Название проекта: Stiify

Автор: Лазимов Даниил

**Идея**

Мой проект заключается в том, чтобы сделать декстопную читалку с перемещением книг по категориям (домашняя страница, любимые, хочу прочитать, прочитанные), возможностью открывать txt, fb2 книги, и запоминанием ранее добавленных книг(БД), возможностью изменять шрифт, размер, выравнивание.

**Реализация**

В ходе реализации своего проекта я использовал основные классы: Sys, SQLite3, uic, QtGui , QFont, QtWidgets QApplication, QMainWindow, QWidget, QListWidgetItem, QListWidget, PushButton, QFileDialog, QMessageBox. Использовал qss для дизайна проекта, так же использовал встроенный класс QXml из QtCore для открытия книги формата fb2. В ходе проекта я сделал кастомный виджет который содержит в себе название книги или файла, там же есть кнопки для перемещения по категориям, удалением из читалки и кнопкой для открытия самой книги, использовал pyqtSignal для связи каждого кастомного виджета с его прописанным функционалом.

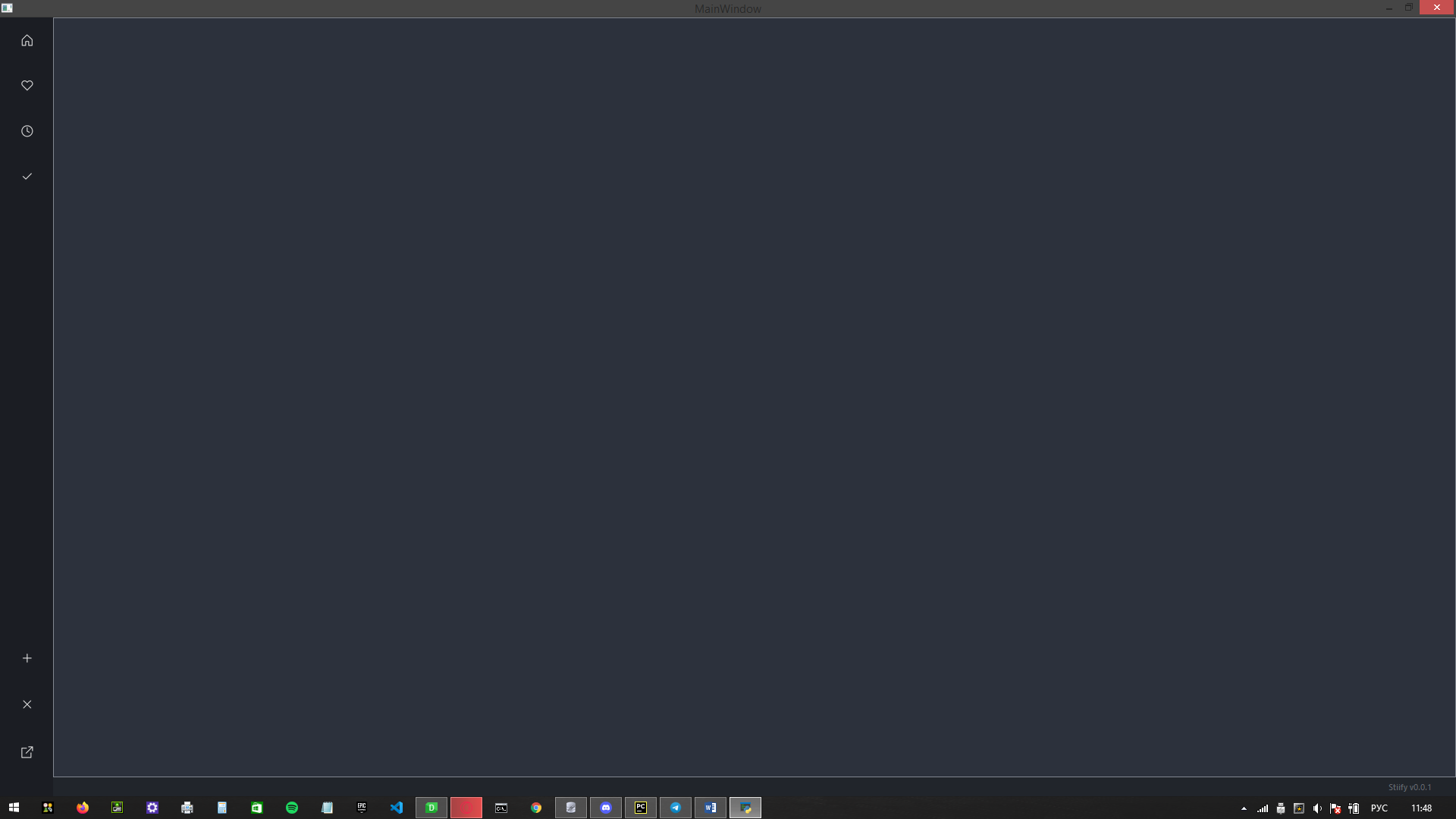
**Описание технологий**

Файл fb2 – частный случай xml. Основным элементом его структуры, как и для html, является тэг. xml в отличии от html не использует язык CSS, в нашем случае это значит то, что в fb2 файле нет указаний на то, как отформатирован текст. QXmlStreamWriter нужнty для записи сформированного документа. Класс QXmlStreamReader производит несколько важных действий. Во-первых, определяет и устанавливает нужный декодер. Во-вторых, отделяет тэги от содержимого. В-третьих, выделяет свойства тэгов. Нам остаётся только обрабатывать разделённые данные. Для считывания данных используется функция readNext(). Все фрагменты, прочитанные ей относятся к одному из 5 типов: StartDocument, EndDocument, StartElement, EndElement и Characters. Из них 2 первых для определения начала и конца файла, 2 следующих для считывания тэгов и последний получения заполнителя. Появление StartElement значит, что прочитан открывающийся или строчный тэг. Соответственно EndElement сигнализирует о считывании закрывающегося тэга. Для контроля структуры документа будем хранить список всех открытых тэгов в списке. Поэтому в случае StartElement добавляет к token имя текущего тэга и удалять его в случае EndElement. Кроме того, открывающиеся (или строчные) тэги могут содержать атрибуты. Атрибуту будут считаны и сохранены в sr как массив строк. Доступ к ним можно получить с помощью метода sr.attributes(). Нам они понадобятся для добавления рисунков в текст. Так при обнаружении тэга image необходимо добавить в текст метку этого рисунка. Так как мы записывали свой xml в байтах, то мы декодируем и получаем xml строение и с помощью встроенного метода переводим в текст и выводим. Каждый раз когда человек добавляет файл, я записываю название, путь, категорию и id в БД, при последующем запуске я обращаюсь к БД и создаю виджеты, соответствующие ранее добавленным книгам, в тех категориях которые были.

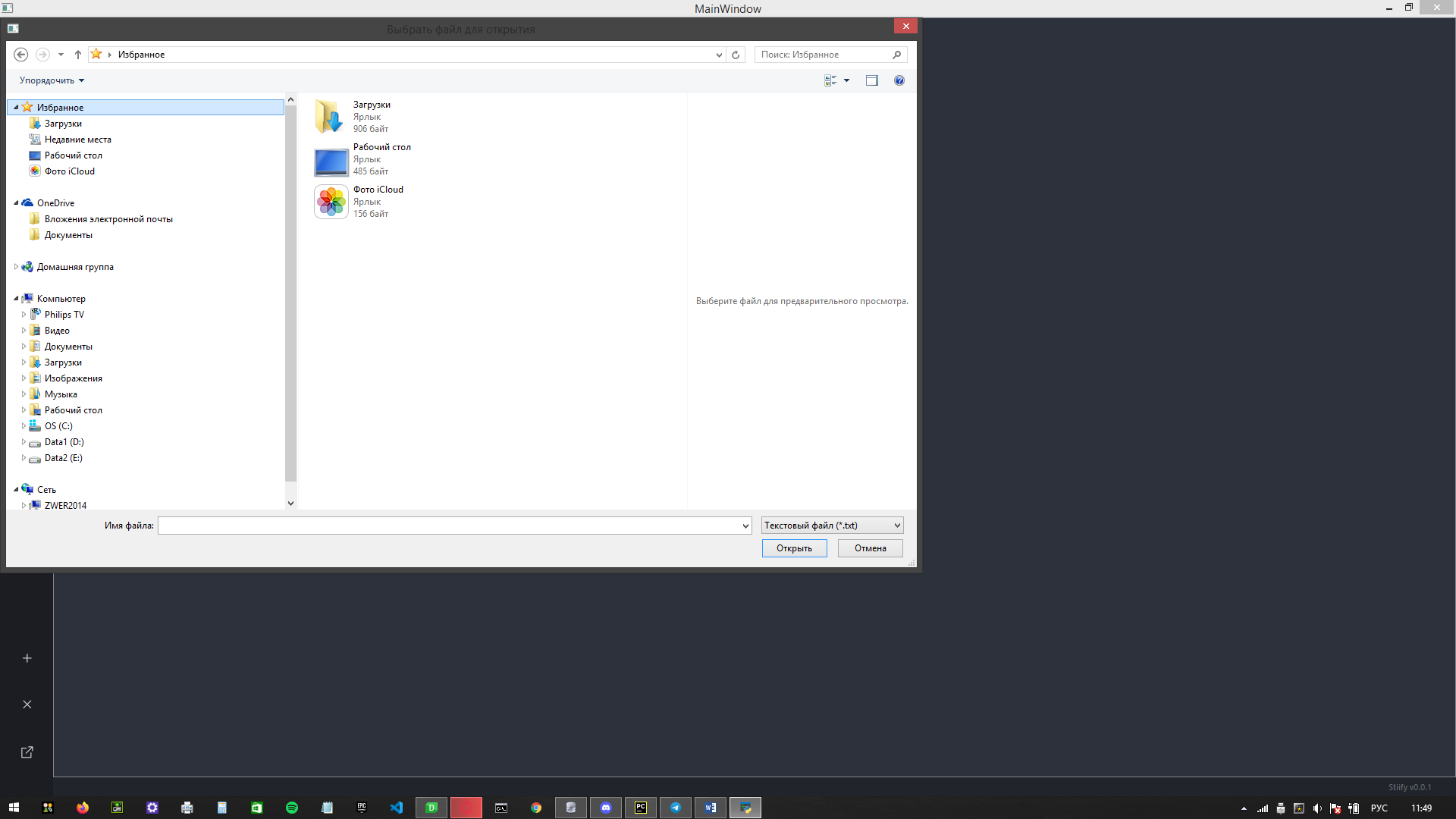
Каждый раз для чтения я создаю окно и отслеживаю для выделенного текста изменения шрифтов и размера, и выравнивание, соответственно динамично изменяю.

Необходимые библиотеки

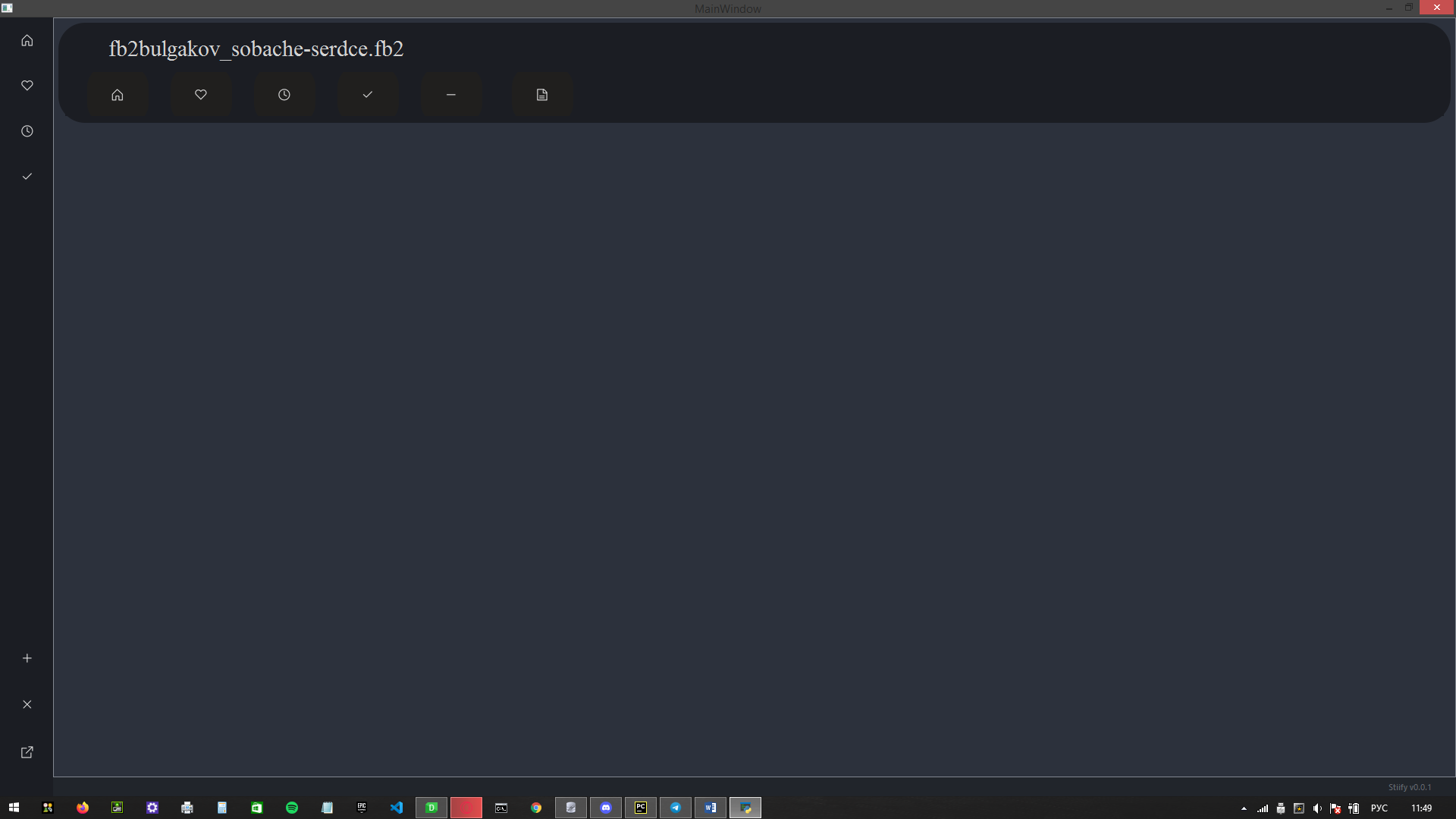
PyQt5==5.15.5  
PyQt5-Qt5==5.15.2  
PyQt5-sip==12.9.0  
PyQt5-stubs==5.15.2.0

Основной интерфейс. 

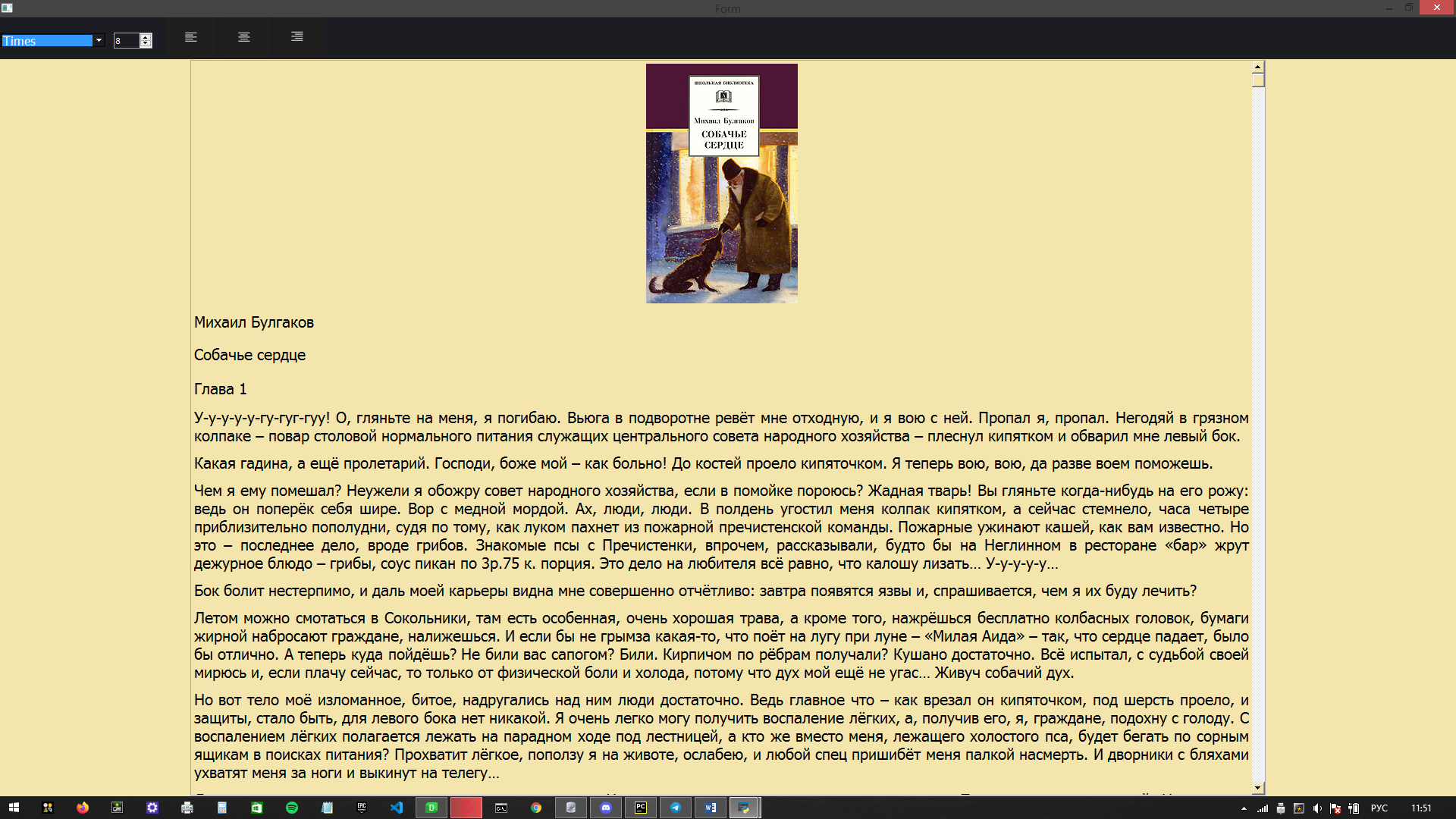
Диалоговое окно



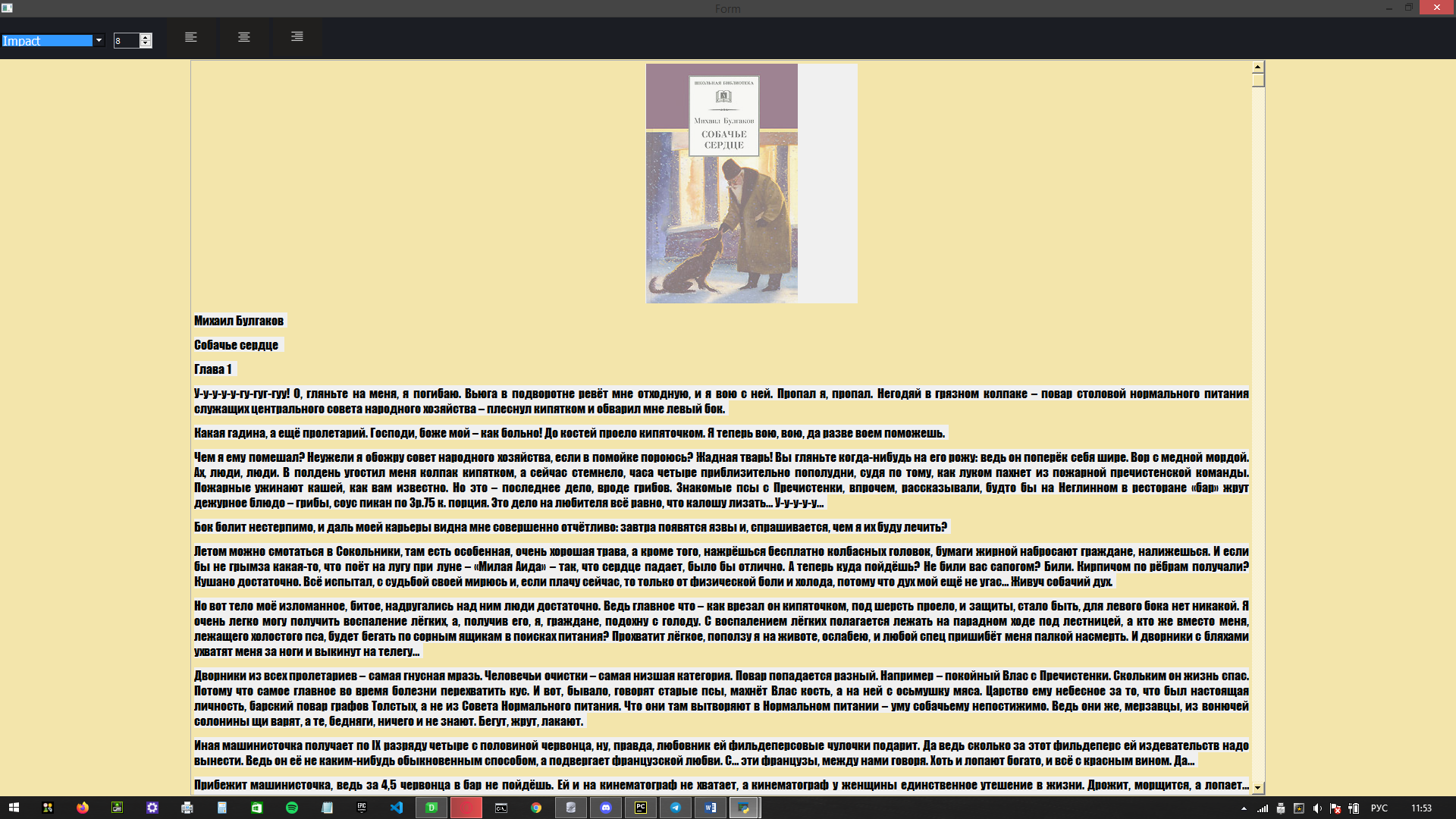
Добавление виджета



Открытие окна по нажатию на кнопку



Изменение шрифта и размера





Выравнивание

