

《Exceptional C++ 中文版》

图书基本信息

书名：《Exceptional C++ 中文版》

13位ISBN编号：9787508314853

10位ISBN编号：7508314859

出版时间：2003-3

出版社：中国电力出版社

作者：萨特 (Sutter Herb)

页数：220

译者：卓小涛

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《Exceptional C++ 中文版》

内容概要

本书详细地讨论了如何编写健壮高效的代码以及模块设计技术。本书采用了自问自答的语言风格，它讨论了实际程序设计中普遍存在的，又非常不易察觉的问题，并且进行了深刻的分析，提出了具有广泛意义的解决方案。本书值得每个想成为优秀c++程序员的人阅读。我相信，任何c++程序员都会在本书的阅读过程中有许多收获，甚至有一种醍醐灌顶，恍然大悟的收获和喜悦。

精彩短评

- 1、好，值得收藏的书
- 2、可以说是对C++的高级技巧总结的最好的一套小册子了 很适合有一段开发经验后看,和之前自己遇到的问题总结类比,会有一种醍醐灌顶的大彻大悟
- 3、类似More Effective C++的Tips小册子，比适合中高级读者阅读，难度比MEC简单一些。
- 4、内容有点老，但是紧凑而深入。
- 5、不好懂
- 6、C++是个令人抓狂的语言，这本书其实还不错
- 7、c++必读
- 8、浏览一遍先
- 9、如何编写异常安全的C++代码
- 10、翻译的不好啊
- 11、读完了，有时间还是必读的，尤其是使用STL和关于Impl
- 12、推荐原著，翻译得欠佳
- 13、这本书和《Effective C++》一样，都是给程序员一些工程经验，但是比《Effective C++》更难一些，两本书有交织的地方，推荐每个C++程序员阅读。
- 14、非常好的总结，大部分已看完
- 15、读过 Effective C++/More Effective C++后，仍能震撼你的作品。
- 16、如果读了E和ME，就没必要读这本书了。GotW式到表现形式导致信息量不多，三星足矣。
- 17、不错的C++书籍
- 18、参考书系列
- 19、当时这一系列的C++书是百读不厌！
- 20、怎样编写异常安全的代码。

章节试读

1、《Exceptional C++中文版》的笔记-第105页

将编译期依存性减至最小。

```
#include <iosfwd>  
前置声明
```

Page109
Pimpl

句柄/实体的一种特殊形式

Pimpl 习惯用法

编译器防火墙

Pimpl 只是一种用来隐藏类的私有成员的非透明指针。

Page119
Fast Pimpl

```
static const size_t sizeofx;  
char x[sizeofx];
```

```
assert  
new (&x[0]) X
```

```
reinterpret_cast<X*>(&x[0])->~X();
```

```
::operator new()  
malloc()
```

```
static void* operator new (size_t);  
static void operator delete (void*);
```

Page127
名字查找和接口规则
Koenig 查找规则
接口规则

```
namespace  
{  
}
```

```
class  
{  
};
```

Page143

名字隐藏

```
d.Base::g(i);
```

```
using B::g;
```

2、《Exceptional C++中文版》的笔记-第171页

对象标识

自动转换

转换操作符

非显式的构造函数

对象生成期

Page189

const

不要声明 const 的按值传递的函数参数

const

mutable

Page196

类型转换

```
`const_cast
```

```
`dynamic_cast
```

```
`reinterpret_cast
```

```
`static_cast
```

bool

```
`typedef ... bool
```

```
`#define bool ...
```

```
`enum bool ...
```

```
`class bool ...
```

传递调用函数

Page206

控制流

3、《Exceptional C++中文版》的笔记-第168页

封装指针成员

```
class
```

```
{
```

```
public: // ...
```

```
protected: // ...
private:
    auto_ptr<T> pimpl_;
};
```

*资源管理类

```
const auto_ptr<T>
`operator *()
`operator -&gt;()
`get()
不能做 =、release、reset 等操作
```

4、《Exceptional C++中文版》的笔记-第80页

重载虚函数

```
using Base::f;
```

Page85
类的关系

Page94
继承的使用和滥用

Liskov 替换规则

不要求更多，也不承诺更少。

Page103
面向对象程序设计

5、《Exceptional C++中文版》的笔记-第27页

编写异常安全的代码

代码的复杂性

6、《Exceptional C++中文版》的笔记-第73页

类的设计与继承

```
T& T::operator +=(const T& other)
{
    // ...
    return *this;
}
```

```
const T operator +(const T& a, const T& b)
{
```

```
T temp(a);
temp += b;
return temp;
}
```

一个设计和风格良好的类

```
class Complex
{
public:
    explicit Complex(double real, double imaginary = 0) : real_(real), imaginary_(imaginary)
    {
    }

    Complex& operator +=(const Complex& other)
    {
        real_ += other.real_;
        imaginary_ += other.imagiary_;
        return *this;
    }

    Complex& operator ++()
    {
        ++real_;
        return *this;
    }
    const Complex operator ++(int)
    {
        Complex temp(*this);
        ++*this;
        return temp;
    }

    ostream& print(ostream& os) const
    {
        return os &&& "(" &&& real_ &&& ", " &&& imaginary_ &&& ")";
    }

private:
    double real_, imaginary_;
};

const Complex operator +(const Complex& lhs, const Complex& rhs)
{
    Complex ret(lhs);
    ret += rhs;
    return ret;
}

ostream& operator &&&(ostream& os, const Complex& c)
```

```
{  
    return c.print(os);  
}
```

7、《Exceptional C++中文版》的笔记-第1页

第一章 泛型程序设计与 C++ 标准库

Page1

异常安全性

迭代器

临时对象

迭代器：

- 有效值
- 有效生存期
- 有效范围
- 非法的内置类型操作

Page4

不区分大小的字符串

stricmp()

```
typedef basic_string<char> string;
```

```
struct ci_char_traits : public char_traits<char>  
{  
    static bool eq(char c1, char c2)  
    {  
        return toupper(c1) == toupper(c2);  
    }  
    static bool lt(char c1, char c2)  
    {  
        return toupper(c1) < toupper(c2);  
    }  
    static int compare(const char* s1, const char* s2, size_t n)  
    {  
        return memicmp(s1, s2, n);  
    }  
    static const char* find(const char* s, int n, char a)  
    {  
        while (n-- > 0 && toupper(*s) != toupper(a))  
        {  
            ++s;  
        }  
        return n > 0 ? s : 0;  
    }  
};
```

```
typedef base_string<char, ci_char_traits> ci_string;
```


Page8

不区分大小写的字符串

LSP Liskov 替换规则

Page11

最大可重用的泛型容器

```
template <typename T, size_t size>
```

```
class fixed_vector
```

```
{  
};
```

Page12

最大可重用的泛型容器

operator=

强类型安全性

Page19

临时对象

const& 常量引用

Page24

使用标准库（或称再谈临时对象）

find

sort

8、《Exceptional C++中文版》的笔记-第151页

内存管理

C++ 的内存区域：

- 常量数据（字符串字面值）
- 栈（自动变量）
- 自由存储（new/delete）
- 堆（malloc/free）
- 全局或静态（静态成员，全局变量，文件范围变量）

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com