**Міністерство освіти і науки України**

**Національний університет харчових технологій**

**кафедра інформаційних технологій,  
штучного інтелекту і кібербезпеки**

**Звіт**

із лабораторної роботи № 4

з дисципліни «Паралельне програмування»

на тему: «Багатопоточна програма з графічним інтерфейсом користувача»

Варіант 6

Виконала:

Студентка групи КН-3-2

Захарченко Альона

Перевірив:

к.т.н., доц. Костіков М. П.

Київ – 2023

**Опис**

Ця програма реалізує паралельні обчислення суми арифметичної прогресії з використанням паралельного циклу Parallel.For бібліотеки .NET. Основний функціонал включає в себе створення графічного інтерфейсу з вкладкою "Програма", де користувач може встановити кількість ітерацій та кількість потоків для паралельних обчислень. Також є можливість змінювати колір фону вкладки.

**Хід роботи**

1. **Запуск програми:**

При запуску програми створюється головне вікно з двома вкладками: "Програма" та "Про автора".

На вкладці "Програма" відображається формула A:1 + 2 + 3 + 4 + 5 + 6 ....

1. **Встановлення параметрів обчислень:**

Користувач може встановити кількість ітерацій (J) та кількість потоків для паралельних обчислень.

Також є можливість вибрати колір фону вкладки ("Червоний", "Зелений", "Синій").

1. **Запуск паралельних обчислень:**

Після встановлення параметрів користувачем, натискання кнопки "Запустити паралельні обчислення" ініціює запуск паралельних обчислень.

Кнопка блокується під час обчислень для уникнення ненавмисних запусків.

1. **Паралельні обчислення:**

Використовуючи Parallel.For, програма розділяє обчислення між вказаною кількістю потоків.

Кожен потік обчислює свою частину суми арифметичної прогресії та виводить результат в текстове поле.

1. **Вивід результатів:**

Після завершення паралельних обчислень програма виводить загальний час виконання та результат суми.

Інформація про кожен потік та його внесок у вивід виводиться в текстове поле.

1. **Завершення обчислень:**

Кнопка "Запустити паралельні обчислення" розблоковується після завершення обчислень.

Результати виводяться у вигляді суми арифметичної прогресії та витрат часу на обчислення.

1. **Зміна кольору вкладки:**

Вибір колір фону впливає на зовнішній вигляд вкладки та текстового поля.

1. **Інформація про автора:**

На вкладці "Про автора" виводиться інформація про автора програми.

1. **Наведення та відведення курсора:**

При наведенні на кнопку "Запустити паралельні обчислення" змінюється колір кнопки для покращення візуального зворотного зв'язку.

Програма взаємодіє з користувачем через графічний інтерфейс, виводячи результати паралельних обчислень та надаючи зручний інтерфейс для введення параметрів обчислень.

**Код програми**

using System;

using System.Drawing;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace zakharchenko\_kn\_3\_2\_lab4

{

public partial class Form1 : Form

{

private bool calculationsRunning = false;

private TabControl tabControl;

private TabPage tabPageProgram;

private NumericUpDown numericUpDownIterations;

private ComboBox comboBoxThreads;

private Button buttonStart;

private TextBox textBoxComments;

public Form1()

{

InitializeComponent();

InitializeUI();

}

private void InitializeUI()

{

// Заголовок головного вікна

this.Text = "Лабораторна робота 4, 2023 рік";

// Додати вкладки

tabControl = new TabControl();

tabControl.Dock = DockStyle.Fill;

tabPageProgram = new TabPage("Програма");

tabPageProgram.BackColorChanged += TabPageColorChanged; // Обробник зміни кольору фону вкладки

InitializeProgramTab();

TabPage tabPageAuthor = new TabPage("Про автора");

tabPageAuthor.Controls.Add(CreateAuthorLabel());

tabControl.TabPages.Add(tabPageProgram);

tabControl.TabPages.Add(tabPageAuthor);

this.Controls.Add(tabControl);

}

private void InitializeProgramTab()

{

tabPageProgram.Controls.Add(new Label { Text = "A:1 + 2 + 3 + 4 + 5 + 6 ....", Location = new Point(10, 10) });

Label labelIterations = new Label { Text = "Кількість ітерацій J:", Location = new Point(10, 40) };

numericUpDownIterations = new NumericUpDown { Minimum = 2, Maximum = int.MaxValue, Value = 1000000, Location = new Point(150, 40) };

Label labelThreads = new Label { Text = "Кількість потоків:", Location = new Point(10, 70) };

comboBoxThreads = new ComboBox { Items = { 2, 4, 8, 16 }, SelectedItem = 2, Location = new Point(150, 70) };

buttonStart = new Button { Text = "Запустити паралельні обчислення", Location = new Point(10, 100) };

buttonStart.MouseEnter += ButtonStart\_MouseEnter;

buttonStart.MouseLeave += ButtonStart\_MouseLeave;

buttonStart.Click += (sender, e) => { StartParallelCalculations((int)numericUpDownIterations.Value, (int)comboBoxThreads.SelectedItem); };

Label labelColors = new Label { Text = "Колір фону вкладки:", Location = new Point(10, 130) };

RadioButton radioButtonRed = new RadioButton { Text = "Червоний", Location = new Point(150, 130) };

RadioButton radioButtonGreen = new RadioButton { Text = "Зелений", Location = new Point(150, 160) };

RadioButton radioButtonBlue = new RadioButton { Text = "Синій", Location = new Point(150, 190) };

radioButtonRed.CheckedChanged += (sender, e) => { tabPageProgram.BackColor = Color.Red; };

radioButtonGreen.CheckedChanged += (sender, e) => { tabPageProgram.BackColor = Color.Green; };

radioButtonBlue.CheckedChanged += (sender, e) => { tabPageProgram.BackColor = Color.Blue; };

textBoxComments = new TextBox { Multiline = true, ScrollBars = ScrollBars.Vertical, Location = new Point(10, 220), Size = new Size(400, 150) };

tabPageProgram.Controls.AddRange(new Control[] { labelIterations, numericUpDownIterations, labelThreads, comboBoxThreads, buttonStart, labelColors, radioButtonRed, radioButtonGreen, radioButtonBlue, textBoxComments });

}

private Label CreateAuthorLabel()

{

Label labelAuthor = new Label

{

Text = "Захарченко Альона Анатоліївна кн-3-2",

Font = new Font("Arial", 16, FontStyle.Bold),

TextAlign = ContentAlignment.MiddleCenter,

Dock = DockStyle.Fill

};

return labelAuthor;

}

private void StartParallelCalculations(int iterations, int threads)

{

if (calculationsRunning)

{

MessageBox.Show("Обчислення вже запущені.");

return;

}

calculationsRunning = true;

buttonStart.Enabled = false; // Блокуємо кнопку під час обчислень

textBoxComments.AppendText($"Початок паралельних обчислень. Кількість потоків: {threads}. Загальна кількість ітерацій: {iterations}.\r\n");

// Засікаємо час початку обчислень

DateTime startTime = DateTime.Now;

// Запускаємо паралельні обчислення

Parallel.For(0, threads, i =>

{

CalculateInParallel(i, iterations / threads);

});

// Засікаємо час завершення обчислень

DateTime endTime = DateTime.Now;

// Обчислюємо затрачений час

TimeSpan elapsedTime = endTime - startTime;

// Виводимо результати

textBoxComments.AppendText($"Обчислення завершено. Затрачений час: {elapsedTime.TotalMilliseconds} мс.\r\n");

textBoxComments.AppendText($"Результат: {CalculateSum(iterations)}\r\n");

// Розблокуємо кнопку після завершення обчислень

buttonStart.Enabled = true;

// Прокручуємо вміст текстового поля до останнього результату

textBoxComments.SelectionStart = textBoxComments.Text.Length;

textBoxComments.ScrollToCaret();

calculationsRunning = false;

}

private void CalculateInParallel(int threadNumber, int iterations)

{

double result = 0;

// Реалізуємо обчислення за формулою A у паралельних потоках

for (int i = 1 + threadNumber; i <= iterations; i += Environment.ProcessorCount)

{

result += i;

}

// Виводимо результати в текстове поле

textBoxComments.Invoke((MethodInvoker)delegate

{

textBoxComments.AppendText($"Потік {threadNumber + 1}: Сума від {1 + threadNumber} до {iterations} = {result}\r\n");

});

}

private double CalculateSum(int iterations)

{

double result = 0;

// Реалізуємо обчислення суми арифметичної прогресії

for (int i = 1; i <= iterations; i++)

{

result += i;

}

return result;

}

private void TabPageColorChanged(object sender, EventArgs e)

{

// Оновлюємо колір текстового поля при зміні кольору вкладки

textBoxComments.BackColor = ((Control)sender).BackColor;

}

private void ButtonStart\_MouseEnter(object sender, EventArgs e)

{

// Змінюємо колір фону та шрифту кнопки при наведенні курсора

Button button = (Button)sender;

button.BackColor = Color.DarkOrange;

button.ForeColor = Color.White;

}

private void ButtonStart\_MouseLeave(object sender, EventArgs e)

{

// Повертаємо колір фону та шрифту кнопки до звичайного після виходу курсора

Button button = (Button)sender;

button.BackColor = SystemColors.Control;

button.ForeColor = SystemColors.ControlText;

}

}

}

**Скріншоти роботи програми**











