#### Министерство науки и высшего образования Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение

#### высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №1

по дисциплине «Алгоритмы и структуры данных»

Выполнил:

студент гр. ЗИСТуд-121

Е.К. Аюбджанов

Приняла:

О.Н. Шамышева

Владимир, 2021 г.

**Задания**

Задание 1.

1.1 При нажатии на кнопку 1 выделенный текст в компоненте TRichEdit меняет свой цвет и прописывается в TLabel.

1.2 При нажатии на CheckBox-ы соответствующие компоненты становятся невидимыми. По умолчанию CheckBox-ы отмечены и соответствующие компоненты видимы.

1.3 При нажатии на RadioButton-ы меняется выравнивание текста в компоненте TMemo.

Задание 2.

2.1 При нажатии на кнопку “Загрузить” в TMemo загружается текст из текстового файла (.txt).

2.2 При нажатии на кнопку “Cохранить” из TMemo текст сохраняется в этот-же тектовый файл .

2.3 При нажатии на кнопку “Очистить все” компоненты TEdit, TLabel, TMemo, TRichEdit – очищаются.

2.4 При нажатии на кнопку “Разрешить” – все компоненты становятся доступны на редактирование.

2.5 При нажатии на кнопку “Запретить” - все компоненты становятся недоступны для редактирования.

Задание 3.

При нажатии на кнопку 3 весь текст из TMemo копируется в TRichEdit, причем каждое предложение располагается с новой строки.

• С такого знака должна начинаться каждая строка в RichEdit-е.

**Выполнение**

Выполнение заданий было реализовано с использованием языка C#.

Выполнение первой задачи

Для начала был создан проект Windows Forms (рисунок 1).

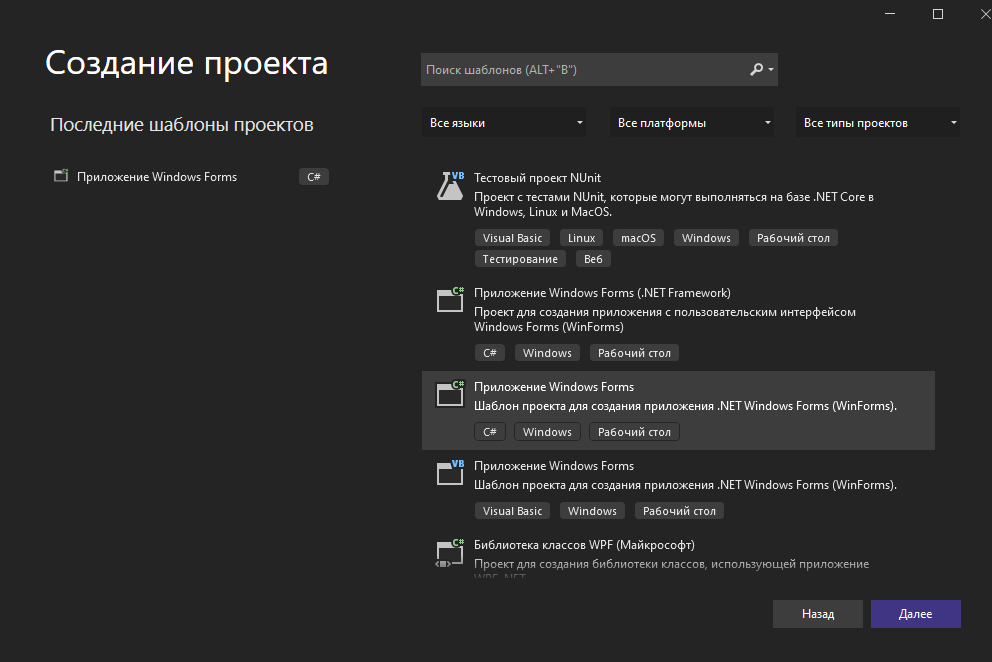


Рисунок 1 – Создание приложения

Далее в макете приложения были расположены и добавлены элементы согласно заданию (рисунок 2).

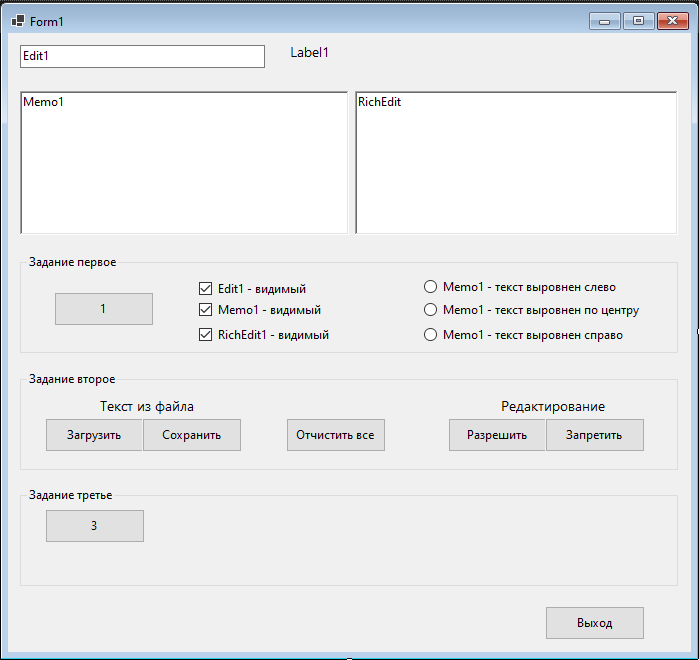


Рисунок 2 – Макет приложения

Следующим шагом было создание двух обработчиков:

1. Обработчик выделения текста.
2. Обработчик кнопки.

На рисунке ниже представлен фрагмент кода, содержащий логику работы обработчиков.

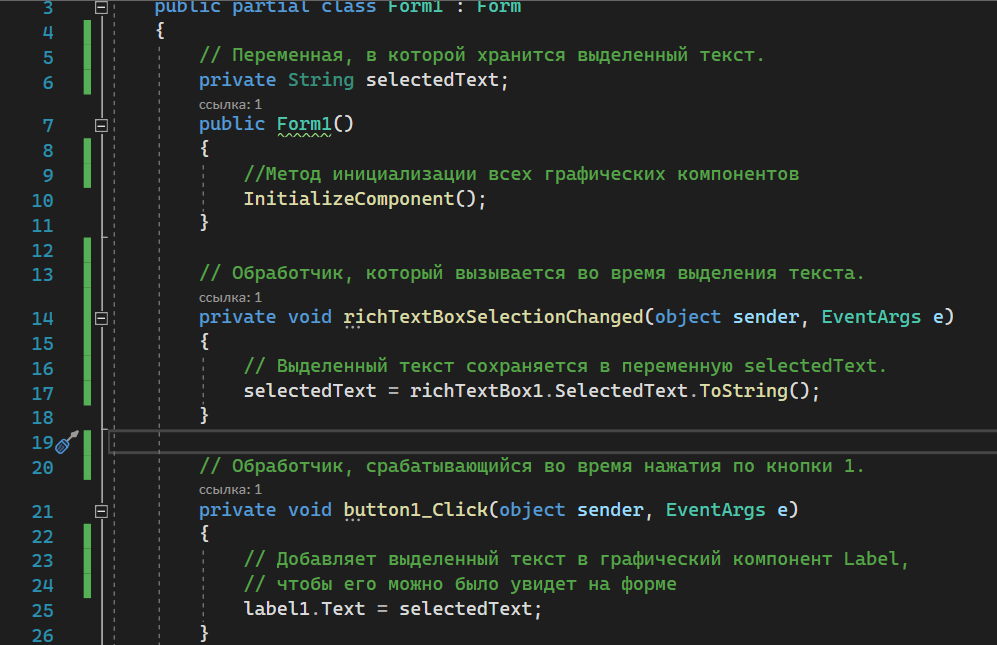


Рисунок 3 – Обработчик кнопки и выделения текста

Ниже приводится демонстрация работы обработчиков.

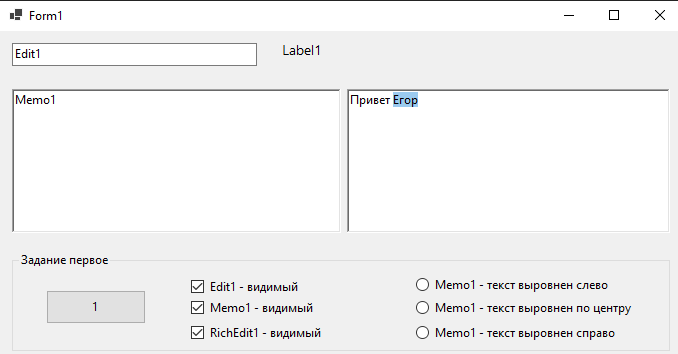


Рисунок 4 – Состояние формы до нажатия первой кнопки

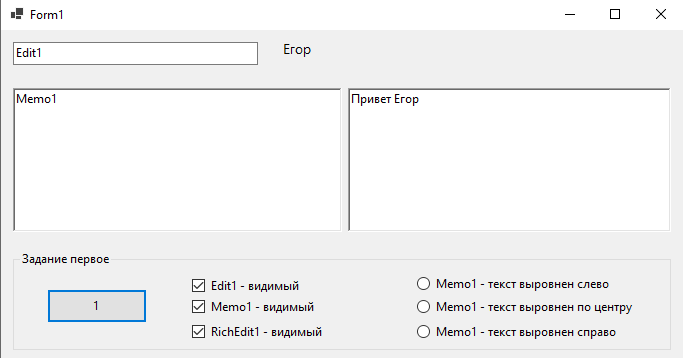


Рисунок 5 – Состояние формы после нажатия первой кнопки

Как видно из рисунков выделенный фрагмент текста появился.

Следующим шагом было добавление обработчиков для компонентов CheckBox. Ниже представлен фрагмент кода, отвечающий за логику видимости компонентов.

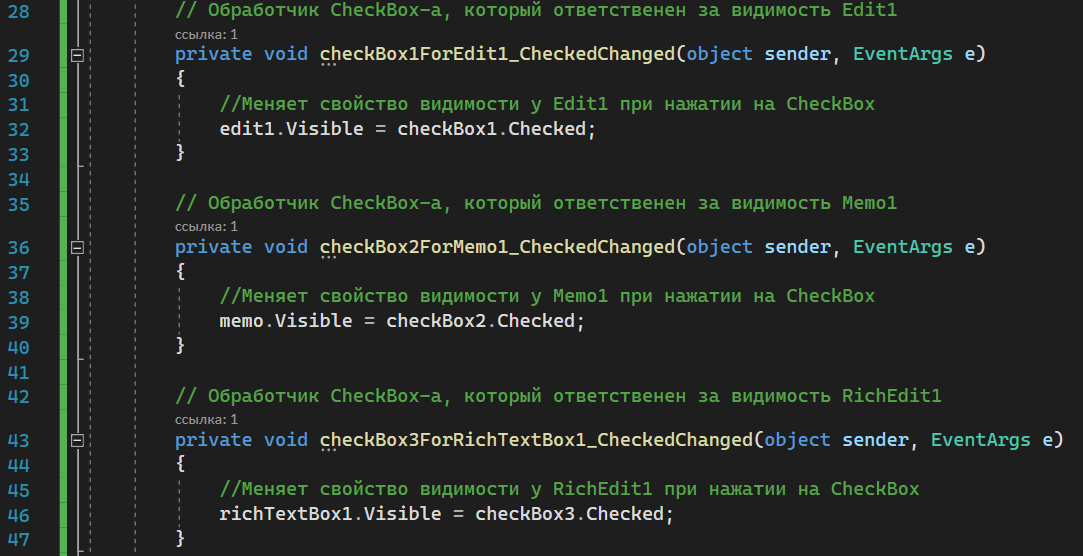


Рисунок 6 – Обработчики компонентов CheckBox

Ниже приводится демонстрация работы обработчиков.

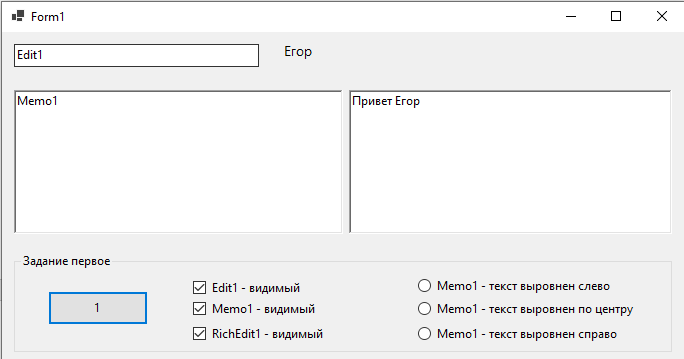


Рисунок 7 – Состояние формы до снятия галочек с CheckBox-ов



Рисунок 8 – Состояние формы после снятия галочек с CheckBox-ов

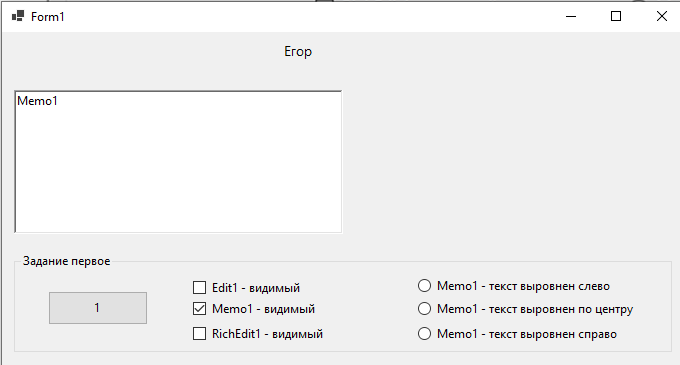


Рисунок 9 – Состояние формы после указания одной галочки в CheckBox-е

Как видно из рисунков логика исчезновения компонентов работает.

Следующим шагом было добавление логики выравнивания текста для компонента TMemo.

Ниже представлен фрагмент кода, демонстрирующих логику обработчиков для компонента RadioButton.

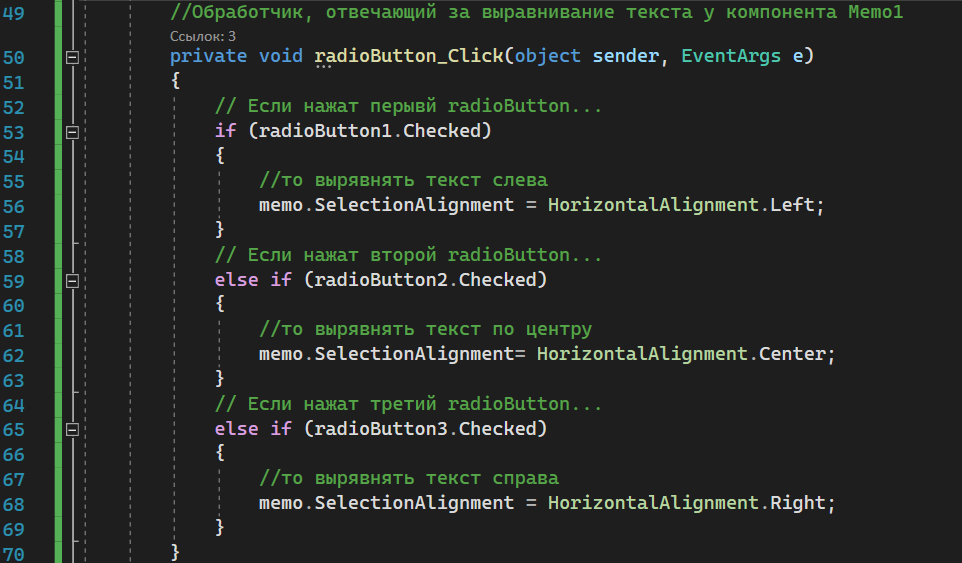


Рисунок 10 – Обработчики компонентов RadioButton

Ниже приводится демонстрация работы обработчика.

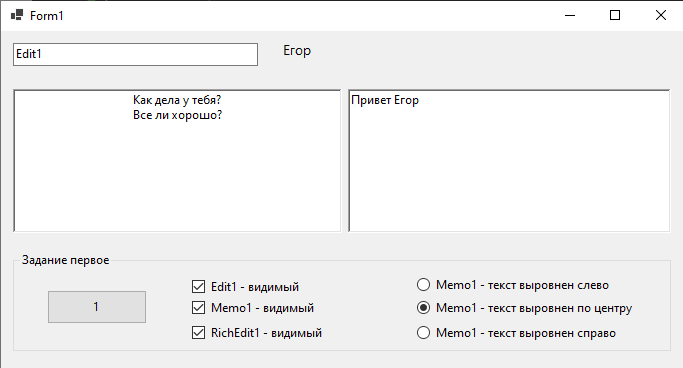


Рисунок 11 – Состояние формы после выравнивания текста по центру

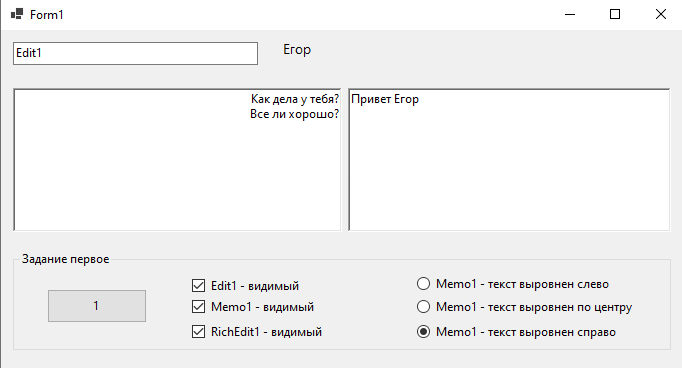


Рисунок 12 – Состояние формы после выравнивания текста справо

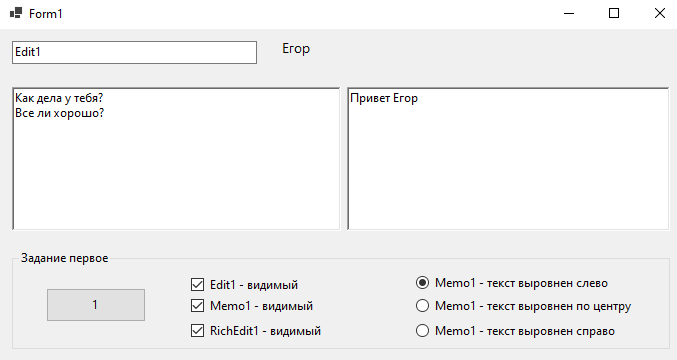


Рисунок 13 – Состояние формы после выравнивания текста слево

Как видно из рисунков логика выравнивания текста работает.

Выполнение второй задачи

Для начала был добавлен обработчик для кнопки «Загрузить». Фрагмент кода обработчика представлен ниже.

Файловый менеджер, с помощью которого осуществлялся поиск нужного файла, был добавлен декларативно через конструктора форм.

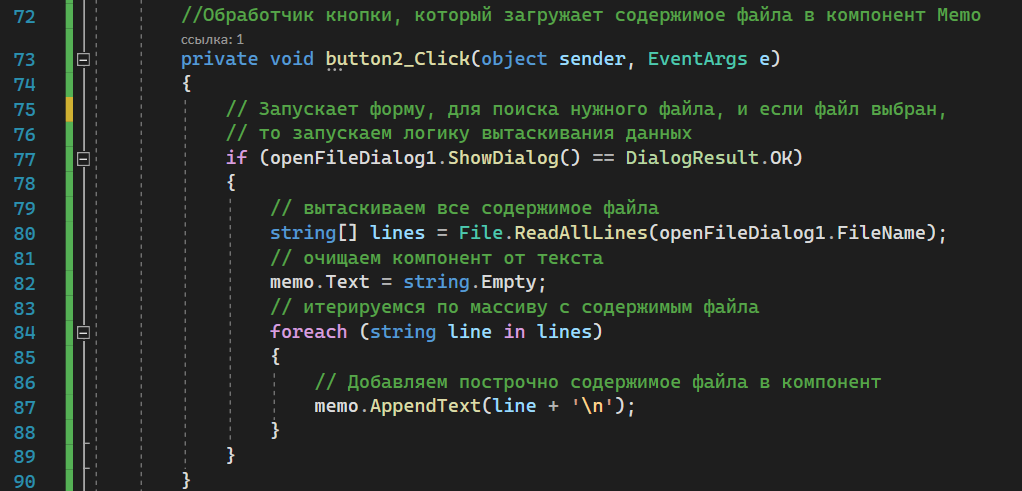


Рисунок 10 – Обработчик кнопки «Загрузить»

Далее был создан файл «text.txt» с содержимым, представленным на рисунке ниже.

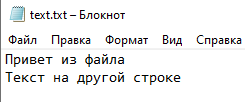


Рисунок 11 – Содержимое файла

Ниже приводится демонстрация работы обработчика.

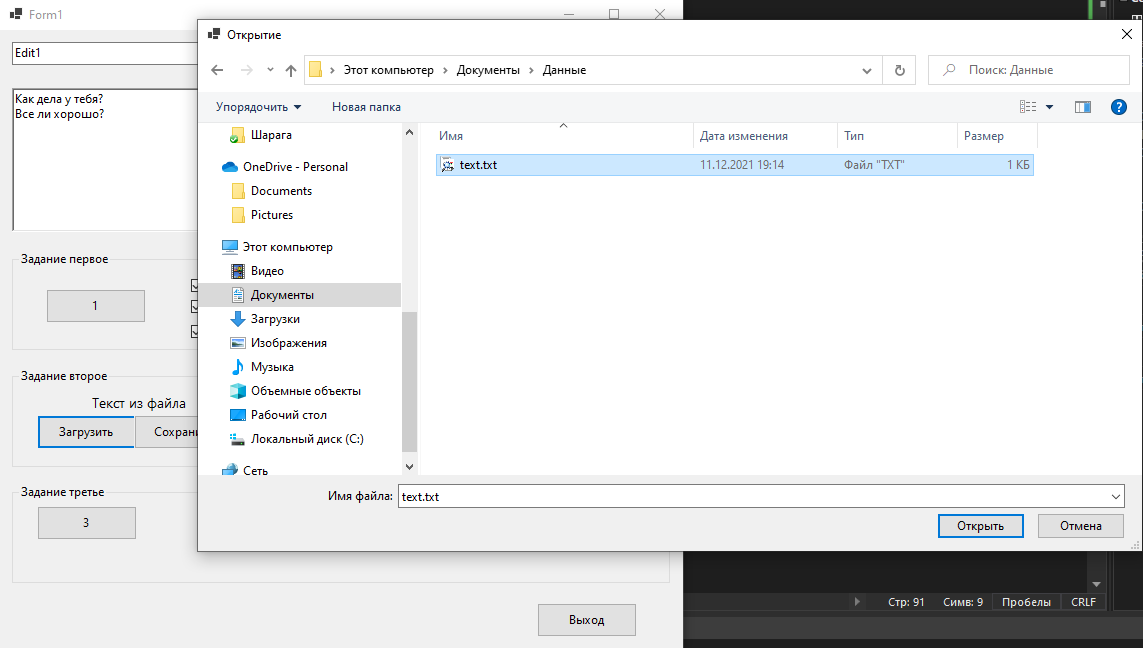


Рисунок 12 – Поиск файла

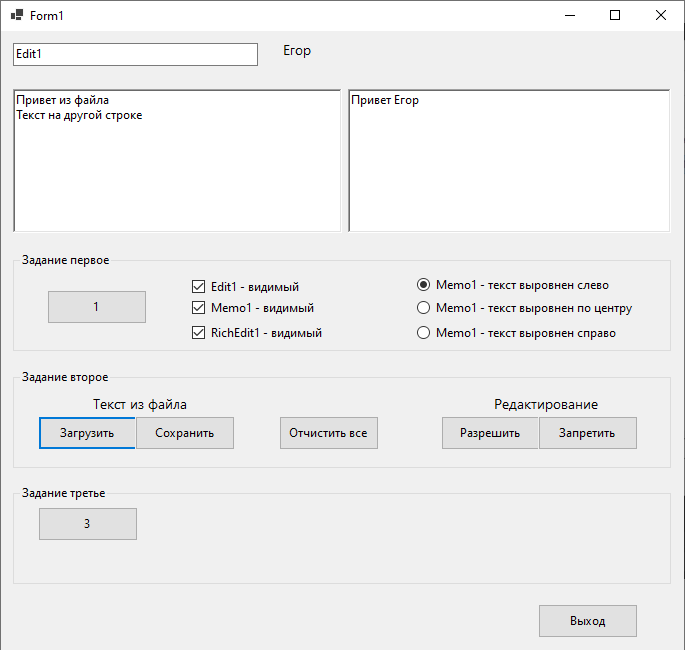


Рисунок 13 – Форма, демонстрирующее содержимое файла

Как видно из рисунков логика загрузки содержимого файла успешно работает.

Следующим шагом было создание обработчика для кнопки «Сохранить», который сохраняет текст из TMemo в файл.

Фрагмент кода обработчика представлен ниже.

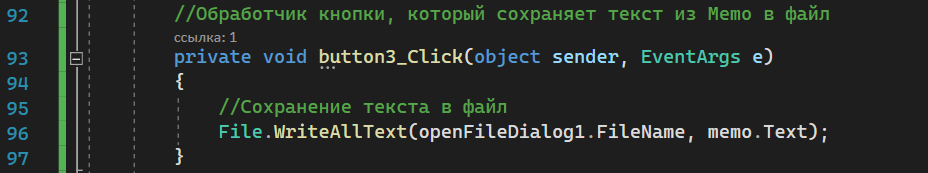


Рисунок 14 – Обработчик кнопки «Сохранить»

Ниже приводится демонстрация работы обработчика.

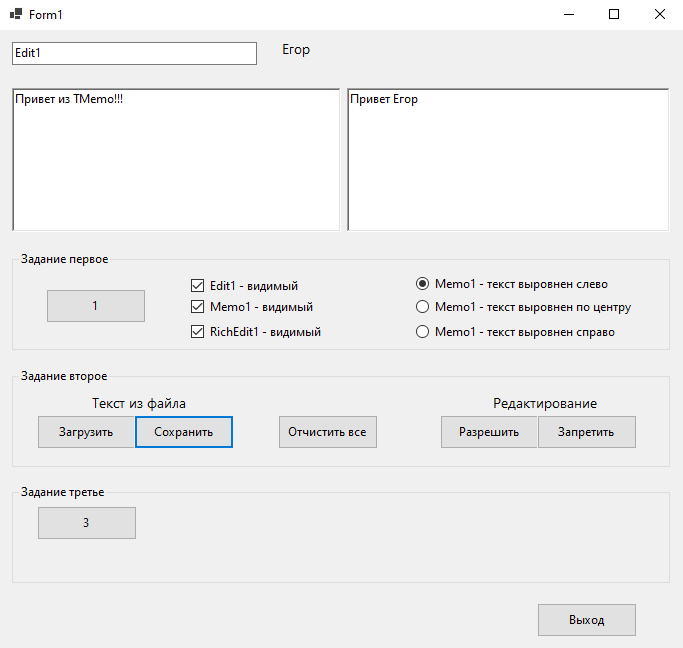


Рисунок 15 – Форма с текстом в TMemo

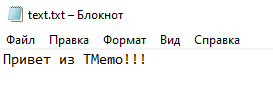


Рисунок 16 – Содержимое файла

Как видно из рисунков логика сохранения текста из TMemo в файл успешно работает.

Следующим шагом было создание обработчика для кнопки «Отчистить все», логика которого убирала весь текст с основных компонентов.

Фрагмент кода обработчиков представлен ниже.

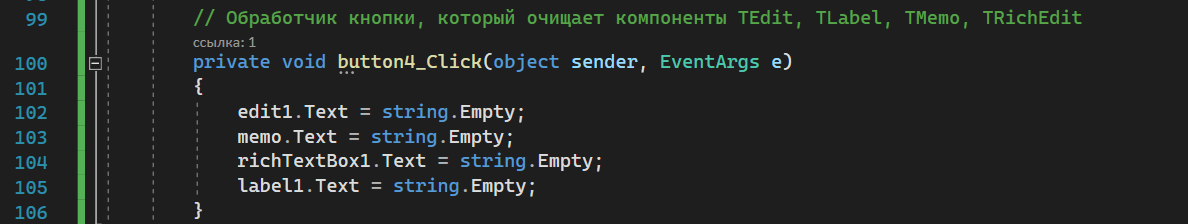


Рисунок 14 – Обработчик кнопки «Отчистить все»

Ниже приводится демонстрация работы обработчика.

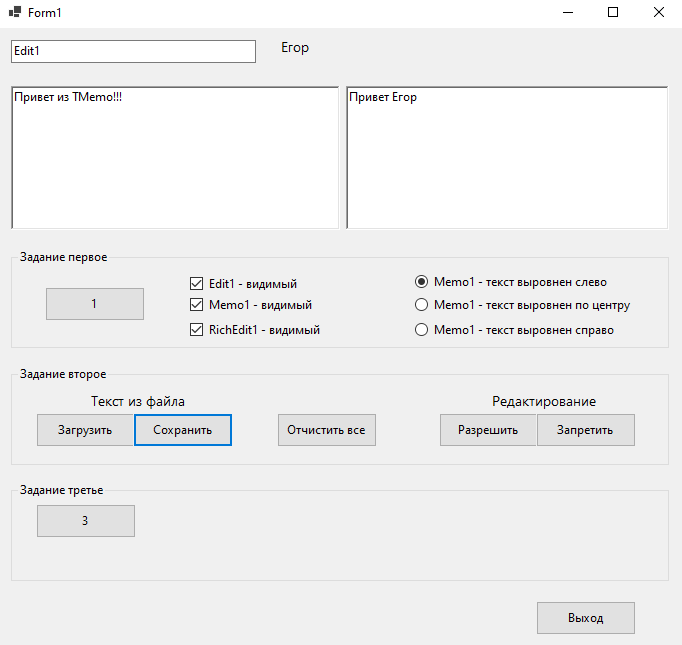


Рисунок 15 – Состояние формы до отчистки

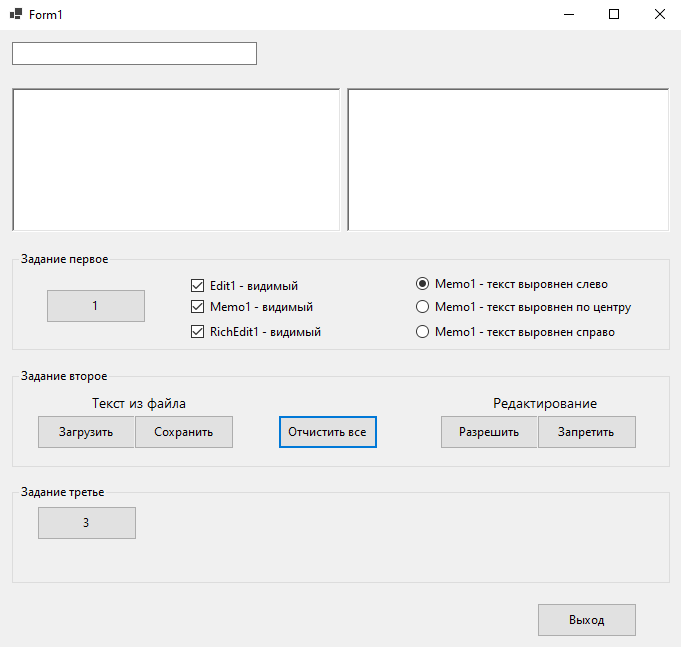


Рисунок 16 – Состояние формы после отчистки

Как видно из рисунков логика отчистки содержимого компонентов успешно работает.

Следующим шагом было создание обработчиков для кнопок «Разрешить» и «Запретить», логика которого регулировала редактирование содержимого компонентов.

Ниже представлен фрагмент кода, отвечающий за логику обработчиков.

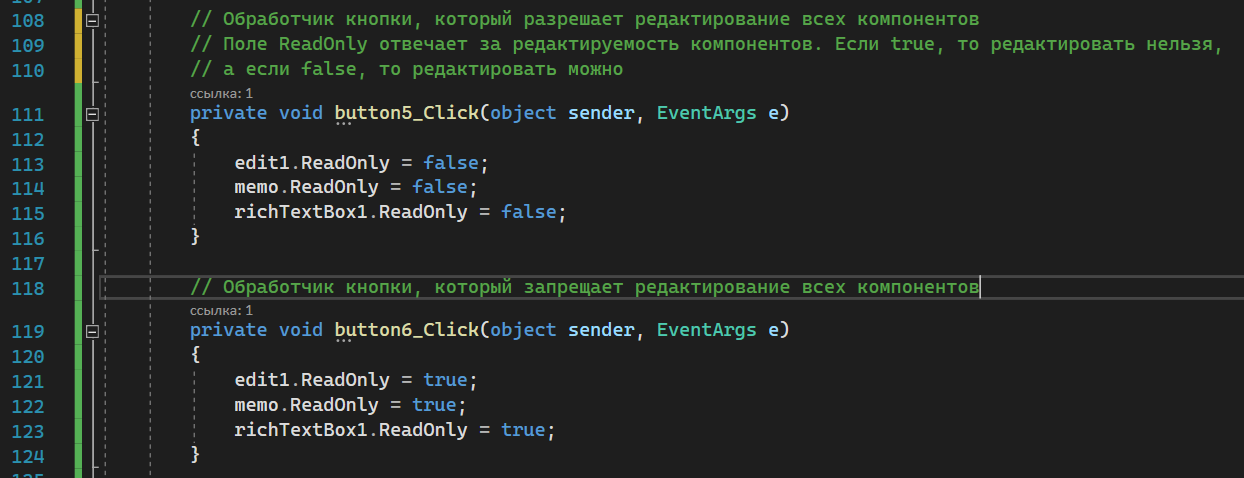


Рисунок 17 – Обработчики кнопок «Разрешить» и «Запретить»

Ниже приводится демонстрация работы обработчика.

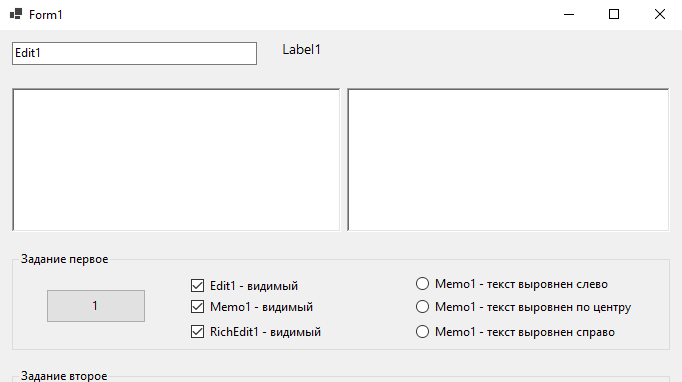


Рисунок 18 – Состояние формы до запрета редактирования

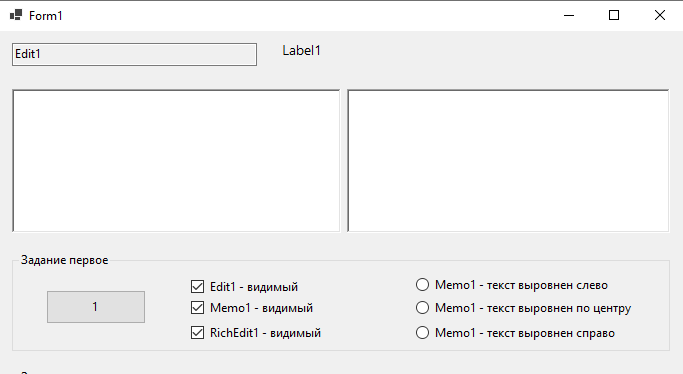


Рисунок 19 – Состояние формы после запрета редактирования

Из рисунков видно, что компонент TEdit потемнел. Это значит, что текст, содержащийся в этом компоненте, нельзя редактировать. Аналогичная ситуация в других компонентах несмотря на то, что визуально они не изменились.

Выполнение третьей задачи

Следующим шагом было создание обработчика для третьей кнопки, логика которого декомпозировала текст из TMemo по предложениям и выводила их в компонент TRichEdit.

Ниже представлен фрагмент кода, отвечающий за логику обработчика.

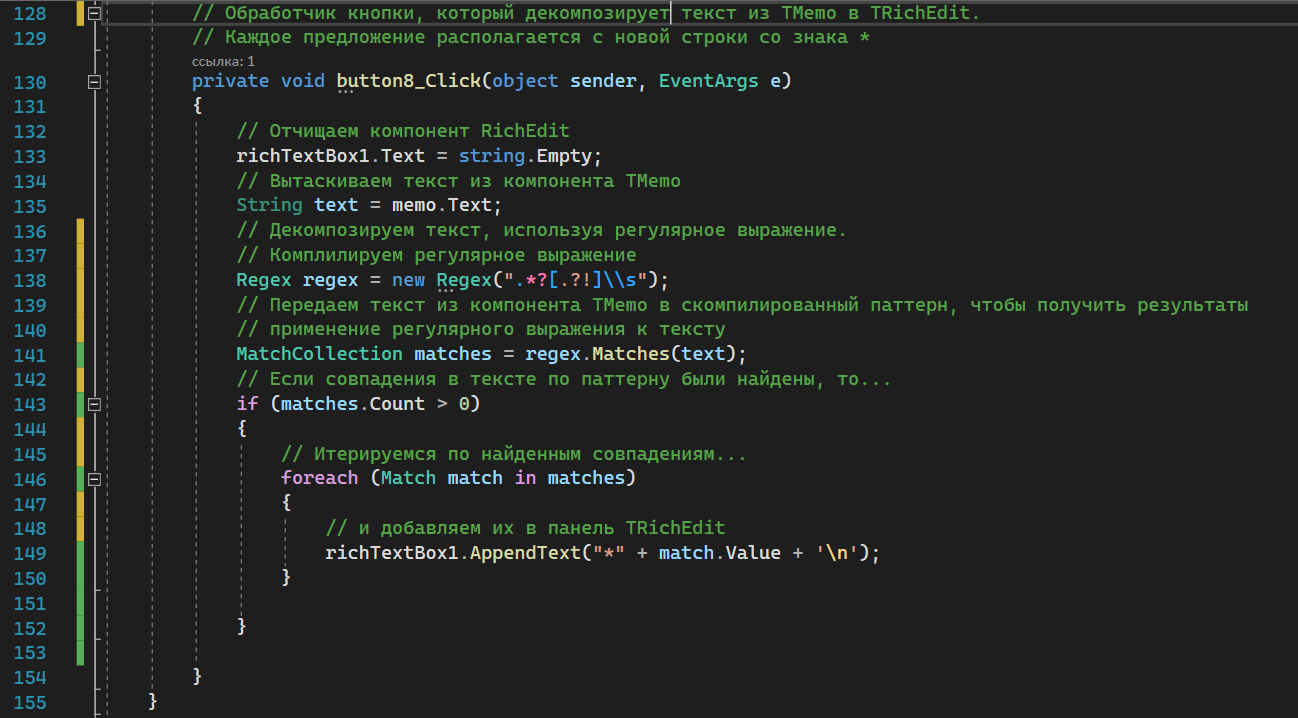


Рисунок 20 – Обработчик третьей кнопки

Отдельно хочется выделить сам паттерн регулярного выражения. Он читается так: ноль или более раз символов в начале текста, затем должен встретится один из символов конца предложения («.», «?», «!»), а после него символ пробела. Данная запись позволит найти все предлолжения в тексте. К тому же режим квантифактора – ленивый. Это нужно для нахождения максимального числа совпадений.

Ниже демонстируется работа обработчика третьей кнопки.

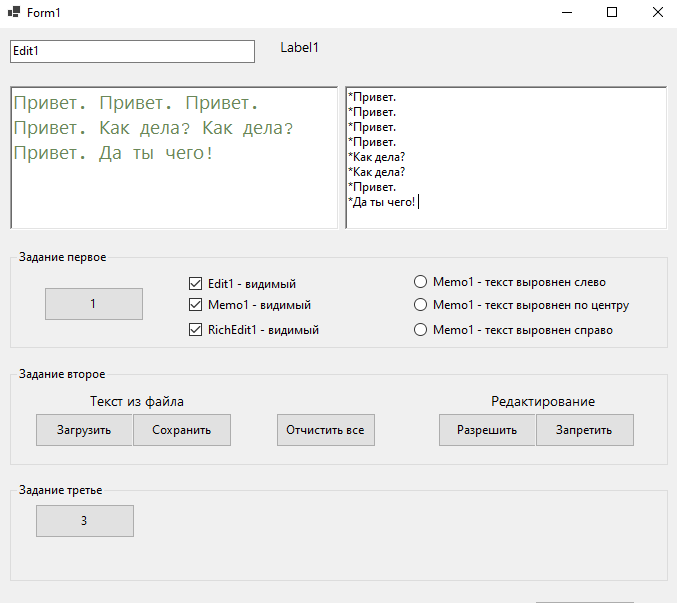


Рисунок 21 – Обработчик третьей кнопки

Как видно из рисунков логика декомпозиции текста работает успешно.

Ниже представлена блок-схема.

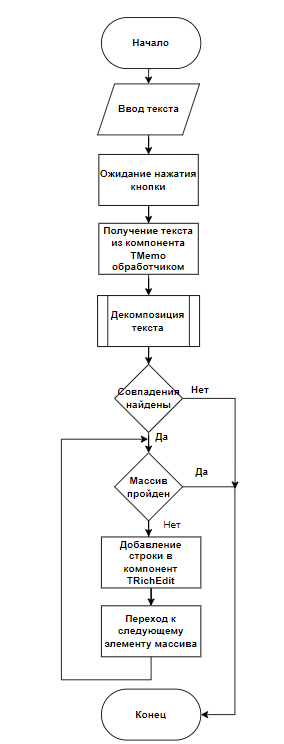


Рисунок 22– Блок-схема алгоритма декомпозиции текста

Для полного ознакомления с кодовой базой можно посетить удаленный репозиторий[[1]](#footnote-1)

**Контрольные вопросы**

1. Перечислить компоненты для работы с текстом и дать им краткую характеристику.

Компонент **TLabel** используется для отображения текста, который играет роль метки и не изменяется пользователем.

**TStaticText** - оконный компонент, отображающий однострочный текст на форме, не подлежащий редактированию.

**TEdit** - окно редактирования для ввода пользователем однострочных текстов.

**TMemo** - многострочный текстовый редактор, позволяющий редактировать текст окна, в которое можно вводить в отличие от компонента TEdit не одну, а множество строк.

Компонент **TRichEdit** представляет собой многофункциональное средство редактирования текстов, позволяющее работать с обогащенным форматом .rtf, т.е. выбирать различные атрибуты форматирования для разных фрагментов текста

2. Для каждого компонента перечислить основные свойства и методы .

Основные свойства **TLabel**.

**Align**. Определяет способ выравнивания элемента внутри контейнера (родительского компонента)

**AutoSize** Если это свойство установлено в true, то вертикальный и горизонтальный размеры компонента определяются размером надписи. Если же AutoSize равно false, то выравнивание текста внутри компонента определяется свойством Alignment

**Alignment**. Управляет горизонтальным выравниваением текста в пределах метки, если свойство AutoSize установлено в false

**Anchors**. Определяет привязку данного компонента к родительскому при изменении размеров последнего.

**Caption**. Строка текста, отображаемая меткой

**Color**. Определяет цвет фона метки

**FocusControl.** Определяет оконный компонент, получающий фокус при нажатии пользователем клавиши быстрого доступа метки

**Font**. Определяет атрибуты шрифта

**Layout.** Определяет выравнивание текста в поле метки по вертикали

**WordWrap**. Указывает, переносится ли текст на новую строку, если он превышает ширину метки, а высота метки позволяет разместить в ней несколько строк

Методы

**Create** Создает и инициализирует экземпляр компонента

**Destroy** Вызов деструктора объекта. Вместо Destroy всегда лучше использовать Free

**Free** Уничтожает объект и освобождает выделенную под него память. Автоматически вызывает Destroy, если объект существует.

**Alignment** Показывает способ выравнивания текста (значение Alignment) компонента

**Hide** Делает компонент невидимым

**Refresh** Немедленно перерисовывает компонент на экране, вызывая метод Repaint

**Repaint** Немедленно перерисовывает изображение компонента на экране

**SetBounds** Задает сразу 4 свойства: Left, Top, Width и Height. Генерирует события OnCanResize, OnConstrainedResize, OnResize, обработчики которых могут изменить заданные значения

**Show** Делает видимым невидимый компонент

**Update** Немедленно перерисовывает компонент

Основные свойства **TStaticText**.

**BorderStyle** Определяет стиль бордюра надписи

**Brush** Кисть - определяет цвет и стиль заполнения фона окна. Свойство только для чтения

Методы

**Hide, Repaint, Show**

Основные свойства **TEdit**.

**AutoSelect** Определяет, будет ли автоматически выделяться весь текст при передаче фокуса в окно редактирования

**AutoSize** Указывает, изменяется ли автоматически высота компонента, подстраиваясь под размер текста. По умолчанию false - не подстраивается

**BorderStyle** Указывает, ограничена ли клиентская область компонента одинарной бордюрной линией

**CharCase** Указывает, приводится ли принудительно текст к верхнему или нижнему регистрам

**Color** Указывает цвет фона компонента

**Font** Определяет атрибуты шрифта

**HideSelection** Определяет,сохраняется ли визуальный обзор выделенного текста при переходе фокуса к другому компоненту

**MaxLength** Указывает максимальное количество символов, которое пользователь может вводить в компонент

**Modified** Указывает, редактировался ли пользователем текст в компоненте

**PasswordChar** Указывает символ, замещающий фактические символы, напечатанные в компоненте. Используется для ввода паролей

**ReadOnly** Указывает, может ли пользователь изменять тест в компоненте

**SelLength** Определяет количество выделенных символов в строке

**SelStart** Указывает позицию первого выделенного символа в тексте или, если выделения нет, то позицию курсора

**SelText** Текст, выделенный в окне

**Text** Текст в окне редактирования

Методы

**Clear** Удаляет текст из окна

**ClearSelection** Удаляет текст, выделенный в окне

**CopyToClipboard** Копирует выделенный текст в компоненте редактирования в Clipboard в формате CF\_TEXT

**CutToClipboard** Переносит выделенный текст в Сlipboard в формате CF\_TEXT и уничтожает его в окне

**PasteFromClipboard** Переносит в окно текст из буфера Clipboard

**SelectAll** Выделяет весь текст в окне редактирования

Основные свойства **TMemo**.

**CaretPos** Указывает строку и символ расположения курсора.Доступ только для чтения

**Lines** Строки текста - объект типа TStrings. Доступно только во время выполнения

**SelText** Текст, выделенный в окне

**Text** Текст окна в виде одной строки

**WantRetuns** Указывает, можно ли вставить в текст символы возврата каретки

**WantTabs** Указывает, можно ли вставить в текст символы табуляции

**WordWrap** Указывает, переносится ли текст на новую строку, если он превышает ширину компонента

Методы

**Clear, ClearSelection, ClearUndo, CopyToClipboard, CutToClipboard, PasteFromClipboard, SelectAll**

3. Чем отличается класс TEdit от класса TStaticText.

В TEdit можно редактировать содержимое, а в TStaticText – нельзя.

4. Чем отличается класс TMemo от TRichEdit.

В TRichEdit можно выбырать различные атрибуты форматирования для разных фрагментов текста, а в TMemo только для всего текста

1. URL: https://github.com/InquisitorGor/FirstLaba [↑](#footnote-ref-1)