**ЛАБОРАТОРНАЯ РАБОТА № 3–4**

по курсу «Основы программирования на C++»

**Разработка текстового редактора средствами Qt**

Используя функционал Qt, разработать текстовый редактор с поддержкой расцветки синтаксиса.

**Часть 1**

1. Предоставить возможность открытия нескольких файлов посредством использования виджета вкладок (QTabWidget).
2. Редактор *должен производить анализ расширения файлов при открытии*. Если файл имеет расширение файлов исходных кодов программ (например, .h, .cpp или .hpp – расширения файлов исходных кодов программ на C++), то при открытии необходимо выполнить расцветку синтаксиса с использованием класса QSyntaxHighlighter.
3. *Все расцветки должны быть зарегистрированы в xml-файле настроек приложения*. В данном файле регистрируются расширения файлов для данного языка программирования и соответствующие ключевые слова и регулярные выражения. (Как минимум должна быть предоставлена расцветка синтаксиса C++). Пример xml-файла настроек редактора:

<?xml version="1.0" encoding="utf-8"?>

<settings>

<syntaxes>

<syntax id = "python"

name = "Язык программирования Python"

ext\_list="py">

<!-- Ключевые слова -->

<rule>

<pattern value = "\bclass\b"/>

<format foreground="#00008B" font\_weight="75"/>

</rule>

<!-- Однострочный комментарий -->

<rule>

<pattern value = "#[^\n]\*"/>

<format foreground="#008000" font\_weight="75"/>

</rule>

</syntax>

</syntaxes >

</settings>

**Часть 2**

1. Реализовать меню (QMenu), содержащее следующие пункты:

**Файл:**

*Новый* (Создание нового документа)

*Открыть* (Открытие существующего документа на основе QFileDialog)

*Сохранить* (Сохранение документа из активной вкладки)

*Сохранить как…* (Сохранение в заданном формате на основе QFileDialog)

*Сохранить все* (Сохранить все открытые документы)

*Закрыть* (Закрыть текущий документ)

*Закрыть все* (Закрытие всех открытых документов)

*Выход* (Закрыть редактор с диалогом сохранения несохраненных документов)

**Правка:**

*Вырезать*

*Копировать*

*Вставить*

*Удалить*

*Выделить все*

**Вид:**

*Показать проводник* (Создание обозревателя каталогов с использованием QDockWidget и QFileSystemModel. При выборе документа в обозревателе открывать документ в новой вкладке.)

*Показать обозреватель открытых документов* (Создание обозревателя открытых документов на основе QDockWidget и QStringListModel. При щелчке по имени документа должна становиться активной его вкладка.)

1. Перед закрытием приложения должно выводиться сообщение с предложением сохранить измененные документ(-ы). Окно сообщения разработать на основе классов QDialog и QTableWidget. В первом столбце таблицы выводить имя файла, во втором – его расположение. При закрытии отдельной вкладки предлагать сохранять измененный файл с использованием диалогового окна сообщений (QMessageBox).
2. Возможные ошибки обрабатывать выводом окна критических сообщений.
3. Для открытия и сохранения файлов использовать стандартное диалоговое окно выбора файлов (QFileDialog).
4. Продублировать пункты меню Новый, Открыть, Сохранить, Сохранить все в панели инструментов Tool Bar (QToolBar). Изображения для рисунков соответствующих кнопок хранить в ресурсах.

**Литература**

1. Пример реализации расцветки: <https://doc.qt.io/qt-5/qtwidgets-richtext-syntaxhighlighter-example.html>

2. Пример создания виджета для нумерации строк: <https://doc.qt.io/qt-5/qtwidgets-widgets-codeeditor-example.html>