## 13. Паралельне виконання. Багатопоточність

**Мета:** Ознайомлення з моделлю потоків *Java*.

Організація паралельного виконання декількох частин програми.

**1 ВИМОГИ**

**1.1 Розробник**

Інформація про розробника:

* Кулик Данііл Ігорович
* НТУ “ХПІ” КІТ-118в
* Варіант 11

**1.2 Загальне завдання**

1. Використовуючи програми рішень попередніх задач, продемонструвати можливість паралельної обробки елементів контейнера: створити не менше трьох додаткових потоків, на яких викликати відповідні методи обробки контейнера.
2. Забезпечити можливість встановлення користувачем максимального часу виконання (тайм-аута) при закінченні якого обробка повинна припинятися незалежно від того знайдений кінцевий результат чи ні.
3. Для паралельної обробки використовувати алгоритми, що не змінюють початкову колекцію.
4. 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 Засоби ООП**

**Були розроблені додаткові функції для класу-контейнера. Розроблене діалогове меню та можливість зчитування даних з файлу.**

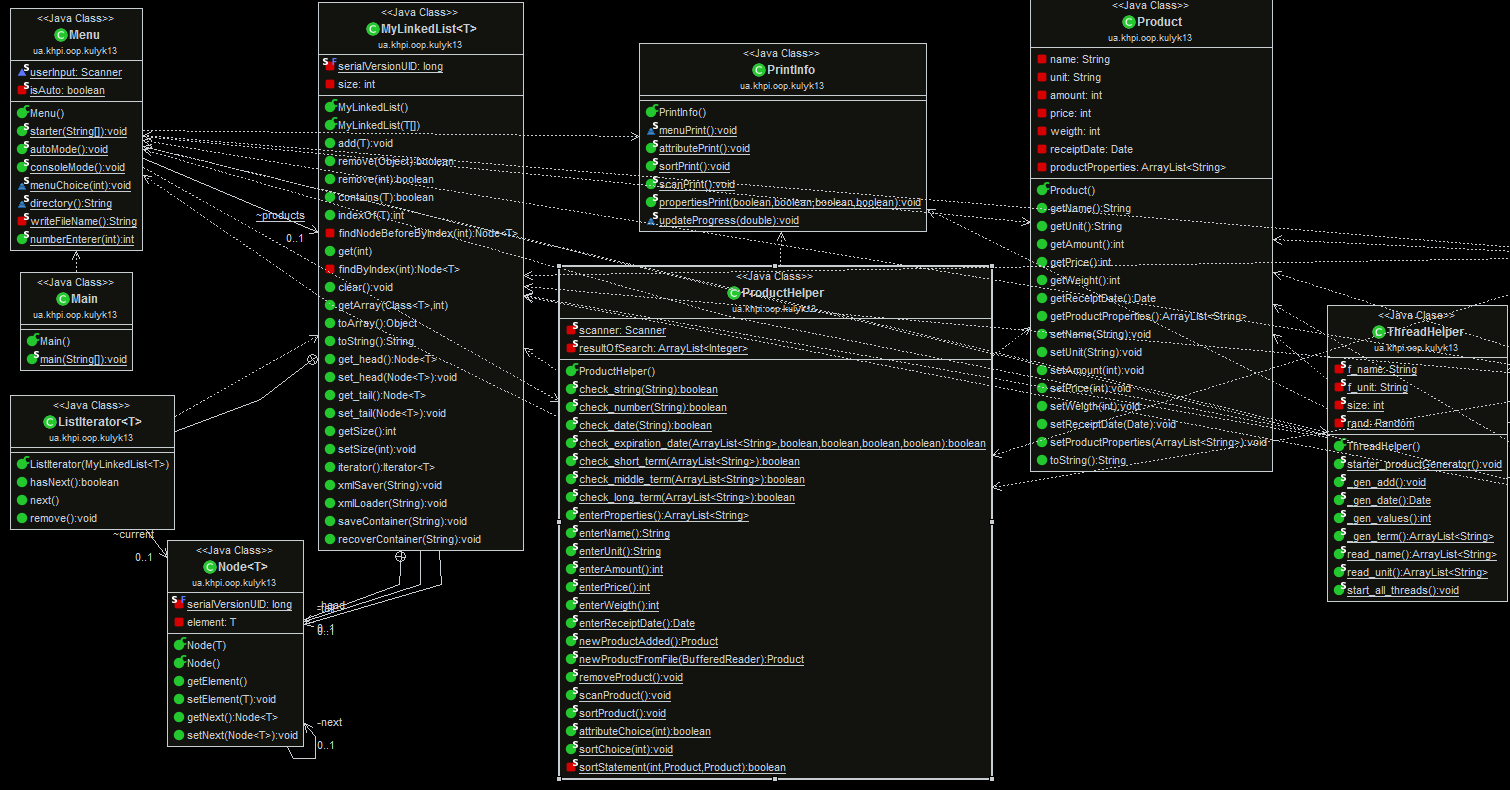
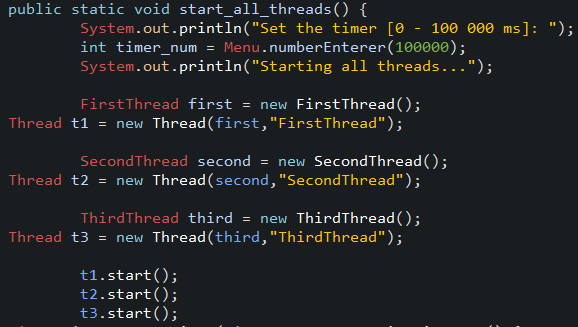
**2.2 Ієрархія та структура даних** Рисунок 1 – Діаграма класів **2.3 Важливі фрагменти програми** 

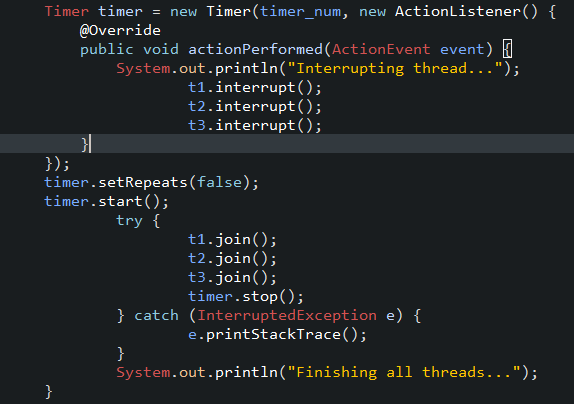
Рисунок 2 – Запуск трьох потоків   
  


Рисунок 3 – **Створення таймеру**

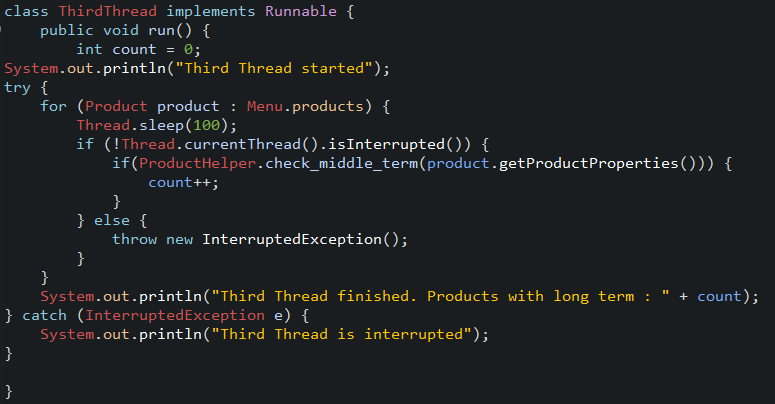


Рисунок 4 – **Один із потоків**

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

Програма дозволяє створювати об’єкти – товари – , 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.

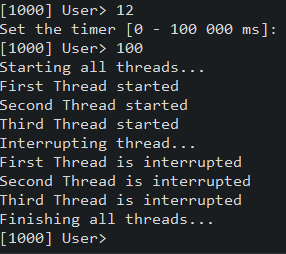


Рисунок 5 – Спрацьовування потоків у випадку не встигання таймеру  
 **ВИСНОВКИ**

В даній лабораторній роботі **організував паралельне виконання декількох частин програми.**