МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"  
  
(ФГАОУ ВО ПНИПУ)  
 *ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ  
КАФЕДРА ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ*УДК 004.65

**ОТЧЁТ ПО ПРОЕКТУ**(**РАБОТЕ)  
РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА ДЕЛОПРОИЗВОДИТЕЛЯ**

студенты 1 курса группы РИС-24-3Б  
специальности 09.03.04 Программная инженерия  
Черемных Т.Д. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 студент подпись  
Шитов А.А. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 студент подпись  
Полякова О.А./ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 руководитель подпись

Пермь 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc198412149)

[Раздел 1 Анализ предметной области и проектирование 5](#_Toc198412150)

[1.1 Анализ предметной области 5](#_Toc198412151)

[1.2 Анализ аналогичных существующих разработок 6](#_Toc198412152)

[1.3 Проектирование 6](#_Toc198412153)

[1.3.1 UML диаграмма класса 6](#_Toc198412154)

[1.3.2 Требования к информационной и программной совместимости 8](#_Toc198412155)

[1.4 Выбор и обоснование среды разработки 9](#_Toc198412156)

[1.4.1 Язык программирования 9](#_Toc198412157)

[1.4.2 Среда разработки 9](#_Toc198412158)

[Раздел 2 Разработка приложения 10](#_Toc198412159)

[2.1 Запуск программы в области уведомлений системы 10](#_Toc198412160)

[2.2 Реализация работы с буфером 11](#_Toc198412161)

[2.3 Защита от множественного запуска приложения 12](#_Toc198412162)

[2.4 Обработка текста 14](#_Toc198412163)

[2.5 Создание формы 17](#_Toc198412164)

[2.6 Тестирование системы 18](#_Toc198412165)

[ЗАКЛЮЧЕНИЕ 19](#_Toc198412166)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_Toc198412167)

# ВВЕДЕНИЕ

При использовании внешних источников информации во время составления электронных документов, скопированный текст может иметь форматирование, отличное от принятых стандартов.

В таком случае большая часть текстовых редакторов предоставляет две опции:

* Сохранение исходного формата скопированного текста – в таком случае, агрегация данных из множества различных источников приводит к определенному беспорядку в документе, в большей мере за счет разного шрифта текста.
* Применение форматирования документа к скопированному тексту, в этом случае вместе с приведением шрифта к нужному формату, будут утеряны списки, выделение текста цветом, полужирные начертания и другое форматирование, которое улучшает читаемость текста.

В связи с этим появляется частая необходимость ручного изменения формата для сохранения структуры скопированного текста. Данная проблема ощущается наиболее остро при долгосрочной работе с объемными исследованиями, где сбор информации может занимать значительную долю времени.

Целью данного проекта (работы) является разработка автоматизированного рабочего места делопроизводителя.

В рамках данного проекта (работы), можно сформулировать список задач, выполнение которых необходимо для достижения поставленной цели.

Поставленные задачи:

* Изучить предметную область.
* Определить структуру приложения.
* Определить среду разработки, а также основной инструментарий, который будет использоваться при проектировании, разработке и реализации приложения.
* Спроектировать структуру программного продукта, его классы, путем создания диаграмм.
* Разработать приложение.
* Провести тестирование и отладку приложения.
* Подготовить конечную версию приложения.

Объектом исследования является работа делопроизводителя, исходя из этого можно сделать вывод, что предметом исследования является автоматизация рабочего места делопроизводителя.

Инструментами для достижения поставленной цели в данной работе будут служить различные теоретические (анализ, дедукция, моделирование) и эмпирические (наблюдение и исследование конкретных явлений) методы исследования.

Конечным результатом выполнения данного дипломного проекта (работы) будет являться достижение поставленной цели, путём выполнения всех поставленных задач, а также готовый программный модуль для автоматизации рабочего места делопроизводителя, получение студентами практического опыта и теоретических знаний и навыков, непосредственно касающихся профиля обучения, в том числе, практических навыков в анализе и изучении предметной области, проектировании, реализации, разработке, тестировании и отладке программного продукта. А также демонстрация уже имеющихся у студента практических и теоретически знаний, навыков и умений.

# Раздел 1 Анализ предметной области и проектирование

## 1.1 Анализ предметной области

В современном мире, все большее число задач требует проведения глубоких исследований и анализа больших объемов информации, представленной в электронном виде. Эта тенденция особенно актуальна при подготовке исследовательских работ в различных областях знаний.

Долгосрочная работа по сбору данных включает в себя обширную исследовательскую работу, требующую многократного обращения к электронным источникам, включая научные публикации, статьи, отчеты, статистические данные, нормативные документы и другие ресурсы, доступные в электронном формате.

В процессе исследований, пользователи часто прибегают к копированию фрагментов данных из различных источников для последующей обработки и включения в свои работы. Несмотря на широкое распространение метода "copy-paste", его ручное применение влечет за собой ряд проблем, выходящих за рамки простого копирования информации.

Основная проблема заключается в необходимости ручного форматирования скопированного текста. При копировании из разных источников, форматирование текста (шрифты, размеры, отступы, стили) может существенно отличаться, что приводит к нарушению единообразия и профессионального вида итогового документа. Пользователи вынуждены тратить значительное время на приведение скопированного текста к единому стилю, что отвлекает их от основной задачи – анализа и интерпретации информации.

Необходимо разработать программное решение, которое позволит автоматически приводить скопированный текст к единому стилю, что в долгосрочной перспективе существенно повысит эффективность пользователя при работе с поиском информации, позволив сосредоточиться на содержательной части работы, а не на технических аспектах форматирования текста.

## 1.2 Анализ аналогичных существующих разработок

В рамках анализа и подготовки к проектированию были проанализированы аналогичные программные решения создаваемому программному модулю, наиболее популярными среди которых стали:

* Встроенный буфер обмена Windows (Win+V) – удобен для истории и базового копирования, но не предоставляет возможности автоматически форматировать текст.
* Ditto – данное решение предлагает базовые возможности форматирования (удаление форматирования, преобразование регистров), но не автоматизирует их по умолчанию. Нужно явно выбрать действие перед вставкой. Можно использовать внешние скрипты, но это требует настройки.
* PureText – бесплатная и простая программа, которая удаляет всё форматирование из текста, скопированного в буфер обмена.
* CopyQ – благодаря поддержке скриптов можно реализовать практически любое автоматическое форматирование текста, помещенного в буфер обмена.

Анализ данных решений позволил сформировать следующий вывод – несмотря на предоставленные функциональные особенности, наиболее популярные решения либо не предоставляют возможностей форматирования скопированного текста, либо так или иначе требуют вмешательства пользователя.

## 1.3 Проектирование

## 1.3.1 UML диаграмма класса

Для наглядной визуализации структуры и логики будущего программного модуля была построена UML диаграмма класса.

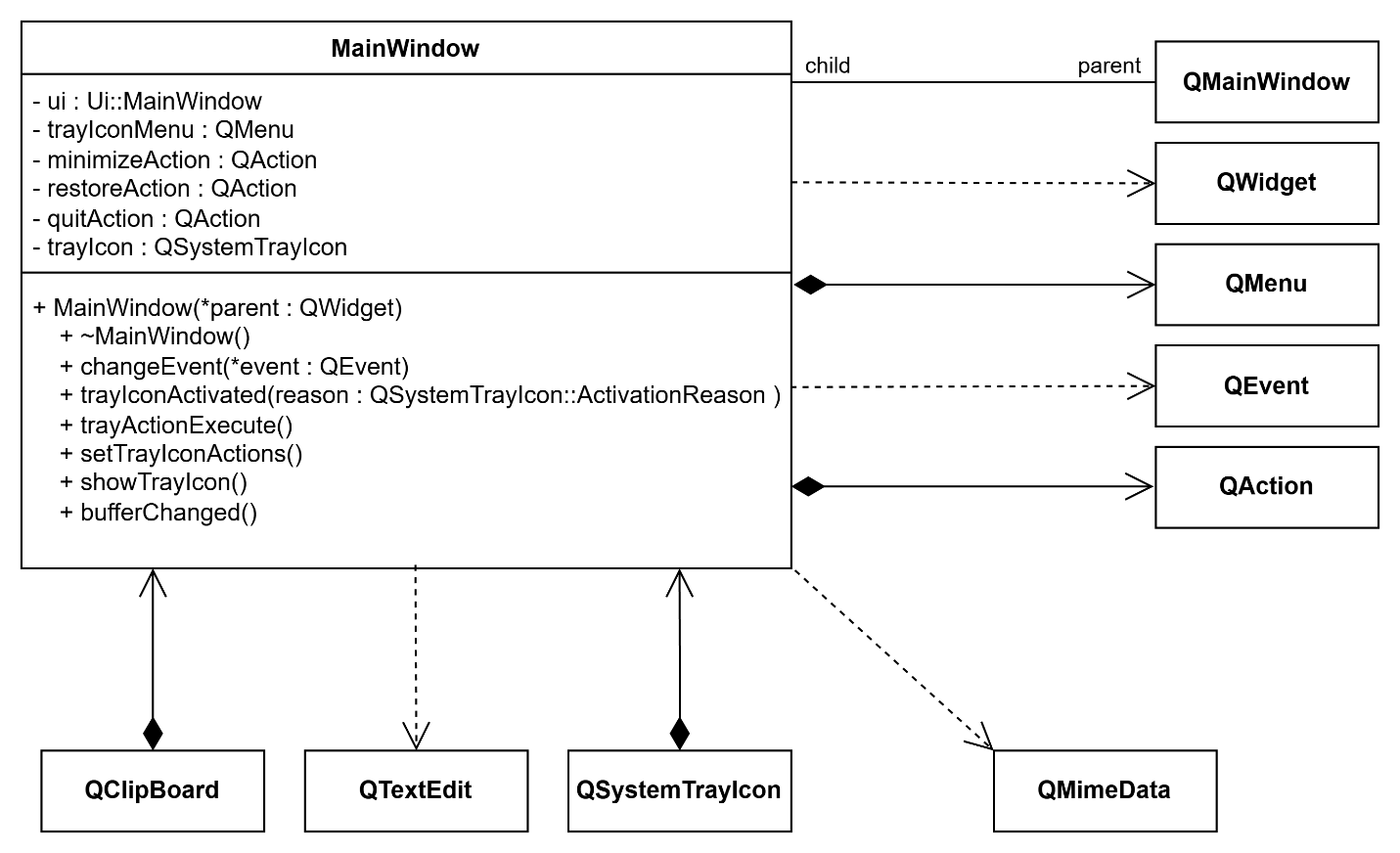


Рисунок 1 - Диаграмма класса.

Краткое описание. Для реализации фоновой работы программы были определены переменные:

* trayIconMenu – меню управления приложения в области уведомлений.
* MinimizeAction – событие сворачивания окна формы приложения в область уведомления.
* restoreAction – событие разворачивания окна формы приложения из области уведомления.
* quitAction – событие выхода из приложения через область уведомлений.
* trayIcon – иконка приложения в области уведомлений.

Для программной реализации необходимых функции были определены следующие методы:

* changeEvent – иконка приложения в области уведомлений.
* trayIconActivated – слот для отладки работы приложения в фоновом режиме.
* setTrayIcon – метод для настройки иконке приложения в области уведомлений.
* showTrayIcon – метод отображения иконки приложения в области уведомлений.
* bufferChanged – слот для работы с буфером обмена.

Также для реализаций необходимого функционала будут использованы следующие классы:

* QmainWindow – предоставляет основу для создания оконного приложения с меню, панелями инструментов и центральным виджетом.
* QWidget – базовый класс для всех объектов пользовательского интерфейса в Qt.
* QMenu – представляет выпадающее меню, которое можно прикрепить к окну или другому виджету.
* QEvent – базовый класс для всех событий в Qt, таких как нажатия кнопок или движения мыши.
* QAction – представляет действие, которое может быть выполнено пользователем, например, выбор пункта меню или нажатие кнопки панели инструментов.
* QClipboard – обеспечивает доступ к системному буферу обмена для копирования и вставки данных.
* QTextEdit – предоставляет виджет для редактирования и отображения текста с форматированием.
* QSystemTrayIcon – позволяет добавить значок в системную область уведомлений и обрабатывать взаимодействия с ним.
* QMimeData – используется для хранения данных в различных MIME-типах, для перетаскивания (drag and drop) или буфера обмена.

## 1.3.2 Требования к информационной и программной совместимости

Основным условием реализации визуальной и графической части программного модуля – является использование кроссплатформенных библиотек и фреймворков, поэтому целевыми операционными системами для программного модуля являются – Windows, Linux и MacOS.

Помимо этого, работа системы не должна мешать работе других сетевых приложений или компрометировать безопасность ПК пользователя.

## 1.4 Выбор и обоснование среды разработки

Исходя из поставленных задач, были выбраны следующие инструменты.

### 1.4.1 Язык программирования

Главным условием выполнение работы – использование экосистемы языка C++, исходя из этого, данный язык программирования будет основным при разработке.

### 1.4.2 Среда разработки

Основные критерии для выбора среды разработки:

* Кроссплатформенность, для поддержки работы на целевых операционных системах;
* Распространение на бесплатной основе;
* Поддержка разработки функциональных и графических программ на языке C++.

Qt обеспечивает кроссплатформенность, позволяя создавать приложения, которые могут быть развернуты на различных операционных системах с минимальными изменениями кода. Также, Qt предоставляет обширный набор инструментов и библиотек, включая готовые классы для работы с буфером обмена (QClipboard) и областью уведомления системы (QSystemTrayIcon), что значительно сокращает время разработки и снижает сложность кода. Qt основан на языке C++. Эти факторы делают Qt наиболее привлекательным выбором на фоне других фреймворков.

Для реализации необходимого функционала будут использованы следующая библиотека:

* Libxml2 внешняя библиотека для обработки XML. Предоставляет инструменты для **чтения XML-файлов, записи XML-файлов, обработки XML-данных.**

# Раздел 2 Разработка приложения

## 2.1 Запуск программы в области уведомлений системы

Для реализации работы программы в фоновом режиме, в основном потоке main() был убран вызов окна, созданного ранее класса MainWindow, сгенерированный конструктором.

После чего в реализации класса MainWindow были описаны функции настройки вариантов взаимодействия с иконкой приложения в области уведомлений и добавление значка в область уведомлений системы, с указанием необходимых свойств и подключением сигнала для возможности взаимодействия (Рисунок 2).

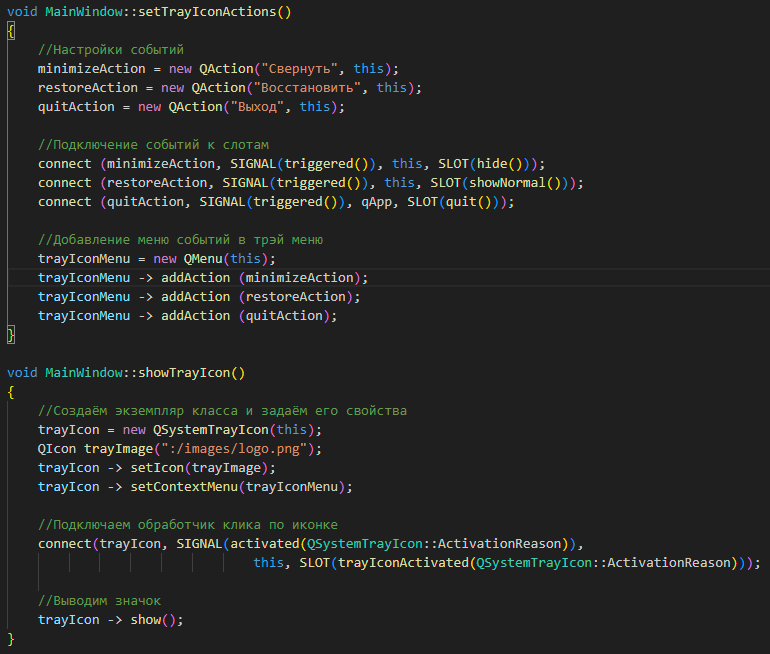


Рисунок 2 – Реализация иконки приложения в области уведомлений

Далее в конструктор класса MainWindow были добавлены вызовы для соответствующих функций, для отображения приложения в области уведомлений сразу после его запуска.

Также был определен слот активации иконки приложения в области уведомлений (Рисунок 3) (метод trayActionExecute() использовался для отладки).

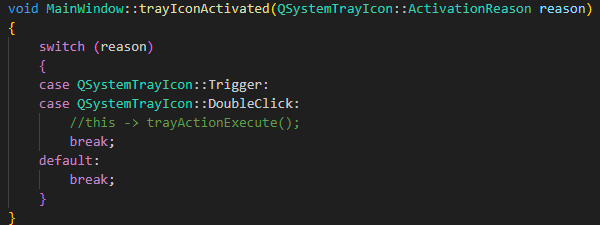


Рисунок 3 – Слот активации иконки приложения

Чтобы исключить сворачивание окна приложения в панель задач вместо области уведомлений был переопределено событие сворачивания окна приложения (Рисунок 4).

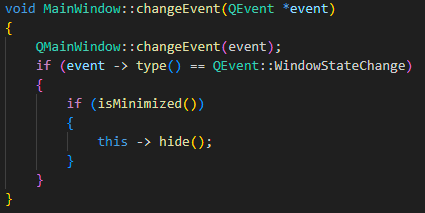


Рисунок 4 – Переопределение события сворачивания окна

## 2.2 Реализация работы с буфером

Для наиболее полной подготовки к работе с содержимым буфера была определена глобальная строковая переменная myString, которая будет хранить прошлое содержимое буфера обмена, для сравнения с новым содержимым, при его изменении. В конструкторе подключаем сигнал изменения содержимого буфера к слоту с указанием очередной логики работы, в слоте определяем конструкцию, которая исключает возможность возникновения рекурсии (Рисунок 5).

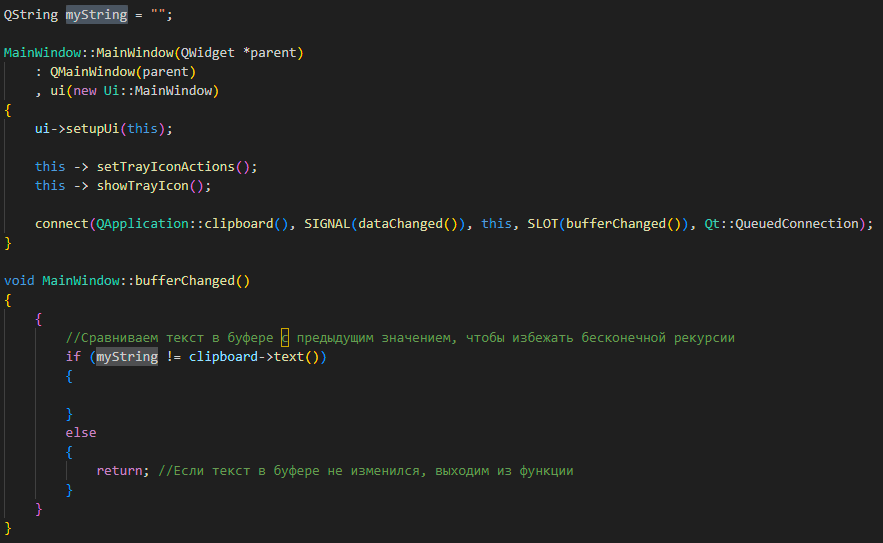


Рисунок 5 – Реализация работы с буфером

## 2.3 Защита от множественного запуска приложения

Ввиду непосредственного взаимодействия приложения с системным буфером обмена, необходимо было исключить возможность запуска нескольких экземпляров в рамках одной учетной записи, охранив при этом принцип кроссплатформенности. Такую возможность предоставляют разделяемая память и семафоры (Рисунок 6).

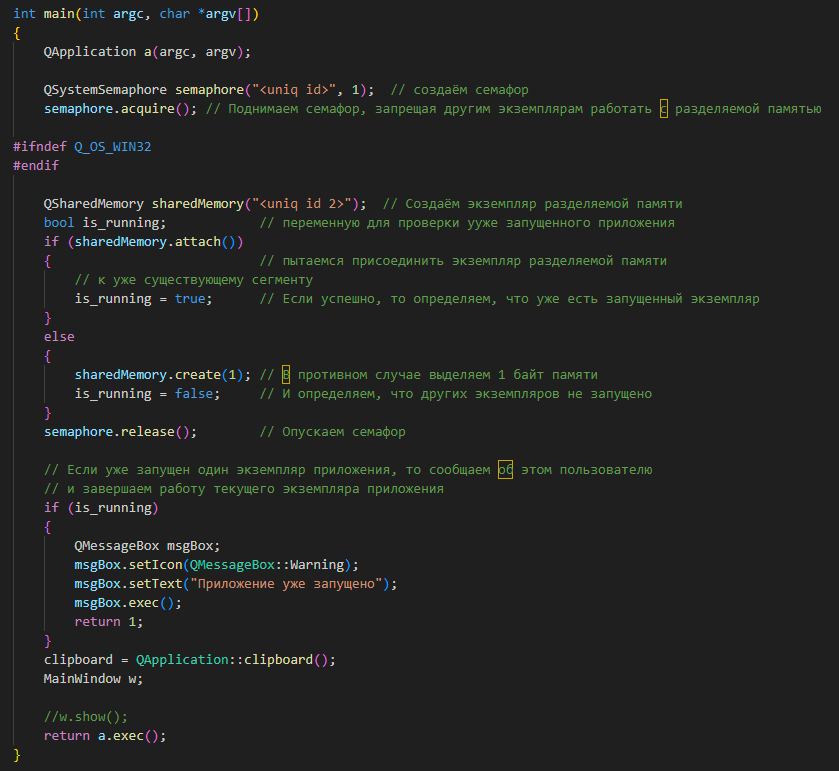


Рисунок 6 – Защита от повторного открытия окна

Логика представленного выше кода следующая, семафоры – призваны регулировать количество одновременно запущенных потоков, однако в случае одновременного выделения памяти для них (одновременного запуска двух экземпляров приложений) оба из них потерпят неудачу, в таком случае разделяемая память выступает гарантом однозначного показателя, что семафор действительно запущен, поэтому проверка проводится именно по разделяемой памяти, после проверки от семафора можно избавиться. В случае повторного запуска выводим сообщение с соответствующей информацией и завершаем работу дублирующего приложения.

## 2.4 Обработка текста

Содержимое буфера обмена проверяется на соответствие формату, если буфер содержит язык разметки гипертекста (HTML), текст обрабатывается с использование метода modifyHtmlFont (Рисунок 7), логика работы которой будет описана ниже.

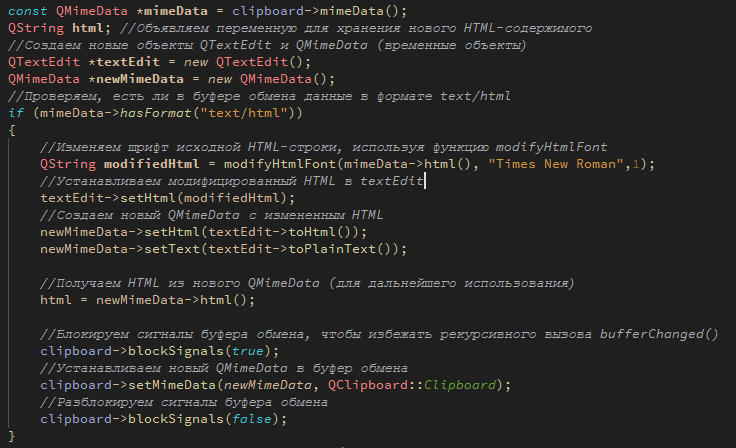


Рисунок 7 – считывание строки HTML формата

В случае содержания в буфере текста без какого-либо формата (PlainText) программа устанавливает для текста HTML теги для изменения шрифта текста (Рисунок 8). После изменённый текст возвращается в буфер.

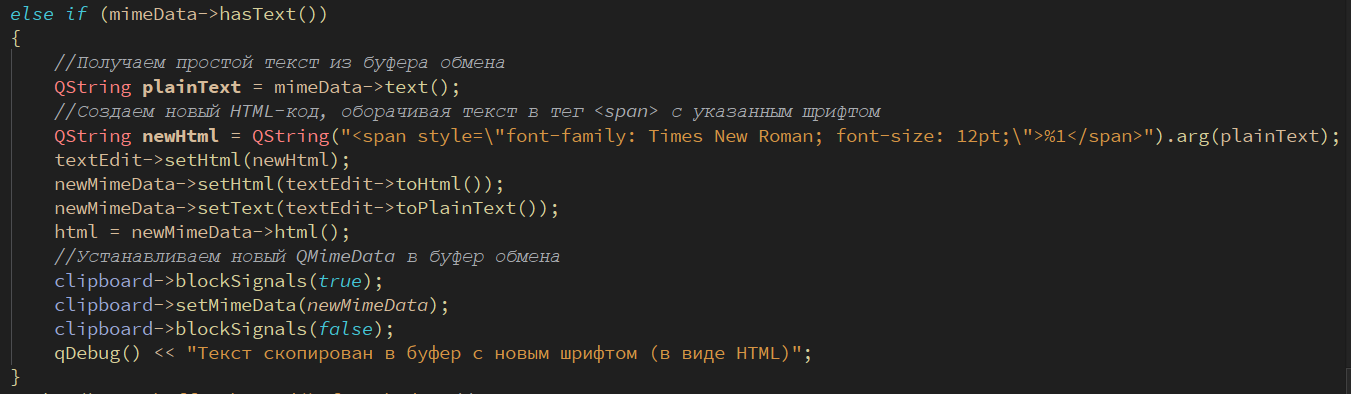


Рисунок 8 – считывание и обработка строки PlainText формата

Для обработки текста в программа используется технология парсинга – синтаксического анализа текста, реализуемая средствами библиотеки Libxml2. Логика программы следует следующему алгоритму: в первую очередь HTML строка считывается в HTML документ (Рисунок 9).

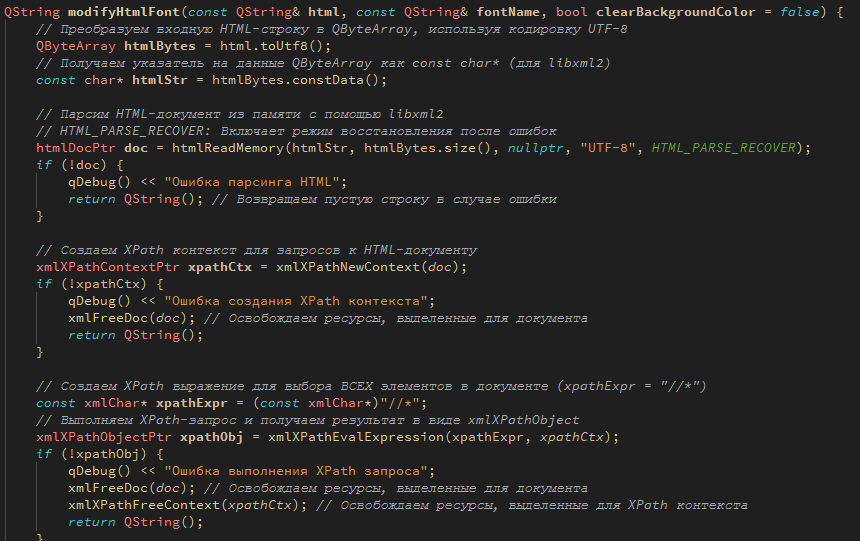


Рисунок 9 – Чтение HTML строки

После чего происходит сохранение исходного формата скопированной HTML строки (Рисунок 10).

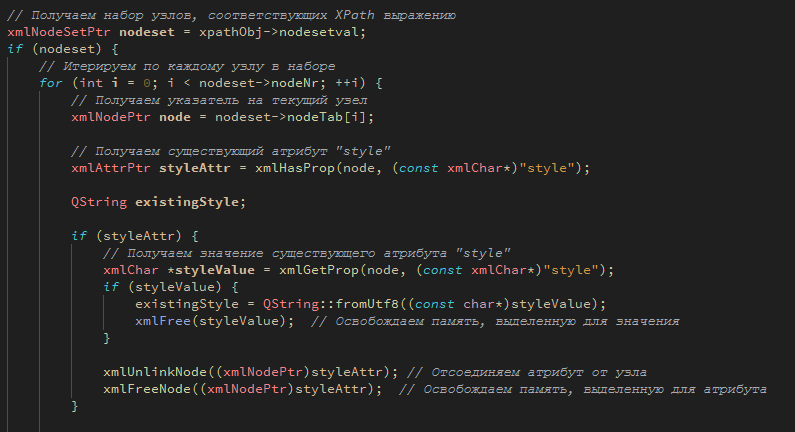


Рисунок 10 – Сохранение исходного формата текста

Далее производится удаление предустановленных шрифтов и установка необходимого шрифта (Рисунок 11).

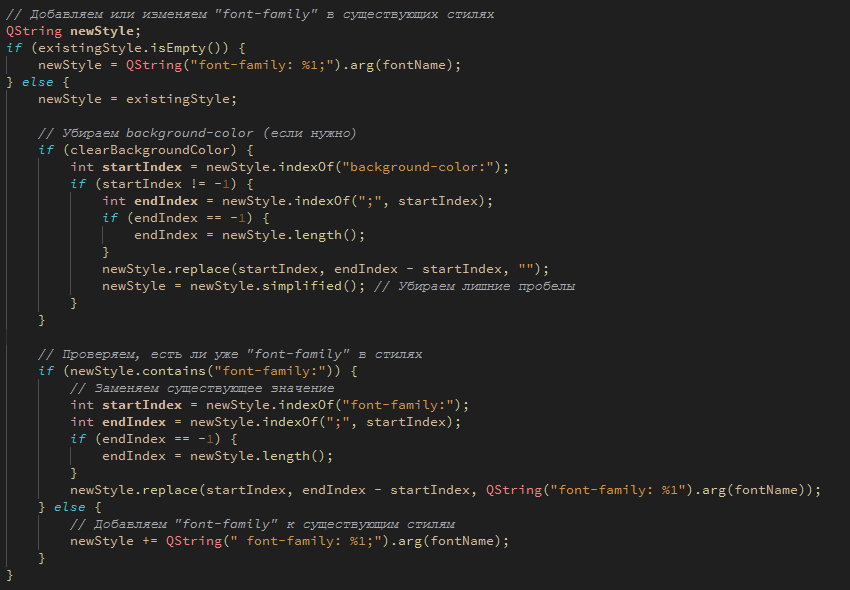


Рисунок 11 – Применение необходимого формата HTML к тексту

Заключительным этапом является обратная запись изменённого HTML документа в строку (Рисунок 12).

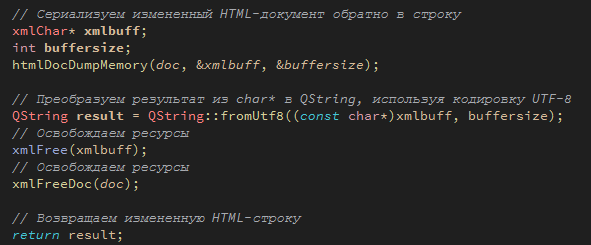


Рисунок 12 – Обратная запись измененной HTML строки

## 2.5 Создание формы

Работа формы основывается на использования QTextEdit виджета, в который добавляется строковая переменная, содержащая в себе измененный текст с HTML тегами, которые способствуют отображению текста с сохранением формата (Рисунок 13).



Рисунок 13 – запись текста в виджет

В последствии после настройки формы через графический конструктор фреймворка, при передаче обработанного текста элементу на форме, пользователь будет иметь доступ к истории скопированного текста (Рисунок 14).

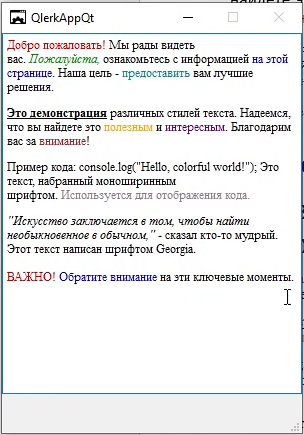


Рисунок 14 – форма истории буфера обмена

## 2.6 Тестирование системы

При разработке программного обеспечения применялось модульное (Unit) тестирование, при котором проверялись только что написанные функциональные блоки кода, изолированно от остальной части программы, что позволило протестировать весь код программы поэтапно непосредственно во время разработки.

# ЗАКЛЮЧЕНИЕ

В результате проделанной работы была достигнута основная цель – разработано автоматизированное рабочее место делопроизводителя путем создания программной решения, обладающего необходимым функционалом для решения, определенного ранее спектра задач.

Используемый фреймворк предоставляет много возможностей для масштабирования разработанного решения, при расширении функциональных возможностей уже реализованной обработки текста, потенциально можно будет добавить больше надстроек в интерфейс приложения для выбора пользователем таких свойств применяемого формата как – семейство шрифтов, кегель, отступы и др.

Также полезным дополнением к разработанному приложению будет создание установщика, и автоматическое добавление программы в автозагрузку системы, для исключения пользовательского участия в запуске программы.

Разработанное приложение позволяет экономить значительную часть времени при долгосрочном сборе информации в исследовательских целях, автоматически улучшая читаемость собранной информации. При написании объемных статей, рефератов, или других научных работ, срок составления которых может исчисляться годами, фоновая работа разработанного решения привнесет значительный вклад в эффективность работы пользователя.

В заключение, можно сделать вывод, что был получен практический и теоретический опыт в разработке программного обеспечения. Были освоены логика работы и новые инструменты взаимодействия с системным буфером обмена, форматами текста, возможности перевода графического интерфейса в фоновый режим, работа с семафорами и разделяемой памятью, на базе фреймворка QT.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32-2017. Отчёт о научно-исследовательской работе. Структура и правила оформления [Текст]. – Введ. 2017-01-07. – Минск: Издательство «Международный совет по стандартизации, метрологии и сертификации», 2017. – 16 с. – (Система стандартов по информации, библиотечному и издательскому делу)
2. ГОСТ РИСО/МЭК 12119-2000 «Информационная технология. Пакеты программ. Требования к качеству и тестирование
3. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств
4. ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование»
5. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ
6. Баженова, И.Ю. Языки программирования: Учебник / И.Ю. Баженова. - М.: Академия, 2018. - 448 c.
7. Баженова, И.Ю. Языки программирования: Учебник для студентов учреждений высш. проф. образования / И.Ю. Баженова; Под ред. В.А. Сухомлин. — М.: ИЦ Академия, 2018. — 368 c.
8. Гвоздева, В.А. Основы построения автоматизированных информационных систем [Текст] : учебник. - Москва : ИД "ФОРУМ" - ИНФРА-М, 2019. - 318 с. : ил.
9. Гвоздева, Т. В. Проектирование информационных систем. Стандартизация : учебное пособие для вузов / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 252 с. — Режим доступа: https://e.lanbook.com/book/169810, авторизованный
10. Гвоздева, Т. В. Проектирование информационных систем. Стандартизация : учебное пособие для вузов / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 252 с. — Режим доступа: https://e.lanbook.com/book/169810, 26 авторизованный
11. Гвоздева, Т. В. Проектирование информационных систем. Стандартизация : учебное пособие для вузов / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 252 с. — Режим доступа: https://e.lanbook.com/book/169810, авторизованный
12. Довек, Ж. Введение в теорию языков программирования / Ж. Довек, Ж.-Ж. Леви. — М.: ДМК, 2019. — 134 c.
13. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — М. : Издательство Юрайт, 2019. — 155 с.
14. Иванов, В. М. Интеллектуальные системы : учеб. пособие для СПО / В. М. Иванов ; под науч. ред. А. Н. Сесекина. — М. : Издательство Юрайт, 2019. — 93 с.
15. Кубенский, А. А. Функциональное программирование : учебник и практикум для академического бакалавриата / А. А. Кубенский. — М. : Издательство Юрайт, 2019. — 348 с.
16. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 432 с.
17. Мамонова, Т. Е. Информационные технологии. Лабораторный практикум : учеб. пособие для СПО / Т. Е. Мамонова. — М. : Издательство Юрайт, 2019. — 178 с.
18. Попова, Ю. Б. Тестирование и отладка программного обеспечения: пособие / Ю. Б. Попова. – Минск: БНТУ, 2020. – 66 с
19. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общей редакцией Д. В. Чистова. — Москва : Издательство Юрайт, 2020. — 258 с. — (Профессиональное образование). — 18В1Ч 978-5-534-03173-7. — Текст : электронный // ЭБС Юрайт [сайт]. <https://urait.ru/bcode/452680> (Дата обращения 20.03.2023)