

从 C 到 C++

Letian Lin — yingziyu-Lin@outlook.com

Feb 23, 2024

1 引用的概念和应用

定义引用：类型名 & 引用名 = 某变量名；

只是别名，修改引用的同时会修改原值。

定义的时候必须要初始化，不能修改引用的对象，不可引用常量和表达式。

e.g. `int &r = n;`

```
1 double a = 4, b = 5;  
2 double &r1 = a;  
3 double &r2 = r1;  
4 r2 = 10;  
5 cout << a << endl; //10  
6 r1 = b;  
7 cout << a << endl; //5;
```

引用还可以作为函数返回类型，只需要 `return` 相关的变量，就可以 `return` 对应变量的引用。

可以用 `const` 修饰，不能修改用 `const` 修饰的引用。

```
1 int n = 4;  
2 const int &a = n;  
3 a = 10; //编译错误  
4 n = 10; //正常
```

`const T&` 的引用是不可以用来初始化 `T &` 的内容的，二者的类型其实不相同，只能在强制类型转化后进行初始化。

2 const 关键字

2.1 定义常量

```
1 const int MAXVALUE = 10000000; //建议常量使用全大写
2 const string NAME = "yingziyu";
```

2.2 定义常量指针

常量指针指向的内容只读，但其自己可以被修改指向别的内容。

```
1 const int *p = n;
2 *p = 5 //CE
3 n = 5; //OK
4 p = &m; //OK
```

不可将常量指针赋值给普通指针。

常量指针还可以避免函数内部将参数指针的所指的内容修改。

```
1 void myPrint(const char *s){
2     strcpy(s, "this"); //CE
3     printf ("%s", s); //OK
4 }
```

3 动态内存分配

3.1 分配一个变量

用 new 申请内存，delete 释放内存

```
1 int *pn = new int;
2 *pn = 5;
```

3.2 分配一个数组

P = new T[N];

T 是任意类型名，N 是一个整数。P 指向数组的第一位。

3.3 delete

销毁用 new 分配出来的动态空间。

```
1 int *p=new int;  
2 *p = 5;  
3 delete p;  
4 delete p;//CE
```

4 内联函数，函数重载，函数缺省参数