**ЛАБОРАТОРНА РОБОТА #4. АСИНХРОННА МОДЕЛЬ ПРОГРАМУВАННЯ**

Іваницька Анна, МІТ 41

**Завдання для виконання:**

1. Розробити програму за шаблоном Windows Form Application, у якій запускається на виконання асинхронна задача (кнопка Start) виконання довготривалої операції. Виконання асинхронної задачі організувати таким чином, щоб вона виконувалася в окремому потоці й елементи керування форми реагували в цей час на дії користувача.
2. Удосконалити розроблену програму таким чином, щоб асинхронна операція повертала результат обчислень та відображала його на елементі керування форми.
3. Доповнити функціонал розробленої програми можливістю відображати хід виконання операції (відсоток у текстовій та графічній формах) за допомогою відповідних елементів керування форми.
4. Розширити функціонал розробленої програми можливістю дочасно перервати виконання довготривалої операції (кнопка Cancel)

**Хід роботи**

**ВИКОНАНЕ ЗАВДАННЯ**

Код:

using System;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab4TaskAsyncApp

{

public partial class Form1 : Form

{

private CancellationTokenSource \_cancellationTokenSource; // field to control the cancellation

public Form1() // initializes form's components

{

InitializeComponent();

}

private async void btnStart\_Click(object sender, EventArgs e) // asynchronous event handler

{

\_cancellationTokenSource = new CancellationTokenSource();

CancellationToken token = \_cancellationTokenSource.Token;

btnStart.Enabled = false;

btnCancel.Enabled = true;

lblProgress.Text = "0%";

progressBar.Value = 0;

var progress = new Progress<int>(value => // reports progress updates

{

lblProgress.Text = $"{value}%";

progressBar.Value = value;

});

try

{

int result = await Process(100, progress, token); // asynchronous task

if (result == 100)

{

lblProgress.Text = "Task is finished.";

}

}

finally

{

btnStart.Enabled = true;

btnCancel.Enabled = false;

}

}

Task<int> Process(int count, IProgress<int> ChangeProgressBar, CancellationToken cancellTocken)

{

return Task.Run(() =>

{

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

{

if (cancellTocken.IsCancellationRequested) // if cancel was called

{

lblProgress.Text = "Task was canceled.";

progressBar.Value = 0;

return i;

}

ChangeProgressBar.Report(i);

Thread.Sleep(100);

}

return count;

});

}

private void btnCancel\_Click(object sender, EventArgs e) // triggers cancel when Cancel button is clicked

{

if (\_cancellationTokenSource != null)

{

\_cancellationTokenSource.Cancel();

}

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

}

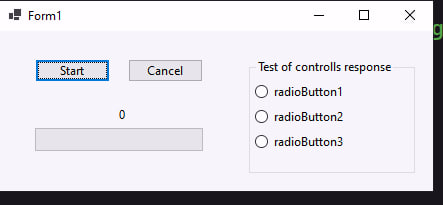
private void groupBox1\_Enter(object sender, EventArgs e)

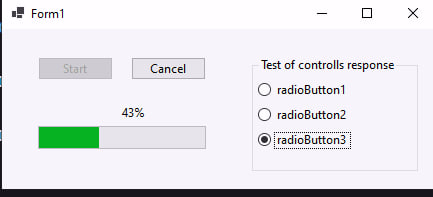
{

}

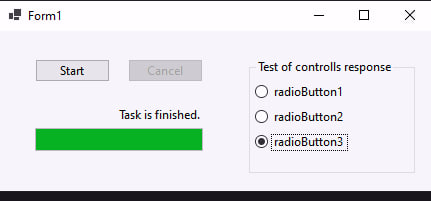
}

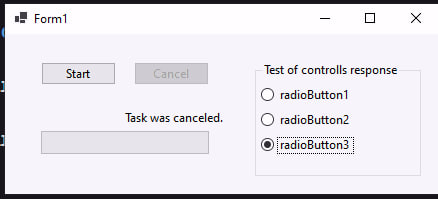
}

Виконання:

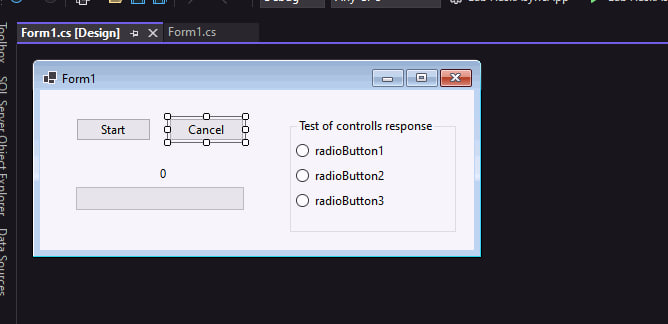
старт програми, з’являється вікно

При натисканні кнопки старт – відображається відсоток виконання у цифровому вигляді та у графічному

При завершенні виконання завдання – виводиться відповідне повідомлення.



Під час виконання завдання можна його перервати, натиснувши кнопку Cancel. Після цього виводиться відповідне повідомлення.



Ось так виглядає редактор самої форми

**Висновок:** у ході виконання цієї лабораторної роботи я зробила Windows Form App за шаблоном і реалізувала функціонал для кнопок. А саме, запуск асинхронного процесу на виконання при натисканні кнопки Start та відображення ступені його виконання. Також, була додана можливість перервати виконання процесу, натиснувши кнопку Cancel. Поки потік виконується, є можливість взаємодіяти з іншими елементами форми.