

Дисциплина «Программирование»

Практическое задание №8

2 марта – 15 марта

«Анализ табличных данных»

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: 15 марта 23:59

Дедлайн проверки: 18 марта 23:59

Дедлайн обсуждения оценок: 20 марта 23:59

Дедлайн выставления флагов: 21 марта 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 оконное приложение (Windows Forms Application), которое будет анализировать данные, представленные в табличной форме.

Задание:

Разработать оконное приложение (.Net Framework 4.7.2 (или 4.8), .NET Core 3.1 или .NET 5) со следующей функциональностью:

1. В основном окне программы необходимо реализовать функционал для чтения данных из CSV-файла с таблицей и корректного отображения данных в виде таблицы внутри оконного приложения. Стоит учитывать то, что:
 - исходные данные могут быть большого объема для удобного отображения их в окне формы;
 - не все поля могут иметь значения (для них придется предусмотреть умалчиваемые значения);
 - в разных загружаемых таблицах количество столбцов и строк различно.

Примеры таблиц можно найти в приложении к заданию (на этих данных программа обязана работать корректно).

2. Необходимо реализовать не только функцию презентации данных (загрузка таблицы и визуализация), но и анализа данных. В частности, аналитический потенциал программы должен позволять пользователю:
 - сравнивать количественные данные и частоту их встречаемости с помощью графиков-гистограмм для заданного или для выделенного столбца таблицы. Подписи столбцов обязательны для соблюдения дружественного интерфейса.
 - строить двумерные графики зависимостей по числовым данным для заданных или выбранных столбцов таблицы (выбираем два столбца: $X \rightarrow Y$).
 - по числовым столбцам считать среднее значение, медиану, среднеквадратичное отклонение и дисперсию.
3. Графики-гистограммы необходимо строить в отдельном окне, которое не блокирует основное окно приложения или другие окна с графиками (из этого следует, что одновременно можно открыть несколько графиков и гистограмм).
4. При изменении размеров окна с гистограммой автоматически перерисовывать сам график, выполняя масштабирование.
5. Для числовых значений предусмотреть возможность менять ширину отображаемых столбцов гистограммы с помощью элемента NumericUpDown. Чем больше их ширина, тем больший спектр значений охватывает каждый столбец, и тем меньше столбцов отображается в данный момент.
6. **[Дополнительный функционал]** Реализовать построение более сложной диаграммы по значениям из выделенной прямоугольной области, как на примере ниже.

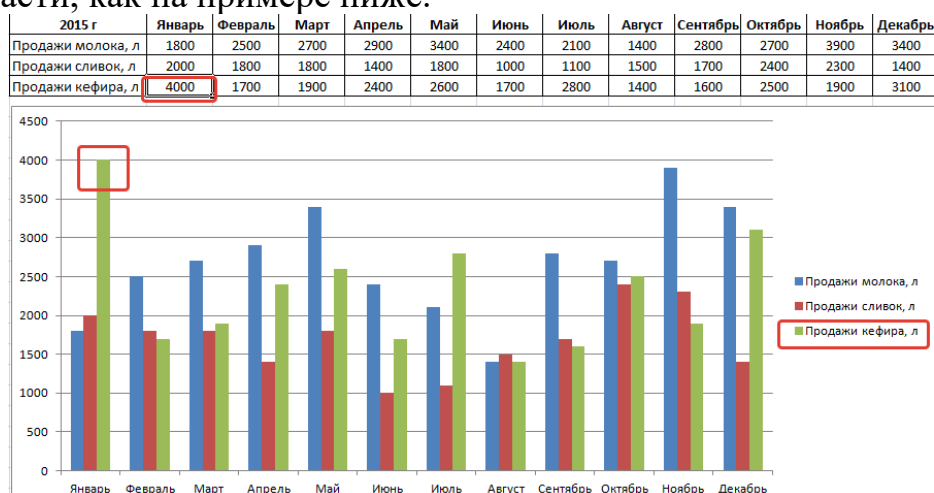


Рисунок 1 – Пример гистограммы

7. **[Дополнительный функционал]** По нажатию на столбец диаграммы менять его цвет на выбранный из оконного диалога.
8. **[Дополнительный функционал]** Реализовать сохранение графиков.

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.
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.6 – п.7 задания.

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>

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.