

# 第23章 指针字面量nullptr

《现代C++语言核心特性解析》 谢丙堃

# 零值整数字面量

- 0既是一个整型常量又是一个空指针常量

```
#ifndef NULL
    #ifdef __cplusplus
        #define NULL 0
    #else
        #define NULL ((void *)0)
    #endif
#endif
```

```
char *p = NULL;
int x = 0;
```

# 零值整数字面量

- 二义性问题

```
void f(int)
{
    std::cout << "int" << std::endl;
}
void f(char *)
{
    std::cout << "char *" << std::endl;
}

f(NULL);
f(reinterpret_cast<char *>(NULL));
```

```
std::string s1(false);
std::string s2(true);
```

## nullptr关键字

- nullptr表示空指针的字面量，是一个std::nullptr\_t类型的纯右值

```
char* ch = nullptr;  
char* ch2 = 0;  
assert(ch == 0);  
assert(ch == nullptr);  
assert(!ch);  
assert(ch2 == nullptr);  
assert(nullptr == 0);
```

# nullptr关键字

- std::nullptr\_t类型的定义

```
namespace std
{
    using nullptr_t = decltype(nullptr);
    // 等价于
    typedef decltype(nullptr) nullptr_t;
}

static_assert(sizeof(std::nullptr_t) == sizeof(void *));
```

## 用std::nullptr创建nullptr

- 例子

```
std::nullptr_t null1, null2;
```

```
char* ch = null1;
```

```
char* ch2 = null2;
```

```
assert(ch == 0);
```

```
assert(ch == nullptr);
```

```
assert(ch == null2);
```

```
assert(null1 == null2);
```

```
assert(nullptr == null1);
```

## nullptr是纯右值

- 例子

```
// nullptr1和nullptr2是左值，可以成功获取对象指针，  
// 并且指针指向的内存地址不同  
std::nullptr_t nullptr1, nullptr2;  
std::cout << "&nullptr1 = " << &nullptr1 << std::endl;  
std::cout << "&nullptr2 = " << &nullptr2 << std::endl;  
  
// 编译失败，取地址操作需要一个左值  
std::cout << "&nullptr = " << &nullptr << std::endl;
```



感谢您的观看  
欢迎关注