



МИНОБРНАУКИ РОССИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»  
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)  
БГТУ.СМК-Ф-4.2-К5-02

Факультет	О	Естественнонаучный
	шифр	наименование
Кафедра	О7	Информационные системы и программная инженерия
	шифр	наименование
Дисциплина		Программирование на языке высокого уровня

## КУРСОВАЯ РАБОТА

### на тему

Объектно-ориентированная разработка программ с графическим  
пользовательским интерфейсом «сверху-вниз»: предварительное  
выявление классов, объектов и их отношений. Вариант: **Текстовый редактор**

Выполнил студент группы И407Б

Альков В. С.

Фамилия И.О.

**РУКОВОДИТЕЛЬ**

Кимсанбаев К. А.

Фамилия И.О.

Подпись

Оценка

«\_\_\_\_»

2022г.

САНКТ-ПЕТЕРБУРГ

2022 г.

## СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	3
ВВЕДЕНИЕ.....	4
1 Постановка задачи .....	5
2 Описание моделей информационной системы.....	6
3 Демонстрация работы.....	14
ЗАКЛЮЧЕНИЕ .....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	22
ПРИЛОЖЕНИЕ А.....	23

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие сокращения и обозначения:

Разработка программы сверху-вниз - алгоритмы подзадач заменяются заглушками, которые представляют собой простейшие операции, имитирующие решение подзадачи, т. е. заглушки получают входные данные и, согласно тесту, формируют выходные данные. Таким образом, сначала проверяется правильность организации взаимосвязи всех подсистем во всех необходимых режимах. При правильной организации интерфейсов результаты работы программы должны быть точно такие же, какими они будут и при полностью завершенной программе. На следующих этапах заглушки поочередно заменяются реальными алгоритмами обработки. В итоге с заменой последней заглушки получается полностью работоспособная программа.

Windows Forms – интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки.

FastColoredTextBox –библиотека, которая предоставляет текстовый элемент, в котором реализованы следующие функции: быстрая подсветка синтаксиса различных языков, отображение номеров строк, поиск по тексту и замена текста, возможность простого добавления нестандартных полос прокрутки.

CustomTabControl – библиотека предоставляющая расширенные возможности стандартного элемента Windows Forms - TabControl, а именно: стили вкладок, кнопка закрытия вкладок, добавления иконок для вкладок.

UtfUnknown – библиотека, предоставляющая классы для работы для определения кодировки текста.

## **ВВЕДЕНИЕ**

Цель работы - разработать программу с использованием методов разработки “сверху-вниз” “Текстовый редактор”.

Разработка «сверху-вниз» начинается с описания взаимосвязи подсистем на самом высоком уровне.

На данный момент существует множество программ для работы с текстовыми файлами, однако все предлагают совершенно разный функционал и разные возможности.

Удобный просмотр текста, поиск по файлу, замена текста, подсветка синтаксиса упрощает взаимодействие пользователя с файлом, поэтому актуальность программы как никогда высока.

В процессе выполнения курсовой работы нам предстоит решить следующие задачи:

- разработать интерфейс взаимодействия пользователя и разработанной программы с помощью Windows Forms;
- реализовать подсветку синтаксиса языков: C, C++, C#;
- реализовать редактирование, просмотр и сохранения файлов;
- реализовать поиск по тексту и его замену;
- реализовать темную тему;
- реализовать перекодирование файла;
- реализовать вывод файла на печать;
- реализовать поддержку открытия нескольких файлов;
- описать взаимосвязи подсистем на самом высоком уровне;
- заменить алгоритмы подзадач заглушками, которые представляют собой простейшие операции, имитирующие решение подзадачи;
- проверить правильность организации взаимосвязи всех подсистем во всех необходимых режимах;
- заменить заглушки реальными алгоритмами обработки;
- протестировать работу программы.

## **1 Постановка задачи**

Задача курсовой работы заключается в разработке программы для работы с текстовыми файлами, позволяющая просмотреть, отредактировать, вывести на печать файлы, подсвечивает синтаксис.

Для выполнения работы разработка будет вестись на высокоуровневом языке программирования C# с применением технологии Windows Forms для создания графического интерфейса.

Для более быстрой подсветки синтаксиса, оптимальной работы темной темы и правильного определения кодировки текста будут использоваться библиотеки “FastColoredTextBox” [4], “CustomTabControl” [5], UtfUnknown [6], которые использует лицензию MIT, позволяющую использовать, копировать, изменять, объединять, публиковать, распространять, сублицензировать и/или продавать копии программного обеспечения.

## 2. Описание моделей информационной системы

Разработанные классы приведены на диаграммах классов на рисунках 1 - 7.

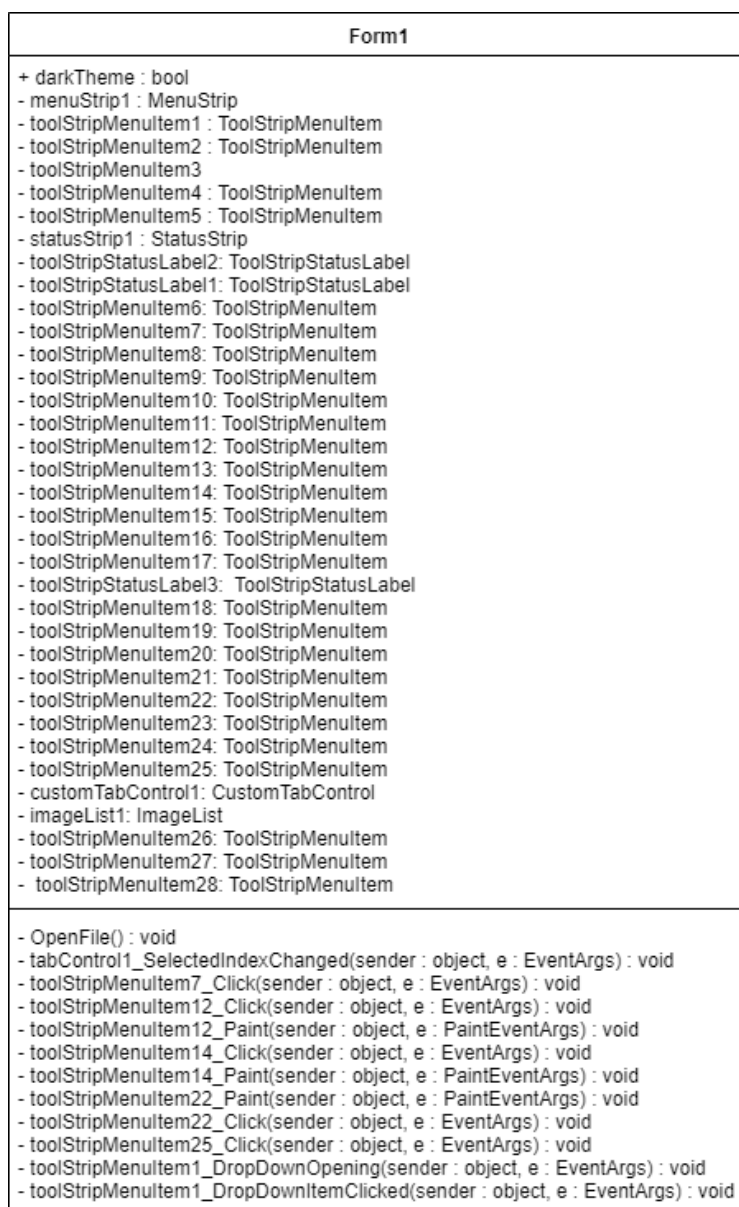


Рисунок 1 - Диаграмма классов (1)

Частичный класс **Form1** представляет графическую форму - фактически то окно, которое мы увидим на экране при запуске проекта. Этот класс определяется как частичный (с модификатором `partial`) и наследуется от встроенного класса **Form**, который содержит базовую функциональность форм.

Класс содержит следующие поля:

- `darkTheme` – поле, которое характеризует состояние темной темы;

- `menuStrip` – элемент интерфейса – главное меню, позволяет взаимодействовать с программой: открывать, сохранять и выводить на печать файлы, включать/отключать темную темы, выбирать язык для подсветки синтаксиса. За каждое действие отвечает определённый пункт меню;

- `toolStripMenuItemi` – пункты главного меню;

- `statusStrip` – элемент интерфейса, статусное поле, отображает полезную информация: кодировку открытого файла, кол-во строк и символов в файле, начало и длину выделения, позицию курсора. Информация разделена на статусные пункты.

- `toolStripStatusLabeli` - статусные пункты `statusStrip`;

Класс содержит следующие методы:

- `OpenFile` – метод, который вызывается для открытия файла;

- `tabControl1_SelectedIndexChanged` – метод, который выполняется по событию изменения выбранного элемента `TabControl` (переход на другую вкладку – выбор другой файла). Выполняет подстановку новых значений в элементы интерфейса: заполняет статусное поле (записывает кодировку выбранного файла, кол-во строк и символов в файле и т.д.).

- `toolStripMenuItem1_DropDownItemClicked` метод, который срабатывает по клику на элементы выпадающего списка пункта меню. Обрабатывает клики во вкладках “Файл” и “Изменить”: “Создать”, “Открыть”, “Сохранить (Ctrl + S)”, “ Сохранить как (Ctrl + Shift + S)”, “Печать”, “Найти (Ctrl + F)”, “Заменить (Ctrl + H)”

- `toolStripMenuItemi_Paint` – добавления отрисовки чего-либо при перерисовке элемента, например: добавить показатель выбранного пункта. У меня есть пункт “Автосохранение”, в нем есть пункт “Включить/отключить текущему файлу”. Этому пункту пририсовывается красный (серый) круг, если автосохранение текущего

файла включено (выключено);

- toolStripMenuItem\_Click – метод, который вызывается по клику на определенный пункт меню и выполняет определенную функцию;
- toolStripMenuItem1\_DropDownOpening – метод, который вызывается во время открытия выпадающего списка. Используется для отключения некоторых функций при отсутствии открытых файлов. Например, отключены: сохранение, печать, выбор языка и т.д.;

CustomTabPage
+ filePath : string + saved : bool + autoSave : bool + textBox: ExFastColoredTextBox + encoding : Encoding + StatusStrip statusStrip - lang : string + Lang : string + selectedInterval + hscrollBar : MyScrollBar + vscrollBar : MyScrollBar - BlueStyle : TextStyle - BoldStyle1 : TextStyle - BoldStyle2 : TextStyle - GrayStyle : TextStyle - MagentaStyle : TextStyle - GreenStyle : TextStyle - BrownStyle : TextStyle + WhiteStyle : TextStyle + BlackStyle : TextStyle + timer : Timer - _darkTheme : bool + DarkTheme : bool
+ CustomTabPage(status : StatusStrip, tBox : ExFastColoredTextBox, openFileDialog : OpenFileDialog) - AdjustScrollbars() : void - AdjustScrollbar(scrollBar : MyScrollBar, max : int, value : int, thumbsize : int) : void - textBox_ScrollbarsUpdated(sender : object, e : EventArgs) : void - scrollbar_Scroll(sender : object, e : ScrollEventArgs) : void - textBox_KeyDown(sender : object, e : KeyEventArgs) : void - textBox_SelectionChanged(sender : object, e : EventArgs) : void - textBox_TextChanged(sender : object, e : TextChangedEventArgs) : void - CustomSyntaxHighlight(e : TextChangedEventArgs) : void - AutoIndentNeeded(sender : object, args : AutoIndentEventArgs) : void - timer_Tick(sender : object, e : EventArgs) : void + Save() : void + SaveAs() : void - setLanguageFromFileName(filename : string) : void + Clear() : void + Convert(chosenEncoding : Encoding) : void

Рисунок 2 - Диаграмма классов (2)



Класс CustomTabPage представляет элемент интерфейса. Объект класса хранит один файл и предоставляет методы по работе с ним. Наследуется от встроенного класса TabPage.

Класс содержит следующие поля:

- filePath – путь к файлу;
- saved – флаг, который показывает сохранен файл или нет;
- autosave – флаг, который показывает включено автоматическое сохранение файла или нет;
- textBox – поле типа ExFastColoredTextBox наследника FastColoredTextBox [3], содержит текст файла;
- encoding – содержит тип кодировки файла;
- statusStrip – поле куда происходит вывод полезной информации: кол-во строк, длина текста и т.д.;
- lang – выбранный язык программирования файла;
- Lang – свойство, с гетерами и сеттерами, в которых по изменению Lang оно присваивается полю lang и производятся действия для перекраски текста под другой язык программирования;
- selectedInterval – выбранный пользователем интервал автоматического сохранения;
- hScrollBar, vScrollBar – полосы прокрутки для ExFastColoredTextBox;
- BlueStyle, BoldStyle1, BoldStyle2, GrayStyle MagentaStyleGreenStyle, BrownStyle, WhiteStyle, BlackStyle – стили для определенных сочетания текста, используются для подсветки текста;
- timer – таймер автоматического сохранения;
- \_darkTheme – поле, которое показывает включена или отключена темная тема у этого элемента;
- DarkTheme – свойство, через которое полю \_darkTheme присваивается значения, выполняет действия по изменению темы

элемента с светлой на темную или наоборот.

Класс содержит следующие методы:

- CustomTabPage – конструктор объекта;
- AdjustScrollbars, AdjustScrollbar – методы для синхронизации полос прокрутки элемента textBox с hScrollBar и vScrollBar;
- textBox\_ScrollbarsUpdated – метод, который выполняется при обновления полос прокрутки textBox, выполняет синхронизацию полос прокруток;
- scrollbar\_Scroll - метод, который выполняется при обновления полос прокрутки hScrollBar и vScrollBar, выполняет синхронизацию полос прокруток;
- textBox\_KeyDown – метод, который вызывается по нажатию клавише, пока активен textbox. Используется для добавления сочетания клавиш;
- textBox\_SelectionChanged – метод, который вызывается при изменении выделения текста в textBox. Выполняет запись полезной информации в statusStrip;
- textBox\_TextChanged - метод, который вызывается при изменении текста в textBox. Фиксирует что файл не сохранен, запускает подсветку синтаксиса, если выбран язык программирования;
- CustomSyntaxHighlight – метод, задающий цвета для разных сочетаний символов для их подсветки;
- AutoIndentNeeded – метод, включающий автоматический отступы, при включенном языке программирования;
- timer\_Tick – метод, который выполняется по срабатыванию таймера timer. Вызывает сохранение файла;
- Save – метод для сохранения файла;
- SaveAs – метод для сохранения файла;
- setLanguageFromFileName – метод, который устанавливает язык

программирования исходя из расширения переданного имени файла;

- Clear – метод, выполняющий очистку текста textBox;
- Convert – метод, который выполняет перекодирование файла в переданную кодировку;

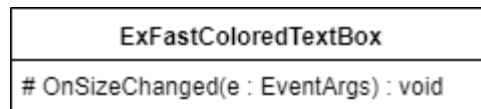


Рисунок 3 - Диаграмма классов (3)

Класс ExFastColoredTextBox. Наследуется от класса FastColoredTextBox [4]. Переопределяет один метод для обновления hScrollBar и vScrollBar при изменении размера элемента;

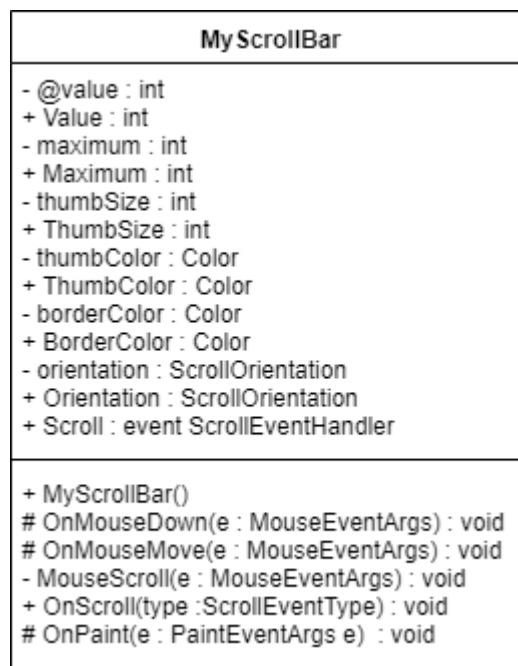


Рисунок 4 - Диаграмма классов (4)

Класс MyScrollBar. Элемент интерфейса – полоса прокрутки. Позволяет изменять внешний вид полосы прокрутки;

Класс содержит следующие поля:

- @value – поле текущего положения полосы прокрутки;
- Value – свойство, через которое происходит получение и изменение поля @value;

- `maximum` – поле максимального значения полосы прокрутки;
- `Maximum` - свойство, через которое происходит получение и изменение поля `maximum`;
- `thumbSize` – размер ползунка;
- `ThumbSize` - свойство, через которое происходит получение и изменение поля `thumbSize`;
- `thumbColor` – поле цвета ползунка;
- `ThumColor` - свойство, через которое происходит получение и изменение поля `thumbColor`;
- `borderColor` – поле цвета границы элемента;
- `BorderColor` - свойство, через которое происходит получение и изменение поля `borderColor`;
- `orientation` – поле ориентации ползунка (вертикальная или горизонтальная);
- `Orientation` - свойство, через которое происходит получение и изменение поля `orientation`;
- `Scroll` – обработчик события скроллинга;

Класс содержит следующие методы:

- `MyScrollBar` – конструктор объекта;
- `OnMouseDown` – действие при нажатии мыши на элемент;
- `OnMouseMove` – действие при движении мыши по элементу;
- `OnScroll` – действие при скроллинге;
- `OnPaint` – отрисовка элемента;

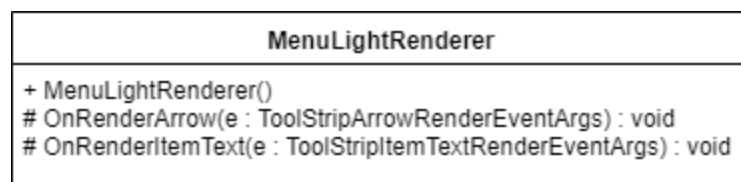


Рисунок 5 - Диаграмма классов (5)

Класс `MenuLightRenderer` – отвечает за отрисовку элементов меню и

статуса светлой темы. Наследуется от ToolStripProfessionalRenderer. С помощью переопределения меняются цвета;

Класс содержит следующие методы:

- MenuLightRenderer – конструктор объекта;
- OnRenderArrow – метод, выполняющий отрисовку стрелок в пунктах меню;
- OnRenderItemText – метод, выполняющий отрисовку текста в пунктах меню;

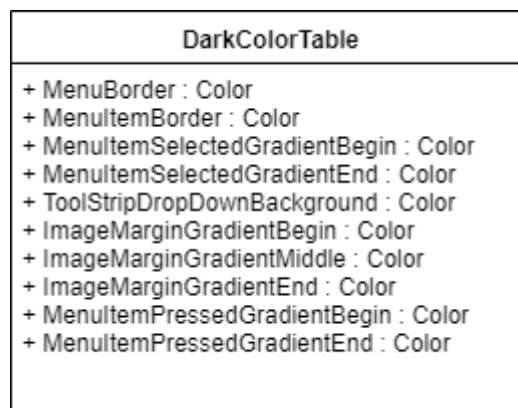


Рисунок 6 - Диаграмма классов (6)

Класс DarkColorTable – отвечает за определения цветов элементов меню и статуса при отрисовке. Наследуется от ProfessionalColorTable. С помощью переопределения свойств меняются цвета;

Класс содержит поля:

- MenuBorder - цвет границы выпадающего списка;
- MenuItemBorder – граница пункта меню;
- MenuItemSelectedGradientBegin, MenuItemSelectedGradientEnd – цвет заливки при наведении на активный элемент (enabled = true);
- ToolStripDropDownBackground - цвет выпадающих элементов;
- ImageMarginGradientBegin, ImageMarginGradientMiddle, ImageMarginGradientEnd - левая часть выпадающих элементов;
- MenuItemPressedGradientBegin, MenuItemPressedGradientEnd – цвет элемента, когда элемент был нажат, то есть открылся выпадающий

список;

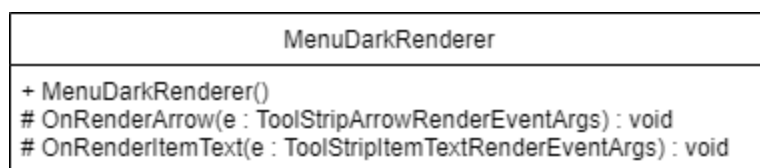


Рисунок 7 - Диаграмма классов (7)

Класс MenuDarkRenderer – отвечает за отрисовку элементов меню и статуса темной темы. Наследуется от ToolStripProfessionalRenderer. С помощью переопределения меняются цвета;

Класс содержит следующие методы:

- MenuDarkRenderer – конструктор объекта;
- OnRenderArrow – метод, выполняющий отрисовку стрелок в пунктах меню;
- OnRenderItemText – метод, выполняющий отрисовку текста в пунктах меню;

### 3. Демонстрация работы



Рисунок 8 – Стартовая страница программы

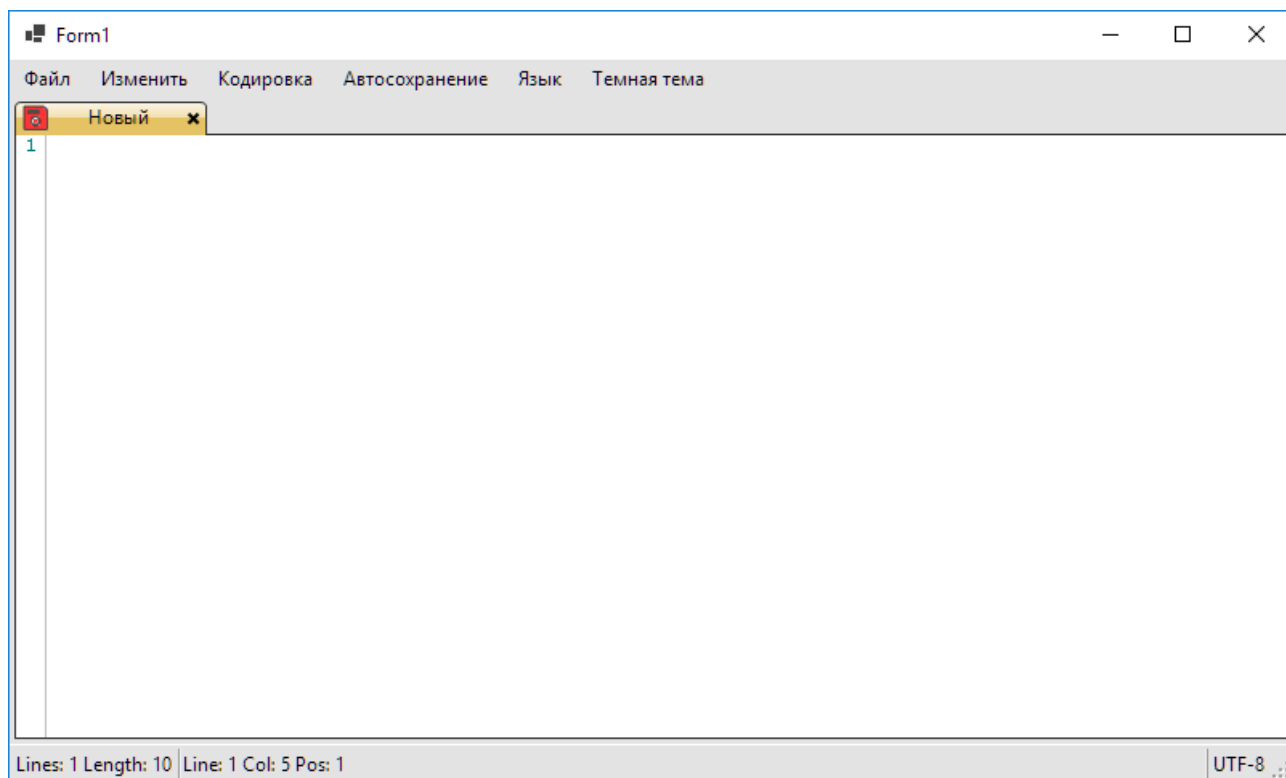


Рисунок 9 – Создан файл

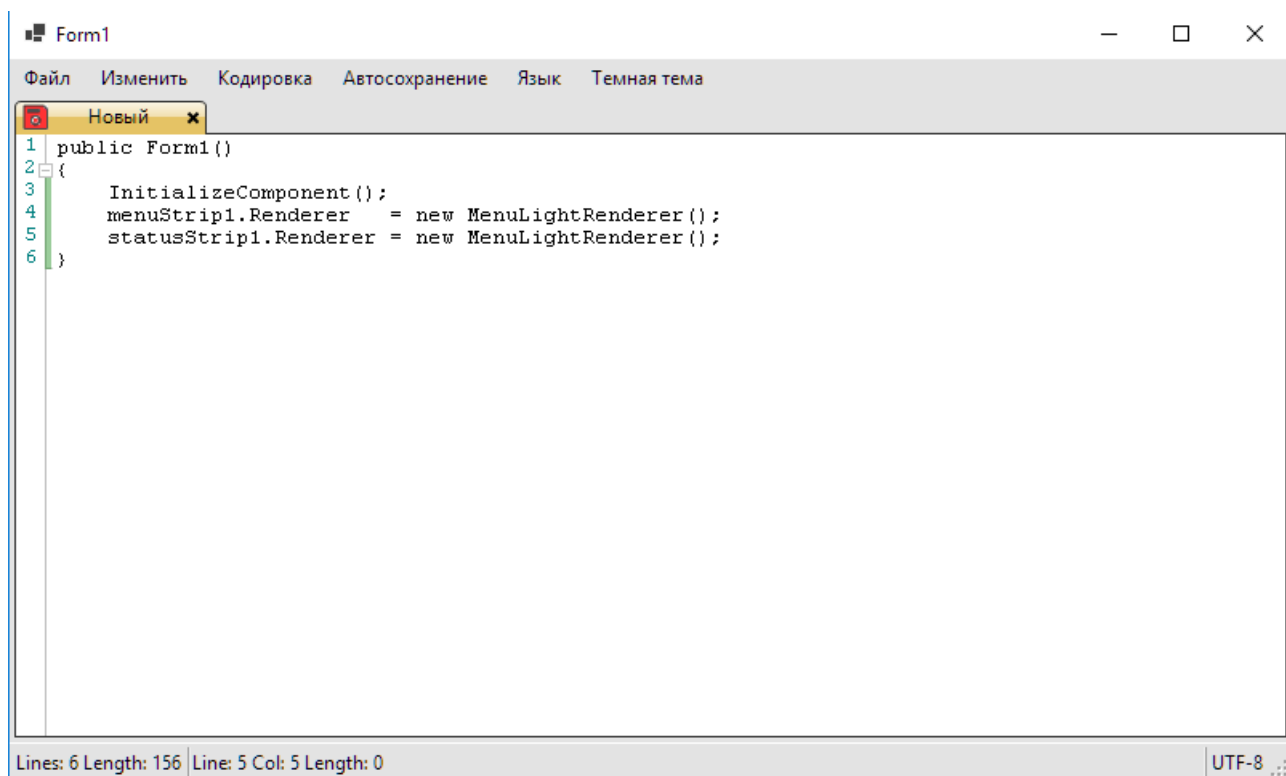


Рисунок 10 – Введен текст

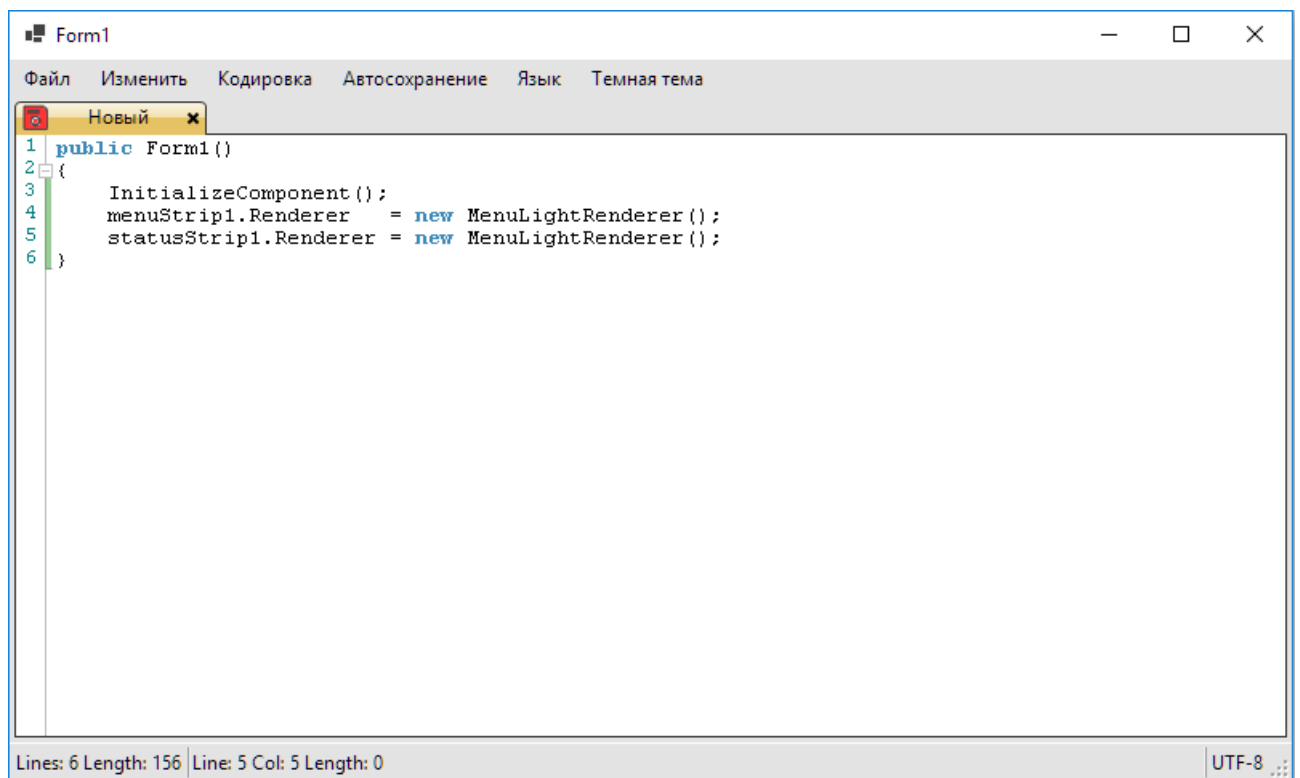


Рисунок 11 – Выбран язык C#

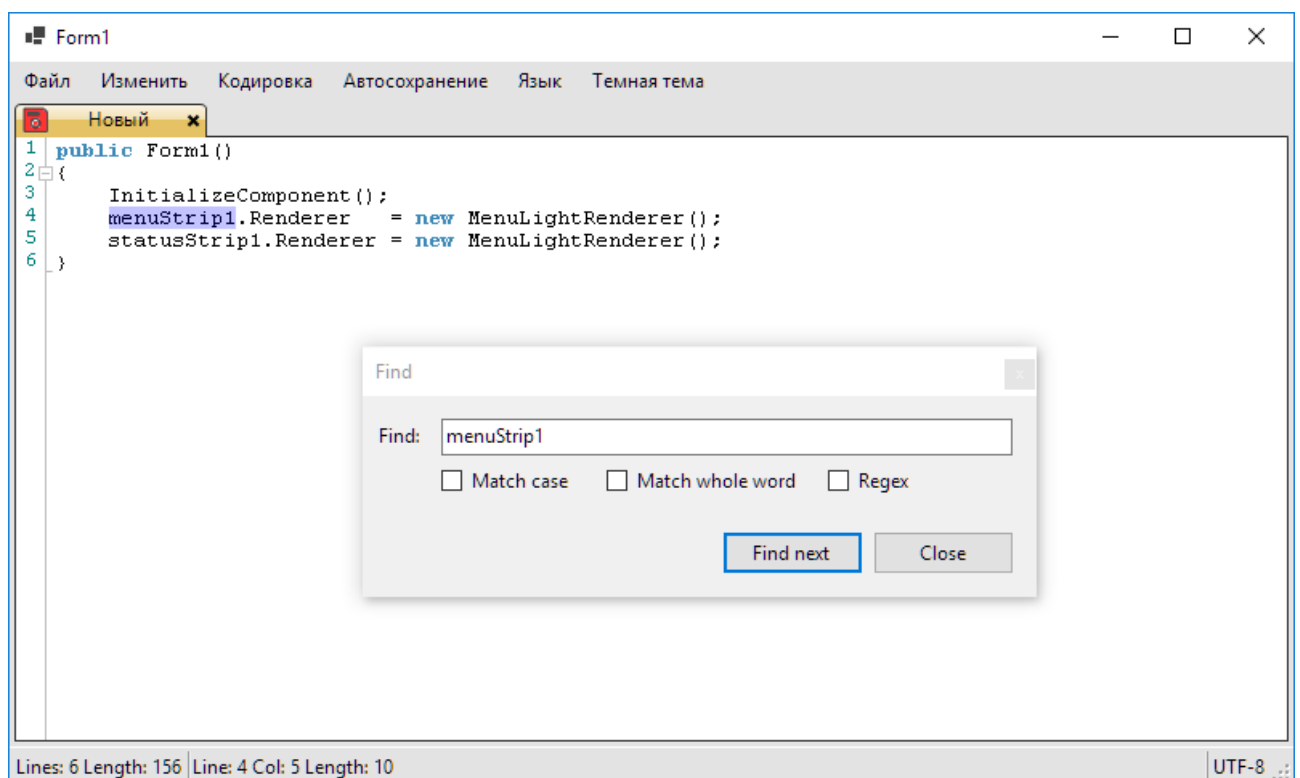


Рисунок 12 – Поиск по файлу



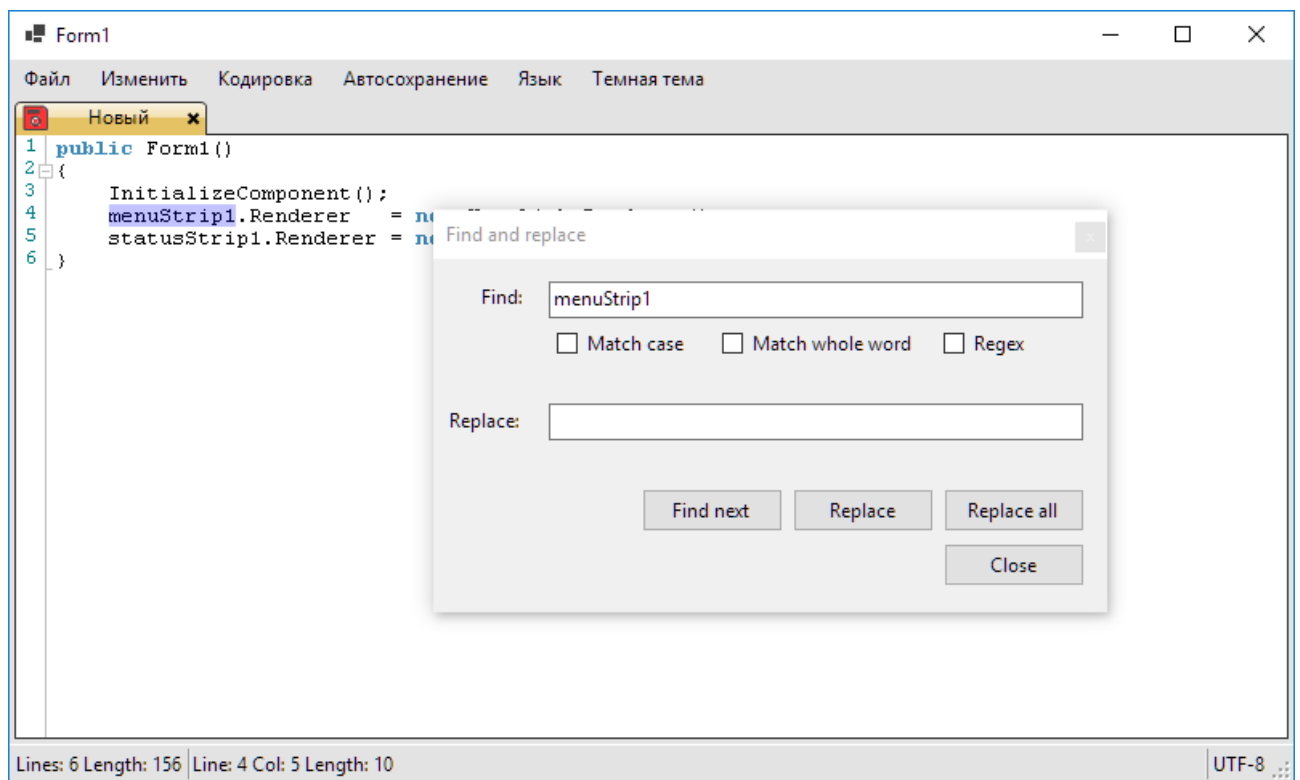


Рисунок 13 – Замена в файле

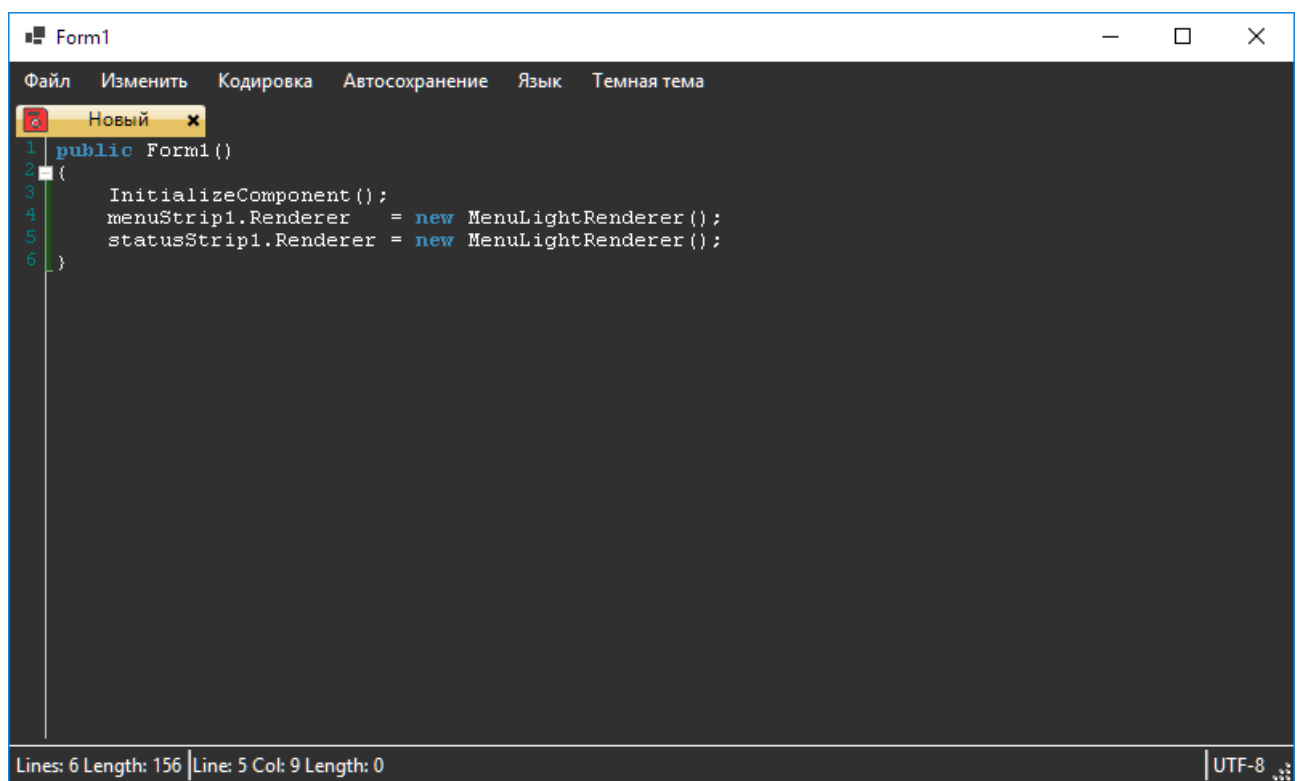


Рисунок 14 – Темная тема

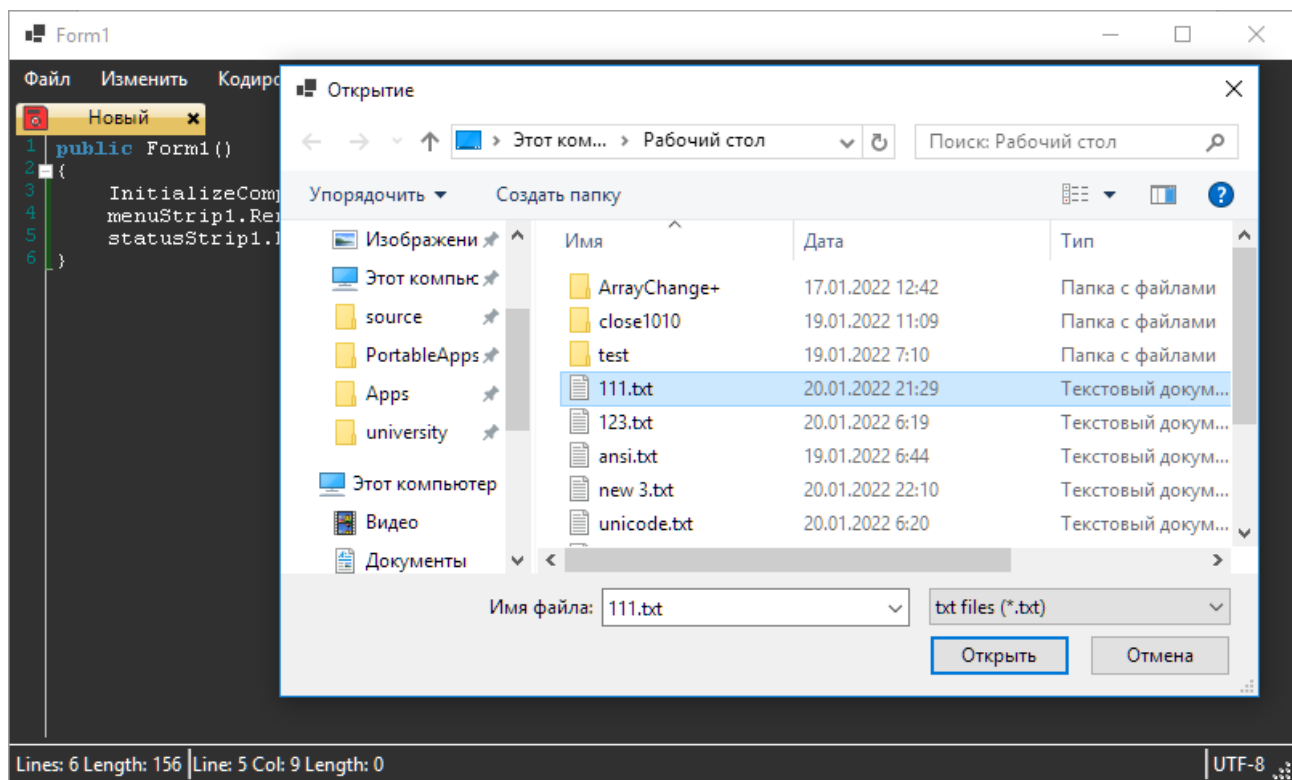


Рисунок 15 – Диалог по открытию текста

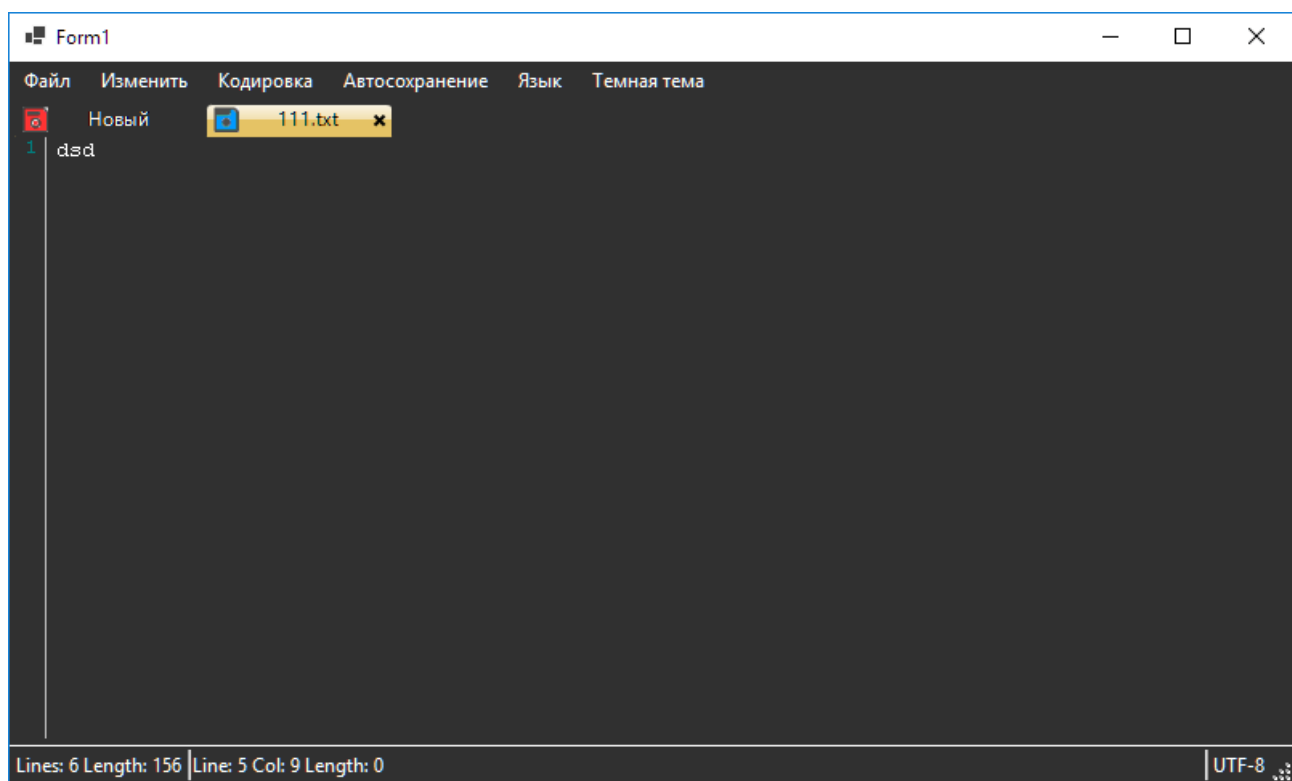


Рисунок 16 – Файл открыт

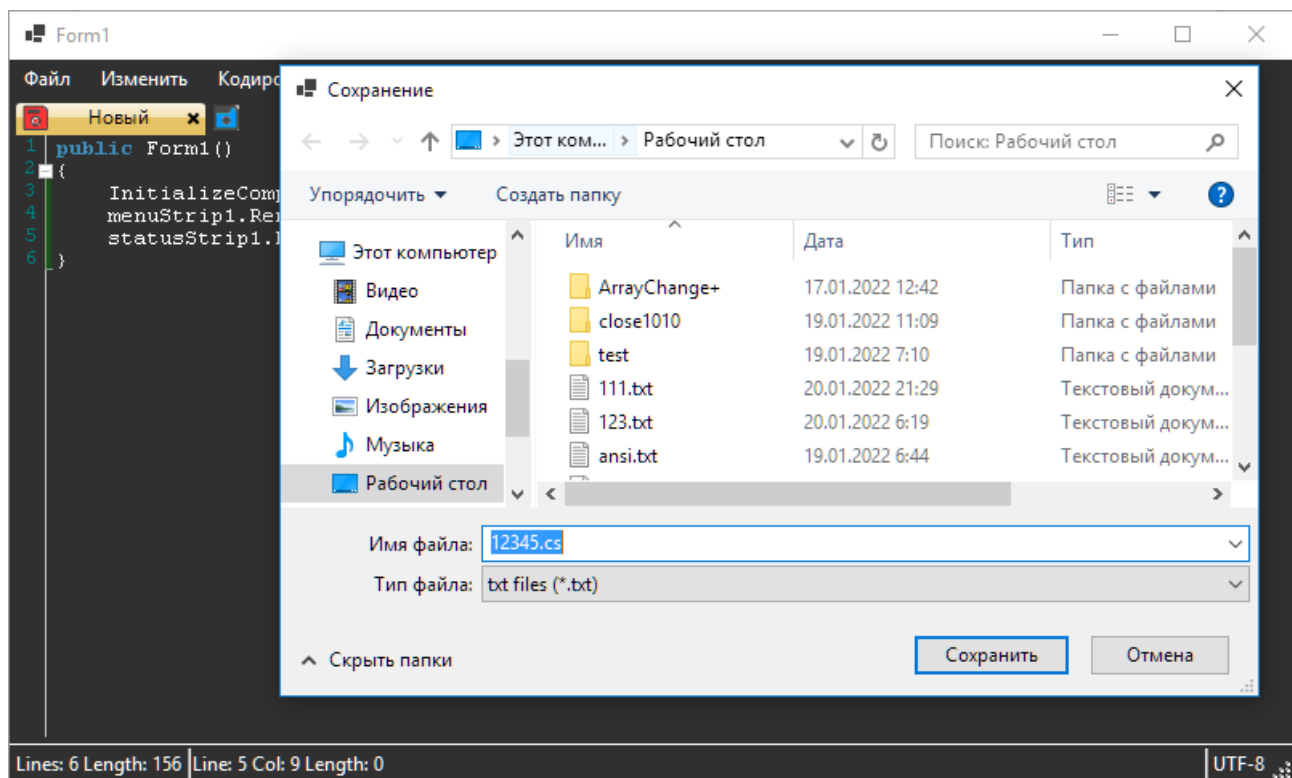


Рисунок 17 – Диалог сохранения файла

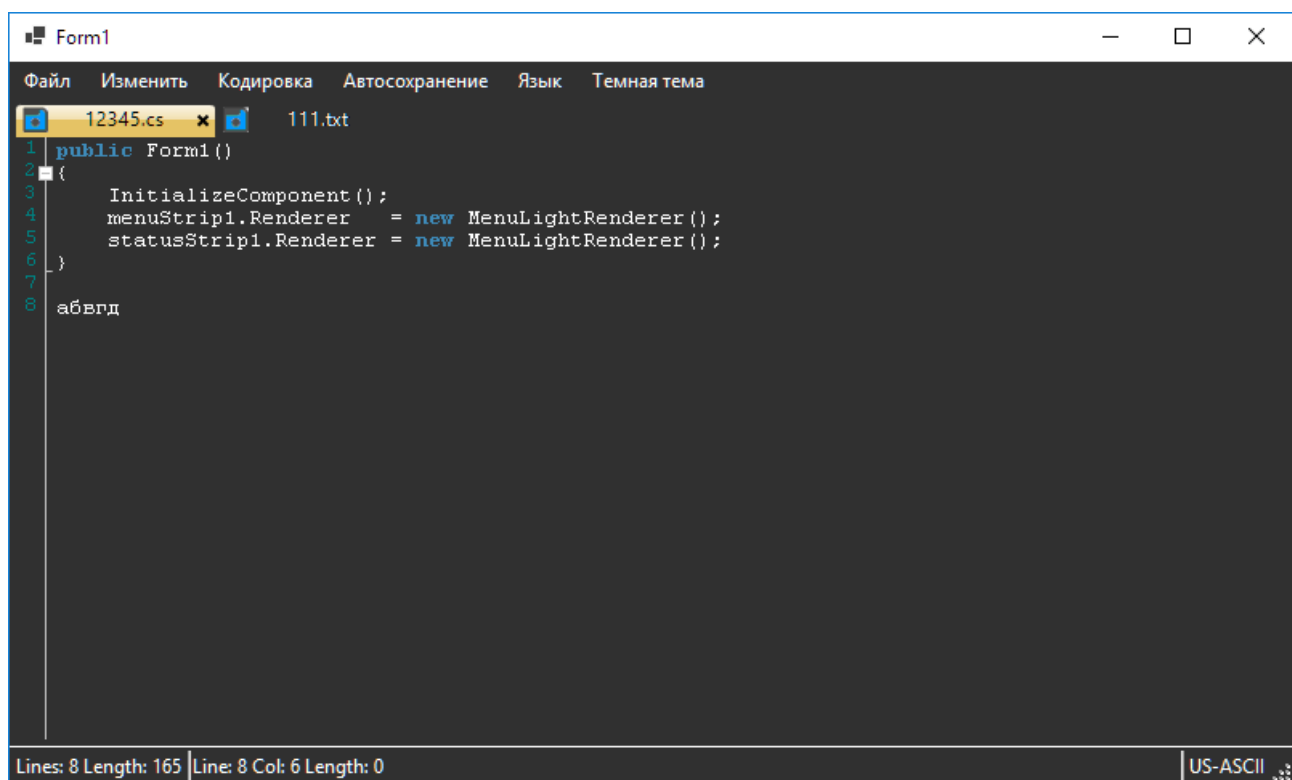


Рисунок 18 – Файл сохранен

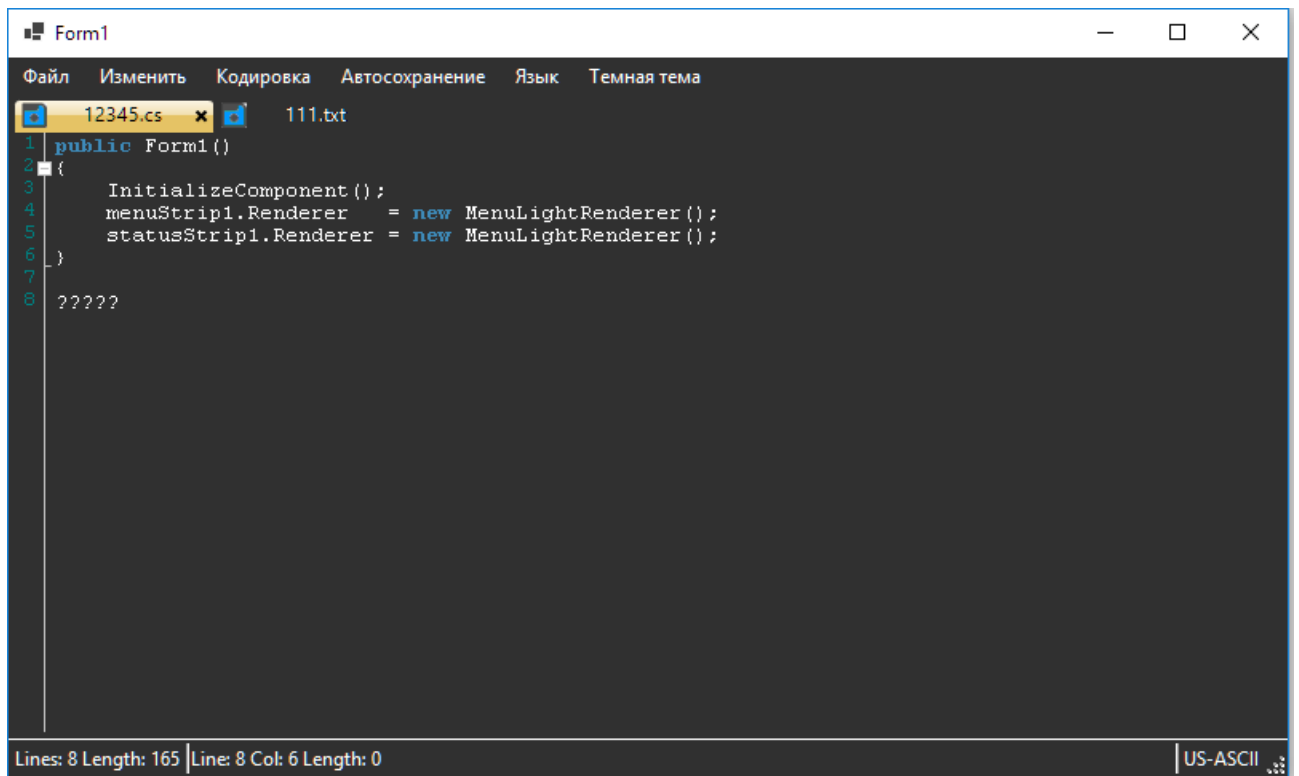


Рисунок 19 – Файл перекодирован в ASCII

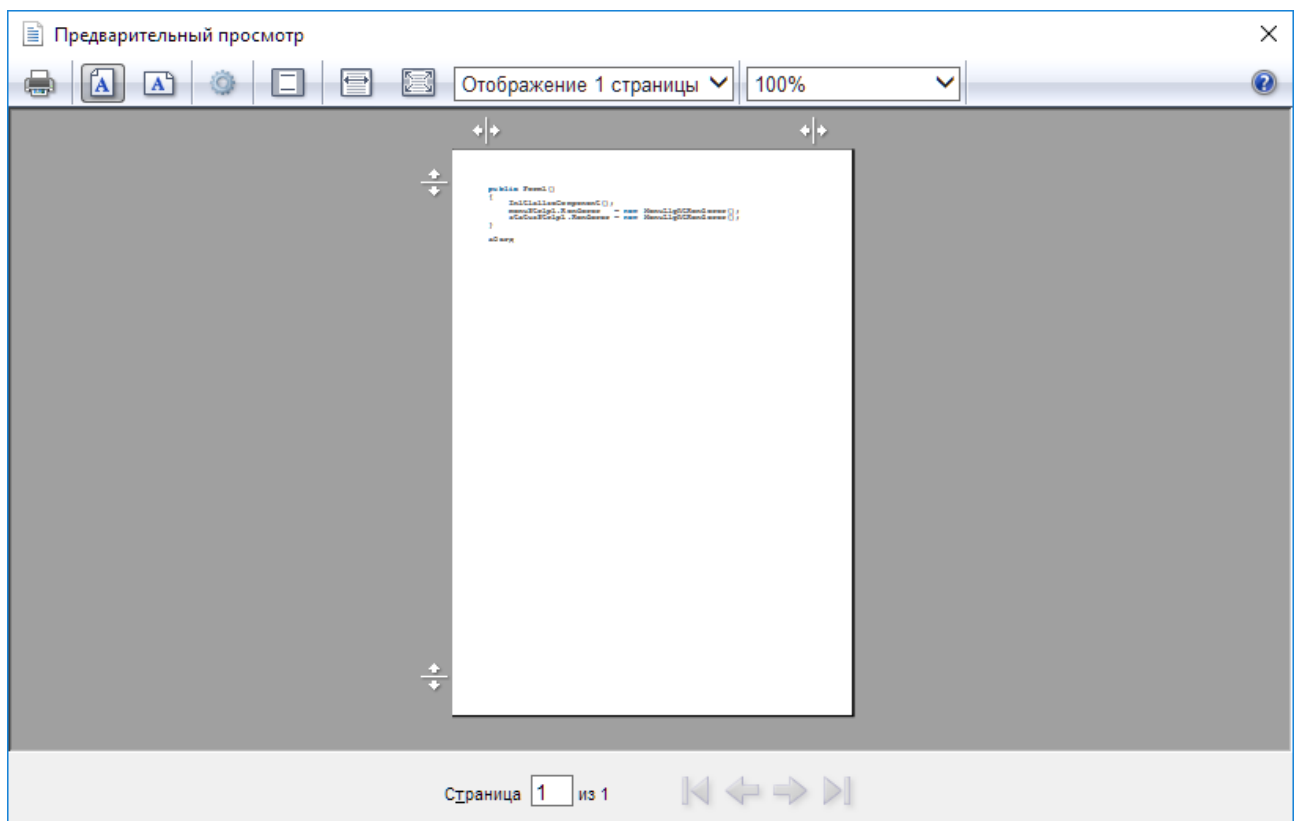


Рисунок 20 – Вывод файла на печать

## **ЗАКЛЮЧЕНИЕ**

В ходе курсовой работы была разработана программа для работы с текстовыми файлами. В процессе выполнения программы были реализованы все запланированные возможности, функции, методы. Были приобретены новые навыки работы с UML – таблицами, работы с фреймворком Windows Forms, усовершенствованы навыки работы с методами и функциями, отработан навык разработки программы методом “сверху-вниз”.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «Язык программирования C# 7 и платформы .NET Core» Троелсен Эндрю – 8 изд.; ООО «Диалектика». 2018 – 1328с.
2. Microsoft Docs [электронный ресурс] Учебник. Создание нового приложения WinForms (Windows Forms .NET) [Электронный ресурс] - URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/get-started/create-app-visual-studio?view=netdesktop-5.0> (Дата обращения: 20.01.2022).
3. Элемент управления MenuStrip (Windows Forms) [Электронный ресурс] - URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/controls/menustrip-control-windows-forms?view=netframeworkdesktop-4.8> (Дата обращения: 20.01.2022).
4. Fast Colored TextBox for Syntax Highlighting. The text editor component for .NET. [Электронный ресурс] - URL: <https://github.com/PavelTorgashov/FastColoredTextBox> (Дата обращения: 20.01.2022).
5. Painting Your Own Tabs - Second Edition [Электронный ресурс] - URL: <https://www.codeproject.com/Articles/91387/Painting-Your-Own-Tabs-Second-Edition-2> (Дата обращения: 20.01.2022).
6. Charset detector build in C# [Электронный ресурс] - URL: <https://github.com/CharsetDetector/UTF-unknown> (Дата обращения: 20.01.2022).

## **ПРИЛОЖЕНИЕ А**

Исходные тексты программы располагаются на прилагаемом электронном носителе.