

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

**Лабораторна робота №6**  
**Паралельне виконання. Багатопоточність.**  
**Ефективність використання.**  
з курсу  
“Об’єктно-орієнтоване програмування”

Виконав:  
студент КН-111  
Петров Дмитро Денисович  
Викладач:  
Гасько Р.Т.

Львів – 2019 р.

## Мета

- Ознайомлення з моделлю потоків Java .
- Організація паралельного виконання декількох частин програми.
- Вимірювання часу паралельних та послідовних обчислень.
- Демонстрація ефективності паралельної обробки.

### 1. Вимоги

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 (таймаута) 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.
5. Забезпечити вимірювання часу паралельної обробки елементів контейнера за допомогою розроблених раніше методів.
6. Додати до алгоритмів штучну затримку виконання для кожної ітерації циклів поелементної обробки контейнерів, щоб загальний час обробки був декілька секунд.
7. Реалізувати послідовну обробку контейнера за допомогою методів, що використовувались для паралельної обробки та забезпечити вимірювання часу їх роботи.
8. 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.

## 2. Опис програми

Ієрархія класів:

- ThreadsTest

## 3. ВИСНОВКИ

Мету було досягнуто в повній мірі. Ресурси до лабораторної роботи знаходяться за посиланням:

<https://github.com/DmytroPetrov/OOPLab>