14. Паралельне виконання. Ефективність використання

Мета: Вимірювання часу паралельних та послідовних обчислень. Демонстрація ефективності паралельної обробки.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кабак Олександр Русланович
- НТУ “ХПІ” 1.КІТ102.8а
- Варіант 5

1.2 Загальне завдання

- Забезпечити вимірювання часу паралельної обробки елементів контейнера за допомогою розроблених раніше методів.
- Додати до алгоритмів штучну затримку виконання для кожної ітерації циклів поелементної обробки контейнерів, щоб загальний час обробки був декілька секунд.
- Реалізувати послідовну обробку контейнера за допомогою методів, що використовувались для паралельної обробки та забезпечити вимірювання часу їх роботи.
- Порівняти час паралельної і послідовної обробки та зробити висновки про ефективність розпаралелювання:
  - результати вимірювання часу звести в таблицю;
  - обчислити та продемонструвати у скільки разів паралельне виконання швидше послідовного.

-

1.3 Задача

5. Прикладна галузь: Довідник покупця. Торговельна точка: назва; адреса; телефони (кількість не обмежена); спеціалізація; час роботи (з зазначенням днів тижня).

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі присутні:

1) Двозв'язний список, що параметризується;
2) Збереження та відновлення об'єктів без протоколу серіалізації та з ним;
3) Спілкування з користувачем за допомогою меню та автоматичний режим для перегляду;
4) Регулярні вирази для коректного запису даних в базу;
5) Пошук елементів за допомогою регулярних виразів;
6) Багатопоточність та паралельне виконання завдань (з використанням таймерів).

2.2 Важливі фрагменти програми

```
public static long cParallel() {
    long time_start = System.currentTimeMillis();
    System.out.println("Запуск всех потоков...");
    Thread1 first = new Thread1();
    Thread t1 = new Thread(first,  name: "Первий поток");

    Thread2 second = new Thread2();
    Thread t2 = new Thread(second,  name: "Второй поток");

    Thread3 third = new Thread3();
    Thread t3 = new Thread(third,  name: "Третий поток");

    t1.start();
    t2.start();
    t3.start();

    try {
        t1.join();
        t2.join();
        t3.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    System.out.println("Завершены все потоки...");
    return System.currentTimeMillis() - time_start;
}
```

Рис. 1 - Метод для виконання паралельної багатопоточності

```java
public static long comparisonSequential() {
    long time_start = System.currentTimeMillis();
    System.out.println("Запустить последовательность");
    try {
        ThreadHelper.test1();
        ThreadHelper.test2();
        ThreadHelper.test3();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    System.out.println("Последовательность окончена");
    return System.currentTimeMillis() - time_start;
}
```

Рис. 2 - Метод для виконання послідовно багатопоточності

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма дозволяє створювати об’єкт – список магазинів, що заносяться у запис каталогу. Користувач може додавати інші магазини до списку, видаляти елементи вибірково, а також очистити весь масив одним викликом відповідної кнопки меню, переглядати магазини за номерами телефону(короткими або українські). Також присутня можливість серіалізувати /десеріалізувати об’єкти з файлу, використовувати багатопоточність для вирішення певних задач.

```
Запуск всех потоков...
Первий поток запущен
Второй поток запущен
Третий поток окончен
Остальных номеров: 999011
Украинских номеров: 991
6-значних номеров: 925
Завершены все потоки...
Запустить последовательность
Первий поток окончен
6-значных номеров: 925
Второй поток окончен
Украинских номеров: 991
Третий поток окончен
Остальных номеров: 999011
Последовательность окончена



Времени затрачено на последовательное выполнение: 1320

Времени затрачена на паралельное выполнение: 801
```

Рис.3 - Порівняння послідовного виконання з паралельним при 1 млн об'єктів

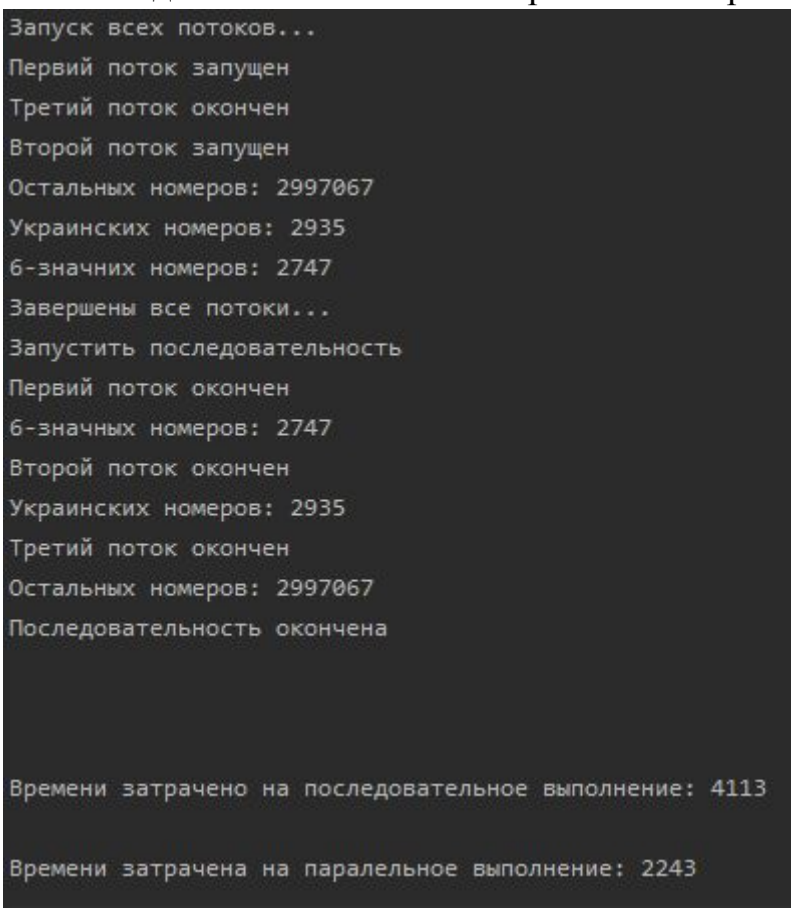

Рис. 4- Порівняння послідовного виконання з паралельним при 3 млн об'єктів

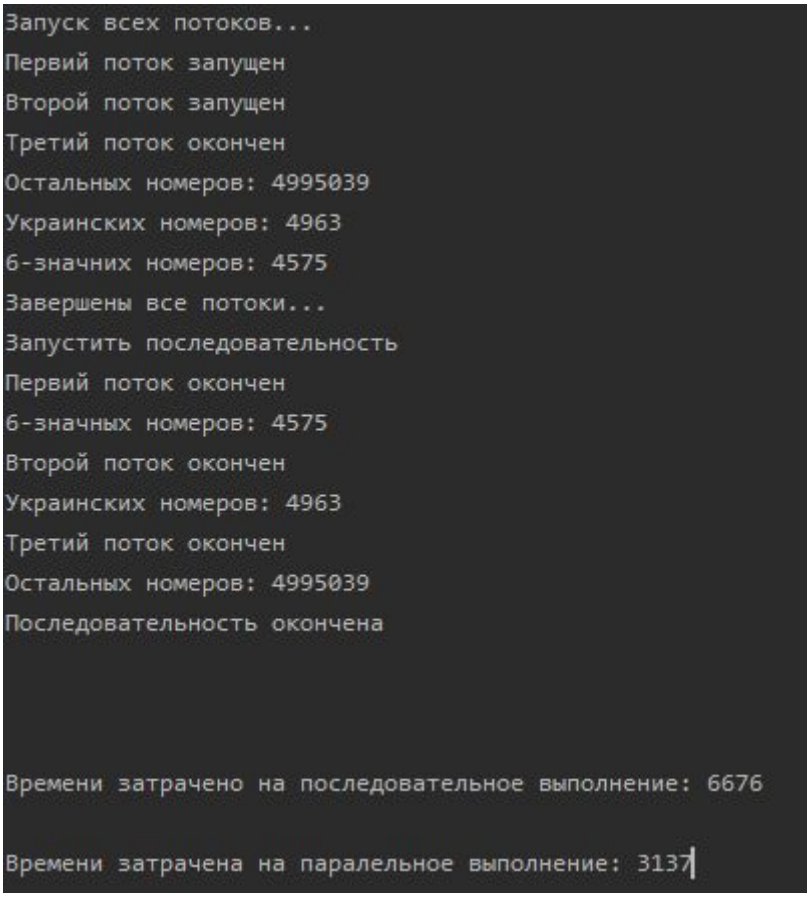

Рис.5 - Порівняння послідовного виконання з паралельним при 5 млн об'ектів

Отримуємо таблицю на основі 3 різних значень:

Кількість магазинів	Послідовне	Паралельне	Коефіцієнт
1 млн	1.320	0.801	1.64
3 млн	4.113	2.243	1.83
5 млн	6.676	3.137	2.12

Висновки: в даній лабораторній роботі створена та виконана паралельна та послідовна обробка елементів контейнера за допомогою класу Thread та створеного ThreadHelper для виконання умов задачі. Вимірявши час виконання роботи доведено , що паралельне виконання швидше ніж послідовне пропроціонально підвищенню к-сті об’ектів. Доведено, що паралельне виконання використовувати ефективніше.