13. Паралельне виконання. Багатопоточність

Мета: Ознайомлення з моделлю потоків Java. Організація паралельного виконання декількох частин програми.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Куйдін Михайло Андрійович
- НТУ “ХПІ” 1.КІТ102.8а
- Варіант 10

1.2 Загальне завдання

- Використовуючи програми рішень попередніх задач, продемонструвати можливість паралельної обробки елементів контейнера: створити не менше трьох додаткових потоків, на яких викликати відповідні методи обробки контейнера.

- Забезпечити можливість встановлення користувачем максимального часу виконання (таймаута) при закінченні якого обробка повинна припинятися незалежно від того знайдений кінцевий результат чи ні.

- Для паралельної обробки використовувати алгоритми, що не змінюють початкову колекцію.

- Кількість елементів контейнера повинна бути досить велика, складність алгоритмів обробки колекції повинна бути зіставна, а час виконання приблизно однаковий, наприклад:

  1. пошук мінімуму або максимуму;
  2. обчислення середнього значення або суми;
  3. підрахунок елементів, що задовольняють деякій умові;
  4. відбір за заданим критерієм;
  5. власний варіант, що відповідає обраній прикладної області.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі присутні об’єктно-орієнтовані методи:
Інкапсуляція – захист даних від неправомірного користування.

2.2 Ієрархія та структура даних

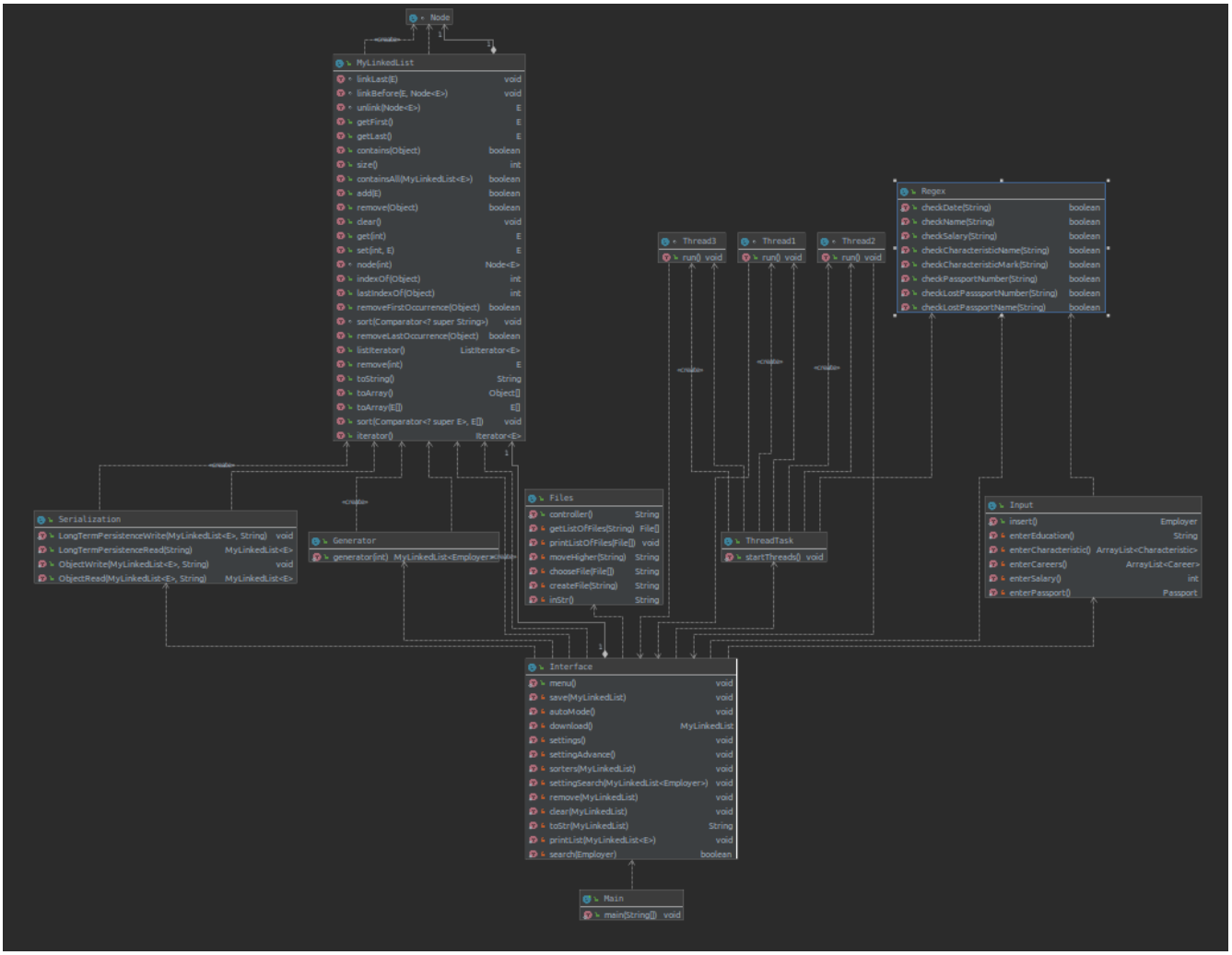

Рисунок 1 – Діаграма класів

2.3 Важливі фрагменти програми

```
class Thread1 implements Runnable {
    public void run() {
        int count = 0;
        System.out.println("First Thread started");
        try {
            for (Employer elem : Interface.object) {
                if (!Thread.currentThread().isInterrupted()) {
                    if (elem.getSalary() > count) {
                        count = elem.getSalary();
                    }
                } else {
                    throw new InterruptedException();
                }
            }
            System.out.println("Max salary : " + count);
        } catch (InterruptedException e) {
            System.err.println(e.getMessage());
        }
    }
}
```

Рисунок 2 – Приклад створеної нитки

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма дозволяє створювати об’єкти – “записи в розкладі” , що заносяться у запис каталогу, тобто створюється масив об’єктів. Користувач може додавати об’єкти до масиву, видаляти елементи вибірково, а також очистити увесь масив одним викликом відповідної кнопки меню. Також присутня можливість серіалізувати/десеріалізувати об’єкти з файлу.

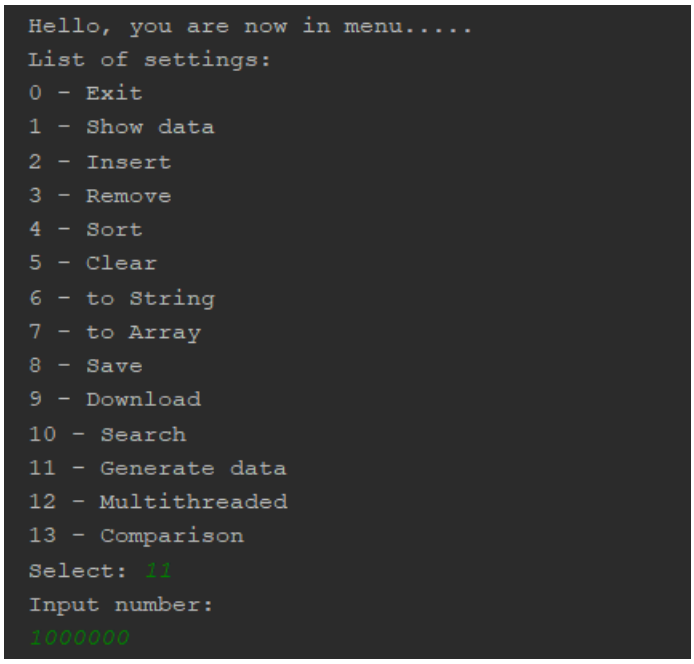

Рисунок 3 – Створюємо мільйон елементів, для тестування

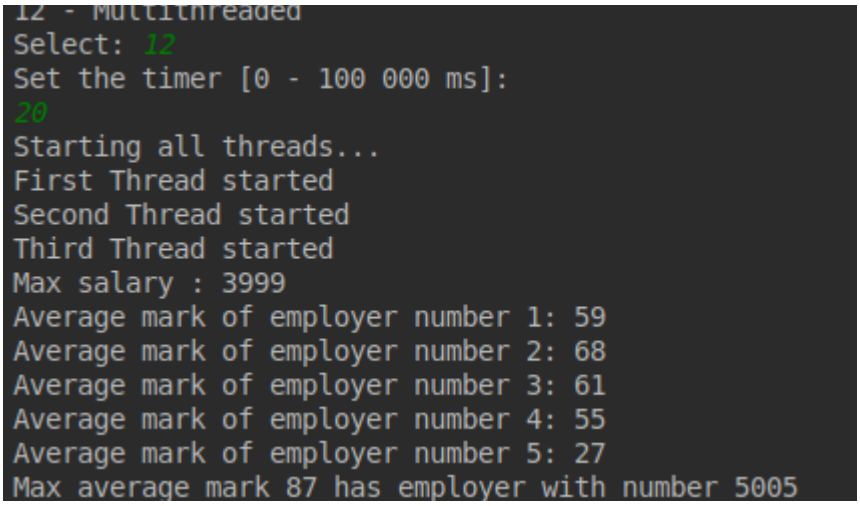

Рисунок 4 – початок роботи та результат

4. ВИСНОВКИ

В даній лабораторній роботі були ознайомлені з механізмом багатопотоковості для декількох функцій програми.