

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ  
УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №14**

**Колекції в Java.**

**Мета з дисципліни.**

**«Об’єктно-орієнтоване програмування»**

**Виконала:**

студентка групи КН-108

Гетьман Соломії

**Викладач:**

Гасько Р.Т.

# Мета

- Ознайомлення з бібліотекою колекцій Java SE
- Використання колекцій для розміщення об'єктів розроблених класів.

# Задача

1. Розробити консольну програму для реалізації завдання обробки
2. Для розміщення та обробки даних використовувати контейнери (колекції) і алгоритми
3. Забезпечити обробку колекції об'єктів: додавання, видалення, пошук, сортування згідно
4. Передбачити можливість довготривалого зберігання даних: 1) за допомогою стандартної серіалізації; 2) не використовуючи протокол серіалізації.
5. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах за результатом обробки параметрів командного рядка.

# Код програми

```
package
ua.lpnuai.oop.hetman13;

import java.util.Random;
class Summation extends Thread {
    private int[] arr;
    private int low, high, partial;
    public Summation(int[] arr, int low, int high)
    {
        this.arr = arr;
        this.low = low;
        this.high = Math.min(high, arr.length);
    }
    public int getPartialSum()
    {
        return partial;
    }
    public void run()
    {
        partial = sum(arr, low, high);
    }
    public static int sum(int[] arr)
    {
```

```

        return sum(arr, 0, arr.length);
    }
    public static int sum(int[] arr, int low, int high)
    {
        int total = 0;
        for (int i = low; i < high; i++) {
            total += arr[i];
        }
        return total;
    }
    public static int parallelSum(int[] arr)
    {
        return parallelSum(arr,
Runtime.getRuntime().availableProcessors());
    }
    public static int parallelSum(int[] arr, int threads)
    {
        int size = (int) Math.ceil(arr.length * 1.0 / threads);
        Summation[] sums = new Summation[threads];
        for (int i = 0; i < threads; i++) {
            sums[i] = new Summation(arr, i * size, (i + 1) * size);
            sums[i].start();
        }
        try {
            for (Summation sum : sums) {
                sum.join();
            }
        } catch (InterruptedException e) { }
        int total = 0;
        for (Summation sum : sums) {
            total += sum.getPartialSum();
        }
        return total;
    }
}

public class Main {
    public static final String ANSI_RESET = "\u001B[0m";
    public static final String ANSI_RED = "\u001B[31m";
    public static final String ANSI_GREEN = "\u001B[32m";
    public static final String ANSI_PURPLE = "\u001B[35m";
    public static final String ANSI_CYAN = "\u001B[36m";
    public static void main(String[] args) {
        Random ran = new Random();
        int[] arr = new int[100000000];
        for (int i = 0; i < 100000000; i++){
            arr[i] = ran.nextInt(100) + 1;
        }
        long start = System.currentTimeMillis();
        System.out.println(Summation.sum(arr));
        System.out.println(ANSI_RED + "Single: " +

```

```

(System.currentTimeMillis() - start) + " milliseconds" + ANSI_RESET);
    start = System.currentTimeMillis();
    System.out.println(Summation.parallelSum(arr));
    System.out.println(ANSI_GREEN + "Parallel: " +
(System.currentTimeMillis() - start) + " milliseconds" + ANSI_RESET);
    start = System.currentTimeMillis();
    try{
        Thread.sleep(1000);
    } catch(InterruptedException e){}
    System.out.println(Summation.parallelSum(arr));
    System.out.println(ANSI_PURPLE + "Parallel with sleeping for 1
sec: " + (System.currentTimeMillis() - start) + " milliseconds" +
ANSI_RESET);
    start = System.currentTimeMillis();
    try{
        Thread.sleep(2000);
    } catch(InterruptedException e){}
    System.out.println(Summation.parallelSum(arr));
    System.out.println(ANSI_CYAN + "Parallel with sleeping for 2
sec: " + (System.currentTimeMillis() - start) + " milliseconds" +
ANSI_RESET);
    }
}

```