



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА № 8
з дисципліни «Мова програмування Java»
Тема: «Parallel»

Виконала:

Студентка групи IA-31
Соколова Поліна

Мета роботи – ознайомлення з поняттям розпаралелювання та потоків виконання, практичне освоєння алгоритму запуску потоку виконання, синхронізації, блокування та спілкування між ними.

Middle Task:

1. Напишіть консольний додаток для обчислення суми всіх елементів масиву (з 1_000_000 цілих елементів, значення яких генеруються випадковим чином в діапазоні від 0 до 100), використовуючи фреймворк ForkJoin. Для цього:

- створіть та ініціалізуйте масив розмірністю 1_000_000 елементів;
- опишіть рекурсивну задачу для поділу масиву на дві частини і виклик дляожної частини рекурсивного завдання для подальшого розподілу;
- продовжуйте розподіл масиву, поки в підмасиві не залишиться менше 20 елементів, і тоді обчисліть суму цих елементів.

Хід роботи

Спочатку у програмі створюється масив із мільйоном елементів, значення генеруються випадковим чином у діапазоні від 0 до 100. Далі використовується ForkJoinPool, який керує групою потоків і відповідає за розподіл роботи між ними. Основна робота виконується у класі SumTask, що успадковує RecursiveTask<Long> і реалізує рекурсивний поділ масиву.

Якщо розмір підмасиву більший за встановлений поріг, тобто 20 елементів, підмасив ділиться на дві частини. Дляожної частини створюються окремі підзадачі. Одна з них запускається асинхронно за допомогою fork(), а друга обчислюється поточним потоком через compute(). Після завершення обчислень програма викликає join() для отримання результату першої частини та підсумовує результати обох підзадач.

Результат:

```
Сума елементів масиву: 49995416
Час виконання: 42 мс
```

Модульне тестування пройшло успішно:

✓ ParallelArraySumTest (org.example)	83 ms
✓ testArrayWithThresholdSize()	50 ms
✓ testRandomArrayComparison()	25 ms
✓ testSmallArray()	3 ms
✓ testArrayNegativeValues()	1 ms
✓ testLargeArray()	4 ms

Висновок: у результаті виконання лабораторної роботи я реалізувала паралельне обчислення суми елементів великого масиву елементів за допомогою фреймворку ForkJoin та на практиці переконалася, як рекурсивний поділ задачі і механізми fork() і join() дозволяють ефективно використовувати багатопотоковість і значно прискорюють обчислення порівняно з послідовним підходом.

Посилання на GitHub репозиторій:

https://github.com/sokolovapolina230/Java_labs/tree/main