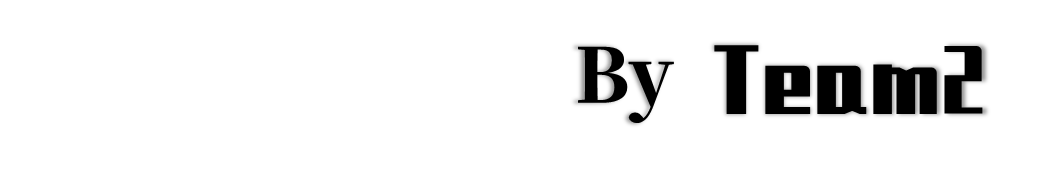
****

****

形状, 箭头

描述已自动生成

****

**目录**

[1 界面测试 2](#_Toc120778725)

[2 文件操作测试 9](#_Toc120778726)

[3 字符串查找替换测试 15](#_Toc120778727)

[4 括号配测试 22](#_Toc120778728)

1. 界面测试
   1. 原开发需求

利用QT编写一个IDE的界面，包括最上面的菜单栏、工具栏，中间的文本输入框，底部的编译日志输出框，左部的文件视图框。（这是最基本的框架，之后会迭代补充）

底部的编译日志输出框与左部的文件视图框要求可以用按钮来控制显示与关闭，也就是可以伸出来，也可以隐藏。

菜单栏先做两个菜单，要求点击之后有下拉菜单显示出来，具体功能可以先不写

工具栏也先写2-3个工具，每个工具都要用图标显示，鼠标悬停到图标上可以显示提示文字

底部的编译日志输出框与左部的文件视图框的编写原理应该相同，建议先写

底部的编译日志输出框，编译日志输出框比较重要

参考设计：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

* 1. 从原始需求中提取需求

（1）在界面的最上面要包含菜单栏与工具栏，菜单栏与工具栏是用户直接对整个IDE进行操作的一个窗口，是很重要的一部分。必须要符合需求。

（2）菜单栏要求点击之后有下拉菜单显示出来，点击对应的菜单显示相应的功能。

（3）工具栏每个工具都要用图标显示，鼠标悬停到图标上可以显示提示文字。

（4）在界面的中间是文本输入框，在界面的底部是编译日志输出框，在界面的左部是文件视图框。

（5）底部的编译日志输出框与左部的文件视图框要求可以用按钮来控制显示与关闭，也就是可以伸出来，也可以隐藏。

* 1. 根据需求制定测试用例并附上测试结果

针对（1）的测试用例：

运行整个程序，直接观察界面的最上面是否存在菜单栏与工具栏。测试时需要注意菜单栏与工具栏应该位于界面的最上面的部分而且应该使用左对齐的方式，这是符合习惯的方式，如果菜单栏与工具栏的位置不对或者对齐方式不对，应该属于不符合习惯的错误，应该修改。

测试背景：程序可以正确运行且已经可以显示界面

前置条件：无

重要级：高

优先级：高

测试类型：手工测试

预期输出结果：菜单栏与工具栏的位置是界面的最上面而且对齐方式为左对齐。

实际输出结果：

图形用户界面, 文本, 应用程序

描述已自动生成

测试结果：

测试无错误

针对（2）的测试用例：

点击菜单栏，观察其是否有下拉菜单弹出，进而观察下拉菜单中是否有相应的操作显示。

注意要观察菜单与其下拉菜单中的操作是否为对应关系，如果不是对应关系，属于逻辑错误，应该改正。

测试背景：程序可以正确运行且已经可以显示界面

前置条件：无

重要级：高

优先级：高

测试类型：手工测试

预期输出结果：点击菜单栏，其确实有下拉菜单弹出，而且菜单与其下拉菜单中的操作为对应关系。

实际输出结果：

图形用户界面, 文本, 应用程序

描述已自动生成

但是“工具”与“帮助”菜单没有下拉菜单

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

测试结果：

测试有错误，具体为：但是“工具”与“帮助”菜单没有下拉菜单

，需要编程人员去修改。

针对（3）的测试用例：

观察工具栏的每个工具是否都用图标显示，并将鼠标悬浮于图标上，观察图标是否可以提示文字。

测试背景：程序可以正确运行且已经可以显示界面

前置条件：无

重要级：高

优先级：高

测试类型：手工测试

预期输出结果：工具栏的每个工具都用图标显示，并且将鼠标悬浮于图标上的时候，每个图标都可以显示提示文字

实际输出结果：

图形用户界面, 文本, 应用程序

描述已自动生成

但是出现了功能与提示不相符的情况。

图形用户界面, 应用程序, Teams

描述已自动生成

图形用户界面, 应用程序, Teams

描述已自动生成

测试结果：

测试有错误。具体为：控制底部的编译日志输出框与左部的文件视图框显示与隐藏的按钮提示文字与功能不相符，分别为”textView”,”edit”。应该修改为“隐藏/显示编译框”与“隐藏/显示文件视图”

针对（4）的测试用例：

观察文本输入框是否在界面的中间，观察编译日志输出框是否在界面的底部，观察文件视图框是否在界面的左部。测试时需要注意文本输入框、编译日志输出框与文件视图框是否在指定位置，这是符合习惯的方式，如果菜单栏与工具栏的位置不对或者对齐方式不对，应该属于不符合习惯的错误，应该修改。

测试背景：程序可以正确运行且已经可以显示界面

前置条件：无

重要级：高

优先级：高

测试类型：手工测试

预期输出结果：文本输入框在界面的中间，编译日志输出框在界面的底部，文件视图框在界面的左部。

实际输出结果：

图形用户界面, 文本, 应用程序

描述已自动生成

测试结果：

测试无错误。

针对（5）的测试用例：

查看底部的编译日志输出框与左部的文件视图框是否可以用按钮来控制显示与隐藏。

测试背景：程序可以正确运行且已经可以显示界面

前置条件：无

重要级：高

优先级：高

测试类型：手工测试

预期输出结果：底部的编译日志输出框与左部的文件视图框可以用按钮来控制显示与隐藏。

实际输出结果：

底部的编译日志输出框单独隐藏

图形用户界面, 文本, 应用程序

描述已自动生成

文件视图框单独隐藏

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

底部的编译日志输出框与文件视图框一起隐藏

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

测试结果：

测试无问题

1. 文件操作测试
   1. 原开发需求

文件的操作有新建文件，文件另存为，删除文件，打开文件，关闭文件

点击新建文件按钮可以出现以下提示页面，询问用户要将文件保存在哪里，文件名是什么，保存为什么类型的文件

图形用户界面, 应用程序

描述已自动生成

点击另存为按钮也会出现与点击新建文件按钮后一样的界面询问用户要将文件保存在哪里，文件名是什么，保存为什么类型的文件

点击删除文件后，首先出现是否确认删除界面，询问用户是否要将文件删除，点击确认后要将储存在本地的文件删除。

点击打开文件按钮出现类似下图的界面，主要功能是让用户选择想要打开的文件

选择好后点击打开，则可以打开文件。

图形用户界面, 应用程序

描述已自动生成

点击关闭文件按钮就将当前文件关闭

* 1. 测试用例

1. 新建有名字的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：客户端已启动

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 点击“新建文件”按钮
2. 输入文件名字
3. 点击“ok”

预期结果：成功创建名字为输入的名称的文件

实际结果：与预期一致

1. 新建名字空白的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：客户端已启动

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 点击“新建文件”按钮
2. 点击“ok”

预期结果：提示“未输入文件名”

实际结果：成功创建文件，与预期不符

1. 创建名字重复的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：客户端已启动

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1、点击“新建文件”按钮

2、输入文件名字

3、点击“ok”

4、重复上述操作建立名字相同的文件

预期结果：提示“工作区文件名字重复”

实际结果：成功创建文件，与预期不符

1. 打开文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：客户端已启动

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 点击“打开”按钮
2. 选择文件路径
3. 打开文件

预期结果：左侧展开文件树，右侧打开文件

实际结果：与预期一致

1. 打开与工作区重名文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：客户端已启动

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 创建文件并保存
2. 关闭文件窗口
3. 新建同名文件

4、打开第一步中创建的文件

预期结果：提示“工作区文件名字重复”

实际结果：成功创建和打开文件，与预期不符

1. 保存刚创建的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：客户端已启动

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

点击“保存”按钮

预期结果：弹出保存路径选择

实际结果：点击保存logo进行操作时，未弹出保存路径选择窗口，与预期不符

1. 保存已另存为的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：已新建文件并另存为

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 编辑文件后，点击保存按钮
2. 编辑文件后，点击保存logo

预期结果：

1后成功保存文件和修改

2后成功保存文件和修改

实际结果：

1后弹出选择保存路径按钮

2后无法保存文件变更

与预期不符

1. 保存打开的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：已打开文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1、编辑文件后，点击保存按钮

2、编辑文件后，点击保存logo

预期结果：

1后成功保存文件和修改

2后成功保存文件和修改

实际结果：

1后弹出保存路径选择窗口，与预期不符

2后成功保存文件和修改

1. 删除已打开文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：已打开文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 右键点击文件树中文件
2. 点击“删除文件”按钮

预期结果：

成功删除文件

实际结果：

与预期一致

1. 删除已打开且源文件放入回收站的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：已打开文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 将源文件放入回收站

2、右键点击文件树中文件

3、点击“删除文件”按钮

预期结果：

第一步时提示“文件已打开”或第二步时提示“找不到源文件”

实际结果：

第二步时将文件识别为文件夹，仅有“删除文件夹”选项，与预期不符

1. 关闭已打开文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：工作区已有打开的文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

点击“关闭”按钮

预期结果：

关闭文件，若有未保存的更改则弹出确认窗口

实际结果：

不管是否存在未保存更改都弹出确认窗口

1. 关闭新建的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：已新建文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

点击“关闭”按钮

预期结果：

关闭文件，若有未保存的更改则弹出确认窗口

实际结果：

与预期一致

1. 关闭新建后已另存为的文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：已新建并保存文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

点击“关闭”按钮

预期结果：

关闭文件，若有未保存的更改则弹出确认窗口

实际结果：

另存为文件时若文件为空白，则直接关闭，若文件中有内容，则不论是否进行修改都弹出确认窗口；

保存文件时，由于无法保存空白文件，因此情况为不论是否进行修改都弹出确认窗口；

1. 另存为文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：工作区有文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1. 点击“另存为”按钮

2、选择保存路径

预期结果：

成功保存到目标路径

实际结果：

与预期一致

1. 另存为或保存重名文件

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：工作区有文件

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

1、点击“另存为”按钮

2、选择保存路径，文件名与路径中已有文件相同

预期结果：

提示“存在重名文件”或询问是否覆盖

实际结果：

询问是否替换文件，与预期一致

1. 工作区没有文件时进行文件操作

重要级：高

优先度：高

测试背景：测试文件操作功能

前置条件：客户端已打开

测试环境：IDE客户端

测试阶段：系统测试

测试步骤：

点击“另存为”，“关闭”或“保存”按钮

预期结果：

提示“工作区没有文件”

实际结果：

程序闪退，与预期不符

1. 字符串查找替换测试
   1. 原开发需求

字符串或者字符的查找与替换功能

1.字符串查找

点击查找按钮后，需要弹出一个类似下图的页面：

图形用户界面, 应用程序, 电子邮件

描述已自动生成

上面的页面中，只要做红色框中的内容，其他内容不需要做。

（1）提供全局查找功能，对于输入的字符或字符串，点击“全局查找”可以找出全文中含有多少个该字符串，并高亮所有字符串。

如图：

图形用户界面, 文本, 应用程序

描述已自动生成

（2）提供顺序查找功能，对于选中的字符或字符串，可首先高亮显示全文第一个该字符串，点击“next”按钮可找到并高亮下一个字符串。

如图：

图片包含 图示

描述已自动生成

2.字符串替换

（1）提供替换功能，点击替换按钮后，会弹出如下界面，这个页面需要自己去设计：

图形用户界面, 文本, 应用程序

描述已自动生成

其中红色框部分是首先要设计出来的，并且把功能实现。

蓝色框部分是在红色框部分设计好后并且把功能实现以后再去加入蓝色框包裹内容的布局，并且再把功能实现。

简单来说就是先把红色框里的内容做出来，有余力之后再把蓝色框里的内容做出来。

* 1. 测试用例设计

查找部分

对查找部分的测试主要侧重于功能测试，即黑盒测试，在完全不考虑程序内部结构和内部特性的情况下在程序接口进行测试，只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数锯而产生正确的输出信息，并且保持外部信息的完整性。

使用等价类划分法和边界值分析法设计以下测试用例：

（1）验证查找等按钮功能是否实现

（2）验证查找快捷键是否实现

（3）验证是否支持回车键查找

（4）验证是否支持查找内容的复制，粘贴，编辑操作

（5）关闭查找页面再点开后，验证原先的查找条件是否清空

（6）查找合法的内容，验证全局查找的结果个数是否正确

（7）查找合法的内容，验证全局查找的高亮结果是否正确

（8）查找合法的内容，验证查找上/下一个的结果是否正确

（9）查找合法的内容，在文本的第一个查找结果上再次点击“查找上一个”

（10）查找合法的内容，在文本的最后一个查找结果上再次点击“查找下一个”

（11）查找合法的内容，验证查找上/下一个一轮之后是否提示“查找达到了搜索的起始点”

（12）查找内容为空格，查看系统如何处理

（13）查找合法的内容，查看是否能区分大小写结果

（14）边界值验证，查找内容为空，验证系统如何处理

替换部分

对替换部分的测试主要侧重于功能测试，即黑盒测试，在完全不考虑程序内部结构和内部特性的情况下在程序接口进行测试，只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数锯而产生正确的输出信息，并且保持外部信息的完整性。

使用等价类划分法和边界值分析法设计以下测试用例：

（1）关闭页面再点开后，验证原先的替换内容是否清空

（2）查找指定的字符并全局替换，验证替换结果是否正确

（3）选中指定字符串并替换，验证替换结果是否正确

（4）不选定字符串，点击替换，验证系统是否提示

（5）将选中小/大写字母替换为小/大写字母，查看是否能区分大小写

（6）替换内容与选中字符串相同，验证系统是否提示

（7）全局替换与全局查找的内容相同，验证系统是否提示

（8）边界值验证，替换内容为空，验证系统如何处理

* 1. 测试用例与测试结果

查找部分

（1）查找等按钮功能是否实现

测试背景：测试查找的按钮功能

前置条件：程序有“全局查找”“顺序查找”“替换”“全局替换”四个按钮

重要级：中

优先级：高

测试步骤：依次点击四个按钮

预期结果：四个按钮均能打开对应界面

测试结果：仅有全局查找可打开功能界面，且该界面涵括查找和替换的所有功能，与预期不符

（2）查找快捷键是否实现

测试背景：测试查找的按键功能

前置条件：程序有基本的查找功能

重要级：低

优先级：低

测试类型：手工测试

测试步骤：按下“ctrl+f”键

预期结果：可打开查找页面

测试结果：按下“ctrl+f”键并无反应，与预期结果不符

（3）验证是否支持回车键搜索

测试背景：测试查找的按键功能

前置条件：程序有基本的查找功能

重要级：低

优先级：低

测试类型：手工测试

测试步骤：在查找文本框输入内容后，按下回车键

预期结果：可以显示查找内容

测试结果：在查找文本框输入内容后，按下回车键并无反应，与预期结果不符

（4）验证是否支持查找内容的复制，粘贴，编辑操作

测试背景：测试查找的文本编辑功能

前置条件：程序有基本的查找功能

重要级：低

优先级：低

测试类型：手工测试

预期结果：查找内容可以复制，粘贴，编辑操作

测试结果：与预期相同

（5）关闭查找页面再点开后，验证原先的查找条件是否清空

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：低

优先级：低

测试类型：手工测试

测试步骤：关闭查找页面再点开后

预期结果：原先的查找条件被清空

测试结果：查找框中仍是原先的查找条件，与预期结果不符

（6）查找合法的内容，验证全局查找的结果个数是否正确

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：高

优先级：高

测试类型：手工测试

测试步骤：现有代码文件包含6个“int”字符串，点击全局查找“int”

预期结果：查找个数为6

测试结果：测试结果与预期结果相同

（7）查找合法的内容，验证全局查找的高亮结果是否正确

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：现有代码文件包含6个“int”字符串

预期结果：全局查找“int”，有6处含“int”的字段高亮显示

测试结果：全局查找“int”，只有最后一个“int”字段高亮显示，与预期结果不符

（8）查找合法的内容，验证查找上一个的结果是否正确

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：高

优先级：高

测试类型：手工测试

测试步骤：现有代码文件包含6个“int”字符串，点击“查找上一个”或“查找下一个”按钮

预期结果：跳转到上/下一个“int”字段

测试结果：测试结果与预期结果相同

（9）查找合法的内容，在文本的第一个查找结果上再次点击“查找上一个”

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：在文本的第一个查找结果上再次点击“查找上一个”

预期结果：跳转到文本的最后一个匹配结果

测试结果：测试结果与预期结果相同

（10）查找合法的内容，在文本的最后一个查找结果上再次点击“查找下一个”

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：在文本的最后一个查找结果上再次点击“查找下一个”

预期结果：跳转到文本的第一个匹配结果

测试结果：测试结果与预期结果相同

（11）查找合法的内容，验证查找上/下一个一轮后是否提示“查找达到了搜索的起始点”

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：点击查找上/下一个一轮后再次点击查找上/下一个

预期结果：提示“查找达到了搜索的起始点”

测试结果：未有提示，仍继续顺序查找，与预期结果不符

（12）查找内容为空格，查看系统如何处理

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：当前代码文件包含156个空格，全局查找的内容为空格

预期结果：有156个匹配结果

测试结果：测试结果与预期结果相同，暂无修改建议

（13）查找合法的内容，查看是否能区分大小写结果

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：高

优先级：中

测试类型：手工测试

测试步骤：当前代码文件包含6个“int”，0个“INt”，分布全局搜索二者

预期结果：全局搜索“int”有6个匹配结果，全局搜索“INt”有0个匹配结果

测试结果：全局搜索“int”和“INt”均有6个匹配结果，系统无法识别大小写，与预期结果不符

（14）边界值验证，查找内容为空，验证系统如何处理

测试背景：测试查找功能的完善性

前置条件：程序有基本的查找功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：查找内容输入为空，点击全局搜索

预期结果：系统不作反应或提示“当前查找内容为空”

测试结果：显示有0个查找结果，与预期结果不符

总结：对于测试用例（1）（2）（3）（5）（7）（11）（13）（14），测试结果与预期结果不同，建议修改。

替换部分

（1）关闭替换页面后再点开，验证原先的替换内容是否清空

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：关闭替换页面后再点开

预期结果：原先的替换条件被清空

测试结果：查找框中仍是原先的替换内容，与预期结果不符

（2）查找指定字符串并替换，验证替换结果是否正确

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：高

优先级：高

测试类型：手工测试

测试步骤：在当前的代码文件中查找“a”并全局替换为“b”

预期结果：原先的“a”被替换为“b”

测试结果：测试结果与预期结果相同

（3）选中指定的字符并全局替换，验证替换结果是否正确

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：高

优先级：高

测试类型：手工测试

测试步骤：在当前的代码文件中选中“a”并替换为“b”

预期结果：原先的“a”被替换为“b”

测试结果：测试结果与预期结果相同

（4）不选定字符串，点击替换，验证系统是否提示

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：不选中文本内容，点击全局替换为“INt”

预期结果：提示“没有输入”

测试结果：测试结果与预期结果相同

（5）将选中小/大写字母替换为小/大写字母，查看是否能区分大小写

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：在当前的代码文件中选中“inT”并替换为“INt”

预期结果：原先的“inT”被替换为“INt”

测试结果：测试结果与预期结果相同，暂无修改建议

（6）替换内容与选中字符串相同，验证系统是否提示

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：中

优先级：中

测试类型：手工测试

测试步骤：在当前的代码文件中选中“int”并替换为“int”

预期结果：系统提示“替换内容与当前相同”或者不做反应

测试结果：不做反应，测试结果与预期结果相同

（7）全局替换与全局查找的内容相同，验证系统是否提示

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：高

优先级：高

测试类型：手工测试

测试步骤：在当前的代码文件中全局查找“int”并全局替换为“int”

预期结果：系统提示“替换内容与当前相同”或者不做反应

测试结果：程序未能响应直至程序异常终端，与预期结果不符

（8）边界值验证，替换内容为空，验证系统如何处理

测试背景：测试替换功能的完善性

前置条件：程序有基本的替换功能

重要级：中

优先级：低

测试类型：手工测试

测试步骤：在当前的代码文件中选中查找“b”，在替换框内不输入字符，点击替换

预期结果：系统提示“输入内容为空”或者不做反应，与预期不符

总结，对于测试用例（1）（7）（8），测试结果与预期结果不同，建议修改。

1. 括号配测试
   1. 原开发需求

（1）用户打出一个“{”，程序自动匹配“}”，这个操作包括(),[]

（2）在“()”中再次按下“)”键支持光标向右移动一位，退出括号,这个操作包括[]

（3）当光标放在括号位置时，该括号以及与该括号匹配的括号会有光标提示,如图：

图形用户界面

描述已自动生成

* 1. 测试用例与测试结果

（1）在输入框键入左括号“{”,“[”,“(”

测试背景：文本框已经实现了最基本的文本编辑功能

前置条件：无

重要级：高

优先级：高

测试类型：手工测试

测试步骤：

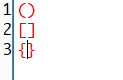
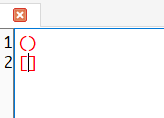
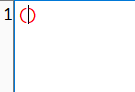
①鼠标点击将光标定位在输入框中；

②分别输入三种左括号“{”,“[”,“(”。

预期结果：

成功键入了左括号，相应的右括号也在光标右侧出现。

实际结果：



左括号被键入，并且正确匹配出相应的右括号，同时光标位于两侧的括号之间。与预期相符合。

（2）在输入框键入左括号“}”,“]”,“)”

测试背景：文本框已经实现了最基本的文本编辑功能

前置条件：无

重要级：高

优先级：高

测试类型：手工测试

测试步骤：

①鼠标点击将光标定位在输入框中；

②分别输入三种右括号“}”,“]”,“)”。

预期结果：

仅仅单独键入右括号，不匹配产生左括号。

实际结果：





“]”和“)”可以被正常输入，但是键入“}”后程序结束并显示以上的报错信息。与预期不符合。

（3）删除键入的左括号与自动匹配的右括号

测试背景：文本框已经实现了最基本的文本编辑功能

前置条件：已经在输入框键入左括号并正确匹配相应的右括号

重要级：高

优先级：高

测试类型：手工测试

测试步骤：

①鼠标点击将光标定位于想要删除的字符右侧；

②按下退格键。

预期结果：

光标左侧的单个字符被删除，且其他的字符都不会受到影响，删除左括号后匹配出的右括号依然存在。

实际结果：





删除“}”,“]”,“)”,“[”,“(”都没有任何影响，但是删除“{”后软件停止运行并显示如上的报错信息。与预期不符合。

（4）键入多层嵌套的“{”,“}”并删除

测试背景：文本框已经实现了最基本的文本编辑功能

前置条件：已经在输入框键入左括号并正确匹配相应的右括号

重要级：高

优先级：高

测试类型：手工测试

测试步骤：

①在输入框中连续键入多个“{”;

②按下退格键删除不同数量的“{”与匹配生成的“}”。

预期结果：

对于任意数量的“{”和“}”都能够在按下退格键后正确删除光标左侧的单个字符。

实际结果：





对于“}”无论如何操作都能正常删除；对于“{”，如果和下一个“{”之间存在“}”，删除掉最后一个“{”时软件会停止运行并显示如上的报错信息。与预期不符合。

（5）匹配出相应的右括号以后再次输入右括号

测试背景：已经实现了在文本框输入左括号自动匹配出相应右括号的功能

前置条件：无

重要级：低

优先级：低

测试类型：手工测试

测试步骤：

①在文本框分别键入“[”和“(”；

②再次键入相对应的右括号“]”和“)”。

预期结果：

并没有再多输入一个右括号，而是直接将光标移动到了刚才自动匹配的右括号外。

实际结果：



和普通的输入一样，额外出现了一个右括号，并且光标也并未移动到右括号的外侧。与预期不符。

（6）将光标移动到“(”左侧

测试背景：文本框已经实现了最基本的文本编辑功能

前置条件：无

重要级：低

优先级：低

测试类型：手工测试

测试步骤：

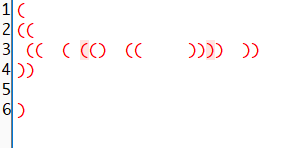
①在文本框中输入若干嵌套或并列的“(”；

②将光标移动到不同的“(”左侧。

预期结果：

该“(”以及与它相匹配的“)”文本背景会显示为淡红色，并且括号间的匹配关系完全正确。

实际结果：

所有的括号匹配关系均正确，并且文本背景显示为淡红色。与预期相符。