以访问常对像中的数据成员，但是不能修改数值。
2. 对于特殊的需求例如需要修改对象中某个数据成员的值可以在数据成员前加mutable ，形如mutable int XXX;
3. 常对象成员（将对象中的成员声明为const）
4. 直接在数据成员之前加const表示为常数据成员。
5. 成员函数在末尾加const表示为常成员函数。
6. 类中存在常成员函数和非const成员函数，可以使用非const成员函数引用常成员函数达到修改使用的目的。

附表

|  |  |  |
| --- | --- | --- |
| 数据成员 | 非const成员函数 | Const成员函数 |
| 非const数据成员  Const数据成员  Const对象 | 可以引用，也能改变值  可以引用，不能改变值  不允许引用 | 能引用，不能改变值  能引用，不能改变值  能引用，不能改变值 |

PS:常成员函数不能调用非const成员函数。

1. 常指针

\*+const+指针名称表示该指针为常指针变量.

可以类名+\*+const+指针名称直接对指针变量进行初始化。

虽然指向一个对象不变，但是可以改变对象的值。

用const固定指针指向的地址可以避免误操作。

1. 指向常对象的指针变量

Const + 类名 +\* 指针变量名

指向常变量的指针只能是指向常变量的指针变量。

通过指向常对象的的指针变量访问非const数据成员时不能改变该成员的值。

实参为const变量的地址不能用指向非const 型变量的指针作为形参。

希望在调用函数时对象的值不被改变，应把形参定义为指向常对象的指针变量，并以对象的地址作实参。

1. 对象的常引用

对象在函数中的引用形如void fun（Time &t）

如果不想在函数中修改传入实参的值，则可以用常引用，形如void fun（const Time &t）；