Задание 13. Многопоточное получение и сохранение данных. *(Многопоточность. Синхронизация потоков).*

Реализуйте двухпоточное сохранение большого объема данных: один поток читает данные порциями, второй сохраняет в текстовый файл.

В качестве источника данных необходимо использовать специальный модуль FabulousCity.DataAccess.dll (dll и целиком проект прилагается). Модуль содержит класс PersonProvider – провайдер данных. Его метод

Person[] FabulousCity.DataAccess.PersonProvider.GetPersons(int idFrom, int count)

возвращает порцию данных: count записей, начиная с idFrom. Общая нумерация записей начинается с 1. При окончании данных метод вернет пустой массив.

```
PersonProvider provider = new PersonProvider();
Person[] persons = provider.GetPersons(1, 100);
```

Предполагается, что все данные сразу считать невозможно, не хватит оперативной памяти, поэтому читаемые блоки данных необходимо сохранять в ограниченной по длине очереди.

Используя класс Stopwatch, сравните скорость сохранения всех данных в однопоточном (получил блок, сохранил, получил - сохранил и т.д.) и многопоточном режимах.

Порядок полученных записей должен сохраняться (каждый экземпляр класса Person содержит свойство Id, при сохранении в файле порядок полученных Id должен сохраняться, переставлять записи нельзя)

Ограничения:

1. В учебных целях нельзя использовать коллекции из пространства имен System.Collections.Concurrent.
2. Суммарное количество объектов класса Person в очереди не должно превышать 100 000 экземпляров.
   Например, если очередь ограничена 100 элементами Person[], то каждый элемент очереди – массив Person[] – должен быть ограничен 1000.
   Если очередь ограничена 1000 элементами Person[], то каждый элемент очереди – массив Person[] – должен быть ограничен 100.
   Если очередь ограничена 100 000 элементами, то каждый элемент очереди – объект класса Person.
   Выбор соотношения этих параметров лежит на разработчике. Выбор должен быть обоснован.
   Для удобства максимальная длина очереди и размер блока в очереди должны задаваться константами.

Для желающих:

1. Дополнительно реализуйте алгоритм сохранения с использование стандартной коллекции `BlockingCollection<T>`
2. Есть также асинхронный метод получения массива Person в классе PersonProvider
   async Task<Person[]> GetPersonsAsync(int idFrom, int count)

Примечания:

Размер сохраняемого текстового файла будет порядка 850 Mb.

В классе Person переопределен метод ToString() для нормального вывода содержимого класса

Person является immutable record, экземпляр Person неизменяем.

Время получения порции Person непостоянно.