

Дисциплина «Программирование»

Практическое задание №2

13 -19 октября

«Файловый менеджер»

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. 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.

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.

Дедлайн загрузки работы: 19 октября 23:59
Дедлайн проверки: 22 октября 23:59
Дедлайн обсуждения оценок: 23 октября 23:59
Дедлайн выставления флагов: 25 октября 23:59

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 консольное приложение – «Файловый менеджер».

Допустима как реализация приложения-эмулятора командной строки, так и приложения с набором нумерованных текстовых меню. При выборе первого варианта обязательно требуется реализация команды “help” с подробным описанием возможных действий. Не обязательно реализовывать 11 различных команд, одна команда может выполнять несколько операций, в зависимости от передаваемых параметров.

Алгоритм работы приложения:

В начале работы программы пользователь выбирает операцию, которую хочет выполнить. Среди доступных операций должны присутствовать:

1. просмотр списка дисков компьютера и выбор диска;
2. переход в другую директорию (выбор папки);
3. просмотр списка файлов в директории;
4. вывод содержимого текстового файла в консоль в кодировке UTF-8;
5. вывод содержимого текстового файла в консоль в выбранной пользователем кодировке (предоставляется не менее трех вариантов);
6. копирование файла;
7. перемещение файла в выбранную пользователем директорию;
8. удаление файла;
9. создание простого текстового файла в кодировке UTF-8;
10. создание простого текстового файла в выбранной пользователем кодировке (предоставляется не менее трех вариантов);
11. конкатенация содержимого двух или более текстовых файлов и вывод результата в консоль в кодировке UTF-8.

После выбора операции программа должна выполнять запрошенную операцию с файловой системой компьютера и сообщать о результате.

Дополнительные требования (критерии оценивания):

1. Для проверки в систему PeerGrade должен быть загружен архив с решением. Тип проекта должен быть Console Application .Net Core 3.1. 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.
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 30 строк.
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 разбирательств с обращением к исходному коду программы.
6. Должно быть реализовано повторение решения. То есть, после окончания выполнения очередной операции, пользователю предлагается завершить программу или ввести новую команду.
7. Текст программы должен быть документирован. Названия переменных и методов должны быть на английском языке и отражать суть хранимых значений / выполняемых действий.
8. Для получения отличной оценки более 8 студенту необходимо реализовать больший функционал, чем описано в задании выше. Вам необходимо будет самостоятельно продумать, как сделать программу лучше и полезнее по сравнению с тем, что указано в основной спецификации.

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

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.