**Міністерство освіти і науки України**

**Національний університет харчових технологій**

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

**Звіт**

із лабораторної роботи № 6

з дисципліни «Паралельне програмування»

на тему: «Синхронізація потоків у багатопоточній програмі»

Варіант 6

Виконала:

Студентка групи КН-3-2

Захарченко Альона

Перевірив:

к.т.н., доц. Костіков М. П.

Київ – 2023

**Опис**

Ця програма на C# використовує Windows Forms для створення графічного інтерфейсу користувача. Вона призначена для генерації випадкового масиву чисел, їх сортування та збереження у файл. Програма також підтримує можливість скасування обчислень у випадку потреби.

**Хід роботи**

1. Інтерфейс користувача:

Користувач бачить вікно з кнопками "Запуск" та "Скасувати", міткою "Довжина масиву M:" та полем для введення довжини масиву. Також є текстове поле для виведення коментарів та статусів.

1. Введення довжини масиву:

Користувач може ввести бажану довжину масиву в поле "Довжина масиву M:". Введене значення перевіряється на коректність та мінімальну допустиму величину.

1. Запуск генерації та сортування:

Після введення довжини масиву користувач натиска на кнопку "Запуск".

Додаток створює два окремих потоки: один для генерації випадкового масиву чисел, інший для сортування цього масиву.

1. Генерація чисел:

В потоці генерації випадкової послідовності чисел генерується масив чисел в заданому діапазоні (від 6 до 6000).

Згенерований масив чисел записується у файл "generated.txt".

1. Сортування чисел:

У другому потоці згенерований масив чисел сортується за методом B.

Відсортований масив записується у файл "sorted.txt".

1. Відображення результатів:

Після завершення генерації та сортування, виводяться повідомлення про завершення відповідних процесів.

Автоматично відкривається провідник з відкриттям файлу "generated.txt" або текстовий редактор для перегляду файлу "sorted.txt".

1. Скасування операцій:

Користувач може скасувати обчислення, натиснувши кнопку "Скасувати". Це приведе до завершення всіх потоків та виведення відповідного повідомлення.

1. Закриття програми:

При закритті вікна програми викликається подія FormClosing, яка може скасувати обчислення та виходити з програми.

**Код програми**

using System;

using System.ComponentModel;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Threading;

using System.Windows.Forms;

namespace zakharchenko\_kn\_3\_2\_lab6

{

public partial class Form1 : Form

{

private const int MinArrayLength = 10000;

private const int DefaultArrayLength = 1000000;

private int[] generatedArray;

private readonly object lockObject = new object();

private bool isCancelled = false;

// Додаємо елементи форми

private Button btnStart;

private Button btnCancel;

private TextBox txtComments;

private TextBox txtArrayLength;

private Label lblArrayLength;

public Form1()

{

InitializeComponent();

// Ініціалізація елементів форми

InitializeComponents();

Text = "ЛР № 6, автор: Захарченко кн-3-2";

}

private void InitializeComponents()

{

btnStart = new Button();

btnStart.Text = "Запуск";

btnStart.Click += btnStart\_Click;

btnCancel = new Button();

btnCancel.Text = "Скасувати";

btnCancel.Click += btnCancel\_Click;

txtComments = new TextBox();

txtComments.Multiline = true;

txtComments.ScrollBars = ScrollBars.Vertical;

txtComments.Dock = DockStyle.Fill;

lblArrayLength = new Label();

lblArrayLength.Text = "Довжина масиву M:";

lblArrayLength.AutoSize = true;

txtArrayLength = new TextBox();

txtArrayLength.Text = DefaultArrayLength.ToString();

Controls.Add(btnStart);

Controls.Add(btnCancel);

Controls.Add(lblArrayLength);

Controls.Add(txtArrayLength);

Controls.Add(txtComments);

// Розмістити елементи на формі

int margin = 10;

int buttonWidth = 80;

lblArrayLength.Location = new Point(margin, margin);

txtArrayLength.Location = new Point(lblArrayLength.Right + margin, margin);

lblArrayLength.Location = new Point(ClientSize.Width - 2 \* margin - lblArrayLength.Width - txtArrayLength.Width - buttonWidth, margin);

txtArrayLength.Location = new Point(lblArrayLength.Right + margin, margin);

btnStart.Location = new Point(txtArrayLength.Right + margin, margin);

btnCancel.Location = new Point(btnStart.Right + margin, margin);

txtComments.Location = new Point(margin, btnStart.Bottom + margin);

txtComments.Height = ClientSize.Height - txtComments.Top - margin;

txtComments.Width = ClientSize.Width - 2 \* margin - SystemInformation.VerticalScrollBarWidth;

}

private void GenerateNumbers(int length)

{

lock (lockObject)

{

// Генерація чисел

Random random = new Random();

generatedArray = new int[length];

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

{

generatedArray[i] = random.Next(6, 6001);

}

// Запис у файл

string filePath = Path.Combine(Path.GetTempPath(), "generated.txt");

File.WriteAllText(filePath, string.Join(",", generatedArray.Select(num => num.ToString())));

UpdateComments("Генерацію завершено");

// Автоматично відкрити папку з файлом і самий файл для перегляду

System.Diagnostics.Process.Start("explorer.exe", "/select," + filePath);

}

}

private void SortNumbers()

{

// Сортування чисел за методом B

Array.Sort(generatedArray);

// Запис у файл

string filePath = Path.Combine(Path.GetTempPath(), "sorted.txt");

File.WriteAllLines(filePath, generatedArray.Select(num => num.ToString()));

UpdateComments("Сортування завершено");

System.Diagnostics.Process.Start("notepad.exe", filePath);

}

private void CancelProcessing()

{

MessageBox.Show("Обчислення скасовано", "Скасування", MessageBoxButtons.OK, MessageBoxIcon.Information);

UpdateComments("Обчислення скасовано.");

}

private void EnableButtons(bool enable)

{

btnStart.Enabled = enable;

btnCancel.Enabled = !enable;

}

private void UpdateComments(string comment)

{

if (txtComments.InvokeRequired)

{

txtComments.BeginInvoke(new Action(() => UpdateComments(comment)));

}

else

{

txtComments.AppendText(comment + Environment.NewLine);

}

}

private void btnStart\_Click(object sender, EventArgs e)

{

isCancelled = false;

txtComments.Text = "";

if (!int.TryParse(txtArrayLength.Text, out int arrayLength) || arrayLength < MinArrayLength)

{

MessageBox.Show("Некоректна довжина масиву. Мінімум 10 тис.");

return;

}

EnableButtons(false);

Thread generateThread = new Thread(() =>

{

UpdateComments($"Початок генерації. Довжина масиву: {arrayLength}");

GenerateNumbers(arrayLength);

lock (lockObject)

{

Monitor.Pulse(lockObject);

}

});

generateThread.Start();

Thread sortThread = new Thread(() =>

{

lock (lockObject)

{

Monitor.Wait(lockObject);

}

if (!isCancelled)

{

SortNumbers();

}

});

sortThread.Start();

}

private void btnCancel\_Click(object sender, EventArgs e)

{

isCancelled = true;

CancelProcessing();

EnableButtons(true);

}

private void txtArrayLength\_Validating(object sender, CancelEventArgs e)

{

if (!int.TryParse(txtArrayLength.Text, out int arrayLength) || arrayLength < MinArrayLength)

{

MessageBox.Show("Некоректна довжина масиву. Мінімум 10 тис.");

e.Cancel = true;

}

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

isCancelled = true;

}

}

}

**Скріншоти роботи програми**





















