**C语言集成开发环境需求规格说明文档**

**2019年8月**

**目录**

[1.项目描述 1](#_Toc18171973)

[1.1项目名称： 1](#_Toc18171974)

[1.2项目背景： 1](#_Toc18171975)

[2.需求分析： 2](#_Toc18171976)

[2.1功能性需求 2](#_Toc18171977)

[2.2性能性需求 2](#_Toc18171978)

[3.界面设计： 3](#_Toc18171979)

[3.1原型设计 3](#_Toc18171980)

[3.2实际界面（初版）： 3](#_Toc18171981)

[4.软件架构： 4](#_Toc18171982)

[5.数据结构设计： 4](#_Toc18171983)

[6.相关接口： 5](#_Toc18171984)

# 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。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

提供对C程序的基本编辑功能：复制、粘贴、剪切，使得用户可以在IDE上进行代码的书写以及后续的修改工作。

支持单、多文件编辑。

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。

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。

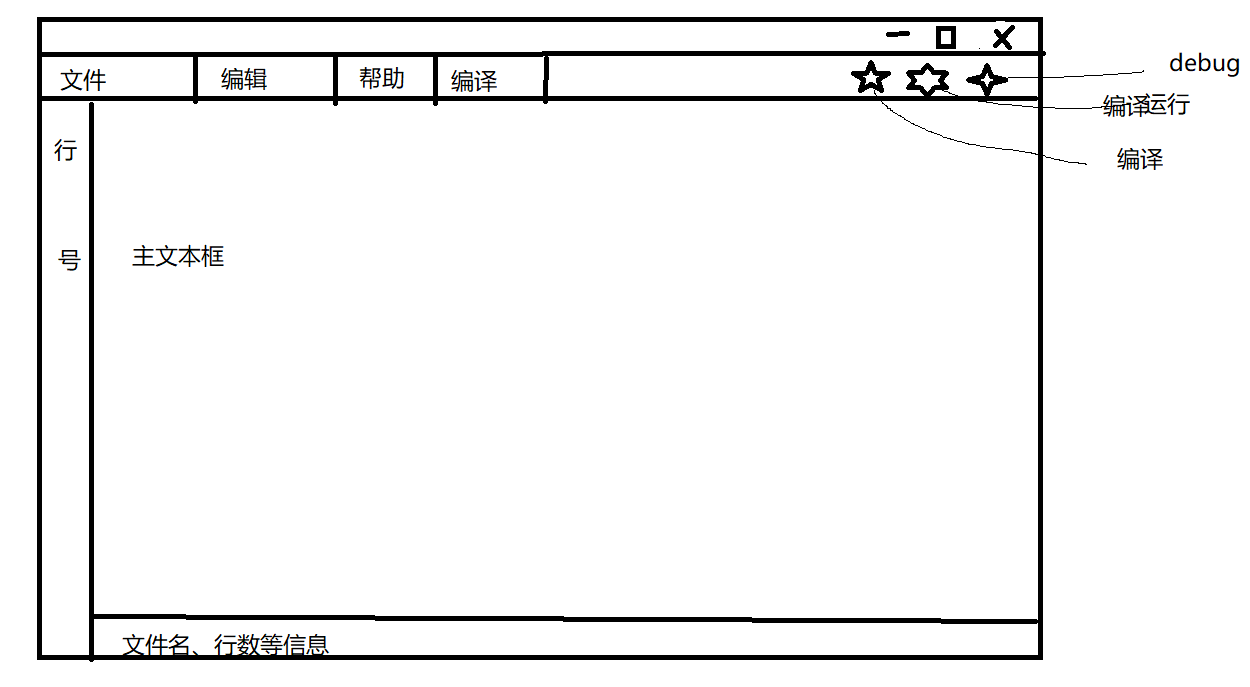
## 2.2性能性需求

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。

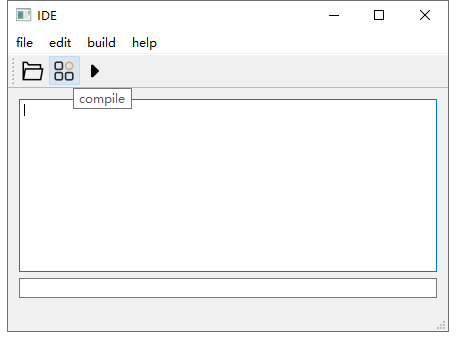
能够对C语言源程序的代码行数进行显示，便于对代码进行相应的操作和后续的管理。

# 3.界面设计：

## 3.1原型设计



## 3.2实际界面（初版）：



# 4.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。

# 5.数据结构设计：

class MainWindow: public QMainWindow {  
private:  
bool changed;//文本保存前是否改动  
QString filename;//文件名  
QTextEdit \*text1;//中央控件  
  
//文件menu  
QMenu \*FileMenu;  
QAction \*NewFile;  
QAction \*OpenFile;  
QAction \*SaveFile;  
  
//编辑menu  
Qmenu \*EditMenu;  
QAction \*Copy;  
QAction \*Paste;  
QAction \*Cut;  
  
//帮助menu;  
QMenu \*Help;  
  
//编译menu  
QMenu \*CompileMenu;  
QAction \*Compile;  
QAction \*Run;  
  
}  
  
class QMenu {  
private:  
char\* name;//菜单名称  
int num;//功能数   
public:  
QMenu(char\* aname) {  
strcpy(this->name, aname);  
this->num = 0;  
}  
  
void addAction(QAction buf);//关联功能  
void addbar(void);//加分隔符分隔不同功能  
}  
  
class QAction {  
private:  
char\* name;//动作命名  
char\* shortcut;//快捷键  
MainWindow self;//指向主体  
public:  
QAction(char\* aname, MainWindow aself) {  
strcpy(this->name, aname);  
this->self = aself;  
}  
  
void setShortcut(tr());//设置快捷键  
}；

# 6.相关接口：

file\_new()//创建文件  
file\_open()//打开文件  
file\_save()//保存文件  
edit\_copy()//复制  
edit\_paste()//粘贴  
edit\_cut()//剪切  
add\_bracket()//判断括号类型并自动补齐  
find\_keyword()//寻找相关关键字匹配  
compile\_com()//开始编译  
compile\_run()//运行