

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра автоматизації та інформаційних систем

Навчальна дисципліна
«ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ»

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ № 6

Виконав
студент групи КН-23-1
Іщенко.Є.В
Перевірила
доцент кафедри АІС
Істоміна Н. М.

Кременчук 2025

Лабораторна робота № 6

Тема: Використання класів та параметричних потоків

Мета: набути навичок розпаралелювання програмного коду на основі використання потоків і класів мовою C#.

Хід роботи:

Під час лабораторної роботи необхідно виконати такі дії:

1. Створіть консольний застосунок з кодом згідно з прикладом. У звіті наведіть власний код та «прінтскрін» роботи програми.

2. Змініть код так: головний потік очікує натиснення клавіші для завершення; усі потоки працюють із затримкою – 400 мс; ім'я потоку стало ST; ім'я делегата – st; слово Second писалось у такому самому рядку, що й слово Primary; на кожні 2 Primary приходилося одне Second. У звіті наведіть власний код і «прінтскрін» роботи програми з внесеними змінами.

3. Визначте рядки, які містять: створення нового класу; визначення змінних нового класу; створення вторинного потоку; створення методу вторинного потоку.

Завдання 1:

Створимо консольний застосунок з кодом згідно з прикладом.

```

using System;
using System.Threading;

namespace LB6PROPolynkoKN23
{
    class Program
    {
        static void Sec()
        {
            while (true)
            {
                Console.WriteLine(new string(' ', 10) + "Second");
                Thread.Sleep(200);
            }
        }

        static void Main()
        {
            ThreadStart sec = new ThreadStart(Sec);
            Thread thread = new Thread(sec);
            thread.Start();

            while (true)
            {
                Console.WriteLine("Primary");
                Thread.Sleep(200);
            }
        }
    }
}

```

Рисунок 1.1 – Код

На рисунку 1.2 – наведена робота консольного застосунку.

```

Primary      Second
Primary      Second
Primary      Second
Primary      Second
Primary      Second
Primary      Second
Primary      Second
Primary

```

Рисунок 1.2 – Робота потоків

Завдання 2:

Створимо програмний застосунок до завдання, що проводить перемноження двох довільних матриць (із глибиною та шириною не менше, ніж 250 елементів) згідно з варіантом 2 перебору циклу.

```

using System;
using System.Threading;

namespace LB6PROPolynkoKN23
{
    class Program
    {
        static bool isRunning = true;

        static void Sec()
        {
            while (isRunning)
            {
                Thread.Sleep(800);
                Console.Write(" Second");
            }
        }

        static void Main()
        {
            ThreadStart st = new ThreadStart(Sec);
            Thread ST = new Thread(st);
            ST.Start();

            int primaryCount = 0;

            while (isRunning)
            {
                Console.Write("Primary");
                primaryCount++;

                if (primaryCount % 1 == 0)
                {
                    Console.WriteLine();
                }

                Thread.Sleep(400);

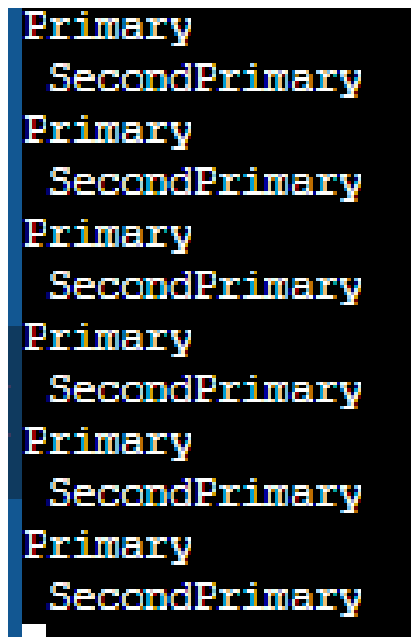
                if (Console.KeyAvailable)
                {
                    Console.ReadKey();
                    isRunning = false;
                }

                ST.Join();
                Console.WriteLine("\nЗавершення програми...");
            }
        }
    }
}

```

Рисунок 1.3 – Код 2

На рисунку 1.4 – наведена робота зміненого коду.



```

Primary
SecondPrimary
Primary
SecondPrimary
Primary
SecondPrimary
Primary
SecondPrimary
Primary
SecondPrimary
Primary
SecondPrimary

```

Рисунок 1.4 – Робота потоків у розширеному коді

Завдання 3:

1. Створення нового класу:

```
class Program
```

2. Визначення змінних нового класу:

```
static bool isRunning = true; // Прапорець для завершення потоку
```

3. Створення вторинного потоку:

```
ThreadStart st = new ThreadStart(Sec); // Ім'я делегата - st
```

```
Thread ST = new Thread(st); // Ім'я потоку - ST
```

```
ST.Start();
```

4. Створення методу вторинного потоку:

```
static void Sec()
```

Висновки:

На цій лабораторній роботі ми використовували класи та параметричні потоки, набули навичок розпаралелювання програмного коду на основні використання потоків і класів мовою C#. Створили два консольних застосунки та провели перевірку роботи потоків. Зазвичай, основний потік працює по чергово з вторим потоком, але інколи основний потік спрацьовує частіше.

