

C里const关键字相关问题 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/241/2021\\_2022\\_C\\_\\_\\_E9\\_87\\_8Ccons\\_c97\\_241857.htm](https://www.100test.com/kao_ti2020/241/2021_2022_C___E9_87_8Ccons_c97_241857.htm) const 最早想法是用于取代预处理器#define 这个宏，从而形成常量的概念。针对常量const对象，const指针及指向const的指针，函数const类型参数，const函数返回类型，const类成员，及const成员函数，及对const最后理解的一些总结来描述 const。 const对象和const类型的对象对于这两个概念的描述如下

```
1. int const object. //object是一个const量是不可以被修改 object = 2.error
2. const int object. //object是 const int型他所存放的内容不可以被修改
```

对于1，2这两种const用于对象，表述虽然不同但是效果是一样的。因为对象本身存放着内容对对象的改变就是对于对象内容的改变，同样改变后者也是在改变前者。所以语义上一样的。

const指针 和 指向const的指针 及两者结合 对于三个概念描述如下

```
1. int* const p. //指针p是const不能被修改 例如p. //修改p本身会error //修改p指向内容 *p = 2. //ok
2. const int* p. //p是指向一个整形常量的指针指向的内容不可以改变 p. //ok // *p = 2. //error
3. const int* const p. //指针p本身是不能被修改并且p所有有效的内容也不能被 //修改 *p = 2. error 和 p. error
```

const 参数修饰 和 参数返回类型的const修饰

```
1.const 参数修饰 此时函数参数修饰 const的具体用法 中用法是一样的 例如
void fun( const int i ) { i. } //error不能修改常量i
2.const修饰函数返回类型用法也是类似于 中,仅仅修饰的对象变化变成一个返回对象 例如：
const int fun() { static int i. return i. } int res = (fun()) //error不能修改常量返回对象
```

const类成员 和 const

成员函数 1. const成员 类const成员在构造期间会允许被初始化并且在以后不能被改变。我们就可以知道类const成员和一般const 变量是有所不同的，类const成员是对应于每个对象而言才有意义。因为他在构造期被初始化，只有当类实例化后才会进行构造。所以类const成员可以这样描述: 在类的每一次实例化时被初始化，在这个对象的生存周期中不可改变。 2. const 成员函数 描述: void class::memberfun() const {}. //此时这个const修饰的this所有类成员变量都不允许在这个函数体作用后被修改。这在设计上会带来一些好处，能防止你意外的处理带来的问题。 总结: const 常量 一般编译器不会分配空间只是维护一张表。而当extern 外部引用这个常量或者 “ & ” 对这个常量取地址时，编译器才会为其分配地址。const本身的机制比较复杂。 const 记忆法则 const修饰后面一个最近名称。我曾初学的时候被const 修饰搞的糊里糊涂，后来慢慢的总结我觉得这样理解最容易的。 例子: const int i. 此时const 仅仅修饰int 表明 i不是一个常量但是i的内容是常量。因为c/c 表达 对i的改变就是对i内容的改变所以 i也类似一个const。大家不妨可以用指针const修饰试试理解会有帮助的我想。对于所有非const 类型可以无条件转化为 const类型，但是后者不能自动转化为前者除非显式的强制转化去掉const性。这样做是有意义的，因为const类型是非const的一个子集是一种特殊，由普遍转化为特殊是合理的，就象模板特化，继承的向上映射都是有意义的。 记住所有const修饰的内容并不是永远不可改变，如果人为的强制转化编译器是不会提醒的。因为它没有义务这么做，所以我们对其转化时要小心。 在const类成员函数处理时，我们引入了mutable修饰类成员变量，经过其修

饰的成员变量可以在const类成员函数中被修改，编译器是允许的。而其他未被mutable修饰的成员还是按照const规则不能在const成员函数中被改变。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)