

13. Паралельне виконання. Багатопоточність

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

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кононенко Дмитро Олексійович
- НТУ “ХПІ” 1.KIT102.8a
- Варіант 8

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

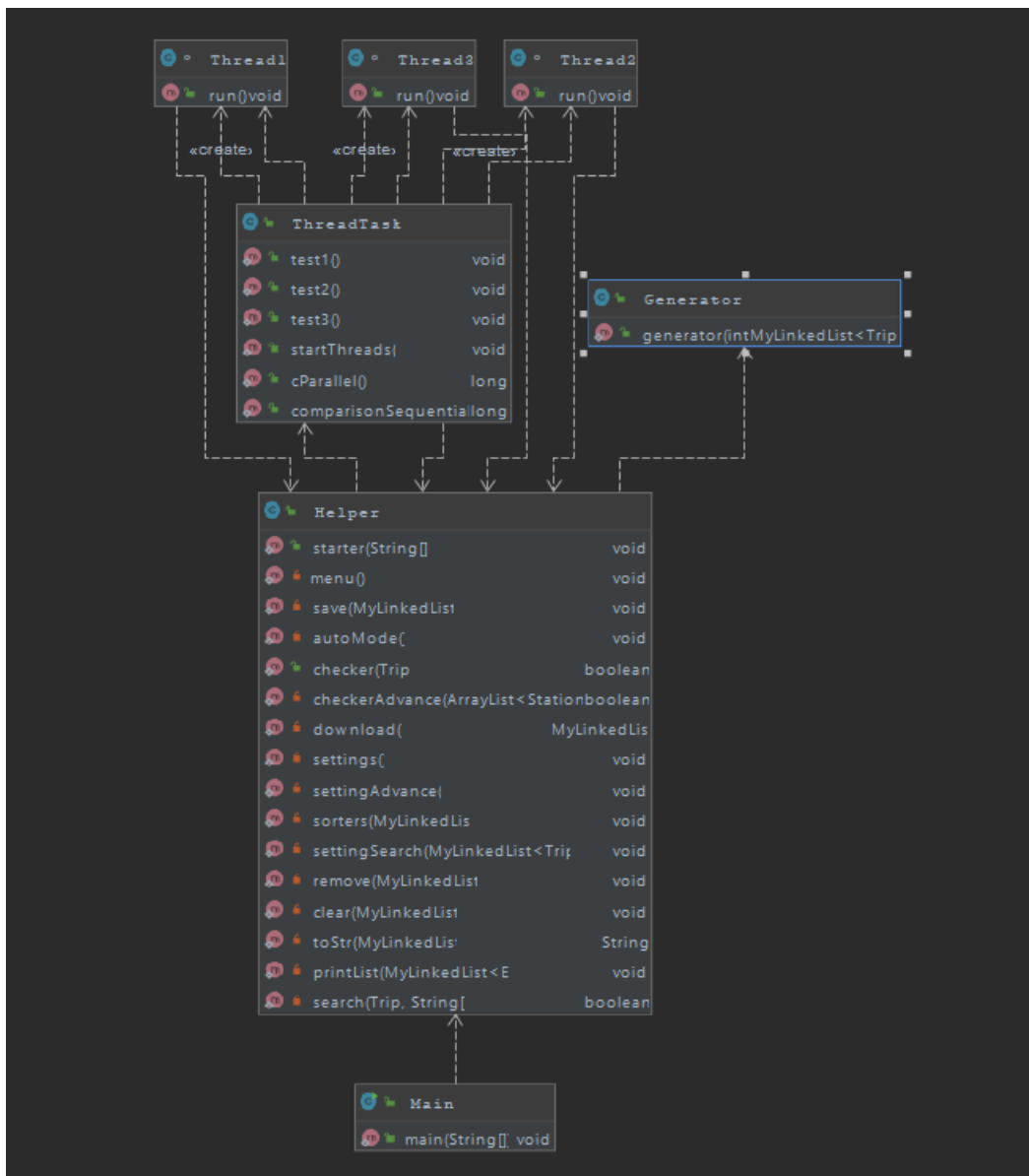


Рисунок 1 – Діаграма класів

2.3 Важливі фрагменти програми

```
class Thread1 implements Runnable {  
    public void run() {  
        int count = 0;  
        System.out.println("First Thread started");  
        try {  
            for (Trip elem : Helper.object) {  
                if (!Thread.currentThread().isInterrupted()) {  
                    if (Regex.checkWeekends(elem.getDays())) {  
                        count++;  
                    }  
                } else {  
                    throw new InterruptedException();  
                }  
            }  
            System.out.println("Number of trips in weekends : " + 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.

```
Hello, you are now in menu.....
List of settings:
0 - Exit
1 - Show data
2 - Insert
3 - Remove
4 - Sort
5 - Clear
6 - to String
7 - to Array
8 - Save
9 - Download
10 - Search
11 - Generate data
12 - Multithreaded
13 - Comparison
Select: 11
Input number:
1000000
```

Рисунок 3 – Створюємо мільйон елементів, для тестування

```
Set the timer [0 - 100 000 ms]:
Input number:
10000
Starting all threads...
First Thread started
Third Thread started
Second Thread started
Earliest trip is at : 00:00
Average number of sits: 37
Number of trips in weekends : 166763
Finishing all threads...
```

Рисунок 4 – початок роботи та результат

4. ВИСНОВКИ

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