鉴定文件 标识：PT-R4444-TN

公开

测试项目软件

鉴定测试记录

总页数：35

**中国科学院卫星软件评测中心**

2024年07月09日

测试项目软件

鉴定测试记录

**编写： 陈俊亦 日期：20240707**

**校对： 陈俊亦 日期：20240708**

目 录

[1 范围 2](#_Toc161644960)

[1.1 标识 2](#_Toc161644961)

[2 测试记录 3](#_Toc161644962)

[2.1 文档审查测试 3](#_Toc161644963)

[附录A 文档审查单 4](#_Toc161644964)

[A.1 软件文档齐套性检查单 4](#_Toc161644965)

[A.2 需求类文档审查单 4](#_Toc161644966)

[A.3 设计类文档审查单 7](#_Toc161644967)

[A.4 手册类文档审查单 9](#_Toc161644968)

[附录B 代码审查单 11](#_Toc161644969)

[附录C 静态分析结果记录 27](#_Toc161644970)

[附录D 未覆盖情况 33](#_Toc161644971)

# 

# 范围

## 标识

1. 本文档的标识号：PT-R4444-TN；
2. 本文档的标题：测试项目软件测试记录；
3. 本文档适用的范围：定义“测试项目软件”软件的测试记录。

系统内各软件总体情况如下表所示。

表5‑4被测软件基本信息表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 软件所在子系统/设备名称 | 软件名称 | 软件类型 | 重要/安全等级 | 运行环境 | 开发环境 | 编程语言 | 版本 | 代码规模  （软件规模/代码行） | 接收日期 | 研制单位 |
|  | 测试项目 | 测试项目 | 新研 | C | gcc | Linux | C | 1.1 | 17777 | 2024-06-15 | 上海微小卫星工程中心 |

# 测试记录

## 文档审查

### 文档审查

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| 测试用例名称 | | 文档审查 | | | 标 识 | | YL\_DC\_WDSC\_001 | |
| 追踪关系 | | 软件测试依据：测评大纲  测试需求分析：（6.2.1.1）文档审查  测试需求标识：XQ\_DC\_WDSC | | | | | | |
| 测试用例综述 | | 测试人员阅读文档，依据文档检查单对软件文档进行审查，审查文档内容是否完整、文档描述是否准确、文档格式是否规范、文档是否文文一致 | | | | | | |
| 用例初始化 | | 开发方已提交被测文档 | | | | | | |
| 前提和约束 | | 提交的文档出自委托方受控库，是委托方正式签署外发的 | | | | | | |
| 测试步骤 | | | | | | | | |
| 序号 | 输入及操作 | | 期望结果与评估标准 | | | 实测结果 | | 通过  与否 |
| 1 | 按照测试需求中文档齐套性检查单检查需求类、设计类、用户类、测试类文档是否齐套 | | 文档齐套性检查单全部通过，软件文档齐套 | | | 文档齐套性检查单全部通过，软件文档齐套 | | 未执行 |
| 2 | 按照测试需求中文档需求规格说明、设计文档等审查单，对相关文档进行审查 | | 文档满足完整性、准确性、规范性和一致性的要求 | | | 文档检查单全部审查通过，文档内容完整、文档描述准确、文档格式规范、文档文文一致 | | 未执行 |
| 执行状态 | | 未执行 | | 测试时间 | | 2024-05-30 | | |
| 测试人员 | | 陈俊亦 | | 监测人员 | | 陈俊亦 | | |
| 问题单标识 | |  | | | | | | |
| 备注 | | / | | | | | | |

## 静态分析

### 静态分析

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| 测试用例名称 | | 静态分析 | | | 标 识 | | YL\_SA\_JTFX\_001 | |
| 追踪关系 | | 软件测试依据：测评大纲  测试需求分析：（6.2.2.1）静态分析  测试需求标识：XQ\_SA\_JTFX | | | | | | |
| 测试用例综述 | | 依据委托方的要求进行静态分析，验证软件质量度量和编码规则是否满足军标要求 | | | | | | |
| 用例初始化 | | 已获取全部被测件源代码程序，静态分析工具准备齐备 | | | | | | |
| 前提和约束 | | 提交的代码出自委托方受控库，是委托方正式签署外发的 | | | | | | |
| 测试步骤 | | | | | | | | |
| 序号 | 输入及操作 | | 期望结果与评估标准 | | | 实测结果 | | 通过  与否 |
| 1 | 使用LDRA TestBed软件和Klocwork软件工具对被测软件全部源程序进行静态分析，并配合人工以及检查单进行分析 | | 静态审查单全部通过，且源代码满足编码规则和质量度量要求 | | | 静态度量结果符合国军标要求，静态分析审查单全部通过 | | 未执行 |
| 执行状态 | | 未执行 | | 测试时间 | | 2024-05-23 | | |
| 测试人员 | | 陈俊亦 | | 监测人员 | | 陈俊亦 | | |
| 问题单标识 | |  | | | | | | |
| 备注 | | / | | | | | | |

## 代码审查

### 代码审查

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| 测试用例名称 | | 代码审查 | | | 标 识 | | YL\_CR\_DMSC\_001 | |
| 追踪关系 | | 软件测试依据：测评大纲  测试需求分析：（6.2.3.1）代码审查  测试需求标识：XQ\_CR\_DMSC | | | | | | |
| 测试用例综述 | | 通过人工审查及借助工具辅助分析的方式开展代码审查，审查代码编程准则的符合性、代码流程实现的正确性、代码结构的合理性以及代码实现需求的正确性；人工审查中发现的问题，审查人员应及时记录 | | | | | | |
| 用例初始化 | | 代码已提交 | | | | | | |
| 前提和约束 | | 提交的代码出自委托方受控库，是委托方正式签署外发的 | | | | | | |
| 测试步骤 | | | | | | | | |
| 序号 | 输入及操作 | | 期望结果与评估标准 | | | 实测结果 | | 通过  与否 |
| 1 | 通过人工审查及借助工具辅助分析的方式开展代码审查，审查代码编程准则的符合性、代码流程实现的正确性、代码结构的合理性以及代码实现需求的正确性；人工审查中发现的问题，审查人员应及时记录 | | 代码设计正确，满足审查单要求，无不符合项 | | | 代码设计正确，满足审查单要求，无不符合项 | | 未执行 |
| 执行状态 | | 未执行 | | 测试时间 | | 2024-05-30 | | |
| 测试人员 | | 陈俊亦 | | 监测人员 | | 陈俊亦 | | |
| 问题单标识 | |  | | | | | | |
| 备注 | | / | | | | | | |

## 功能测试

### 连续的功能测试

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| 测试用例名称 | | 真实的用例 | | | 标 识 | | YL\_FT\_FFFF\_001 | |
| 追踪关系 | | 软件测试依据：测评大纲  测试需求分析：（6.2.4.1）连续的功能测试  测试需求标识：XQ\_FT\_FFFF | | | | | | |
| 测试用例综述 | | 123123 | | | | | | |
| 用例初始化 | | 软件正常启动，正常登录进软件 | | | | | | |
| 前提和约束 | | 软件正常启动，各界面显示工作正常 | | | | | | |
| 测试步骤 | | | | | | | | |
| 序号 | 输入及操作 | | 期望结果与评估标准 | | | 实测结果 | | 通过  与否 |
| 1 | 图片如下:    表格如下:   |  |  |  |  | | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | | 星表参数 | 已知星表 | 星表注入 | 0xbc100000 | | 星表参数 | 临时星表 | 星表注入 | 0xbc160000 | | 星表参数 | 超亮点源 | 星表注入 | 0xbc1c0000 | | 主控软件目标文件 | 主控软件目标文件 | 软件代码重构 | 0xbc000000 | | FPGA代码目标文件 | FPGA代码目标文件 | FPGA重构 | 存储在FPGA的NORFLASH芯片上 | | | 预期可以 | | | 图片如下:    表格如下:   |  |  |  |  | | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | | 星表参数 | 已知星表 | 星表注入 | 0xbc100000 | | 星表参数 | 临时星表 | 星表注入 | 0xbc160000 | | 星表参数 | 超亮点源 | 星表注入 | 0xbc1c0000 | | 主控软件目标文件 | 主控软件目标文件 | 软件代码重构 | 0xbc000000 | | FPGA代码目标文件 | FPGA代码目标文件 | FPGA重构 | 存储在FPGA的NORFLASH芯片上 | | | 通过 |
| 执行状态 | | 已执行 | | 测试时间 | | 2024-06-06 | | |
| 测试人员 | | 陈俊亦 | | 监测人员 | | 陈俊亦 | | |
| 问题单标识 | |  | | | | | | |
| 备注 | | / | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| 测试用例名称 | | 用例2号 | | | 标 识 | | YL\_FT\_FFFF\_002 | |
| 追踪关系 | | 软件测试依据：测评大纲  测试需求分析：（6.2.4.1）连续的功能测试  测试需求标识：XQ\_FT\_FFFF | | | | | | |
| 测试用例综述 | | 123123 | | | | | | |
| 用例初始化 | | 软件正常启动，正常登录进软件 | | | | | | |
| 前提和约束 | | 软件正常启动，各界面显示工作正常 | | | | | | |
| 测试步骤 | | | | | | | | |
| 序号 | 输入及操作 | | 期望结果与评估标准 | | | 实测结果 | | 通过  与否 |
| 1 | 123 | | 123 | | | 312 | | 通过 |
| 执行状态 | | 已执行 | | 测试时间 | | 2024-06-06 | | |
| 测试人员 | | 陈俊亦 | | 监测人员 | | 陈俊亦 | | |
| 问题单标识 | |  | | | | | | |
| 备注 | | / | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| 测试用例名称 | | 用例3号 | | | 标 识 | | YL\_FT\_FFFF\_003 | |
| 追踪关系 | | 软件测试依据：测评大纲  测试需求分析：（6.2.4.1）连续的功能测试  测试需求标识：XQ\_FT\_FFFF | | | | | | |
| 测试用例综述 | | 123123 | | | | | | |
| 用例初始化 | | 软件正常启动，正常登录进软件 | | | | | | |
| 前提和约束 | | 软件正常启动，各界面显示工作正常 | | | | | | |
| 测试步骤 | | | | | | | | |
| 序号 | 输入及操作 | | 期望结果与评估标准 | | | 实测结果 | | 通过  与否 |
| 1 | 123 | | 123 | | | 312 | | 未通过 |
| 执行状态 | | 未执行 | | 测试时间 | | 2024-06-06 | | |
| 测试人员 | | 陈俊亦 | | 监测人员 | | 陈俊亦 | | |
| 问题单标识 | |  | | | | | | |
| 备注 | | / | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| 测试用例名称 | | 用例4号 | | | 标 识 | | YL\_FT\_FFFF\_004 | |
| 追踪关系 | | 软件测试依据：测评大纲  测试需求分析：（6.2.4.1）连续的功能测试  测试需求标识：XQ\_FT\_FFFF | | | | | | |
| 测试用例综述 | | 123123 | | | | | | |
| 用例初始化 | | 软件正常启动，正常登录进软件 | | | | | | |
| 前提和约束 | | 软件正常启动，各界面显示工作正常 | | | | | | |
| 测试步骤 | | | | | | | | |
| 序号 | 输入及操作 | | 期望结果与评估标准 | | | 实测结果 | | 通过  与否 |
| 1 | 123 | | 123 | | | 312 | | 未通过 |
| 执行状态 | | 已执行 | | 测试时间 | | 2024-05-25 | | |
| 测试人员 | | 陈俊亦 | | 监测人员 | | 陈俊亦 | | |
| 问题单标识 | | PT\_R4444\_1 | | | | | | |
| 备注 | | / | | | | | | |

# 文档审查单

## 软件文档齐套性检查单

软件文档齐套性检查单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 审查内容 | 期望结果 | 审查结果 | 备注 |
| 1 | 是否具有依据文件（研制总要求等） | Y |  |  |
| 2 | 是否具有开发计划（包括质量保证计划和配置管理计划） | Y |  |  |
| 3 | 是否具有需求规格说明 | Y |  |  |
| 4 | 是否具有设计说明 | Y |  |  |
| 5 | 是否具有（包括概要设计和详细设计） | Y |  |  |
| 6 | 是否具有单元测试计划 | Y |  |  |
| 7 | 是否具有单元测试说明 | Y |  |  |
| 8 | 是否具有单元测试报告 | Y |  |  |
| 9 | 是否具有配置项测试计划 | Y |  |  |
| 10 | 是否具有配置项测试说明 | Y |  |  |
| 11 | 是否具有回归测试计划 | Y |  |  |
| 12 | 是否具有回归测试说明 | Y |  |  |
| 13 | 是否具有回归测试报告 | Y |  |  |
| 14 | 是否具有用户手册 | Y |  |  |
| 15 | 是否具有研制总结报告 | Y |  |  |
| 16 | 是否具有产品规格说明 | Y |  |  |
| 17 | 是否具有版本说明 | Y |  |  |
| 18 | 是否具有评审报告 | Y |  |  |
| 审查结果说明：Y—是；N—否；NA—不适用 | | | |  |

## 需求类文档审查单

需求类文档审查单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 文档名称 | | 需求规格说明-回归 | | | | |
| 序号 | 审查项 | 审查内容 | 审查结果（填√） | | | 备注 |
| 通过 | 未通过 | 不适用 |
| 1 | 完整性 | 标识描述本文档所适用系统和软件的完整标识，包括其标识号、名称、缩略名、版本号和发布号 | Y |  |  |  |
| 2 | 系统概述概述本文档适用的系统和软件的用途，描述系统与软件的一般特性 | Y |  |  |  |
| 3 | 文档概述概述了本文档的用途和内容，并描述与它的使用有关的保密性方面的要求 | Y |  |  |  |
| 4 | 引用文档应列出文档的编号、标题、编写单位、修订版及日期 | Y |  |  |  |
| 5 | CSCI能力需求标识出所要求的每一个CICI能力，并列出与该能力有关的需求，包括所需的CSCI行为 | Y |  |  |  |
| 6 | CSIC的外部接口需求应使用唯一的标识符标识CSCI每一个外部接口，并描述为实现该接口提出的该CSCI的需求。 | Y |  |  |  |
| 7 | CSCI内部接口需求应指明施加于CSCI内部接口的需求 | Y |  |  |  |
| 8 | 保密性需求应指明与维护保密性有关的CSCI需求 | Y |  |  |  |
| 9 | 安全性需求应指明关于防止或尽可能降低对人员、财产和物理管径产生以外危险的CSCI安全性需求 | Y |  |  |  |
| 10 | 环境适应性需求应指明CSCI的运行环境需求 | Y |  |  |  |
| 11 | 计算机资源需求应指明CSCI必须使用的计算机硬件的需求、计算机硬件资源的使用需求、必须使用或必须被纳入本CSCI的计算机软件需求、必须使用的计算机通信方面的需求 | Y |  |  |  |
| 12 | 设计和实现约束应指明约束CSCI的设计和实现的需求 | Y |  |  |  |
| 13 | 人员相关需求描述与使用或保障本CSCI的人员有关的CSCI需求，包括人员的数量、职能等级、工作周期、必须的培训以及其他信息 | Y |  |  |  |
| 14 | 训练相关需求应指明与训练相关的CSCI需求 | Y |  |  |  |
| 15 | 软件保障需求应指明与软件保障考虑有关的CSCI需求 | Y |  |  |  |
| 16 | 包装需求应指明为了交付而对CSCI进行包装、标记和处理的需求 | Y |  |  |  |
| 17 | 需求的优先顺序和关键性应指明本规格说明中各需求的优先次序、关键性或表示其相对重要性的权重 | Y |  |  |  |
| 18 | 合格性规定应定义一组合格性检验方法，针对每个需求制定确定需求得到满足所使用的方法 | Y |  |  |  |
| 19 | 需求可追踪性应描述从本规格说明中的每一个CSCI需求所涉及的系统/子系统需求的可追踪性，以及从分配给本CSCI的每一个系统/子系统需求，到所涉及的CSCI需求的可追踪性 | Y |  |  |  |
| 20 | 准确性 | 对各项需求项目的描述准确无歧义 | Y |  |  |  |
| 21 | 列举的各项需求均具备可测试性 | Y |  |  |  |
| 22 | 在已知的条件限制下，所有的需求都应为可实现的需求 | Y |  |  |  |
| 23 | 一致性 | 各项需求之间无相互矛盾或冲突 | Y |  |  |  |
| 24 | 需求规格说明与其他软件文档之间无相互矛盾或冲突 | Y |  |  |  |

# 代码审查单

C语言代码审查单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 审查内容 | | 审查结果（填√） | | | 备注 |
| 通过 | 未通过 | 不适用 |
|  | 存储类 | 未使用内存的内容是否影响系统安全  其处理是否得当 | √ |  |  |  |
|  | 动态申请内存是否判断成功后使用 | √ |  |  |  |
|  | 动态分配的内存是否及时释放 | √ |  |  |  |
|  | 防止内存操作越界 | √ |  |  |  |
|  | 声明定义 | 变量是否正确定义 | √ |  |  |  |
|  | 不同标识符的书写是否清晰可辨 | √ |  |  |  |
|  | 数组声明时，其大小是否定义好 | √ |  |  |  |
|  | 共用体、结构体、枚举的所有成员变量是否进行了初始化 | √ |  |  |  |
|  | 指针使用 | 是否存在指针越界（指针指向的地址不超过了有效地址范围） | √ |  |  |  |
|  | 不应将局部变量的地址赋值给外部指针 | √ |  |  |  |
|  | 是否对参数指针进行赋值 | √ |  |  |  |
|  | 参数变量的传递是否使用了无类型指针 | √ |  |  |  |
|  | 是否使用了野指针 | √ |  |  |  |
|  | 类型转换 | 数据类型转换是否正确 | √ |  |  |  |
|  | 隐式的整型转换是否改变了基础类型的符号 | √ |  |  |  |
|  | 整型或浮点的隐式转换是否缩小了基础类型的大小 | √ |  |  |  |
|  | 指针变量赋予非指针变量时是否进行了强制转换 | √ |  |  |  |
|  | 是否使用无实质作用的类型转换 | √ |  |  |  |
|  | 比较判断 | 是否进行了浮点相等比较（不能进行浮点相等比较，也不能与0值进行相等比较，包括不等于的比较，只能进行大于小于的比较） | √ |  |  |  |
|  | 是否对无符号数进行大于或等于零或小于零的比较 | √ |  |  |  |
|  | 变量有效性判断时范围是否与文档一致 | √ |  |  |  |
|  | 是否使用指针进行大于或小于的逻辑比较 | √ |  |  |  |
|  | 数学运算 | 选择的运算符合理吗？运算符的优先结合次序正确吗？ | √ |  |  |  |
|  | 应对运算中可能存在的数据溢出（如除数为零、大数相加减乘除、数据连续加、数据连续减、根号下为负数等）情况进行处理 | √ |  |  |  |
|  | 是否将bool类型用做内置运算符的操作数？（除了&&、||、!、==、!=、&和条件运算符） | √ |  |  |  |
|  | 是否将enum类型用作内置运算符的操作数？（除了[]、=、==、!=、&、<、<-、>、->运算符） | √ |  |  |  |
|  | 运算符!、&&或||的每个操作数是否都是bool类型？ | √ |  |  |  |
|  | 对变量进行移位运算是否超出变量长度 | √ |  |  |  |
|  | 是否将越界整数赋给整型变量 | √ |  |  |  |
|  | 数组下标是否是大于等于零的整型数 | √ |  |  |  |
|  | 非枚举类型变量是否使用了枚举类型的值 | √ |  |  |  |
|  | 是否存在一个很大的数与一个很小的数进行+，-运算 | √ |  |  |  |
|  | 是否给无符号类型变量赋负值 | √ |  |  |  |
|  | 数据精度是否满足运算要求 | √ |  |  |  |
|  | 需要时，应对运算进行正确的限幅（输出的结果是否有上下限要求） | √ |  |  |  |
|  | 变量的定义类型和使用类型是否一致 | √ |  |  |  |
|  | 函数使用 | 函数是否返回了定义在函数内部的局部变量 | √ |  |  |  |
|  | 有返回值函数的返回语句是否带有返回值 | √ |  |  |  |
|  | 全部所需的参数是否已传送给每一个被调用的模块（参数传的全不全） | √ |  |  |  |
|  | 被传送的参数值是否正确的设置 | √ |  |  |  |
|  | 函数定义与声明一致 | √ |  |  |  |
|  | 函数的错误返回值都必须进行处理 | √ |  |  |  |
|  | 实参和形参所使用的变量类型和个数是否一致 | √ |  |  |  |
|  | 程序语言的使用 | 文件、数据库和注册表等打开后，在对其进行操作之后是否进行了关闭 | √ |  |  |  |
|  | 用于表示字符串的数组是否以‘\0’结束 | √ |  |  |  |
|  | 函数宏参数的定义是否使用括号 | √ |  |  |  |
|  | for循环中控制变量是否为局部变量 | √ |  |  |  |
|  | for循环中控制变量是否为整型变量 | √ |  |  |  |
|  | for循环中是否对循环控制变量进行修改 | √ |  |  |  |
|  | 寄存器使用 | 寄存器使用是否按照芯片手册的操作要求 | √ |  |  |  |
|  | 宏扩展或子程序调用是否使用了已使用着的寄存器而未保存数据 | √ |  |  |  |
|  | 默认使用的寄存器的值是否正确（注意每个寄存器的初始默认值） | √ |  |  |  |
|  | 可维护性 | 嵌套的IF、ELSE、WHILE等语句是否已正确地缩进、是否少括号，是否忘写了括号 | √ |  |  |  |
|  | 注释准确并且有意义 | √ |  |  |  |
|  | 注释语句应不少于20 | √ |  |  |  |
|  | 是否存在嵌套注释 | √ |  |  |  |
|  | 标号和子程序名符合代码的逻辑含义吗 | √ |  |  |  |
|  | 避免使用不易理解的数字，用有意义的标识来替代，涉及物理状态或者含有物理意义的常量，不应直接使用数字，必须用有意义的枚举或宏来代替 | √ |  |  |  |
|  | 注意运算符的优先级，并用括号明确表达式的操作顺序，避免使用默认优先级 | √ |  |  |  |
|  | 逻辑 | 代码是否做了设计规定的内容 | √ |  |  |  |
|  | 全部设计是否均已实现（和详细设计的一致性） | √ |  |  |  |
|  | 每个循环是否执行正确的次数 | √ |  |  |  |
|  | 是否存在死循环的风险（例如当等待某变量为一特定值才退出时存在风险；循环语句中必须有超时保护） | √ |  |  |  |
|  | 不能使用未赋值的变量 | √ |  |  |  |
|  | Switch语句必须有default分支 | √ |  |  |  |
|  | 必须处理程序所能遇到的各种出错情况 | √ |  |  |  |
|  | 软件多余物 | 是否有不可能执行到的代码 | √ |  |  |  |
| 是否有即使不执行也不影响程序功能的指令 | √ |  |  |  |
| 是否有未引用的变量、标号和常量 | √ |  |  |  |
|  | 有无未引用的宏 | √ |  |  |  |
|  | 其他 | 应根据看门狗触发周期正确设置其清除周期（不能超过狗咬时间） | √ |  |  |  |
|  | 应根据需要监视的流程，正确设置清除看门狗的位置（在该清除的地方要清除，比如初始化阶段、各个进程等；在不该清除的地方不能清除，比如在需求中要求等待狗咬的位置等） | √ |  |  |  |
|  | 对于全局变量的读写要合理 | √ |  |  |  |
|  | 代码审查覆盖项 |  |  |  |  |  |

代码审查项（C语言）

|  |  |  |
| --- | --- | --- |
| 序号 | 审查项 | 审查内容 |
|  | Use of Uninitialized Data  ---使用未初始化的数据 | Uninitialized Variable – possible  ---变量可能未初始化 |
|  | Uninitialized Variable  ---变量未初始化 |
|  | Uninitialized Array – possible  ---队列元素可能未初始化 |
|  | Uninitialized Array  ---队列元素未初始化 |
|  | Partialy Uninitialized Array  ---队列元素部分初始化 |
|  | Uninitialized Variable in Constructor – possible  ---构造函数中变量可能未初始化 |
|  | Uninitialized Variable in Constructor  ---构造函数中变量未初始化 |
|  | Uninitialized Heap Use – possible  --使用的堆栈未初始化 |
|  | Concurrency  --并发处理 | Missing unlock for variable  ---变量缺少解锁 |
|  | Deadlock  ---程序死锁 |
|  | Memory Leaks  ---内存泄露 | Inconsistent Freeing of Memory  ---释放内存不一致 |
|  | Memory Leak – possible  ---内存可能泄漏 |
|  | Memory Leak  ---内存泄漏 |
|  | Memory Leak - in destructor  ---在析构函数中，内存泄露 |
|  | Memory Leak - possible in destructor  --在析构函数中，可能内存泄露 |
|  | Memory Leak - in operator=  --赋值操作符=，导致内存泄露 |
|  |
|  | Attempt to Use Memory after Free  ---使用已经释放  的内存 | Use of free memory (access) – possible  ---使用(读写)可能已经释放的内存 |
|  | Use of Freed Memory by Pointer  ---通过指针调用已经被释放的内存 |
|  | Use of free memory (double free) – possible  ---使用可能已经被释放的内存(重复释放) |
|  | Freeing Freed Memory  ---释放已经被释放的内存 |
|  | Use of freed memory (return) – possible  ---使用(返回)可能已经释放的内存 |
|  | Use of Freed Memory on Return  ---使用已经释放的返回内存 |
|  | Use of free memory – possible  ---使用可能已经被释放的内存 |
|  | Use of Freed Memory  ---使用已经被释放的内存 |
|  | Use of free memory (double free) - no copy constructor  ---释放已经被释放的内存(重复释放)---拷贝  构造函数没有定义 |
|  | Use of free memory (double free) – no operator=  ---释放已经被释放的内存(重复释放)---赋值操作符=没有定义 |
|  | Function returns address of local variable  ---函数返回局部变量(指针)地址 |
|  | Function returns address of local variable  ---函数返回局部变量(内存)地址 |
|  | Function returns address of local variable  ---函数返回局部变量(全局变量)地址 |
|  | Improper Memory Deallocation  ---不合理的内存释放 | Freeing Mismatched Memory - in destructor  ---在析构函数中，释放不匹配的内存 |
|  | Freeing Non-Heap Memory – possible  ---可能释放非堆栈内存 |
|  | Freeing Non-Heap Memory  --释放非堆栈内存 |
|  | Freeing Unallocated Memory – possible  ---释放可能未开辟的内存 |
|  | Freeing Unallocated Memory  ---释放未开辟的内存 |
|  | Freeing Mismatched Memory – possible  ---释放可能不匹配的内存 |
|  | Freeing Mismatched Memory  ---释放不匹配的内存 |
|  | Null Pointer Dereference  ---空指针引用取消 | Null pointer may be dereferenced  ---对空指针引用可能会被取消 |
|  | Null pointer will be dereferenced  ---对空指针引用将取消 |
|  | Null pointer may be passed to function that may dereference it  ---空指针可能会传递给函数，可能会取消对它的引用 |
|  | Null pointer will be passed to function that may dereference it  ---空指针将传递给函数，可能会取消对它的引用 |
|  | Result of function that can return NULL may be dereferenced  ---函数返回NULL 结果，可能会被取消引用 |
|  | Result of function that may return NULL will be dereferenced  ---函数可能返回NULL 结果，将被取消引用 |
|  | Pointer may be dereferenced after it was positively checked for NULL  ---经过对指针是否为空的检查后，对指针的引用可能会被取消 |
|  | Pointer will be dereferenced after it was positively checked for NULL  ---经过对指针是否为空的检查后，对指针的引用将会被取消 |
|  | NULL is dereferenced  ---对NULL 的引用被取消 |
|  | Result of function that may return NULL may be passed to another function that may dereference it  ---函数可能返回NULL 结果，并可能被传递到另一个函数，这可能会取消对它的引用 |
|  | Result of function that may return NULL will be passed to another function that may dereference it  ---函数可能返回NULL 结果，并将被传递到另一个函数，可能会取消对它的引用 |
|  | Pointer may be passed to function that can dereference it after it was positively checked for NULL  ---经过对指针是否为空的检查后，指针可能会被传递给函数，对指针的引用将被取消 |
|  | Pointer will be passed to function that may dereference it after it was positively checked for NULL  ---经过对指针是否为空的检查后，指针将被传递给函数，对指针的引用可能会取消 |
|  | NULL is passed to function that can dereference it  ---将NULL 传递给函数，这种引用将被取消 |
|  | Suspicious dereference of pointer before NULL check  ---在检查指针是否为空之前，指针的引用不可信 |
|  | Suspicious dereference of pointer in function call before NULL check  ---在检查指针是否为空之前，函数调用中指针的引用不可信 |
|  | Inappropriate Iterator Usage  ---不兼容的迭代使用 | Use of invalid iterator  ---使用无效的迭代器 |
|  | Dereference of 'end' iterator  ---取消迭代器调用end()成员函数 |
|  | Use of iterator with inappropriate container object  ---使用迭代器的不适当容器对象 |
|  | Unreachable Code  ---不可达代码 | Unreachable code  ---不可达代码 |
|  | Mismatched Return Types  ---不匹配的返回类型 | Operator= return a reference to \*this  赋值操作符=，返回内容引用了\*this |
|  | Non-void function does not return value  ---非Void 函数没有返回值 |
|  | Non-void function implicitly returning int does not return value  ---非Void 函数明确返回int 值，实际没有返回值 |
|  | Void function returns value  ---Void 函数有返回值 |
|  | Non-void function returns void value  ---非Void 函数返回void 结果 |
|  | Implicitly int function returns void value  ---明确的int 函数返回void 结果 |
|  | Buffer Overflow  ---缓冲区溢出 | Buffer Overflow - Array Index Out of Bounds  ---队列索引出超出边界，导致缓冲区溢出 |
|  | Buffer Overflow - Local Array Index Out of Bounds  ---局部队列索引超出边界，导致缓冲区溢出 |
|  | Buffer Overflow from Unvalidated Input  ---无效的输入导致缓冲区溢出 |
|  | Unvalidated User Input Causing Buffer Overflow - Non-Null Terminated String  ---无效的用户输入(非空终断字符串)导致缓冲区溢出 |
|  | Buffer Overflow - Non-null Terminated String  ---非空终断字符串导致缓冲区溢出 |
|  | Buffer Overflow - Non-null Terminated String  ---非空终断字符串可能导致缓冲区溢出 |
|  | Buffer Overflow - Array Index Out of Bounds  ---队列索引超出边界导致缓冲区溢出 |
|  | Mapping function failed  ---映射函数失败 |
|  | Buffer overflow in mapping character function  ---字符映射函数导致缓冲区溢出，使用MultiBytetoWideChar 和WidecharToMultiByte函数容易导致缓冲区溢出 |
|  | Buffer Overflow in Bound String Copy  --使用邦定的字符串拷贝，导致缓冲区溢出 |
|  | Buffer Overflow in Bound sprintf  ---使用邦定的函数sprintf，导致缓冲区溢出 |
|  | Usage of 'gets'  --使用gets 函数，该函数不检查内存边界，容易导致内存溢出 |
|  | Buffer Overflow in Unbound String Copy  ---使用未邦定的字符串拷贝，导致缓冲区溢出 |
|  | Buffer Overflow in Unbound sprintf  ---使用未邦定的函数sprintf，导致缓冲区溢出 |
|  | Buffer Overflow - Array Index may be out of Bounds  --局部队列索引可能超出边界，导致缓冲区溢出 |
|  | Input format specifier error  ---输入格式符错误 |
|  | Print functions format  ---打印函数格式问题 | Incompatible type of a print function parameter  ---打印函数中使用不匹配的参数类型 |
|  | Too few arguments in a print function call  ---调用打印函数的参数太少 |
|  | Unknown format specifier in a print function call  ---在打印函数中使用未知格式符 |
|  | Scan functions format  --扫描函数格式问题 | Incompatible type of a scan function parameter ---扫描函数参数使用不匹配类型 |
|  | Too few arguments in a scan function call  ---调用扫描函数的参数太少 |
|  | Unknown format specifier in a scan function call  ---调用扫描函数中，使用未知格式的修饰符 |
|  | Unvalidated User Input  ---不可用的用户输入 | Use of Unvalidated Integer in Memory Allocation  ---内存开辟过程中使用无效的整数 |
|  | Use of Unvalidated Data in a Format String.  ---在固定格式的字符串中，使用无效数据 |
|  | Use of Unvalidated Integer as Array Index  ---使用无效的整数作为队列索引 |
|  | Use of Unvalidated Integer in Loop Condition  ---在迭代条件中使用无效的整数 |
|  | Command Injection  ---命令行注入 |
|  | Loading File without Use of Absolute Path  ---加载文件没有使用绝对路径 |
|  | Use of Unvalidated String Data  ---使用无效字符串数据 |
|  | Weak Encryption  ---弱加密 | Use of Poor Encryption  ---使用弱加密 |
|  | Use of Poor Encryption  ---使用弱加密 |
|  | Insecure (Constant) Temporary File Name in Call to CreateFile  ---调用函数CreateFile 导致不安全的固定文件名 |
|  | Modification of Temporary File Name before Call to CreateFile  ---调用函数CreateFile 将改变临时文件名 |
|  | Missing Temporary File Name in Call to CreateFile  ---调用函数CreateFile 导致丢失临时文件名 |
|  | Insecure Temporary File Name in Call to CreateFile  ---调用函数CreateFile 导致不安全的临时文件名 |
|  | Missing Secure Temporary File Names in Call to CreateFile  ---调用函数CreateFile 将丢失安全的临时文件名 |
|  | Registry Manipulation  ---注册表篡改 | HKEY\_LOCAL\_MACHINE Used as 'hkey' Parameter for Registry Manipulation Function  --注册时使用宏HKEY\_LOCAL\_MACHINE作为参数，将导致非管理员获得权限 |
|  | DNS Spoofing  ---DNS 欺骗 | Use of INADDR\_ANY in sin\_addr.s\_addr field of struct sockaddr\_in Structure Used for Call to bind Function  ---调用函数bind 时，结构体sockaddr\_in 中sin\_addr.s\_addr 被设置为INADDR\_ANY |
|  | Suspicious Code Practices  ---不可信的代码操作 | Command Injection into Shell Execution  ---通过命令行注入Shell 再执行，无法限制输入命令内容和长度 |
|  | Format String Vulnerability  ---使用固定格式的字符串，但是没有定义字符串长度可能导致内存溢出 |
|  | Use of Privilege Elevation  ---使用提升权限 |
|  | Use of Dangerous Process Creation  ---危险进程创建，容易通过环境变量运行任意命令 |
|  | Missed conversion from network to host byte order  ---接收时，没有将网络字节顺序转换为主机字节顺序 |
|  | Missed conversion from host to network byte order  ---发送时，没有将主机字节顺序转换为网络字节顺序 |
|  | Missed conversion from network to host byte order  --读取时，没有将网络字节顺序转换为主机字节顺序 |
|  | Missed conversion from host to network byte order  ---写入时，没有将主机字节顺序转换为网络字节顺序 |
|  | Suspiciously placed semicolon  ---可疑的分号放置 |
|  | Assignment in condition (call)  ---调用的条件语句中有赋值操作 |
|  | Assignment in condition  ---在条件语句中赋值操作 |
|  |

# 静态分析结果记录

质量度量要求

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 软件名称 | |  | | 程序行总数（Loc） |  |
| 软件版本 | |  | | 代码行总数（Loc） |  |
| 文件总数(个) | |  | | 注释行总数（Loc） |  |
| 模块总数（个） | |  | | 空白行总数（Loc） |  |
| 软件总注释率 | |  | | 需求分析功能点数 |  |
| 度量元 | 描述 | | 度量指标 | 不满足指标的函数单元个数 | 占文件/函数总个数百分比 |
| 文本度量 | 软件单元（每个函数）的语句数 | | ≤200行 |  |  |
| 扇出数 | 函数调用下层函数个数 | | <7 |  |  |
| 模块最大圈复杂度 | 各模块圈复杂度的最大值 | | ≤80 |  |  |
| 圈复杂≥10的比例 | 模块中圈复杂度≥10的个数占总模块数的个数和百分比； | | ≤10 |  |  |
| 圈复杂≥20的比例 | 模块中圈复杂度≥20的个数占总模块数的个数和百分比； | | ≤20 |  |  |

编码规则不满足情况

|  |  |  |
| --- | --- | --- |
| 规则名称 | 规则描述 | 问题来源 |
| R-1-1-1 | 禁止通过宏定义改变关键字和基本类型含义 |  |
| R-1-1-2 | 禁止将其他标识宏定义为关键字和基本类型 |  |
| R-1-1-3 | 用typedef自定义的类型禁止被重新定义 |  |
| R-1-1-4 | 禁止重新定义C或C++的关键字 |  |
| R-1-1-5 | 禁止#define被重复定义 |  |
| R-1-1-6 | 函数中的#define和#undef必须配对使用 |  |
| R-1-1-7 | 以函数形式定义的宏，参数和结果必须用括号括起来 |  |
| R-1-1-8 | 结构、联合、枚举的定义中必须定义标识名 |  |
| R-1-1-9 | 结构体定义中禁止含有无名结构体 |  |
| R-1-1-10 | 位定义的有符号整型变量位长必须大于1 |  |
| R-1-1-11 | 位定义的整数型变量必须明确定义是有符号还是无符号的 |  |
| R-1-1-12 | 位定义的变量必须是同长度的类型且定义位禁止跨越类型的长度 |  |
| R-1-1-13 | 函数声明中必须对参数类型进行声明，并带有变量名 |  |
| R-1-1-14 | 函数声明必须与函数原型一致 |  |
| R-1-1-15 | 函数中的参数必须使用类型声明 |  |
| R-1-1-16 | 外部声明的变量，类型必须与定义一致 |  |
| R-1-1-17 | 禁止在函数体内使用外部声明 |  |
| R-1-1-18 | 数组定义禁止没有显示的边界限定 |  |
| R-1-1-19 | 禁止使用extern声明对变量初始化 |  |
| R-1-1-20 | 用于数值计算的字符型变量必须明确定义是有符号还是无符号 |  |
| R-1-1-21 | 禁止在#include语句中使用绝对路径 |  |
| R-1-1-22 | 禁止头文件重复包含 |  |
| R-1-1-23 | 函数参数表为空时，必须使用void明确说明 |  |
| R-1-2-1 | 循环体必须用大括号括起来 |  |
| R-1-2-2 | If、else if、else必须用大括号括起来 |  |
| R-1-2-3 | 禁止在头文件前有可执行代码 |  |
| R-1-2-4 | 引起二义性理解的逻辑表达式，必须使用括号显式说明优先级顺序 |  |
| R-1-2-5 | 逻辑判别表达式中的运算项必须要使用括号 |  |
| R-1-2-6 | 禁止嵌套注释 |  |
| R-1-3-1 | 禁止指针的指针超过两级 |  |
| R-1-3-2 | 函数指针的使用必须加以&明确说明 |  |
| R-1-3-3 | 禁止对参数指针进行赋值 |  |
| R-1-3-4 | 禁止将局部变量地址做为函数返回值返回 |  |
| R-1-3-5 | 禁止使用或释放未分配空间或已释放的指针 |  |
| R-1-3-6 | 指针变量被释放后必须置为NULL |  |
| R-1-3-7 | 动态分配的指针变量定义时如未被分配空间必须初始化为NULL |  |
| R-1-3-8 | 动态分配的指针变量第一次使用前必须进行是否为NULL的判别 |  |
| R-1-3-9 | 空指针必须使用NULL，禁止使用整型数0 |  |
| R-1-3-10 | 禁止文件指针在退出时没有关闭文件 |  |
| R-1-4-1 | 在if-else if语句中必须使用else分支 |  |
| R-1-4-2 | 条件判定分支如果为空，必须以单独一行的分号加注释进行明确说明 |  |
| R-1-4-3 | 禁止使用空switch语句 |  |
| R-1-4-4 | 禁止对bool量使用switch语句 |  |
| R-1-4-5 | 禁止switch语句中只包含default语句 |  |
| R-1-4-6 | 除枚举类型列举完全外，switch语句必须要有default |  |
| R-1-4-7 | Switch中的case和default必须以break或return终止，共用case必须加以明确注释 |  |
| R-1-4-8 | Switch语句的所有分支必须具有相同的层次范围 |  |
| R-1-5-1 | 禁止从复合语句外goto到复合语句内，或由下向上goto |  |
| R-1-5-2 | 禁止使用setjmp/longjmp |  |
| R-1-6-1 | 禁止将浮点常数赋值给整型变量 |  |
| R-1-6-2 | 禁止将越界整数赋值给整型变量 |  |
| R-1-6-3 | 禁止在逻辑表达式中使用赋值语句 |  |
| R-1-6-4 | 禁止对逻辑表达式进行位运算 |  |
| R-1-6-5 | 禁止在运算表达式中或函数调用参数中使用++或—操作符 |  |
| R-1-6-6 | 对变量进行移位运算禁止超出变量长度 |  |
| R-1-6-7 | 禁止移位操作中的移位数为负数 |  |
| R-1-6-8 | 数组禁止越界使用 |  |
| R-1-6-9 | 数组下标必须是大于等于零的整型数 |  |
| R-1-6-10 | 禁止对常数值做逻辑非的运算 |  |
| R-1-6-11 | 禁止非枚举类型变量使用枚举类型的值 |  |
| R-1-6-12 | 除法运算中禁止被零除 |  |
| R-1-6-13 | 禁止在sizeof中使用赋值 |  |
| R-1-6-14 | 缓存区读取操作禁止越界 |  |
| R-1-6-15 | 缓存区写入操作禁止越界 |  |
| R-1-6-16 | 禁止使用已被释放了的内存空间 |  |
| R-1-6-17 | 被free的指针必须指向最初malloc、calloc分配的地址 |  |
| R-1-6-18 | 禁止使用gets函数，应使用fgets函数替代 |  |
| R-1-6-19 | 使用字符串赋值、拷贝、追加等函数时，禁止目标字符串存储空间越界 |  |
| R-1-7-1 | 禁止覆盖标准函数库的函数 |  |
| R-1-7-2 | 禁止函数的实参和形参类型不一致 |  |
| R-1-7-3 | 实参和形参的个数必须一致 |  |
| R-1-7-4 | 禁止使用旧形式的函数参数表定义形式 |  |
| R-1-7-5 | 函数声明和函数定义中的参数类型必须一致 |  |
| R-1-7-6 | 函数声明和函数定义中的返回类型必须一致 |  |
| R-1-7-7 | 有返回值的函数必须通过返回语句返回 |  |
| R-1-7-8 | 禁止无返回值函数的返回语句带有返回值 |  |
| R-1-7-9 | 有返回值的函数的返回语句必须带有返回值 |  |
| R-1-7-10 | 函数返回值的类型必须与定义一致 |  |
| R-1-7-11 | 具有返回值的函数，其返回值如果不被使用，调用时应有（void）说明 |  |
| R-1-7-12 | 无返回值的函数，调用时禁止再用（void）重复说明 |  |
| R-1-7-13 | 静态函数必须被使用 |  |
| R-1-7-14 | 禁止同一个表达式中调用多个顺序相关函数 |  |
| R-1-7-15 | 禁止在函数参数表中使用省略号 |  |
| R-1-7-16 | 禁止使用直接或间接自调用函数 |  |
| R-1-8-1 | 禁止不可达语句 |  |
| R-1-8-2 | 禁止不可达分支 |  |
| R-1-8-3 | 禁止使用无效语句 |  |
| R-1-8-4 | 使用八进制数必须明确注释 |  |
| R-1-8-5 | 数字类型后缀必须使用大写字母 |  |
| R-1-9-1 | For循环控制变量必须使用局部变量 |  |
| R-1-9-2 | For循环控制变量必须使用整数型变量 |  |
| R-1-9-3 | 禁止在for循环体内部修改循环控制变量 |  |
| R-1-9-4 | 无限循环必须使用while(1)语句，禁止使用for(;;)等其他形式的语句 |  |
| R-1-10-1 | 浮点数变量赋值给整型变量必须强制转换 |  |
| R-1-10-2 | 长整数变量赋值给短整数变量必须强制转换 |  |
| R-1-10-3 | Double型变量赋给float型变量必须强制转换 |  |
| R-1-10-4 | 指针变量的赋值类型必须与指针变量类型一致 |  |
| R-1-10-5 | 将指针量赋予非指针变量或非指针量赋予指针变量，必须使用强制转换 |  |
| R-1-10-6 | 禁止使用无实质作用的类型转换 |  |
| R-1-11-1 | 变量禁止未赋值就使用 |  |
| R-1-11-2 | 变量初始化禁止隐含依赖于系统的缺省值 |  |
| R-1-11-3 | 结构体初始化的嵌套结构必须与定义一致 |  |
| R-1-11-4 | 枚举元素定义中的初始化必须完整 |  |
| R-1-12-1 | 禁止对逻辑量进行大于或小于的逻辑比较 |  |
| R-1-12-2 | 禁止对指针进行大于或小于的逻辑比较 |  |
| R-1-12-3 | 禁止对浮点数进行是否相等的比较 |  |
| R-1-12-4 | 禁止对无符号数进行大于等于零或小于零的比较 |  |
| R-1-12-5 | 禁止无符号数与有符号数之间的直接比较 |  |
| R-1-13-1 | 禁止局部变量与全局变量同名 |  |
| R-1-13-2 | 禁止函数形参与全局变量同名 |  |
| R-1-13-3 | 禁止变量名与函数名同名 |  |
| R-1-13-4 | 禁止变量名与标识名同名 |  |
| R-1-13-5 | 禁止变量名与枚举元素同名 |  |
| R-1-13-6 | 禁止变量名与typedef自定义的类型名同名 |  |
| R-1-13-7 | 禁止在内部块中重定义已有的变量名 |  |
| R-1-13-8 | 禁止仅依赖大小写区分的变量 |  |
| R-1-13-9 | 禁止仅依赖小写字母“l”与数字“1”区分的变量 |  |
| R-1-13-10 | 禁止仅依赖小写字母“O”与数字“0”区分的变量 |  |
| R-1-13-11 | 禁止单独使用小写字母“l”或大写字母“O”作为变量名 |  |
| R-1-13-12 | 程序外部可改写的变量，必须使用volatile类型说明 |  |
| R-1-13-13 | 禁止在表达式中出现多个同一volatile类型变量的运算 |  |
| R-1-13-14 | 禁止将NULL做为整型数0使用 |  |
| R-1-13-15 | 禁止给无符号类型变量赋负值 |  |
| R-1-13-16 | 用于表示字符串的数组必须以‘\0’结束 |  |
| R-2-1-1 | 含有动态分配成员的类，必须编写拷贝构造函数，并重载赋值操作符 |  |
| R-2-1-2 | 虚拟基类指针转换为派生类指针必须使用dynamic cast转换 |  |
| R-2-1-3 | 菱形层次结构的派生设计，对基类派生必须使用virtual说明 |  |
| R-2-1-4 | 抽象类中的复制操作符重载必须是保护的或私有的 |  |
| R-2-2-1 | 构造函数中禁止使用全局变量 |  |
| R-2-2-2 | 类中必须明确定义缺省构造函数 |  |
| R-2-2-3 | 单参数构造函数必须使用explicit声明 |  |
| R-2-2-4 | 类中所有成员变量必须在构造函数中初始化 |  |
| R-2-2-5 | 派生类构造函数必须在初始化列表中说明直接基类构造函数 |  |
| R-2-3-1 | 具有虚拟成员函数的类，析构函数必须是虚拟的 |  |
| R-2-3-2 | 析构函数中禁止存在不是由自身捕获处理的异常 |  |
| R-2-4-1 | 基类虚拟函数的参数缺省值在派生类重写函数中禁止被改变 |  |
| R-2-4-2 | 派生类对基类虚拟函数重写的声明必须使用virtual显示说明 |  |
| R-2-4-3 | 禁止非纯虚函数被纯虚函数重写 |  |
| R-2-5-1 | 禁止将不相关的指针类型强制转换为对象指针类型 |  |
| R-2-5-2 | 指针或引用的类型转换中禁止移除const或volatile属性 |  |
| R-2-6-1 | 使用new分配的内存空间，用完后必须使用delete释放 |  |
| R-2-6-2 | 必须使用delete[]释放new[]分配的内存空间 |  |
| R-2-6-3 | 被delete的指针必须指向最初new分配的地址 |  |
| R-2-7-1 | 函数中固定长度数组变量的传递必须使用引用方式 |  |
| R-2-7-2 | 定义为const的成员函数禁止返回非const的指针或引用 |  |
| R-2-7-3 | 禁止可导致非资源性对象数据被外部修改的成员函数返回 |  |
| R-2-8-1 | 捕获的顺序必须按由派生类到基类的次序排序 |  |
| R-2-8-2 | 每个指定的抛出必须由与之匹配的捕获 |  |
| R-2-8-3 | 异常抛出的对象必须使用引用方式捕获 |  |
| R-2-8-4 | 缺省捕获必须放在所有指定捕获之后 |  |
| R-2-8-5 | 禁止显式直接抛出NULL |  |
| R-2-9-1 | 模板的声明、定义与实现必须在同一个文件之中 |  |

# 未覆盖情况

表D-1未覆盖语句统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 未覆盖语句 | 未覆盖原因 | 影响分析 | 补充测试情况 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

表D-2未覆盖分支统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 未覆盖分支 | 未覆盖原因 | 影响分析 | 补充测试情况 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

表D-3目标码未覆盖语句统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 未覆盖语句 | 未覆盖原因 | 影响分析 | 补充测试情况 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

表D-4目标码未覆盖分支统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 未覆盖分支 | 未覆盖原因 | 影响分析 | 补充测试情况 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

——文档结束——