УТВЕРЖДЕН
А.В.00001-01 33 01-1-ЛУ

Оформление лабораторной работы в рамках ГОСТ 19

Основы работы с библиотекой Qt

Инв.№	
Подп. и дата	
Взам. Инв.	
Инв. № дубл.	
Подп. и дата	

Жёсткий диск

Руководство программиста

А.В.00001-01 33 01-1-ЛУ
(Твердотельный накопитель)

10 листов

2022

АННОТАЦИЯ

В данном программном документе приведено описание работы программы, представленной для выполнения студентом второго курса направления «Компьютерная безопасность» факультета МИЭМ им.Тихонова НИУ ВШЭ.

В данном программном документе, в разделе «Назначение программы» приведено описание назначения программы, возможности данной программы, а также её основные характеристики и скорость работы.

В данном программном документе, в разделе «Условия применения», указаны условия, необходимые для выполнения программы (требования к необходимым для данной программы техническим средствам, и другие программы, общие характеристики входной и выходной информации, а также требования и условия организационного, технического и технологического характера).

В данном программном документе, в разделе «Описание задачи» указаны определения задачи и методы её решения.

Оформление программного документа «Описание программы» произведено по требованиям ЕСПД (ГОСТ 19.101-77 [1)], ГОСТ 19.103-77 [2)], ГОСТ 19.104-78* [3)], ГОСТ 19.105-78* [4)], ГОСТ 19.106-78* [5)], ГОСТ 19.502-78* [6)], ГОСТ 19.604-78* [7)]).

[1)] ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов
[2)] ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов
[3)] ГОСТ 19.104-78* ЕСПД. Основные надписи
[4)] ГОСТ 19.105-78* ЕСПД. Общие требования к программным документам
[5)] ГОСТ 19.106-78* ЕСПД. Общие требования к программным документам, выполненным печатным способом
[6)] ГОСТ 19.502-78* ЕСПД. Описание применения. Требования к содержанию и оформлению
[7)] ГОСТ 19.604-78* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1. Назначение программы

Программа создает текстовый редактор с базовыми возможности для обработки текстовых документов.

1.2. Возможности программы

Программа предоставляет доступ к использованию текстового редактора с базовыми возможностями (сохранение, редактирование, создание нового документа). Позволяет редактировать файлы с программным кодом C++20.

1.3. Основные характеристики программы

Программа содержит методы для работы рабочей области текстового редактора, подсветки синтаксиса C++20 и QtCreator, обеспечения настройки/загрузки состояния программы из ini-файла, поиска переменных и функций с помощью ctags, сворачивание/разворачивание кусков кода с помощью cscope, построение графа вызовов функций с помощью cflow.

1.4. Ограничения, накладываемые на область применения программы

Программа имеет прикладное применение в качестве текстового редактора с базовыми возможностями (сохранение, редактирование, создание нового документа). Позволяет редактировать файлы с программным кодом C++20. Не обеспечивает компиляцию кода и исправление синтаксических, орфографических ошибок.

2. УСЛОВИЯ ПРИМЕНЕНИЯ

2.1. Требования к техническим (аппаратным) средствам

В состав используемых технических средств должны входить:

1. Процессор 5-ого поколения Intel Core или AMD Ryzen
2. ОЗУ более 4 Мбайт

2.2. Требования к программным средствам (другим программам)

Наличие компилятора MinGW-w64 C++ GNU
QtCreator 5.15

2.3. Общие характеристики входной информации

Программа не имеет входных данных.

2.4. Общие характеристики выходной информации

Первая программа реализует работу с QtCreator.

2.5. Требования и условия организационного характера

Для обеспечения грамотной работы программ не требуется каких-либо особых условий организационного характера.

2.6. Требования и условия технического характера

Реализация данных программ не требует каких-либо особых условий технического характера.

2.7. Требования и условия технологического характера

Для работы программ не требуется обеспечения каких-либо особых требований и условий технологического характера.

3. ОПИСАНИЕ ЗАДАЧИ

3.1. Определение задачи

Разработать программу с графическим интерфейсом пользователя «Текстовый редактор». Программа должна обеспечивать подсветку синтаксиса для языка программирования C++20, а также библиотеки Qt.

Интерфейс программы должен содержать меню, область «Обозреватель», выполняющую навигацию по всем открытым документам, область «Редактор», содержащую вкладки с открытыми документами. Для реализации механизма навигации использовать ctags.

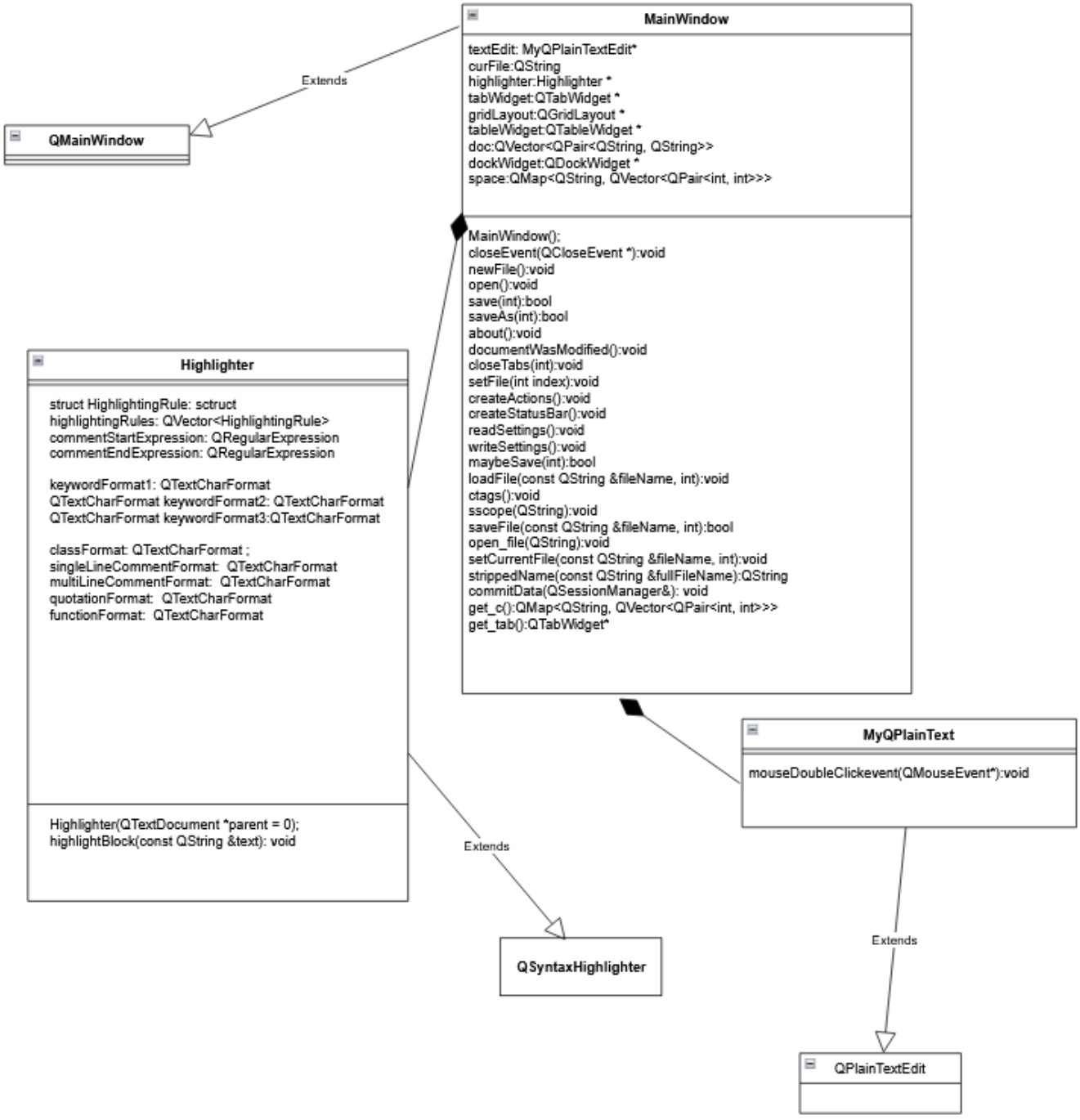
Настройки/состояние программы загружаются/сохраняются из/в ini-файла.

Предусмотреть возможность восстановления работы программы при сбоях.
Область «Редактор», содержащая вкладки с открытыми документами, должна иметь возможность «сворачивать» блоки кода, определяющие тело функции. Для реализации механизма использовать cscope.

Добавить пункт главного меню «Инструменты». В меню «Инструменты» добавить пункт «Граф вызовов» - создающий новое окно, в котором строится граф вызовов для текущей вкладки «Редактора». Для построения графа вызовов использовать QGraphicsScene. Если текущая вкладка не открыта или пуста, пункт меню недоступен. Информацию, необходимую для построения графа вызовов получить с помощью cflow.

Добавить в окно «Граф вызовов» программу, содержащую главное меню «Файл» с пунктами «Открыть» и «Выход». При выборе «Открыть» открывается диалоговое окно с выбором файла в формате xml для чтения графа и отображения его в окне программы.

3.2. Методы решения задачи



UML-диаграмма программы. Рис.1

Реализован класс MainWindow, обеспечивающий работу графической оболочки.

Public:

QMap<QString, QVector<QPair<int, int>>> get_c() – доступ к данным cscope.

QTabWidget * get_tab()– доступ к окну документа.

void loadFile(*const* QString &fileName, int) – загрузка текста открываемого в рабочую область редактора файла.
void cscope(QString) – парсинг файла cscope для реализации сворачивания/разворачивания строк кода.

Protected:

void closeEvent(QCloseEvent *event) *override*; - закрытие файлов().

Private slots:

void newFile() – добавление нового пустого файла с именем «untitled.txt».
void open() – открытие существующего файла.
bool save(int) – промежуточная функция сохранения файлов. Вызывает либо SaveAs(int), либо SaveFile() в зависимости от файла (было ли ранее сохранение или нет).
bool saveAs(int) – сохранение нового файла, не имевшего сохранения ранее на компьютере.
bool saveFile(const QString&, int) – сохранение файла в систему.
void documentWasModified() – проверка на изменение документа.

void closeTabs(int)– закрытие файлов в панели редактора.

void setFile(int) – загрузка данных файла при открытии.

void graph() – создание графа по открытому файлу.
void saveGraph() – сохранение графа в формате xml в новом файле формата .xml
void xml() – открытие файла формата .xml в виде графа в новом окне.

Private:

void createActions() – функция с описанными сигналами, создание графической оболочки.
void createStatusBar() – строка состояния, нижняя панель графической оболочки.
void readSettings() – настройки размеров окна, настройка ini-файла.
void writeSettings() - настройки размеров окна, создание ini-файлов.
bool maybeSave(int) – промежуточная функция для проверки на сохранения, закрывающегося файла.
void open_file(QString) – открытие файла из ini-файла.
void ctags() – парсинг файла ctags для реализации поиска переменных/функций по всем файлам.
void setCurrentFile(*const* QString &fileName, int index) – ставит статус на изменение файла.
MyQPlainTextEdit *textEdit; - текстовое поле.
QString curFile; - текущий файл.
Highlighter *highlight; - объект подсветки.
QTabWidget *tabWidget; - окно файла.
QGridLayout *gridLayout; - окно файла на панели.

QTableWidget *tableWidget; - таблица ctags данных.
QDockWidget *dockWidget; - ячейки таблицы ctags данных.
QVector<QPair<QString, QString>> doc; - данные ctags.
QMap<QString, QVector<QPair<int, int>>> space; - данные cscope
QVector<QPair<QPair<std::string, int>, QPair<std::string, int>>> data; - данные cflow.
int maxTab = 0; - глубина дерева
QGraphicsScene *scene; - новое окно под граф
QGraphicsView *view; - объект для отрисовки графа

Добавлен класс Highlighter для подсветки синтаксиса файлов с расширениями .cpp и .h.

Добавлен класс MyQPlainTextEdit наследующий QPlainTextEdit, в котором перегружена функция двойного нажатия на область рабочего окна в целях отслеживания команды скрытия/раскрытия блока кода.

Использована утилита ctags (утилита, собирающая информацию об именах и позициях переменных, функций и процедур, встречающихся в исходном коде компьютерных программ. При этом генерируется тег-файл). Данные тег-файла считываются, обрабатываются и добавляются в ячейки таблицы справа на панели.

Использована утилита cscope (утилита, предназначенная для навигации по исходному коду. Используется для поиска в исходном коде функций, объявлений, определений и регулярных выражений по заданной строке текста). Обеспечивает сворачивание/разворачивание функций. В генерирующемся файле находятся названия функций ($имя) и номера строк, начало({) и конец(}). С помощью данных о конце, начале функции и статуса следующей после названия функции строки(isVisible()), выставляется статус блока (setVisible(true/false)).

Использована утилита cflow – инструмент для статического анализа кода языка C, с помощью которого могут быть созданы отношения между вызовами функций. Сформированный в файле cflow.txt список считывается в вектор пар из пар. В первой паре записывается родитель и его глубина вхождения, во второй – ребенок и его глубина вхождения.

Граф отрисован с помощью библиотеки QGraphicsScene. В новое окно добавлены текст с помощью addText() и отрисованы линии с помощью addLine(). Для координат сформирован отдельный вектор из начал и концов требуемых прямых.

Реализована возможность сохранение графа вызовов в файле формата .xml, а также считывание содержимого файлов .xml и последующее открытие считанных данных в виде графа вызовов в новом окне. Работа с xml-файлами реализована с помощью библиотеки QXmlStreamWriter.

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

4.1. Сведения о входных данных

Пример выполнения программы реализован внутри main.cpp и не требует пользовательского вмешательства.

4.2. Сведения о выходных данных

Графическая оболочка текстового редактора с описанными ранее возможностями.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Изм.	*Номера листов (страниц)*				*Всего листов (страниц) в докум.*	*№ докум.*	*Входящий № сопрово-дительно-го докум. и дата*	*Подп.*	*Дата*
	изменён-ных	*заменё-ных*	*новых*	*аннулиро-ванных*					

Подп. и дата	
Инв. № дубл.	
Взам. Инв.	
Подп. и дата	
Инв.№	