Лабораторная работа №2

Синхронизация данных для многопоточности

Работа с потоками в .NET

Работа с потоками
https://learn.microsoft.com/ru-ru/dotnet/standard/threading/using-threads-and-threading

Пул потоков
https://learn.microsoft.com/ru-ru/dotnet/api/system.threading.threadpool?view=net-8.0

Синхронизация потоков в .NET

Mutex
https://learn.microsoft.com/ru-ru/dotnet/api/system.threading.mutex?view=net-8.0

Semaphore
https://learn.microsoft.com/ru-ru/dotnet/api/system.threading.semaphore?view=net-8.0

Monitor (и в него встроены события)
https://learn.microsoft.com/ru-ru/dotnet/api/system.threading.monitor?view=net-8.0

Работа с потоками в Java

Работа с потоками
https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Thread.html

Пул потоков
https://www.baeldung.com/thread-pool-java-and-guava

Синхронизация потоков в Java

ReentrantLock
https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/concurrent/locks/ReentrantLock.html

Semaphore
https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/concurrent/Semaphore.html

Monitor
https://docs.oracle.com/javase/specs/jls/se8/html/jls-14.html#jls-14.19 *(п. 14.19)*

События
https://docs.oracle.com/javase/specs/jls/se8/html/jls-17.html#jls-17.2 *(п. 17.2)*

Задание

Разработать однопоточное и многопоточное приложение для решения СЛАУ на *C#* или *Java*. Входные данные задачи:

- матрица коэффициентов уравнения (*A*)
- вектор правых частей (*B*)

Количество задач (СЛАУ) - не менее 10-ти. Размерность задач - не менее 5000. Задачи описать построчно в текстовом файле. Например:

```
a1.csv,b1.csv
a2.csv,b2.csv
...
an.csv,bn.csv
```

где `an.csv`, `bn.csv` - файлы с входными данными задачи.

Реализовать приложение по следующей схеме:

- поток `Producer` выполняет чтение текстового файла с задачами и помещает их в очередь задач
- потоки `Consumer` берут задачи из очереди и выполняют их. Результаты кладут в отдельный список
- после завершения работы `Producer` и `Consumer` потоков результаты вычислений записываются в файлы `x1.csv`, `x2.csv`, ..., `xn.csv`

Примечание: `Producer` и `Consumer` потоки должны работать параллельно

Произвести замеры времени выполнения однопоточной и многопоточной реализации. Результаты замеров вывести на консоль.

Количество потоков должно быть равно количеству физических или логических ядер процессора, а при использовании пула потоков необходимо ограничить максимальное значение потоков числом физических или логических ядер + 4.

Таблица 1 - Варианты заданий

№	Метод решение СЛАУ	Способ реализации многопоточности	Механизм синхронизации
1	Решение СЛАУ методом Гаусса	Потоки	Semaphore
2	Решение СЛАУ методом Зейделя	Пул потоков	События
3	Решение СЛАУ методом Гаусса-Зейделя	Потоки	Monitor
4	Решение СЛАУ методом LL^t разложения	Пул потоков	Semaphore
5	Решение СЛАУ методом LU разложения	Потоки	События
6	Решение СЛАУ методом LDL^t разложения	Пул потоков	Monitor
7	Решение СЛАУ методом Гаусса	Потоки	Monitor
8	Решение СЛАУ методом Зейделя	Пул потоков	Mutex (ReentrantLock)
9	Решение СЛАУ методом Гаусса-Зейделя	Потоки	События
10	Решение СЛАУ методом LL^t разложения	Пул потоков	Monitor
11	Решение СЛАУ методом LU разложения	Потоки	Semaphore
12	Решение СЛАУ методом LDL^t разложения	Пул потоков	Mutex (ReentrantLock)
13	Решение СЛАУ методом LL^t разложения	Потоки	Mutex (ReentrantLock)
14	Решение СЛАУ методом LDL^t разложения	Пул потоков	События
15	Решение СЛАУ методом Гаусса	Потоки	Mutex (ReentrantLock)