# Тема роботи

Багатопоточність

**Мета**

Ознайомлення з бібліотекою колекцій Java SE
Використання колекцій для розміщення обєктів розроблених класів

**Загальне завдання**
**Вимоги**

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.

## Опис програми

Було запущено 3 основних потоки з обробкою великих стрічок і порівняно виконаннння їх в двох режимах

## Важливі фрагенти програми

```
public static void main(String[] args) throws InterruptedException{
    Scanner scan = new Scanner(System.in);
    System.out.println("Type times tasks will proceed: ");
    int times = scan.nextInt();
    Task1 t1 = new Task1(times);
    Task2 t2 = new Task2(times);
    Task3 t3 = new Task3(times);
    System.out.println("m - MultiPotochniy \ng - Postupovo");
    char ch = scan.next().charAt(0);
    System.out.println("Type limit time in ms: ");
    float limit = scan.nextFloat();
```

## Приклад використання програми

```
Type times tasks will proceed:
400
m - MultiPotochniy
g - Postupovo
m
Type limit time in ms:
2000
Task 1 proceed 400 times
Task 2 proceed 400 times
Task 3 proceed 400 times
MultiPotok - 9.463013

Process finished with exit code 0
```

```
Type times tasks will proceed:
400
m - MultiPotochniy
g - Postupovo
g
Type limit time in ms:
2000
Task 1 proceed 400 times
Task 2 proceed 400 times
Task 3 proceed 400 times
Postupovo - 7.73551

Process finished with exit code 0
```