常函数

我们都知道this指针是指向类对象的, 且这个地址是一直不变的，即const性质的。

可以将它记为 Person \* const this;

那么如果要让指针指向的类对象中的值无法修改时，即

Const Person\* const this;

就要用到常函数了，如

Void c1() const{}

在这个函数中，this指针指向的类对象的值是无法修改的

同时，C++也给我们放宽了一点，提供了mutable关键字来突破const的限制，若类中有些值我们需要修改，便可以声明为mutable，即使是在常函数中，也是可以修改的。

常函数只能在函数的范围内控制类对象不能被修改，如果我想把这个控制范围扩大呢？扩大到整个类对象的范围，就变成了常对象了。即定义一个整个类对象中的值都是无法被修改的。

Const Person p2;

常对象不可以调用普通的成员函数，因为普通的成员函数有修改成员变量的风险，二常变量已经声明了不允许修改对象的属性。

因此，常对象只能调用常函数，

如

Void c2() const{}

Const Person p2;

P2.c2();

代码：  
#include<iostream>

using namespace std;

class Person {

public:

Person()

{

a = 0;

b = 0;

}

void text() const //常函数

{

cout << "a = " << a << endl;

cout << "b = " << b << endl;

}

int a;

int b;

};

void test()

{

Person p1;

p1.text();

const Person p2; //常对象

p2.text();

}

int main()

{

test();

return 0;

}