

1.3.4const常量

- const常量类型表示一个“常值变量”，即值不能被改变的变量。
- 声明一个const型变量只需加上关键字const，以可加在类型说明符前或后。例如：

```
1      const int size = 20;
2      int const size = 20;
```

- 用constt声明的变量必须用常量或常量表达式初始化。
- 一旦变量被声明为const型，禁止任何试图修改该变量的操作。如果知道一个变量赋初值后在生命周期里其值不变，用const限定该变量是程序设计的好习惯，如数组长度。
- 宏定义在预编译时做文本替换，不做类型检查。const在编译时会进行类型检查。

例子

```
1 void eg2_08(){
2     #define size1 20
3     const int size2 = 20;
4
5     int array1[size1] = {1};
6     int array2[size2] = {2};
7
8     for(int i=0; i<size1; i++) cout << array1[i] << ", "
9     cout << endl;
10    for(int i=0; i<size2; i++) cout << array2[i] << ", "
11    cout << endl;
12 }
```

常指针与常值变量指针

- 常指针：地址不能变
- 常值变量指针：值不能变

```
1 double x = 3.14;
2 double* const p1 = &x; //常指针：地址不能变
3 const double* p2 = &x; //常值变量指针：值不能变
```

const常量类型

```
// eg2_08
const int size = 20;
int a[size];
const int b[4] = {1, 2, 3, 4};
size = 100; //错误: 非法修改常值变量
size++; //错误: 非法修改常值变量
b[0] = 10; //错误: 非法修改常数组元素
```

```
// eg2_09
double arr[5] = {1.1, 1.2, 1.3, 1.4, 1.5};
const double x = 12.78;
double* p0 = &x; //错误: 普通指针不能指向常值变量
const double* p1 = &x; //常值变量指针
double* const p2 = arr;
double* const p3 = &x; //错误: 常指针不能指向普通变量

p1 = arr;
p1++;
(*p1)++; //错误: 不能修改常值变量指针所指的变量
(*p2)++;
p2++; //错误: 不能修改常指针
```

```
const int* FunA() {
    static int x = 1;
    ++x;
    return &x;
}

void eg2_10() {
    int y;
    y = *FunA(); //合法
    *FunA() = 2; //非法
}
```

