

ЛАБОРАТОРНА РОБОТА №13
ПАРАЛЕЛЬНЕ ВИКОНАННЯ. БАГАТОПОТОЧНІСТЬ

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

1 ВИМОГИ

1.1 Розробник

- Котенко Сергій Миколайович;
- Студент групи КІТ-118в;
- Варіант 9

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 якої обробка повинна припинятися незалежно від того знайдений кінцевий результат чи ні.

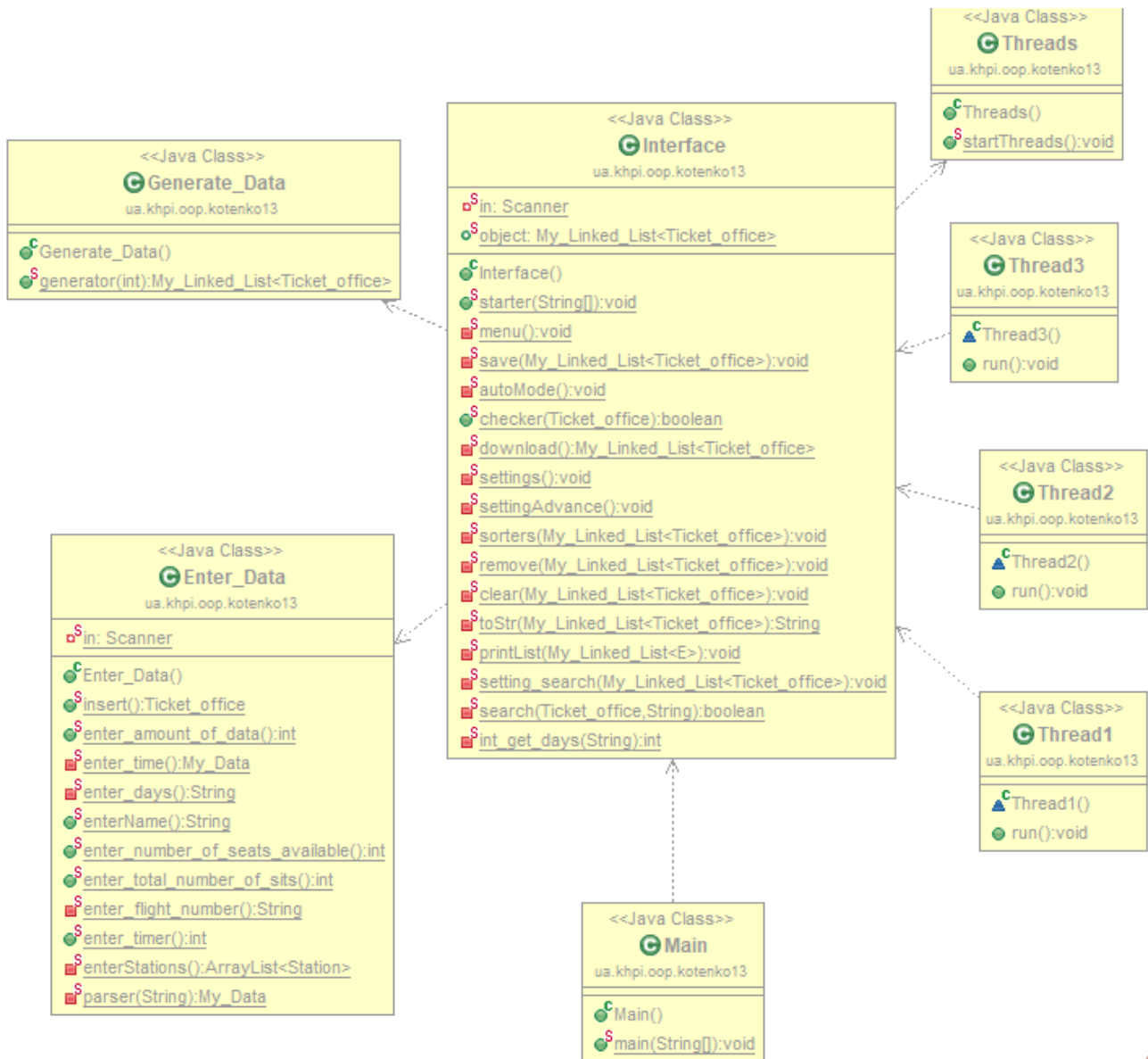
Для паралельної обробки використовувати алгоритми, що не змінюють початкову колекцію.

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

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

2 ОПИС ПРОГРАМИ

2.1 Опис логічної структури



2.2 Фрагменти коду

```
class Thread1 implements Runnable {  
    public void run() {  
        int count = 0;  
        System.out.println("First Thread: ");  
        try {  
            for (Ticket_office elem : Interface.object) {  
                if (!Thread.currentThread().isInterrupted()) {  
                    if (Regex.check_work_days(elem.get_days())) {  
                        count++;  
                    }  
                } else {  
                    throw new InterruptedException();  
                }  
            }  
            System.out.println("Number of trips in monday and friday : " + count);  
        } catch (InterruptedException e) {  
            System.err.println(e.getMessage());  
        }  
    }  
}
```

Рисунок 2.1 – Створений потік

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.1 Ілюстрація роботи програми

```
Menu:  
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
```

Рисунок 3.1 – Меню програми

```
Select: 11  
Enter amount of data:  
1000000
```

Рисунок 3.2 – Створення елементів для перевірки

```
Select: 12
Set the timer [0 - 100 000 ms]:
100000

Second Thread:
Third Thread:
First Thread:
Average number of available sits: 14
First trip is at : 00:00
Number of trips in monday and friday : 333331
```

Рисунок 3.3 – Результат роботи багато потоковості

ВИСНОВОК

В інтегрованому середовищі *Eclipse* розроблена програма мовою Java. Виконання програми дозволяє продемонструвати роботу багато потоковості для декількох функцій програми.