**Міністерство освіти України**

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет інформатики та обчислювальної техніки**

**ЗВІТ**

до лабораторної роботи № 3

на тему “Неблокуючі алгоритми”

з дисципліни “Технології паралельних та розподілених обчислень”

Виконала:

Студентка групи ІП-71

Каспрук Анастасія Андріївна

Київ 2020

**Програмний код:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading;

namespace Lab3

{

class Program

{

static void Main(string[] args)

{

var random = new Random();

int randomNumbersCount = 10;

var randomNumbers = new List<long>(randomNumbersCount);

for(int i = 0; i < randomNumbersCount; i++)

randomNumbers.Add(random.Next(-100, 100));

Console.WriteLine($"Зненерований список чисел: {string.Join(" ", randomNumbers)}");

Console.WriteLine();

Console.WriteLine("Перевiрка роботи першої функцiї");

Console.WriteLine($"Кiлькiсть парних чисел: {CalcElementsOnCondition(randomNumbers, n => n % 2 == 0)}");

Console.WriteLine($"Кiлькiсть непарних чисел: {CalcElementsOnCondition(randomNumbers, n => n % 2 != 0)}");

Console.WriteLine();

Console.WriteLine("Перевiрка роботи другої функцiї");

(long min, long minIndex, long max, long maxIndex) = CalcMinAndMaxWithIndices(randomNumbers);

Console.WriteLine($"Мiнiмальний елемент {min} знаходиться пiд iндексом {minIndex}");

Console.WriteLine($"Максимальний елемент {max} знаходиться пiд iндексом {maxIndex}");

Console.WriteLine();

Console.WriteLine("Перевiрка роботи третьої функцiї");

Console.WriteLine($"Контрольна сума для чисел {CalcControlSum(randomNumbers)}");

Console.WriteLine($"Перевiрка контрольної суми послiдовним алгоритмом: {randomNumbers.Aggregate((f, s) => f ^ s)}");

}

// Паралельна функцiя для знаходження кiлькостi елементiв за умовою

public static long CalcElementsOnCondition(IEnumerable<long> elementsList, Func<long, bool> selector)

{

long elementsCount = 0;

elementsList.AsParallel().ForAll(e =>

{

if(selector(e))

Interlocked.Increment(ref elementsCount);

});

return elementsCount;

}

// Паралельна функцiя для знаходження мiнiмального та максимального елементiв з iндексами

public static (long min, long minIndex, long max, long maxIndex) CalcMinAndMaxWithIndices(IEnumerable<long> elementsList)

{

long min = 0, minIndex = 0, max = 0, maxIndex = 0;

elementsList.AsParallel().Select((elem, index) =>

{

if(elem < Interlocked.Read(ref min))

{

Interlocked.Exchange(ref min, elem);

Interlocked.Exchange(ref minIndex, index);

}

if(elem > Interlocked.Read(ref max))

{

Interlocked.Exchange(ref max, elem);

Interlocked.Exchange(ref maxIndex, index);

}

return elem;

}).ToList();

return (min, minIndex, max, maxIndex);

}

// Паралельна функцiя для знаходження контрольної суми iз використанням XOR

public static long CalcControlSum(IEnumerable<long> elementsList)

{

long controlSum = 0;

elementsList.AsParallel().ForAll(e =>

{

long curSumValue = Interlocked.Read(ref controlSum);

Interlocked.Exchange(ref controlSum, curSumValue ^ e);

});

return controlSum;

}

}

}

**Результат виконання:**

