|  |
| --- |
| IDE需求文档 |

|  |
| --- |
| 陈雨、邱瑞亨  2019-9-10 |

**IDE需求规格说明文档**

**2019年9月**

目录

[1.项目描述 2](#_Toc19033802)

[1.1项目名称： 2](#_Toc19033803)

[1.2项目背景： 2](#_Toc19033804)

[2.需求分析： 2](#_Toc19033805)

[2.1功能性需求 2](#_Toc19033806)

[2.2性能性需求 2](#_Toc19033807)

[3.功能设计： 3](#_Toc19033808)

[3.1文件操作设计： 3](#_Toc19033809)

[3.2编辑操作设计： 3](#_Toc19033810)

[3.3编译设计： 3](#_Toc19033811)

[4.界面设计： 3](#_Toc19033812)

[4.1原型设计： 3](#_Toc19033813)

[4.2实际界面(初版)： 4](#_Toc19033814)

[5.操作设计： 5](#_Toc19033815)

[6.软件架构： 5](#_Toc19033816)

[7.相关接口 5](#_Toc19033817)

# 1.项目描述

## 1.1项目名称：

C语言集成开发环境

## 1.2项目背景：

代码推动着信息时代的发展，目前推出的各种代码编译器使得代码的书写和调试变得十分方便。不同的代码编译器有不同的风格，但最终目的都是提高人们编写的效率。本项目旨在开发出一款功能齐全，却又简约便捷的代码书写编译器。

# 2.需求分析：

## 2.1功能性需求

①提供对C程序的基本编辑功能：复制、粘贴、剪切、全选，支持通过相应的快捷键进行操作。

②支持多文件编辑，可以实现多个文件同时打开进行相关的操作。

③能够对C语言源程序进行编译，并反馈编译结果，给出正确程序的执行结果。程序的编译和执行结果是程序员比较关心的问题，也是判断一个IDE是否有效的关键。

④支持关键字识别，括号自动匹配。关键字的识别和括号匹配减少了代码书写的工作量，提高了工作的效率。

⑤实现对程序的debug。

⑥在编辑过程中可以隐藏/打开程序中的全部注释语句。注释隐藏后，源程序部分可以继续编辑，但注释部分不能更改。注释打开后，原来隐藏的注释重新显示在正确的位置。

## 2.2性能性需求

①操作响应无明显延迟，将源代码编译以及结果显示时间降到最低，减少时间和资源的浪费。

②能够对C语言源程序的代码行数进行显示，便于对代码进行相应的操作和后续的管理。

③支持1000行以下的代码书写编译。

# 3.功能设计：

## 3.1文件操作设计：

实现文件的打开、保存和另存为等功能。不同功能可以利用QAction：：setShortcut设置快捷键进行操作。使用Qt库中得QFile类保存文件名和属性， QString QFileDialog::getOpenFileName()打开文件的默认路径，QTextStream 流类型获取文本内容进行操作。

## 3.2编辑操作设计：

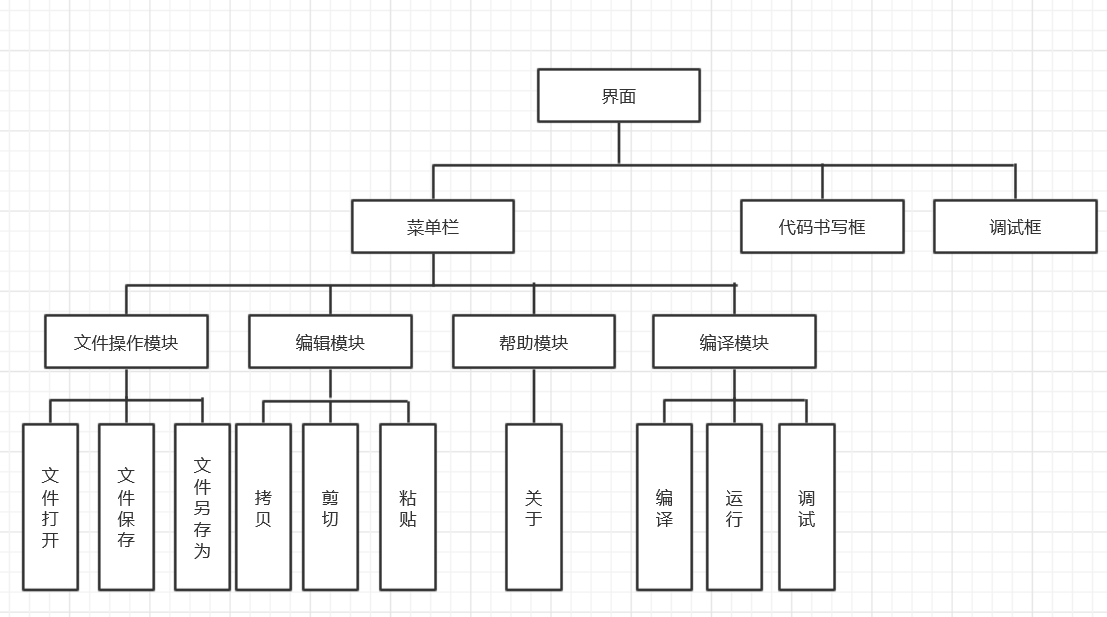
可以实现对文本的拷贝、粘贴、剪切、全选等功能。利用editor->copy(),editor->cut()等进行具体的功能实现。

## 3.3编译设计：

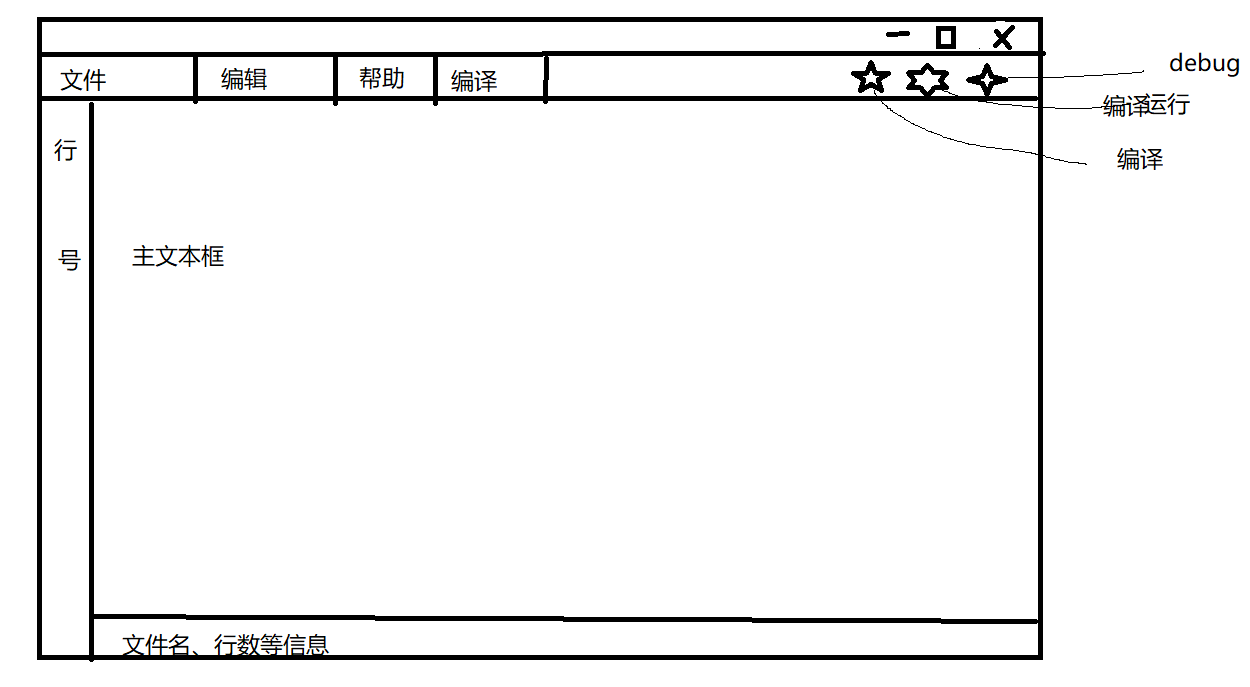
# 4.界面设计：

## 4.1原型设计：

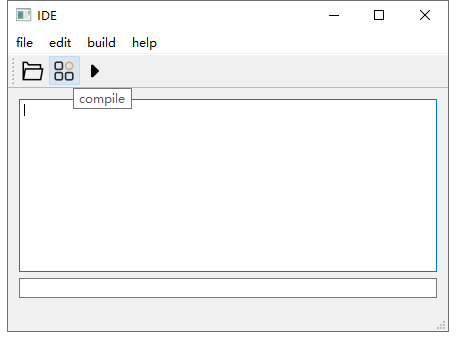
整体设计：



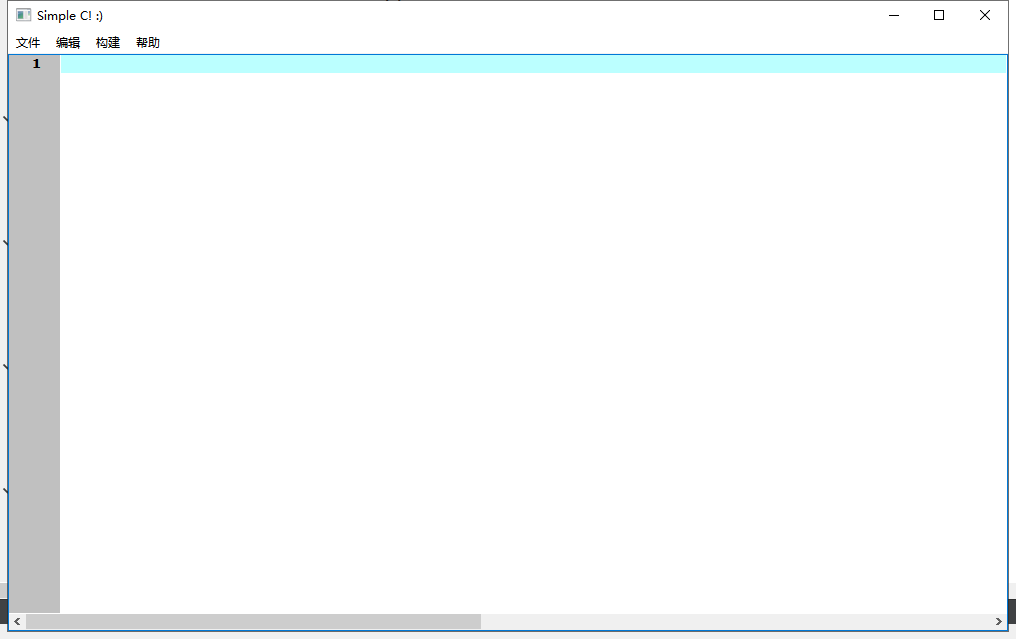
界面设计：



## 4.2实际界面(初版)：



第二版：



# 5.操作设计：

打开IDE界面后，点击空白编辑框即可以进行代码的书写，界面左侧显示代码行数，方便进行调试。

如果需要编译已存在的文件，在菜单栏点击文件选择打开，即可成功显示在编辑框内，同时可以通过预设的快捷键进行操作。

对代码的拷贝、粘贴等功能可以通过鼠标进行实现，同时也支持菜单栏和快捷键的操作。

使用过程中存在的任何问题可以参考帮助文档解决。

代码的编译和运行在菜单栏点击相关菜单栏实现。

# 6.软件架构：

整个窗口视作一个大类MainWindow，继承QMinWindow类。  
使用QTextEdit的实例接收用户的输入，定义所需的QMenu和QAction以及相应接口slot  
各个一级按钮抽象为一个菜单以QMenu数据类型保存于ManWindow内  
各一级按钮下的二级按钮抽象为QAction变量仍存于ManWindow内  
用QMnue的addAction方法将二级按钮与一级按钮关联  
按钮功能用connect函数将鼠标事件与QAction和相应接口关联以实现。

# 7.相关接口

void **on\_open**();

void **on\_about**();

void **on\_exit**();

void **on\_copy**();

void **on\_cut**();

void **on\_paste**();

void **on\_selectall**();

void **on\_save**();

void **on\_compile**();

void **on\_run**();

void **Save\_File**();

void **precomp**();