

Дисциплина «Программирование»

Практическое задание №6

19 января – 1 февраля

«Текстовый редактор “Notepad+”»

Процесс выполнения этого задания состоит из трёх частей:

- 1) реализация программы, согласно описанным в условии требованиям;
- 2) оценивание работ других студентов;
- 3) период «споров».

Период реализации программы:

После выдачи задания Вам необходимо выполнить его и загрузить архив (*.zip) с решением задачи (полностью заархивировать решение, созданное средой разработки) до крайнего срока. Работа должна выполняться студентом самостоятельно. В работе строго запрещается указывать ФИО, а также любую другую информацию, которая может выдать авторство работы. В случае выявления факта деанонимизации работы, работа может быть аннулирована.

Период взаимного оценивания:

После окончания срока, отведенного на реализацию программы, начинается период взаимного оценивания. Вам будет необходимо проверить пять работ других студентов, также выполнявших данное задание, согласно критериям оценивания. Проверка осуществляется анонимно: Вы не знаете, чью работу Вы проверяете, также, как и человек, кому принадлежит решение, не знает, кем была проверена его работа. Помимо оценки Вам необходимо указать комментарий к каждому из критериев. В случае, если Вы снижаете балл, необходимо подробно описать, за что именно была снижена оценка. Также рекомендовано писать субъективные комментарии, связанные с тонкостями программной реализации, предлагать автору работу более оптимальные на ваш взгляд решения. Снимать баллы за субъективные особенности реализации запрещено.

Период споров:

По окончании периода взаимного оценивания, в случае если Вы не согласны с оценкой, выставленной одним из проверяющих, Вы можете вступить с этим студентом в анонимный диалог с целью уточнения причин выставления оценки по тому или иному критерию. В случае, если проверяющий не ответил Вам, или вы не пришли к обоюдному решению об изменении оценки, Вы можете поставить флаг, и работа будет рассмотрена одним из преподавателей.

Флаги, поставленные без предварительного обсуждения с проверяющим или после окончания периода выставления флагов, будут отклонены.

Также допустима перепроверка Вашей работы преподавателем или ассистентом. В таком случае оценки других проверявших работу не учитываются.

Дедлайн загрузки работы: 1 февраля 23:59

Дедлайн проверки: 4 февраля 23:59

Дедлайн обсуждения оценок: 6 февраля 23:59

Дедлайн выставления флагов: 7 февраля 23:59

Возможность поздней сдачи работы в этом задании предоставляться не будет.

Оценивание:

$O_{\text{итог}} = 0,8 * O_{\text{задание}} + 0,2 * O_{\text{проверки}}$, где $O_{\text{задание}}$ – неокруглённая десятибалльная оценка за решение задания выставленная проверяющими с учётом возможной перепроверки преподавателем, а $O_{\text{проверки}}$ неокруглённая десятибалльная оценка, выставленная студентами, чьи работы вы проверяли. Также в случае, если преподаватель обнаружит, что Вы проверили работы некачественно к Вам могут быть применены санкции в виде штрафа до 3 десятичных баллов от $O_{\text{итог}}$ (в таком случае оценка вычисляется по формуле $O_{\text{итог}} = O_{\text{задание}} - \text{Штраф}$).

Необходимо разработать Windows Forms приложение – «Notepad+».

Реализовать с использованием функционала компонента RichTextBox текстовый редактор, позволяющий открывать, редактировать и сохранять текстовые файлы в форматах *.rtf и *.txt, а также любые другие текстовые файлы.

Задание:

Разработать оконное приложение Windows Forms App (.NET 5 или .Net Core 3.1) позволяющее:

1. Открывать, редактировать и сохранять простые текстовые файлы (без сохранения форматирования).
2. Открывать, редактировать и сохранять текстовые файлы формата *.rtf (с сохранением форматирования).
3. Предоставить возможность открывать несколько текстовых файлов в одном окне приложения с возможностью переключения между ними (с несколькими вкладками).

4. При наличии несохраненных изменений в открытых файлах при закрытии приложения предоставить пользователю возможность сохранить файл, не сохраняя или отменить закрытие приложения.
5. Реализовать меню приложения со следующими подменю: Файл, Правка, Формат(для выбора основного шрифта текста), Настройки. Список может быть дополнен на ваше усмотрение.
6. Реализовать автоматическое сохранение открытых файлов. Частота сохранения задаётся в окне Настроек.
7. Реализовать возможность выбора цветовой схемы приложения в настройках.
8. Предоставить возможность пользователю форматировать выделенный фрагмент изменяя стиль текста по следующим параметрам:
 - Курсив
 - Жирный
 - Подчёркнутый
 - ЗачёркнутыйПричём, несколько стилей могут применяться к выделенному фрагменту одновременно.
9. Создать контекстное меню, активируемое правой кнопкой мыши, со следующим функционалом:
 - Выбрать весь текст
 - Вырезать выделенный фрагмент
 - Копировать выделенный фрагмент
 - Вставить сохранённый в буфере обмена
 - Задать формат выделенного фрагмента текста
10. Реализовать возможность использования горячих клавиш для следующих операций:
 - Создание документа в новом окне.
 - Создание документа в новой вкладке
 - Сохранение текущего документа
 - Сохранение всех открытых в окне документов
 - Закрытие приложения
11. При перезапуске приложения должны сохраняться выбранные настройки, а также открываться файлы, которые были открыты в приложении (если они ещё существуют).
12. [Дополнительный функционал] Реализовать журналирование (сохранение в скрытой папке нескольких версий файла) для возможности отката к предыдущей версии. Интервал сохранения задаётся в меню настроек.
13. [Дополнительный функционал] Реализовать возможность отмены и повторения предыдущих действий, также создать горячие клавиши для этих действий Ctrl+Z и Ctrl+Shift+Z соответственно.

14. [Дополнительный функционал] Реализовать возможность сохранения и открытия исходных кодов на языке C#.
15. [Дополнительный функционал] Реализовать автоматическое форматирование кода на языке C# для кода, расположенного в одном файле.
16. [Дополнительный функционал] Реализовать автоматическую подсветку синтаксиса кода на языке (выделение разными цветами, цвета задаются в настройках, ключевых слов, имён классов, переменных, комментариев, и иное на усмотрения разработчика) для кода, расположенного в одном файле.
17. [Дополнительный функционал] Реализовать возможность компиляции исходного кода из одного файла в исполняемый файл с выводом ошибок компиляции в отдельное окно(часть окна). Путь к компилятору указывается с помощью настроек при первой попытке сборки исполняемого файла, в дальнейшем может быть изменён в настройках.

Дополнительные требования (критерии оценивания):

1. Для проверки в систему PeerGrade должен быть загружен архив с решением. Ожидается, что проверка работы будет проводиться, в среде разработки Visual Studio 2019, поэтому в случае выполнения задания с использованием другой среды разработки настоятельно рекомендуется проверить возможность открытия и запуска проекта в этой среде разработки.
2. Текст программы должен быть отформатирован согласно кодстайлу языка C#¹. Для автоматического форматирования в среде Visual Studio достаточно нажать Ctrl+K, D. Идентификаторы должны соответствовать соглашению о именовании C#². Основным требованием, которое требуется соблюдать в данном задании – это написание имён локальных переменных с использованием camelCasing, а при именовании методов и типов PascalCasing.
3. Программа не должна завершаться аварийно (или уходить в бесконечный цикл) при любых входных данных. При некорректных входных данных программа должна выводить сообщение об ошибке и запрашивать ввод заново.
4. Программа должна быть декомпозирована. Каждый из логических блоков должен быть выделен в отдельный метод. Не строго, но желательно, чтобы каждый метод по длине не превышал 40 строк.

¹ <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>

² <https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/naming-guidelines>

5. Интерфейс программы должен быть понятен. Пользователю должны выводиться подсказки о возможных дальнейших действиях и иные необходимые сообщения. Предполагается, что для успешного использования программы не требуется обращения к исходному коду программы.
6. Текст программы должен быть документирован. Необходимо писать, как комментарии перед методами, так и комментарии, поясняющие написанный внутри метода код. Названия переменных и методов должны быть на английском языке и отражать суть хранимых значений / выполняемых действий.
7. Работа должна быть выполнена в соответствии с заданием, критерии оценивания(“Rubric”) доступны для предпросмотра в системе “PeerGrade”.
8. Для получения отличной оценки более 8 студенту необходимо реализовать дополнительный функционал, описанный в п.12 – п.17 задания.
9. Также оценивается общее впечатление, которое производит работа (как с точки зрения пользовательского интерфейса, так и с точки зрения написания кода программы). Эта часть оценки остаётся на усмотрение проверяющего.