

Хід роботи:

1. Пригадати API для здійснення паралельних обчислень. Особливу увагу звернути на такі інтерфейси, класи та методи:

- Runnable;
- Thread;
- run();
- start();
- join().

2. Знайти суму арифметичної прогресії

$$f(n, N) = n + 2n + 3n + 4n + 5n + \dots + Nn,$$

де n - номер варіанту,

N - число достатньо велике для того, щоб спосіб № 2 обчислював результат на протязі кількох секунд (з метою зменшення впливу похибки при замірах часу роботи програми),

наступними способами:

- 1) за допомогою формули розрахунку суми арифметичної прогресії;
- 2) «в лоб» за допомогою оператора циклу, що працює в одному треді;
- 3-7) «в лоб» за допомогою оператора циклу, що працює в декількох тредах ($k=2, 4, 8, 16, 32$).

Порівняти точність (має співпасти для усіх способів) та час t отримання результатів (для випадків $2 - 7$). У звіті навести таблицю та побудувати графік $t(k)$.

3. Відповісти на контрольні питання

Контрольні питання:

1. Чим процес відрізняється від треда?
2. В чому полягає різниця між кооперативною та витискальною багатозадачністю? Яка з цих моделей реалізована у Java?
3. Пояснити діаграму станів об'єкта класу Thread. Чим стан «Running» відрізняється від стану «Runnable»?
4. Коли і як слід використовувати методи run() та start() класу Thread?
5. Чому не слід використовувати deprecated-методи класу Thread?
6. Для чого потрібен метод join()?