

Лабораторная работа №1

Проверочное задание на тему “Расчет средней успеваемости студентов”.

Аудитория: студенты.

Назначение: задание для выполнения студентами, содержащее основную информацию и требования к результату.

Задание предназначено для проверки уровня владения навыками разработки приложений среди студентов, чтобы можно было оценить состояние текущей подготовки, уровень написанного исходного кода, приемы и практики, паттерны которые применяются и т.п.

Описание задачи

Необходимо разработать консольное приложение, которое производит чтение входного файла в формате CSV и путем обработки считанных данных проводит расчеты средних величин, генерирует выходной файл в виде Excel (xlsx) или JSON документа. Приложение должно предоставлять возможность взаимодействия с ним через консоль с помощью команд, в которых можно указать как входной файл, так и выходной файл и его тип (Excel, JSON).

Детали:

1. Входной файл содержит список студентов группы и их оценки по экзаменам за все предыдущие семестры. Оценки должны быть указаны по предметам, чтобы в процессе чтения можно было точно сопоставить предмет и оценку. Следовательно, файл должен содержать строку с заголовком и правильно введенные данные.
2. Обработка данных заключается в расчете среднего балла по каждому студенту, а также средней оценки по каждому предмету.
3. Выходной файл содержит в себе таблицу со всеми студентами, а также средними оценками каждого из них. Внизу таблицы приведены средние баллы по каждому предмету и по группе.
4. Один входной файл - один выходной.

Средства

Язык: C#.

Платформа: .NET Core 2.x или 3.x.

Для расширенного функционала, в частности чтения файла и записи в необходимых форматах можно использовать некоторые пакеты NuGet, представляющие собой библиотеками, для реализации того или иного функционала.

Требования

Приложение должно запускаться на любом ПК (если мы говорим про .NET Core то в частности на любой ОС), а также полностью успешно компилироваться и собираться из исходных файлов.

Чтение и запись файлов должно осуществляться в соответствии с входными данными. Для проведения тестирования необходимо иметь подготовленные входной файл (или файлы) и примеры готовых выходных файлов. Каждый случай должен иметь описание команды (которую необходимо записать в консоли при запуске приложения), которая создает выходной файл. То есть должно быть описание входа и результатов.

Все ошибки в приложении должны логироваться и выводиться в текстовый файл с логами. Исключительные ситуации в процессе работы не должны приводить к прекращению работы приложения.

Исходный код приложения должен быть самодокументирован при помощи комментариев.

Дополнительно:

1. Исходный код приложения должны быть помещен в систему контроля версий Git.
2. Должен быть создан удалённый репозитории на сервисе GitHub, к которому преподаватели имеют доступ для проверки.

Отчетность

Выполненное задание должно сопровождаться документов, в котором должны быть предоставлены сведения о том, каким образом происходила реализация приложения, какие средства, компоненты, библиотеки использовались. Должны присутствовать диаграммы, описывающий процесс обработки внутри приложения. Эти диаграммы не обязательно должны соответствовать какой-либо нотации, основное их предназначение это визуализация черного ящика, которым является разработанное приложение.

Все проведенные испытания (тесты) также должны содержать описание входных и выходных данных - результатов. Все тесты должны быть воспроизводимы.

Исходный код не должен присутствовать в отчете, т.к. он будет размещен в репозитории со всей версионностью и всеми изменениями. Только если необходим привести отрывки кода с особенностями реализации.

Исходные данные

Здесь приведен пример файла, который необходимо создать на основании реальных данных. В примере видно базовое оформление файла и как необходимо вводить в него данные.

Пример входного файла в формате CSV:

```
Фамилия,Имя,Отчество,Математика,Физика,Химия,Черчение,Алгоритмизация
Иванов,Иван,Иванович,4,3,8,6,7
Петров,Петр,Петрович,7,7,8,9,10
```

Пример запроса с параметрами в консоли для запуска приложения с указанием файла для чтения и записи:

```
ConsoleApplication.exe -i InputFile.csv -o OutputFile -f Excel
```

или для формата JSON аналогичная конструкция, но уже с другим значением параметра Format (-f)

```
ConsoleApplication.exe -i InputFile.csv -o OutputFile -f JSON
```

Это всего лишь примеры и они не отражают реальную реализацию, которую можно выбрать в процессе разработки. Однако она должна быть общеиспользуемой и понятной.

Демонстрация работы

Необходимо продемонстрировать работу приложения, его запуск с тестовыми данными, показать исходные данные и результат. Оно должно успешно запускаться во всех режимах, не только в режиме отладки в среде разработки.

Приложение должно успешно запускаться, отработать и завершить работу.

Ссылки по теме

1. <https://docs.microsoft.com/en-us/dotnet/core/> - официальная документация.

Литература

1. .NET Core in Action - <https://www.manning.com/books/dotnet-core-in-action>
2. C# in Depth, Fourth Edition - <https://www.manning.com/books/c-sharp-in-depth-fourth-edition>
3. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов - <https://oz.by/books/more10716308.html>

4. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# - <https://oz.by/books/more1028671.html>
5. C# 5.0 и платформа .NET 4.5 для профессионалов - <https://oz.by/books/more1051976.html>