

C++20高级编程勘误

1

- p3 1.1 常称为->常被称为
- p9 1.1.3 C++操作符重载->C++运算符重载
- p13 表格中间一栏 第二格 true/false(非零或零)->真/假 (非零/零) 第六格 两处 增1->增加1
- p16 下边的警告 旧式风格枚举->旧式风格enum枚举
- p16 1.1.7 括号里的注释应写在页脚
- p21 1.1.11 倒数第二行 三元->三向
- p22 书中代码的第五行
 `result == strong_ordering::less` 应为 `is_lt(result)`
第六行
 `result == strong_ordering::greater` 应为 `is_gt(result)`
第七行
 `result == strong_ordering::equal` 应为 `is_eq(result)`
- p25 4. [[deprecated]] 第二行 不赞成使用->弃用
- p35 1.1.24 第一段最后两行的四处 范围->作用域
- p40 第二段 第二行和第三行 三处 比特->位
- p46 1.1.29 第二行 则会在编译时满足需要对函数求值->则会在需要时在编译期对函数求值

2

- p73 警告中 废弃->弃用
- p73 2.1.2 第二段第二行 等效->相同
- p74 第一段第三行 字面量的池操作->字面量池的操作
- p76 字符串比较 第三行 双重性->对偶性
- p77 第二段第二行 堆中的变量->栈中的变量

3

- p92 3.2.1 1.说明用途的注释 第一段第三行 注释应写在页脚
- p103 5.大写 第四行 下画线->下划线

7

- p181 7.3.2 第二段第三行 发放大块内存中片段的类->发放大块内存中的片段
- p182 7.3.4 最后一行 编写高效的C++代码->编写高效的C++程序

- p195 7.5.7 第三行 已经不赞成使用->正式弃用了

8

- 标题应为 熟悉类和对象
- p214 何时调用拷贝构造函数一节 第二段 第二行与第三行 第三段第一行 三处 inString->value

9

- p227 本章内容第七行 reference->引用

从第十章开始

- 本书的所有“复制构造”全部应改为“拷贝构造”
- 本书的所有“复制赋值运算符”全部应改为“拷贝赋值运算符”
- 本书所有的“运行期”全部应改为“运行时”
- 本书所有的“编译时”全部应改为“编译期”
- 本书所有的“堆栈”全部应改为“栈”

10

- p274 第一行 从技术上讲->严格来讲
- p278 第二段 第三行和第五行 两处 双精度数->double
- p280 倒数第二行 std::unique_ptrs->std::unique_ptr
- p281 译者注 删掉，因为前文出现过
- p288 注意中 下画线->下划线
- p290 第二段第三行 虚编号->虚成员
- p294 10.4.5 第二段 三处矢量->vector
- p295 第一行 矢量->vector
- p304 第二行 子句->声明
- p317 最后一行 它对可以转化为什么有很多的限制->它对什么可以转化至什么有很多的限制
- p318 第一行 通常,... 这里应该另起一段
- p320 练习10.3 基于范围for的循环->基于范围的for循环 两处向量->vector

11

- p321 第一段第三行 实用程序->实用工具
- p321 11.1 第一段第四行 上万行->数万行
- p323 第三段 模块权限->模块主体(module purview)

- p326 11.5 第二行 在任何想要的层次结构中构建模块->用任何的层次结构来组织模块
- p328 第二段 只有一个具有特定分区名称的文件->每个特定分区名称只能用于唯一的文件
- p328 警告下面的一段 应与上文警告中的内容合为一段警告 第二行的“相反”删掉
- p329 第三段 随后通过datamodel:person接口分区文件和datamodel接口文件导入该分区->随后 atamodel:person接口分区文件和datamodel接口文件便都可以导入该分区
- p329 实现分区一节 第一段第一句应改为 分区并不需要在模块的接口文件中生命，也可以在模块的实现分区文件(扩展名为.cpp的常规源码文件)中声明。

12

- p341 12.1 第一行 范型 -> 范式
- 第二行 重用很多不同的值 -> 被很多不同的值重用
- p343 第二行 unique_ptr矢量组中的矢量 -> 使用unique_ptr的vector的vector
- p344 第二行 将讨论 -> 讨论了
- p344 本页代码段中倒数第二个函数void GameBoard::swap->void swap
- p347 注意下边的第一段 两处 矢量组中的矢量->vector的vector
- 还是这段 属性类和对象->熟悉类和对象
- p347 最后一行 注意：之前的类名是::Grid -> 注意::之前的类名是Grid
- p348 注意中 第二行 初始化为0 -> 零初始化
- p349 12.2.2上边的一行 自由存储区(堆) -> 自由存储区
- p350 1的第三段 非常量 ->非const
- p350 2的第三段 concepts -> concept
- p352 第二行 矢量组中的矢量->vector的vector
- p354 注意上边的一段 向量、数组 ->vector、array
- p360 最后一段第二行 “例如”一句 应为 例如，Grid的const char*特化在实现at()方法时，返回的是optional<string>，而非optional<const char*>
- p361 第一段 部分特化->偏特化
- p362 第三行 Gird::DefaultWidth -> Gird<T>::DefaultWidth
- p363 注意中 sentinel值 -> 哨兵值
- p368 注意中 应改为 既然C++支持自动返回类型推导和decltype(auto)，建议使用其中的一种机制，而不是替换函数语法。
- p368 12.4章节编号 应为 12.3.5 这样后续章节的编号就全部错了 需要修改
- p369 12.6 第一段第一句话应改为 C++20引入了概念(concept)，一种用来约束类模板和函数模板的模板类型和非类型参数的命名要求。
- p369 12.6 第二段 std::字符串 -> std::string
- p370 第一个小标题require表达式应为三级标题 1.require表达式 以及p371最后的组合概念表达式 应为 2.组合概念表达式 不用缩进
- p370 最后一行 类型需求-> 类型requirement

- p372 第一行 它要求一个类型既可以是自增，也可以是自减的 -> 它要求一个类型既是可以自增，也是可以自减的
- p372 注意上边的一段 下面的概念要求类型T既是默认的也是可复制的 -> 下面的概念要求类型T既是可以默认构造也是可以拷贝构造的
- p372 最后一行 下面的代码编译的很好 -> 下面的代码可以成功编译 模拟->满足
- p373 第二行 建模->满足
- p373 倒数第四段和最后一段 require -> requires
- p374 第四段 两处require -> requires
- p375 12.6.6的最后一段和本页的最后一段 两处require -> requires

13

- p379 第一段第二行 也有可能提供操作系统特定的API -> 也有可能通过操作系统特定的API
- p382 警告中 明智 -> 审慎
- p386 最后一段第一行 符号化 -> 标志化
- p399 倒数第二段最后一行 下一节 -> 下面

14

- p405 14.1第一段第一行 操作系统界面 -> 操作系统接口
- p405 14.1第一段第四行 任何程序开发者都必须包含错误处理功能 -> 任何程序开发者都必须拥有错误处理的能力
- p407 第四段 调用堆栈 -> 调用栈
- p408 最后一段 除0 -> 除以0
- p416 第一行 编译时评估 -> 编译期求值
- p416 警告 自C++11之后，已不赞成使用异常规范 -> 自C++11之后，已经移除了对异常规范的支持
- p419 14.3.3 第五段第六段 两处 泛型类FileError -> 通用类FileError

15

- 本书的大部分行内的箭头运算符存在排版问题，本章尤其多。很多 -> 显示成了类似 _> 的样子，需要调整，可以按照p437倒数第二行的箭头来改
- p437 第一段第二行 能让自己的类具有内建类型(例如int和double)的类似行为是有益的 -> 能让自己的类具有类似内建类型(例如int和double)的行为是有益的
- p438 15.1.1 第三段 客户->使用者
- p438 15.1.2 最后一行 插入运算符->流插入运算符 最后一句 ->对于这种具有两个意义的运算符，可以重载所有的意义。
- p441 第二列倒数第二栏 三元比较运算符->三向比较运算符
- p443 表格第二列倒数第二栏 解除引用指针-成员 -> 解除引用成员指针

- p445 第五段第一行第二行 两处 三元->三向
- p447 倒数第四段 第二行 解密C++ I/O ->C++ I/O揭秘
- p452 第二段第一行 为const对象调用const operator[], 因此无法增加数组大小 -> 因为调用const operator[]的是const对象, 所以无法增加数组大小
- p454 第三段第一行 普通指针->普通的原始指针 第二行 真正的指针(普通指针) -> 原始指针
- p456 第二段第一行 double表达方式 -> double的表示方式
- p457 15.8.2 第二行代码 加粗
- p466 15.10 第一段 合理性->原因 威力->能力

16

- p470 16.1.1 第二段第三行 跳过->跳过了
- p470 16.1.1 第二段第二行 在继续处理本章和阅读后续章节之前 -> 在继续阅读本章和后续章节之前
- p471 注意中 从技术角度看 -> 严格来讲
- p471 16.2.2 第二行 酌情二字删掉
- p471 16.2.4 第一段中所有的“存储”改为“内存” 悬挂指针->悬空指针
- p477 倒数第二段 双向队列->双端队列
- p478 第一段最后一行 而不总是像vector那样需要堆访问权限 -> 而不是像vector那样总是访问自由存储区
- p479 10 第一段第四行 operator<运算符 -> operator<
- p483 表16-5 第五行第二列 三路比较->三向比较
- p486 7.二叉树搜索算法 标题以及第一句中的 二叉树->二分

17

- p491 17.1 第一行 泛型抽象 -> 通用抽象
- p491 17.1 第三段第一行 所有迭代器都必须可通过复制来构建、赋值 -> 所有迭代器都必须是可以拷贝构造和拷贝赋值的
- p494 第一段第二行 整数矢量 -> int类型的vector
- p495 17.1.2 第二段下面 五个类型别名中 第四行 冒号左边的指针 -> pointer 第五行 冒号左边的引用->reference
- p495 最后一行 向量 -> vector
- p497 17.2.1 第三段 两处 向量->vector
- p501 第三段前两行 应为 代码的第2行通过默认构造函数[1]创建一个MoveableClass实例。第一个 push_pack()的调用触发拷贝构造函数[2], 将mc复制到vector中。在这个操作之后, vector有空间容纳一个元素, 即mc的第一个副本。
- p501 第四段的第二第三句应为 这个大小调整会导致调用移动构造函数[4]将每个元素从旧vector移动到调整大小后的新vector。触发拷贝构造函数[3]将mc第二次复制到vector中。

- p501 第五段 新向量->新vector
- p503 投影一节中 第二段和第三段中 两处 向量->vector
- p504 17.4.2 列表第一项 惰性评估->惰性求值
- p505 续表中 第七行第二列 元组->tuple
- 第八九行第二列 四处 队->pair
- 第十行第二列 第三行 公共范围 ->普通范围
- p506 第二段 惰性评估->惰性求值
- p507 第三段第一行 向量->vector 第三行 值向量 values vector
- p508 17.4.3 第三段第二行 向量 -> vector
- p509 17.5 最后一行 惰性执行->惰性求值
- p509 练习17-1 懒惰地构造 -> 惰性构造
- p510 练习17-2 第一行 向量对->pair的vector 第四行 公共范围 ->普通范围
- p510 练习17-2 向量对 -> pair的vector
- p510 练习17-4 向量 -> vector

18

- p515 第二段 两处 区间 -> 范围
- p516 第一段第二行 区间 -> 范围
- p517 第四段 自由存储区(堆) -> 自由存储区
- p524 算法复杂度和迭代器失效一节 第二行 采用线性复杂度 -> 服从线性复杂度
- p525 预留容量小标题上边一行 std::size() -> std::ssize()
- p533 4.list大小 一节 第一句 公开->暴露
- p537 18.2.6 第一段第二行 自由存储区(堆) -> 自由存储区
- p547 最后一段代码应为

```
map m {
    pair { "Marc G."s, 123 },
    pair { "Warren B."s, 456 },
    pair { "Peter V.M."s, 789 }
}
```

- p550 第二段 可按如下更优美的方式编写代码 -> 可按如下可读性更高的方法编写代码
- 第三段 实现的方式会更优美 -> 实现的方式会更优雅
- p551 本页所有的“节点”应改为“结点”
- p554 本页所有的“iterator”应改为“迭代器”
- p570 练习18-4 双精度值 -> double

19

- p572 最后一段第一行 向量 -> vector
- p574 第一段代码第二行 第一个e 应大写
- p578 第一行 提供->提高
- p579 第一段代码 第一行 C应小写
- p581 中间加粗的代码段第三行 P应小写
- p586 第二个警告的最后一行 应为 即使没有这些问题，也不推荐使用全局变量。
- p589 19.4.7 第二行 但在C++20之后无效 -> 但在C++20之后可以使用

20

- p593 20.1 第二段第二行 跨越最后一个元素->最后一个元素之后
- p597 20.1.3 第五行 掌握类和对象 -> 精通类和对象
- p602 4.技术算法 第三段 闭包区域 -> 闭包作用域
- p603 2.转移 标题应为 转换
- p605 3.拷贝 第一段第三行 [b,e] -> [b,e)
- p607 6.删除 标题应为 擦除
- p608 7.擦除 标题应为 删除
- p608 第二个注意中 $O(n^2)$ -> $O(n^2)$
- p609 9.乱序 一节中 第一段第三行 "播种"他们 -> "设置种子" (保留引号)
- p610 10.抽样 播下一) 设置
- p611 20.2.3 第一段第二句应为 for_each()对范围内的每个元素执行回调，for_each_n()对范围内的前n个元素执行回调。
- p620 20.2.10 第一段第二行 concepts -> 概念
- p620 20.2.10 第二段第二行 "将"字应删掉

21

- 本章所有的"捕捉"应改为"捕获"

22

- p647 本页中间的五项组件 第四个和第五个应改为 日期(Date)(C++20) 时区(Time Zone)(C++20)

23

- p661 第二段第二行 分布器 -> 分布
- 第三行 该分布 -> 分布

24

-
- p671 第一段 可选词汇表类型->optional词汇表类型
 - p676 2.tie()小节中 第一段 两处 引用tuple -> 引用的tuple
 - p676 2.tie()小节中 第一段下方的代码中 第六行应加粗
 - p676 最下方的一行代码 应加粗
 - p677 倒数第二段 三元运算符 -> 三向比较运算符