**《C语言IDE》软件测试方案设计**

编写 崔世昂 2020 年 04 月 20 日

审核 年 月 日

批准 年 月 日

版本控制

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | AMD | 修订者 | 说明 |
| 1.0 | 2020/4/20 | A | 崔世昂 | 添加软件测试总计划 |
| 1.1 | 2020/4/22 | M | 张海汀 | 添加软件兼容性测试计划 |
| 1.2 | 2020/4/22 | M | 崔世昂 | 整合各项测试分计划 |
| 1.3 | 2020/5/2 | M | 崔世昂 | 添加软件需求规格说明书审查计划 |
| 1.4 | 2020/5/3 | M | 崔世昂 | 添加软件测试进度计划 |

注：（A-添加，M-修改，D-删除）

目录

目录

**[《C语言IDE》软件测试方案设计 1](#_Toc8084)**

**[目录 3](#_Toc21535)**

**[1 概述 6](#_Toc29116)**

[1.1 编写目的 6](#_Toc13017)

[1.2 读者对象 6](#_Toc32410)

[1.3 项目背景 6](#_Toc5948)

[1.4 测试目标 6](#_Toc7701)

[1.5 参考资料 6](#_Toc2563)

**[2 测试配置说明 6](#_Toc11452)**

[2.1 测试手段 6](#_Toc6452)

[2.2 测试数据 7](#_Toc32084)

[2.3 测试策略 7](#_Toc7695)

[2.4. 测试通过准则 8](#_Toc14201)

**[3 软件结构介绍 8](#_Toc3990)**

[3.1 概述 8](#_Toc5276)

[3.2 整体功能模块介绍 8](#_Toc11814)

[3.3 整体功能模块关系图 10](#_Toc13736)

[3.4 系统外部接口功能模块关系图 11](#_Toc30963)

[3.5 系统内部接口功能模块关系图 11](#_Toc24334)

**[4 系统测试用例参考 11](#_Toc9331)**

[4.1 XX系统 11](#_Toc5654)

**[5 附录 12](#_Toc16622)**

**[5.1 附录1 审批记录表 12](#_Toc12688)**

**[角色 12](#_Toc31446)**

**[签名 12](#_Toc16751)**

**[日期 12](#_Toc27283)**

**[备注 12](#_Toc3021)**

**[6 进度计划说明 13](#_Toc22493)**

[6.1 进度计划2020/5/3  13](#_Toc4962)

**[7 相关文档（双击可打开） 13](#_Toc13546)**

[1. 功能测试计划说明书  13](#_Toc31357)

[2. 性能测试计划说明书  13](#_Toc30704)

[3. 可靠性测试计划说明书  13](#_Toc28416)

**[4. 兼容性测试计划说明书 14](#_Toc28107)**

[4.1.兼容性测试计划概述 14](#_Toc28391)

[4.2.兼容性测试环境 14](#_Toc14027)

[4.3.兼容性测试策略 14](#_Toc29285)

[4.4.兼容性测试通过准则 15](#_Toc26355)

[备注：测试记录表 15](#_Toc12196)

**[5. 需求规格说明书审查计划 15](#_Toc19762)**

**[6. 回归测试计划说明书 15](#_Toc3483)**

1 概述

1.1 编写目的

编写本测试方案的目的是为软件开发项目管理者、软件工程师、系统维护工程师、测试工程师提供关于《C语言IDE》系统整体系统功能和性能的测试指导。

1.2 读者对象

本测试方案可能的合法读者对象为软件开发项目管理者、软件工程师、测试组、系统维护工程师。

1.3 项目背景

《C语言IDE》是“量子寄点”小组在本科三年级上学期小学期所做的项目，旨在设计、制作一个面向C语言的IDE，完成C语言编写、编译等操作。

1.4 测试目标

按照测试目的来划分，对《C语言IDE》完成 功能测试、性能测试、可靠性测试、兼容性测试、回归测试 五项测试。

1.5 参考资料

《软件测试方法和技术》（第3版） 清华大学出版社 朱少民 主编

2 测试配置说明

2.1 测试手段

在此参照 各项测试分计划 说明 测试方法 和 工具 ，注明执行测试时，必须同时填写 各项测试分计划指定的测试记录表。

2.2 测试数据

参考 “量子奇点”小组《C语言IDE》需求分析文档。（双击可打开）  


2.3 测试策略

在 各项测试分计划 中说明测试策略，可以如下这样说明：

A)系统测试

系统测试目的是在于验证软件的功能和性能及其他特性是否与用户的要求一致，主要是下列类型的测试：

1）用户界面测试：测试用户界面是否具有导航性、美观性、行业或公司的规范性、是否满足设计中要求的执行功能。

功能测试：测试相应模块功能是否正确实现。

强度测试：测试资源（内存、硬盘）敏感的问题。

容量测试：测试大量数据对系统的影响。

容错测试：测试软件系统克服软件、硬件故障的能力。

安全性测试：测试软件系统对非法侵入的防范能力。

配置测试：测试在不同网络、服务器、工作站的不同软硬件配置条件下，软件系统的质量。

安装测试：确保软件系统在所有可能情况下的安装效果和一旦安装之后必须保证正确运行的质量。

升级测试：确保软件升级后，功能一切正常，用户的习惯设置正常保留。

卸载测试：确保软件能否正常卸载，缓存数据清理完成。

兼容性测试：确保软件在不同的操作系统中能否正常运作，功能是否正常

2.4. 测试通过准则

详情参考 各项测试分计划 测试通过准则。

3 软件结构介绍

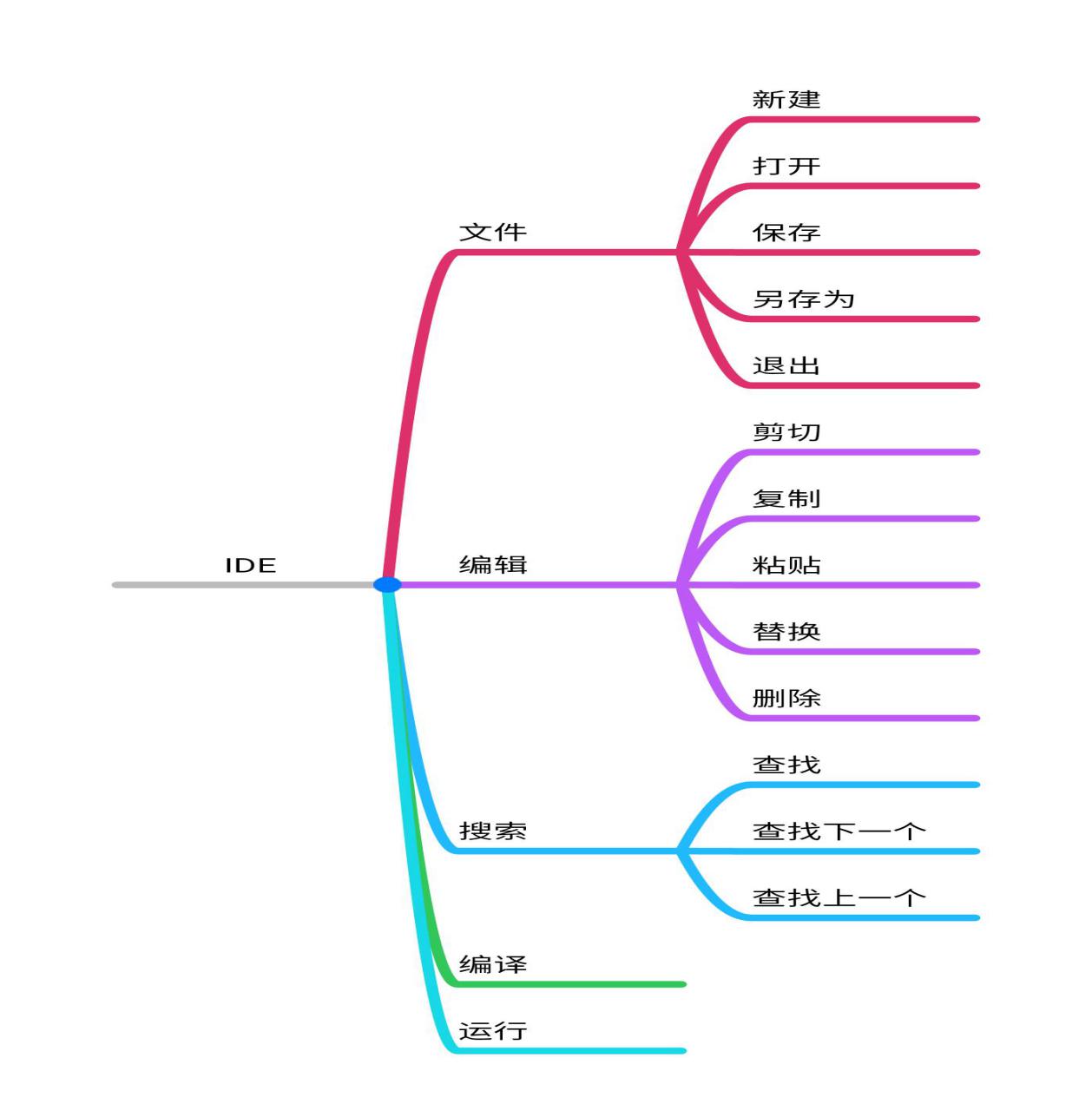
3.1 概述

《C语言IDE》系统主要包含四个子系统，分别是 文件子系统、编辑子系统、搜索子系统、运行子系统 。文件子系统包含 新建 、打开 、保存 、另存为 、退出 模块。编辑子系统包含 剪切 、复制 、粘贴 、替换 、删除 模块。搜索子系统包含 查找 、查找下一个 、查找上一个 模块。运行子系统包括 编译 、运行 、编译运行 模块。

3.2 整体功能模块介绍

在此介绍《C语言IDE》系统的功能模块如下表所示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 系统名称 | 所属系统 | 软件项标识 | 备注 |
| 1 | C语言IDE |  |  |  |
| 2 | 文件子系统 | C语言IDE |  |  |
| 3 | 新建文件 | 文件子系统 |  | 新建一个.c文件 |
| 4 | 保存文件 | 文件子系统 |  | 先判断是否保存，若未保存，则弹出选择保存路径的窗口；若已保存，则提示用户“已保存！” |
| 5 | 另存为文件 | 文件子系统 |  | 选择一个除开默认路径的位置保存 |
| 6 | 打开文件 | 文件子系统 |  | 打开之前已经保存的文件 |
| 7 | 退出文件 | 文件子系统 |  | 关闭文件并退出 |
| 8 | 编辑子系统 | C语言IDE |  |  |
| 9 | 剪切 | 编辑子系统 |  |  |
| 10 | 复制 | 编辑子系统 |  |  |
| 11 | 粘贴 | 编辑子系统 |  |  |
| 12 | 删除 | 编辑子系统 |  |  |
| 13 | 搜索子系统 | C语言IDE |  |  |
| 14 | 查找 | 搜索子系统 |  | 打开查找窗口，可查找代码中的指定字符串，若找到，则将字符串标记并显示，否则提示查找失败 |
| 15 | 下一个 | 搜索子系统 |  | 找到下一个查找的字符段 |
| 16 | 上一个 | 搜索子系统 |  | 找到上一个查找的字符段 |
| 17 | 替换 | 搜索子系统 |  | 打开替换窗口，将所有指定A字符段换成B字符段 |
| 18 | 运行子系统 | C语言IDE |  |  |
| 19 | 编译 | 运行子系统 |  | 打开编译窗口，可对代码进行编译 |
| 20 | 运行 | 运行子系统 |  | 打开运行窗口，可直接运行代码 |
| 21 | 编译运行 | 运行子系统 |  | 运行已编译的代码 |
| 22 | 帮助 | C语言IDE |  |  |

 3.3 整体功能模块关系图

3.4 系统外部接口功能模块关系图

此系统并未与任何外部系统进行通信。

3.5 系统内部接口功能模块关系图

这实在太为难我了，我并不是 软件项目 的实施者或者管理者，且 其文档实在是匮乏 ，要我来描绘出系统内部接口的约定和调用实在是过于困难。在此便不展示。

4 系统测试用例参考

[注：模板是按照系统模块化进行编写的，具体计划可参照执行]

4.1 XX系统

4.1.1 用户界面

4.1.1.1 XX模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 操作步骤 | 预期结果 | 备注 |
|  |  |  |  |
|  |  |  |  |

填写说明：

“测试用例”是根据《需求分析说明书》和《详细设计说明书》来制定的。

“预期结果”是根据《需求分析说明书》和《详细设定说明书》来制定的。

4.1.1.2 XX模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 操作步骤 | 预期结果 | 备注 |
|  |  |  |  |
|  |  |  |  |

4.1.2 功能测试

4.1.2.1 XX模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 操作步骤 | 预期结果 | 备注 |
|  |  |  |  |
|  |  |  |  |

填写说明：

“测试用例”是根据《需求分析说明书》和《详细设计说明书》来制定的。

“预期结果”是根据《需求分析说明书》和《详细设定说明书》来制定的。

4.1.2.2 XX模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 操作步骤 | 预期结果 | 备注 |
|  |  |  |  |
|  |  |  |  |

5 附录

5.1 附录1 审批记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 签名 | 日期 | 备注 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. 进度计划说明

6.1 进度计划2020/5/3  


7 相关文档（双击可打开）

1. 功能测试计划说明书  
   
2. 性能测试计划说明书  
   
3. 可靠性测试计划说明书  
   
4. 兼容性测试计划说明书

4.1.兼容性测试计划概述

兼容性测试是指检查能否兼容各种不同硬件和软件组成的运行环境，在多种不同环境中顺利运行并得到相同的结果。

根据软件的需求文档，我们将仅从软件维度对软件的兼容性进行测试，即只测试软件在多种操作系统下的运行状况和结果。

4.2.兼容性测试环境

我们计划在Windows 7，Windows 10，Ubuntu 这三个常见操作系统环境中测试软件的兼容性。

4.3.兼容性测试策略

4.3.1.安装

软件需求文档没有对安装做出要求，所以我们计划能成功打开界面即为通过本项测试内容。

4.3.2.功能测试

我们将逐模块和功能对软件进行运行测试，依次记录运行成果并与需求文档相对比，运行结果若能与需求文档相符合即为通过该功能的兼容性测试。若每一项功能和需求均能符合设计文档即为通过本项测试内容。

备注：软件的模块和功能列表如下：

|  |  |
| --- | --- |
| 文件 | 新建 |
| 保存文件 |
| 另存为文件 |
| 打开文件 |
| 退出文件 |
| 编辑 | 剪切 |
| 复制 |
| 粘贴 |
| 删除 |
| 搜索 | 查找 |
| 上一个 |
| 下一个 |
| 替换 |
| 运行 | 编译 |
| 运行 |
| 编译运行 |
| 帮助 | 无 |

4.3.3.升级测试

我们计划在安装软件的不同版本后，升级软件并测试升级后软件是否正常运行，并对4.3.2中功能逐个测试并记录它们是否通过测试。如果所有版本均能正常升级且在所有版本升级后每个功能均能通过测试，则记为通过升级测试。

4.3.4.卸载测试

我们将卸载软件，并测试软件是否能够顺利卸载，卸载后是否能够清理所有缓存，若能，则记为通过卸载测试。

4.4.兼容性测试通过准则

若能通过4.2及4.3中所有测试内容即为该软件通过兼容性测试，否则认为该软件未能通过兼容性测试。

备注：测试记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试项目 | 测试环境 | 测试时间 | 测试结果 | 预期结果 | 是否通过本项测试 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. 需求规格说明书审查计划
2. 回归测试计划说明书