Программирование на платформе .Net

**Задание на самостоятельную работу.**

**Цель:** освоение методов работы с данными большого размера (которые не могут быть целиком загружены в оперативную память системы)

**Задача:**

Напишите 32-х битное приложение, которое генерирует файл с данными. Размер сгенерированного файла должен превышать некоторый предварительно сконфигурированный размер (например, 5Гб). Файл должен быть заполнен данными, представляющими информацию о студенте. Структура должна содержать следующую информацию о студенте: фамилия, имя, отчество, дата рождения, факультет, номер студенческого билета, номер специальности. Генерация данных должна производиться на основании некоторых словарей, представляющих перечень возможных значений, того или иного атрибута. Например, словарь фамилий, словарь названий факультетов.

Далее, приложение должно отсортировать записи в сгенерированном файле на основании некоторого атрибута записи (например, сначала по факультету, затем по фамилии, затем по дате рождения). Набор атрибутов, учитываемых в сортировке, определяется студентом самостоятельно, но не менее 3-х. Сортировку можно выполнять, используя любые внешние алгоритмы сортировки.

Для доступа к файлу необходимо использовать технологию Buffered Stream.

Для визуализации результата, приложение должно вывести первые X-записей исходного файла, затем Х-записей отсортированного файла (для вывода может быть использована консоль, либо другие способы визуализации).

**Дополнение.**

Необходимо объяснить, с чем связана необходимость использования технологии Buffered Stream и алгоритмов внешней сортировки в контексте данного задания, указать плюсы и минусы доступа к файлам с использованием технологии Buffered Stream, а также какие классы в .net реализуют данную технологию.

Результат должен быт формализован в виде текстового документа, приложенного к результирующему комиту, либо комментария непосредственно в файле program.cs

**Список литературы:**

1. Сортировка больших объёмов данных, реализация на Java: <https://habrahabr.ru/post/65612/>
2. Алгоритмы сортировки массивов. Внешняя сортировка: <http://www.intuit.ru/studies/courses/648/504/lecture/11473>
3. External merge sort: <https://blogs.msdn.microsoft.com/dhuba/2010/08/24/external-merge-sort/>
4. Файловый и потоковый ввод-вывод: <https://msdn.microsoft.com/ru-ru/library/k3352a4t(v=vs.110).aspx>
5. Fast Binary File Reading with C#: <http://www.codeproject.com/Articles/10750/Fast-Binary-File-Reading-with-C>
6. Efficient File I/O From C#: <https://designingefficientsoftware.wordpress.com/2011/03/03/efficient-file-io-from-csharp/>