

# C++标准程序库读书笔记

## CH3 基本概念

### 命名空间

使用C++标准程序库的任何标识符时，可以有三种选择：

- 1. 直接指定标识符。
- 2. 使用using declaration。
- 3. 使用using directive。

### 头文件

从操作系统的角度视之，新头文件命名方式并不意味着标准头文件没有扩展名，标准头文件的`#include`怎么处理，由编译器决定。C++可以自动添加一个扩展名，甚至可以内建声明，不读入任何文件。

无扩展名这一条件只使用与标准头文件。

### 错误处理和异常处理

标准异常类别：

- 语言本身支持的异常。
- C++标准程序库发出的异常。
- 程序作用域之外发出的异常。

语言本省支持的异常：

- 全局操作符`new`操作失败，会抛出`bad_alloc`异常。
- 执行期间，当一个加诸于`reference`身上的动态型别转换操作失败时，`dynamic_cast`会抛出`bad_cast`异常。
- 执行期间型别辨识过程中，如果交给`typeid`的参数为零或空指针时，`typeid`操作符会抛出`bad_typeid`异常。
- 如果发生非预期的异常，`bad_exception`异常会接手处理，方式如下：

当函数抛出异常规格以外的异常时，`bad_exception`就会调用`unexpected()`

C++标准程序库发出的异常：

- `invalid_argument`表示无效参数。
- `length_error`指出某个行为可能超越了最大极限。
- `out_of_range`指出参数值不在预期范围内。
- `domain_error`指出专业菱领域内的范畴。

程序作用域之外发出的异常

- `range_error`指出内部计算时发生区间错误。
- `overflow_error`指出算术运算发生上溢位
- `underoverflow_error`指出算术运算发生下溢位

异常类别的头文件

- 基础类别`exception`和`bad_exception`定义于`<exception>`
- `bad_alloc`定义于`<new>`
- `bad_cast`，`bad_typeid`定义于`<typeinfo>`
- `ios_base::failure`定义于`<ios>`
- 其他异常类别都定义于

### 配置器

C++标准程序库在许多地方采用特殊对象来处理内存配置和寻址。这样的对象称为配置

器，配置器体现出一一种特定的内存模型，称为一个抽象表征，表现出内存需求之内存低阶调用的转换。