const用法总结

赵英良

1.修饰变量

- const int N=100;
- const double pai=3.1415926;
- N=1;　　X
- pai=3.14;　X
- cout<<N<<" "<<pai<<endl;　V
- int const N=100;
- double const pai=3.1415926;

2.修饰引用

- int a=3;
- const int b=5;
- const int &c=a;　　V
- const int &d=b;　　V
- a=10;　　V
- c=11;　　X
- int &e=b;　　X

3.修饰指针——常内容指针

- const char *pContent;
- 常内容
  - char s1[100],s2[100];
  - const char *pContent=s1;
  - pContent=s2;　　V
  - pContent[0]='a';　*pContent='a';　X
- const　(char)　*pContent;

3.修饰指针——常指针

- char * const pContent;
- pContent是常量
- 常指针
  - char s1[100],s2[100];
  - char * const pContent=s1;
  - *pContent='a'; V
  - pContent=s2; X
- 等价写法
  - (char*) const pContent;
  - const (char*) pContent;

两者都不可变

- const char* const pContent;
- For Example

```
char s1[100]="1234",s2[100];
const char* const pContent=s1;
s1[0]='a';  s2[0]='1';     V
pContent[0]='x';           X
pContent=s2;               X
cout<<s1; cout<<s2              V
```

4.修饰函数

出现在返回值前面

(1) const int fun1();　　//

int a; a=fun1();　　V

const int b=fun1();　V

(2) const int *fun2(int &a);//返回常内容指针

- 调用时

```
int  *q, c=5;
const int *pValue;
pValue=fun2( c );          V
q=fun2(c);                 X
```

(3) int * const fun3(int &a);

- 调用时
- int a=6;
- int * const pValue = fun3(a); V
- const int *r=fun3(a); V
- int *q =fun3(a); V
-

const出现在形参中

```
fun( const int N , int * const q, const int *p );
int  fun4( const int a);
int  fun4( const int a)
{
   int b;
   b=a+1;
   //a=a+1;    X
   return a+b;
}
```

5.const修饰类对象/对象指针/对象引用

- 设POINT是一个类
- const POINT a;　//常量对象
- const POINT * p;//常对象的指针
- POINT * const r;//常量指针
- POINT b;
- CONST POINT &c=b;　//常引用
- 常量对象，那这个对象的所有成员都不能修改
- 常引用，不能通过这个引用名修改对象的成员

- const POINT x; const int a; 有所不同
- const修饰的对象
  - 只能调用对象的const成员函数
  - 不能调用非const成员函数
  - 因为非const成员函数会有修改成员变量的企图

举例

```
class POINT
{    int x,y;
  public:
 POINT(int a,int b){x=a;y=b;}
    void f1();
    void f2() const;
};
void POINT ::f1()
{  x=11;y=12;    //V
  cout<<"function 1 \n";  }
  void POINT ::f2() const
{    //x=21;y=22;     // X
    cout<<"function 2 \n";   }
```

```
int main() // 主函数
{
const POINT b(1,2);
POINT c(3,4);
c.f1();c.f2();    //V
//b.f1();    //X
b.f2();        //V
const POINT *p=new POINT(5,6);
//p->f1();    //X
p->f2();        //V
return 0;
```

后缀const的函数
不能改变对象的成员变量.
也不能调用类中任何非const成员函数

6. const修饰类的成员变量

const修饰类的成员变量，表示成员常量，不能被修改，同时它只能在初始化列表中赋值。

```
class A
{
private:
   const int Value; //成员常量不能被修改
public:
   A(int x): Value(x) {          //只能在初始化列表中赋值
   //Value=10;     //错误
   };
};
```

小结

- int a,b,d[00]
- const int N=10;//常量，值不能改
- const int *p;//常内容指针，指向的单元的内容不能改
- int * const q=d;//常指针,指针的值不能改,不能再指向其它单元
- const int &c=a; //常引用,不能通过常引用修改其值
- const POINT x; //常量对象，只能访问常成员函数
  - 常成员函数，不能修改数据成员的值，也不能访问非const成员函数
  - 常成员，只能在构造函数头部初始化