**Упражнение 1: Создание простейшего текстового редактора**

**Задача 1: Открыть SimpleEditor проект**

**Задача 2: Отобразить окно диалога, для выбора файлы пользователем**

|  |  |
| --- | --- |
| 1. | Открыть файл MainWindow.xaml.  Кнопка **Open** позволяет на пользователю открыть в файл, и кнопка **Сохранить** позволяет пользователю сохранить изменения. Вы будете добавлять код, который реализует логику для этих двух кнопок. |
| 2. | Реализуем обработчик для кнопки OpenButton\_Click  Открыть MainWindow.XAML.CS файл. |
| 4. | Удалить на комментарий, а затем определить в новый частный метод с именем **GetFileName**, который не принимает ни одного параметра и возвращает строку значение, которая содержит имя файла. |
| 5. | В методе объявить переменную с именем **fname**, а затем инициализировать её значением **String.Empty**. |
| 6. | В конце списка с using добавить пространство имен **Microsoft.Win32** |
| 7. | В **GetFileName** методе, после объявления переменной fname, добавьте код:   |  |  | | --- | --- | | а. | Создание экземпляр **OpenFileDialog,** назвав его **openFileDlg**. | | б. | Установите свойство **InitialDirectory**, прописав путь в папке Starter, например, E:\Labfiles\Lab 5\Ex1\Starter, не забудьте про префикс @ перед строкой | | с. | Установите свойство **DefaultExt** на «.txt»; | | д. | Установите свойство Filter на **"Текстовые документы (.txt)|\*.txt"**. | |
| 8. | Добавьте код для выполнения на следующих задач:   |  |  | | --- | --- | | а. | Вызовите метод **ShowDialog** класса **openFileDlg**, а затем сохраните результат. | | б. | Если результат является **true**, присвойте переменной fname значение свойства  **FileName.** | |
| 9. | В конце метода возвращаем значение переменной fname. | |

**Задача 3: Реализация нового класса для чтения и записи текста в файл**

|  |  |
| --- | --- |
| 1. | Добавить новый класс с именем **TextFileOperations** к в FileEditor проекта.  Вы будете использовать этот класс для реализации некоторых распространенных операций с файлами. Это позволит изменить способ чтения или записи файлов, не затрагивая остальную часть приложения.. |
| 2. | В верхней части файла класса, добавьте пространство имен **System.IO** в область видимости. |
| 3. | В класса**TextFileOperations**, добавить метод public static **ReadTextFileContents**. Метод должен принимать в строку параметр с именем *имя\_файла*, и вернуть строку, которая является содержимым файла. |
| 4. | В методе **ReadTextFileContents** добавить код для возврата всего содержимого текстового файла, чей путь будет указан в параметре *FileName*.  **Подсказка**: Используйте в статический **ReadAllText** метод из File класса. |
| 5. | После **ReadTextFileContents** метода добавить public static **WriteTextFileContents**, метод. Метод не должен иметь возвращаемое значение и должен принимать следующие параметры:   |  |  | | --- | --- | | а. | Параметр с именем *имя\_файла* . | | б. | Параметр с именем *текста* . | |
| 6. | В **WriteTextFileContents** методе добавить код для записи текста из параметра в файл, который указан в параметре filename.  **Подсказка**: Используйте в статический **WriteAllText** метод из  класса File. |
| 7. | Построить на решение и исправить  ошибки. |

Задача 4: Обновите обработчики событий MainWindow для использования класса TextFileOperations

1. В списке задач найдите задачу TODO- U**pdate the OpenButton\_Clickmethod**. Дважды щелкните эту задачу.

Эта задача находится в методе OpenButton\_Click класса MainWindow.

2. Удалите комментарий и добавьте код для выполнения следующих задач:

Вызовите метод GetFileName. Сохраните результат метода в члене класса fileName.

Если fileName не является пустой строкой, вызовите статический метод ReadTextFileContents класса TextFileOperations, а затем передайте fileName в качестве параметра. Сохраните результат в свойстве Text элемента управления Textbox в окне WPF.

3. В списке задач найдите задачу **TODO - Update the SaveButton\_Clickmethod**.. Дважды щелкните эту задачу.

Эта задача находится в методе SaveButton\_Click класса MainWindow.

4. В методе SaveButton\_Click удалите комментарий, а затем добавьте код для выполнения следующих задач:

1)Убедитесь, что элемент fileName не является пустой строкой.

2) Если fileName не является пустой строкой, вызовите статический метод WriteTextFileContents класса TextFileOperations. Передайте fileName и свойство Text элемента управления Textbox редактора в качестве параметров.

5.Постройте решение и исправьте все ошибки.

**Упражнение 2: Создание редактор XML Aware**

**Эти приложения , которые контролируют с роботизированным устройством чтения с инструкциямииз в файл и затем закодировать их в качестве XML - документа , прежде чем передавая их в винструкции выполнения модуля на в устройстве. К примеру, представьте , что текстовый файлсодержит в инструкции последующий код примера.**

Move x, 10

Move y, 20

If x < y Add x, y

If x > y Sub x, y

Store 30

В управление приложениями будет завернуть их в виде пары из XML тегов, а следующий код примерпоказывает.

<ControlApplication>

<Instructions Code = "

Move x, 10

Move y, 20

If x < y Add x, y

If x > y Sub x, y

Store 30"

/>

</ControlApplication>

Тем не менее, некоторые из тех данных в этих инструкциях могут содержать символы , такие как «> » и «<» , что может быть неверно истолковано , как XML - теги , а чем данные.

В этом упражнении вы будете модифицировать в WPF приложение , чтобы искать для данных , которыесодержат XML - тег в в текстовом файле , как он будет прочитать в и закодировать эту информацию в видеXML эвакуационных последовательностей , прежде чем отображать его. К примеру, «< » характер будетбыть замененс «& GT; », «> » символ будет быть заменен с «& LT; », и так далее. WPF приложение будетиспользовать в файл поток для чтения в данных.

В основные задачи для этого упражнения являются также следующие:

|  |  |
| --- | --- |
| 1. | Открыть в стартовый проект. |
| 2. | Добавить в новый метод для фильтрации XML - символы в в **TextFileOperations** классе. |
| 3. | Обновление в пользовательский интерфейс для вызова на новый метод. |
| 4. | Реализовать тестовые примеры. |

**Задача 1: Открыть в стартовый проект**

|  |  |
| --- | --- |
| • | Открыть в SimpleEditor решение в на E: \ Labfiles \ Lab 5 \ Ex2 \ Starter папку.  Этот проект является завершена версией о в SimpleEditor проекте из упражнения 1. |

**Задача 2: добавить в новый метод для фильтрации XML - символов в в TextFileOperations класс**

|  |  |
| --- | --- |
| 1. | Просмотрите на задачи списка. |
| 2. | В в целевой список, найдите в **TODO - Реализовать на новый метод в в TextFileOperations класса**задач. Дважды щелкните эту задачу.  Эта задача будет расположена в в **TextFileOperations** классе. |
| 3. | Удалить на комментарий, а затем добавить новую публичную статическую методу с именем**ReadAndFilterTextFileContents** . Метод должен принимать в строку параметр с именем *имя\_файла* ,и вернуться в строку. |
| 4. | В в **ReadAndFilterTextFileContents** методе, добавьте в следующие локальные переменные:   |  |  | | --- | --- | | а. | **StringBuilder** объект с именем **fileContents** , инициализируется в виде нового экземпляра из в**StringBuilder** класса. | | б. | Целое число переменной называется charCode. | |
| 5. | Добавить в заявлении , что инстанцирует в **StreamReader** объект, названный **FileReader** , прииспользовании в *FileName* параметра. |
| 6. | Добавить в **то время как** о том , что читает каждый символ в с **StreamReader** объекта до конца из вфайле не будет достигнут.  **Подсказка** : Используйте в **Read** метод из в **StreamReader** класса для чтения на следующийсимвол из в потоке. Этот метод возвращает -1 , если есть это не больше данных. |
| 7. | В в **то время как** блок, добавьте в **переключатель** заявление , что оценивает в charCodeпеременную.  В с **переключателем** заявления, добавить **тематические** заявления для каждого из тех символов , в вследующей таблице. В каждом заявлении, добавьте в **fileContent StringBuilder** объект с вальтернативном представлении , показанном в в таблице.   | **charCode** | **Стандартное представление** | **Альтернативное представление** | | --- | --- | --- | | 34 | "( прямой кавычек ) | & quot; | | 38 | & ( амперсанд) | & amp ; | | 39 | '( апостроф) | & APOS; | | 60 | <( меньше ) | & lt; | | 62 | > ( больше ) | & gt; | |
| 8. | Добавить в по умолчанию **случай** о том , что присоединяет к фактическому характеру чтения из впотоке до в **fileContent StringBuilder** объекта.  **Примечание** : **Read** метод возвращает в значение чтения из в файл в качестве целого имагазинах оно в в charCode переменной. Вы должны бросить эту переменную в виде символа ,прежде чем вы добавить его в в конце части в **StringBuilder** объект. |
| 9. | В в конце части в способе, возвращают на содержимое из в **fileContent StringBuilder** объекта в видестроки. |
| 10. | Построить на решение и исправить любые ошибки. |

**Задача 3: Обновление в пользовательский интерфейс для вызова на новый метод**

|  |  |
| --- | --- |
| 1. | В в целевой список, найдите в **TODO - Обновление в пользовательский интерфейс дляиспользования на новый метод** задания. Дважды щелкните эту задачу.  Эта задача будет расположен в в **OpenButton\_Click** методе из в **MainWindow** . **xaml** . **cs** класс. |
| 2. | Удалить на комментарий, а затем изменить в строку из кода , который вызывает в **TextFileOperations.ReadTextFileContents** метод , чтобы позвонить в **TextFileOperations. ReadAndFilterTextFileContents**метод вместо этого. Проходит в **FileName** поле , как в параметре, а затем сохранить в результате в в**текстовой** собственности в в редакторе **TextBox** контроль. |
| 3. | Построить на решение и исправить любые ошибки. |
| 4. | Запустите в приложение без отладки. |
| 5. | В в MainWindow окне, нажмите **Открыть** . |
| 6. | В в **открытом** диалоговом окне перейти к в **E: \ Labfiles \ Lab 5 \ Ex2 \ Starter** папку, нажмите кнопку **команды. txt** , а затем нажмите « **Открыть»** . |
| 7. | В в MainWindow окне, убедитесь , что текст в в следующей коде примере будет отображаться в вредактор **TextBox** управлении. |

Move x, 10

Move y, 20

If x &lt; y Add x, y

If x &gt; y &amp; x &lt; 20 Sub x, y

Store 30

Это является текст из тех команд. txt- файл. Обратите внимание на то, что **<** , **>** , и **&** символы были былизаменены с в тексте **и ЛТ;**, **& gt;**, и **& amp ;**,

|  |  |
| --- | --- |
| 8. | Закройте в MainWindow окно и вернуться к визуальной студии. |

**Задача 4: Реализация тестовых примеров**

|  |  |
| --- | --- |
| 1. | В в целевой список, найдите в **TODO - Полный Unit тесты** задач. Дважды щелкните эту задачу.  Эта задача будет расположена в в **TextFileOperationsTest** классе. |
| 2. | Исследовать на **ReadAndFilterTextFileContentsTest** метод, а затем раскомментировать вкомментировал линию.  Этот метод создает три строки:   |  |  | | --- | --- | | а. | **Файла** строка содержит в пути из в предварительно составленном файл , который содержитконкретное содержание. | | б. | **Ожидаются** строка содержит в содержимое из в предварительно составленном файл, включаяформатирование и ускользают символы. | | с. | **Текущая** строка будет инициализирована с помощью вызова в **ReadAndFilterTextFileContents**метода , который вы только что выполнен. |   Тест метод затем использует в **Assert** заявление , чтобы убедиться в том , что эти **ожидаемые** и**фактические** строки являются же.  Этот метод является полным, и не требует не дальнейшей работы. |
| 3. | Выполнить все тесты в в растворе, и убедитесь , что все тесты выполнить правильно. |