

## Лабораторна робота № 3

### Багатопотоковість у C#.

**Мета:** навчитися працювати з потоками та процесами у мові C#.

#### Хід роботи:

#### Завдання 1. Вимоги до програми шифрування файлів

1. Графічний інтерфейс користувача з можливістю вводу ключа для шифрування та діалогу вибору довільного файлу, який буде шифруватись або розшифровуватись.
2. Здатність працювати з файлами довільного розміру та формату.
3. В процесі шифрування, повинен відображатись індикатор прогресу (0-100%) та час, який пройшов від запуску шифрування.
4. По завершенні шифрування повинне бути відображене вікно з інформацією про розмір зашифрованого файлу, його назву та час, затрачений на шифрування.
5. Необхідно запобігти «підвисанню» вікна при здійсненні операції шифрування (шифрування здійснюватись в окремому обчислювальному потоці за допомогою класу BackgroundWorker).
6. Коректна обробка виключень, що можуть виникати під час роботи.
7. Для успішного виконання роботи необхідно ознайомитися з наступними темами: асинхронні операції (клас BackgroundWorker), класи File, FileStream, Timer, DateTime, OpenFileDialog, SaveFileDialog, обробка виключень.

#### Лістинг програми:

##### CripterBib:

```
using System.Security.Cryptography;

namespace CripterBib
{
    public class Crypter
    {
        Aes aes;
        ICryptoTransform cryptoTransform;

        public static long n;
        public static long i;

        public static bool working = false;

        int block_size;

        private Thread t;
        private string res_file_name;
        public Crypter(int block_size)
        {
            aes = Aes.Create();
            aes.KeySize = 128;
            aes.BlockSize = 128;
            aes.Key = System.Text.Encoding.ASCII.GetBytes("1234567890123456");
            aes.IV = System.Text.Encoding.ASCII.GetBytes("1234567890123456");
            aes.Mode = CipherMode.CBC;
            aes.Padding = PaddingMode.PKCS7;
            this.block_size = block_size;
        }
    }
}
```

					ДУ «Житомирська політехніка».22.121.04.000 – Лр3			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи			
Розроб.		Ігліньський І.Ю.						
Перевір.		Чижмотря О.В.						
Керівник								
Н. контр.								
Зав. каф.					ФІКТ Гр. ІПЗ-21-4[2]			
					Лім.	Арк.	Аркушів	
						1	15	

```

    }

    public void EncryptFile(string file_name, string key)
    {
        if (working)
            return;

        aes.Key = System.Text.Encoding.ASCII.GetBytes(key);
        cryptoTransform = aes.CreateEncryptor();

        res_file_name = "Encrypted\\Encrypted_" + file_name;

        for (int i = file_name.Length - 1; i >= 0; i--)
            if (file_name[i] == '\\')
            {
                res_file_name = "Encrypted\\Encrypted_" + file_name.Substring(i + 1,
file_name.Length - i - 1);
                break;
            }

        t = new Thread(() =>
        {
            Crypter.working = true;
            TransformFile(file_name, res_file_name, block_size);
            Crypter.working = false;
        });

        t.Start();
    }

    public void DecryptFile(string file_name, string key)
    {
        if (working)
            return;

        aes.Key = System.Text.Encoding.ASCII.GetBytes(key);
        cryptoTransform = aes.CreateDecryptor();

        res_file_name = "Decrypted\\Decrypted_" + file_name;

        for (int i = file_name.Length - 1; i >= 0; i--)
            if (file_name[i] == '\\')
            {
                res_file_name = "Decrypted\\Decrypted_" + file_name.Substring(i + 1,
file_name.Length - i - 1);
                break;
            }

        t = new Thread(() =>
        {
            Crypter.working = true;
            TransformFile(file_name, res_file_name, block_size + 16);
            Crypter.working = false;
        });

        t.Start();
    }

    public void TransformFile(string file_name, string res_file_name, int block_size)
    {
        using (FileStream fs = new FileStream(file_name, FileMode.Open, FileAccess.Read))
        {
            using (FileStream out_fs = new FileStream(res_file_name, FileMode.Create,
FileAccess.Write))
            {
                Crypter.n = fs.Length / block_size;
                Crypter.i = 0;
                while (true)

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            if (!Crypter.working)
            {
                out_fs.Dispose();
                File.Delete(res_file_name);
                break;
            }

            Byte[] buffer = new Byte[block_size];

            int n = fs.Read(buffer, 0, buffer.Length);

            if (n == 0)
                break;
            byte[] output = new byte[buffer.Length];
            try
            {
                output = cryptoTransform.TransformFinalBlock(buffer, 0,
buffer.Length);
            }
            catch
            {
            }

            out_fs.Write(output, 0, output.Length);
            Crypter.i++;
        }
    }

    public void Stop()
    {
        Crypter.working = false;
    }
}

```

### Coder:

```

using CrypterBib;

namespace CoderForm
{
    public partial class Form1 : Form
    {
        int pointCount = 0;
        string filePath = "";
        string fileName = "";
        string labelText = "";
        Crypter crypter = new Crypter(1024);
        int time = 0;
        bool openFile = false;
        bool enc = true;
        public Form1()
        {
            InitializeComponent();
            OFD.Filter = "All (*.*)|*.*";
            NewFile("", EventArgs.Empty);
        }

        private void SetFile(string s)
        {
            openFile = true;
            filePath = s;
        }
    }
}

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – Лр3	Арк.
		Чижмоторя О.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        for (int i = s.Length - 1; i >= 0; i--)
            if (s[i] == '\\')
            {
                FileLabel.Text = s.Substring(i + 1, s.Length - i - 1);
                break;
            }
        if (enc)
            fileName = "Encrypted_" + FileLabel.Text;
        else
            fileName = "Decrypted_" + FileLabel.Text;
        FileButton.Text = "Змінити";
        ShifrButton.Enabled = true;
        RozshifrButton.Enabled = true;
    }
    private void file_but_Click(object sender, EventArgs e)
    {
        if (OFD.ShowDialog() == DialogResult.OK)
        {
            SetFile(OFD.FileName);
        }
    }

    private void Form1DragEnter(object sender, DragEventArgs e)
    {
        e.Effect = DragDropEffects.Copy;
    }

    private void Form1DragDrop(object sender, DragEventArgs e)
    {
        string[] s = (string[])e.Data.GetData(DataFormats.FileDrop);
        if (s.Length > 1)
        {
            MessageBox.Show("Ви вибрали більше чим один файл!", "Error4ik",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
        SetFile(s[0]);
    }

    private void CodeButtonClick(object sender, EventArgs e)
    {
        enc = true;

        labelText = "Відбувається шифрування файлу";

        crypter.EncryptFile(filePath, GetKey());

        showProgress();
    }

    private string GetKey()
    {
        string key = TextBoxKey.Text;
        while (key.Length != 16)
            if (key.Length > 16)
                key = key.Substring(0, 16);
            else
                key += '\\0';
        return key;
    }

    private void showProgress()
    {
        ShifrButton.Visible = false;
        RozshifrButton.Visible = false;
    }

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – ЛрЗ	Арк.
		Чижомотря О.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

FileButton.Visible = false;
TextBoxKey.ReadOnly = true;
progressBar.Visible = true;
CancelButton.Visible = true;
FileLabel.Font = new Font("Arial", 19);
FileLabel.Top = 175;
point.Start();
pointTimerTick("", EventArgs.Empty);
progress.Start();
TimeLabel.Visible = true;
CancelButton.Enabled = true;
}

private void decodeButtonClick(object sender, EventArgs e)
{
    enc = false;

    labelText = "Відбувається розшифрування файлу";

    crypter.DecryptFile(filePath, GetKey());

    showProgress();
}

private void pointTimerTick(object sender, EventArgs e)
{
    FileLabel.Text = labelText;
    for (int i = 0; i < pointCount; i++)
        FileLabel.Text += ".";
    pointCount++;
    if (pointCount == 4)
        pointCount = 0;

    int mins = time / 120;
    int secs = (time % 120) / 2;

    TimeLabel.Text = $"Час від старту: {mins} хв., {secs} сек.";

    time++;
}

private void ProgressTimerTick(object sender, EventArgs e)
{
    int percentage = (int)((float)Crypter.i / Crypter.n * 1000);

    progressBar.Value = Math.Max(0, Math.Min(percentage, 1000));

    if (!Crypter.working)
    {
        progressBar.Value = 1000;
        progress.Stop();
        point.Stop();
        AfterCrypto();

        string text = "Шифрування";
        string dir = "Encrypted";
        if (!enc)
        {
            text = "Розшифрування";
            dir = "Decrypted";
        }
        MessageBox.Show(
            text + " файлу відбулось успішно!\n" +
            "Файл збережено в папку " + dir,

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "Успіх!",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information
    );
}

private void AfterCrypto()
{
    string text = "Шифрування";
    if (!enc)
        text = "Розшифрування";
    FileLabel.Text = text + " відбулось успішно!";
    FileLabel.Top = 120;
    FileButton.Top = 275;
    SaveFile.Visible = true;
    FileButton.Visible = true;
    FileButton.Text = "Новий файл";
    FileButton.Click -= file_but_Click;
    FileButton.Click += NewFile;
    CancelButton.Enabled = false;
}

private void NewFile(object sender, EventArgs e)
{
    FileLabel.Top = 141;
    FileLabel.Text = "Оберіть файл або перемістіть на форму";
    FileButton.Top = 211;
    FileButton.Text = "Обрати файл";
    FileButton.Click -= NewFile;
    FileButton.Click -= file_but_Click;
    FileButton.Click += file_but_Click;
    FileButton.Visible = true;
    TextBoxKey.Text = "";
    TextBoxKey.ReadOnly = false;
    progressBar.Visible = false;
    CancelButton.Visible = false;
    ShifrButton.Visible = true;
    ShifrButton.Enabled = false;
    RozshifrButton.Visible = true;
    RozshifrButton.Enabled = false;
    SaveFile.Visible = false;
    TimeLabel.Visible = false;
    time = 0;
}

private void SaveFileButtonClick(object sender, EventArgs e)
{
    SFD.FileName = fileName;
    if (SFD.ShowDialog() == DialogResult.OK)
    {
        if (File.Exists(SFD.FileName))
            File.Delete(SFD.FileName);
        if (enc)
            File.Copy("Encrypted\\" + fileName, SFD.FileName);
        else
            File.Copy("Decrypted\\" + fileName, SFD.FileName);
    }
}

private void CancelButtonClick(object sender, EventArgs e)
{
    string temp = "шифрування";

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – Лр3	Арк.
		Чижомотря О.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (!enc)
            temp = "розшифрування";
        var res = MessageBox.Show("Ви впевнені що хочете відмінити " + temp +
"\nПрогресс буде втрачено",
            "Попередження",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Question);
        if (res == DialogResult.Yes)
        {
            progress.Stop();
            point.Stop();
            crypter.Stop();
            NewFile("", EventArgs.Empty);
        }
    }

private void Form1FormClosing(object sender, FormClosingEventArgs e)
{
    if (Crypter.working)
    {
        string temp = "шифрування";
        if (!enc)
            temp = "розшифрування";
        var res = MessageBox.Show(
            "Ви впевнені що хочете закрити вікно?\nПрогресс " + temp + " буде
втрачено",
            "Попередження",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Question);
        if (res == DialogResult.Yes)
        {
            progress.Stop();
            point.Stop();
            crypter.Stop();

            NewFile("", EventArgs.Empty);
        }
        else
            e.Cancel = true;
    }
}
}

```

**Результат виведення:**

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		



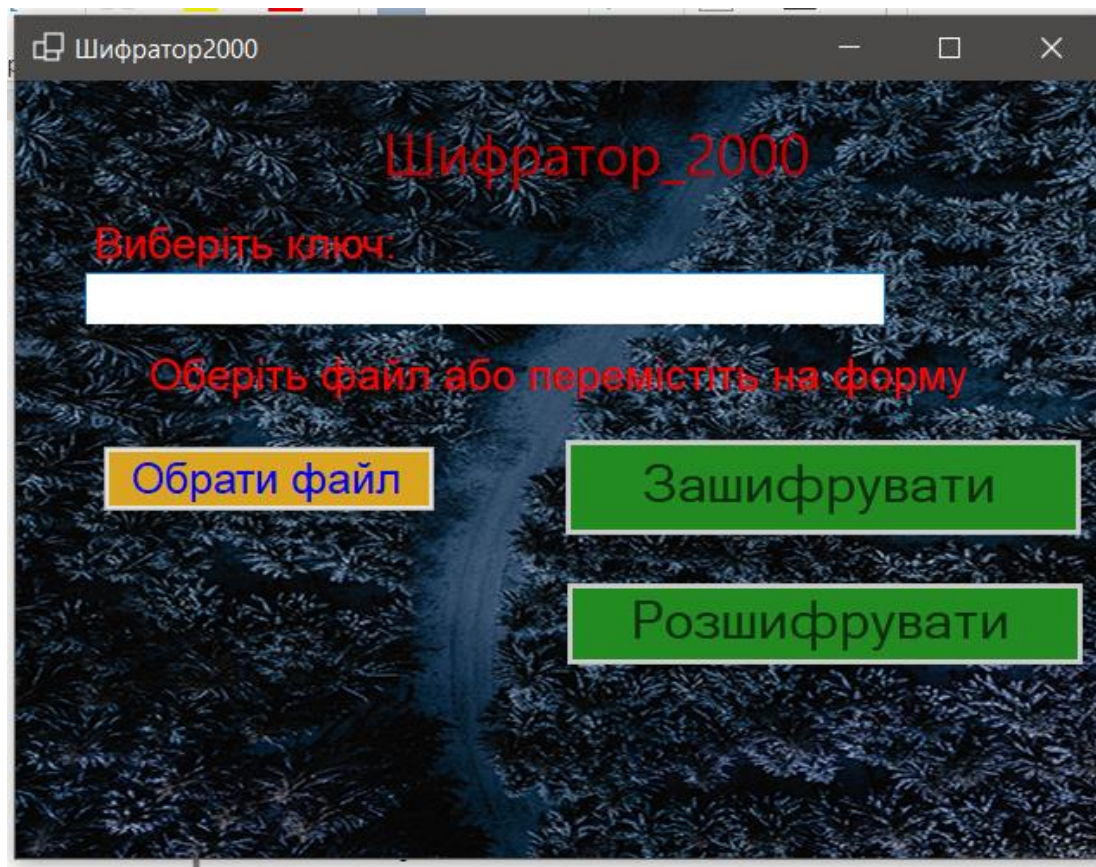


Рис. 1 – Меню старту

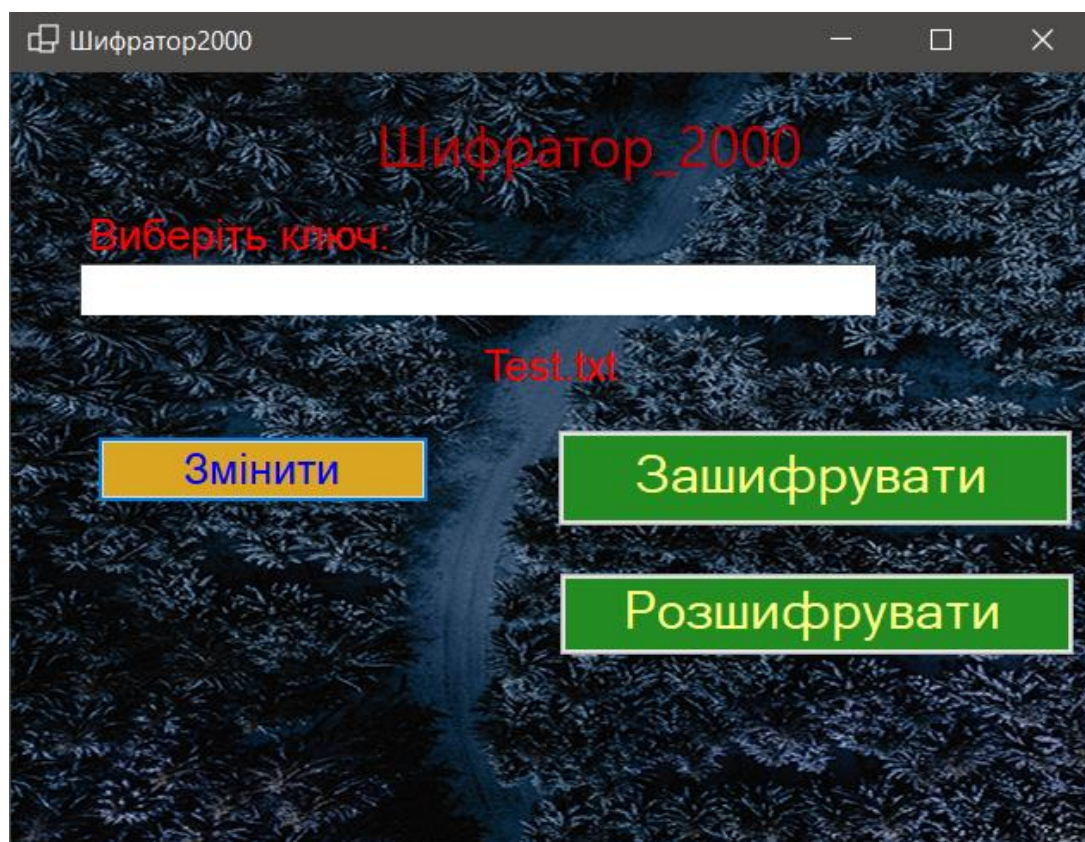


Рис. 2 – Вибраний файл

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



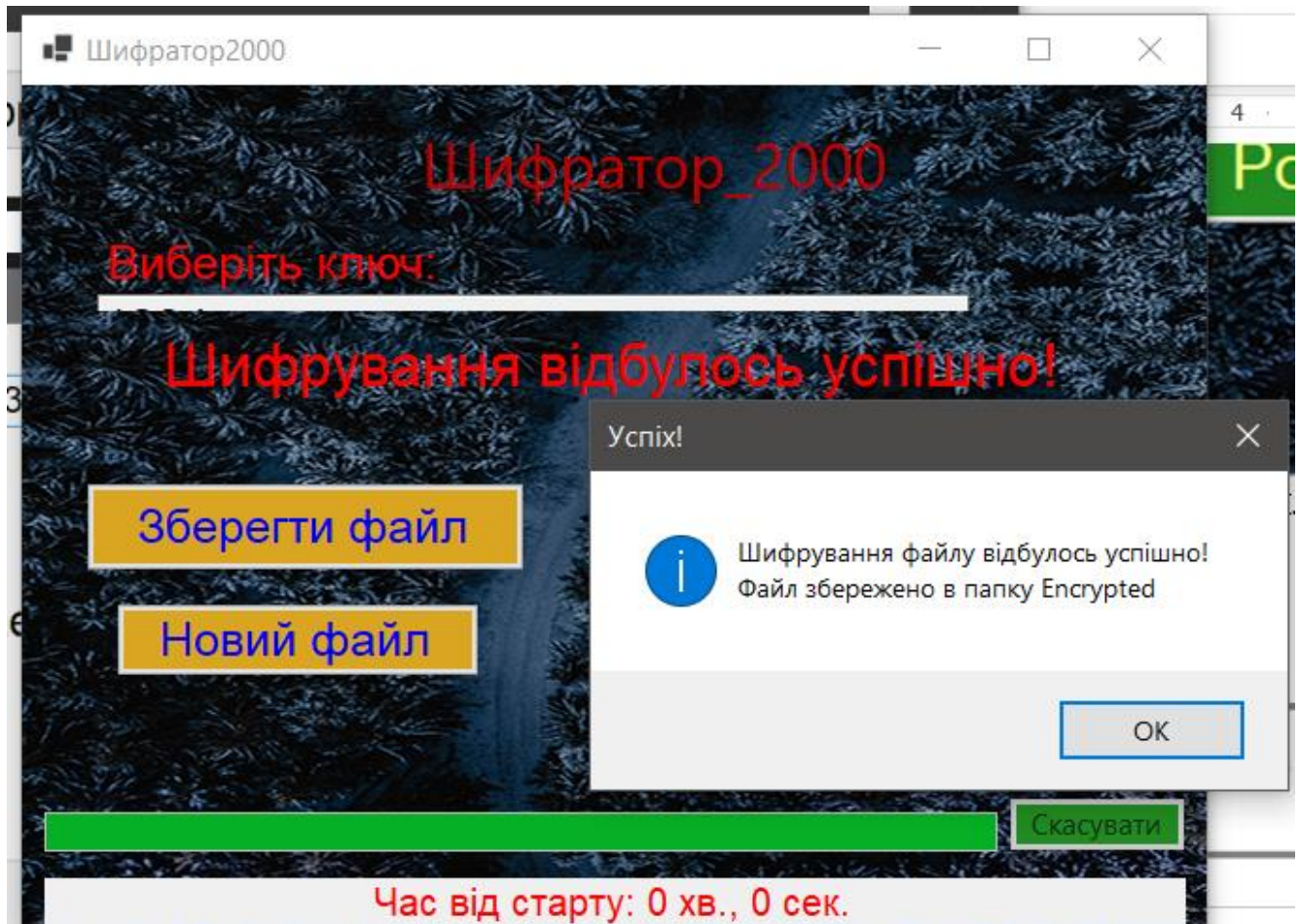


Рис. 3 – Шифрування відбулось

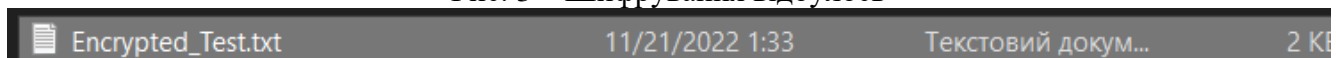


Рис. 4 – Зашифрований файл

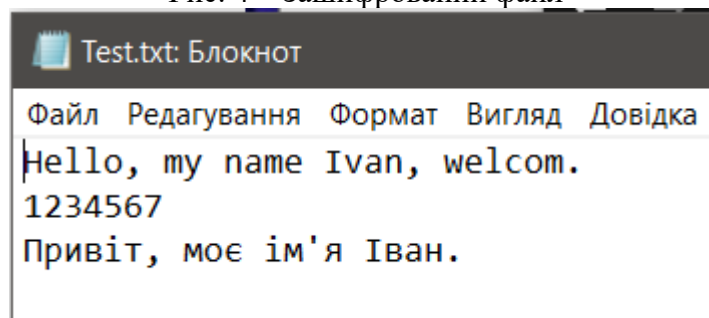


Рис. 5 – Вміст до шифрування

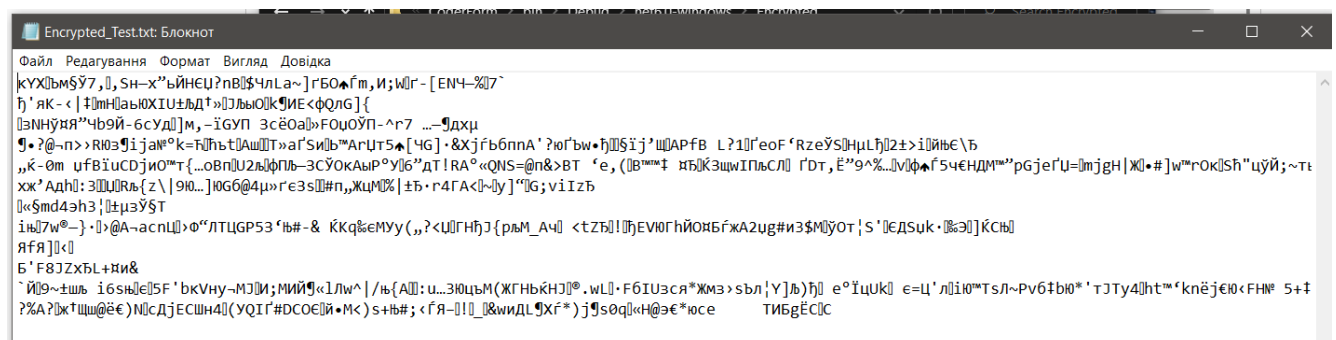


Рис. 6 – Вміст після шифрування

		Ігліський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – Лр3	Арк.
		Чижмоторя О.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

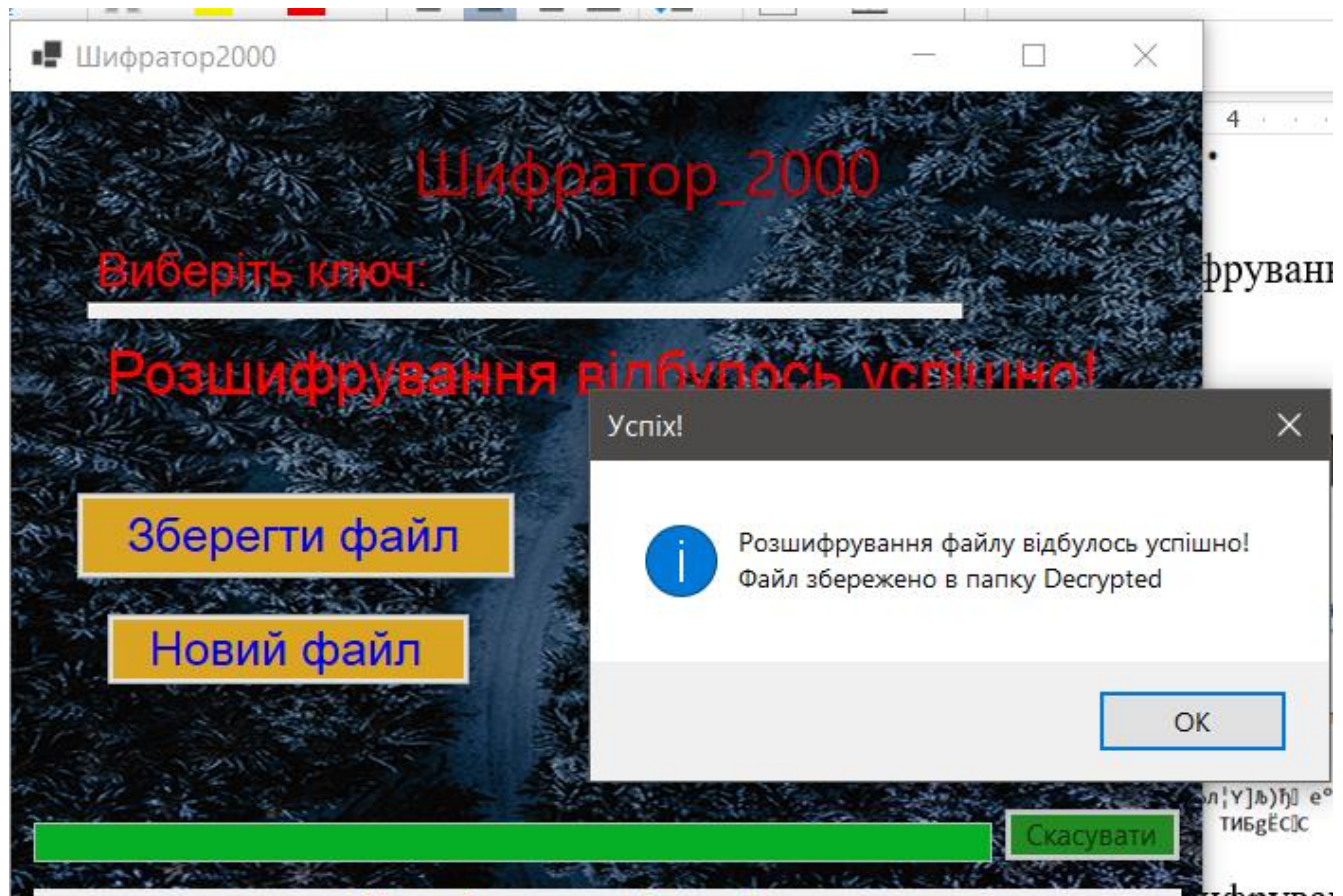


Рис. 7 – Розшифрування відбулось

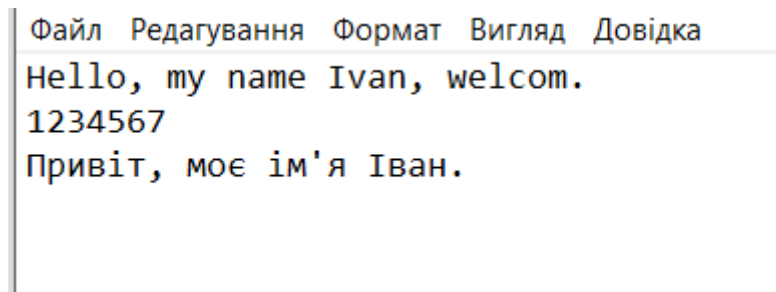


Рис. 8 – Вміст після розшифрування

## Завдання 2. Вимоги до менеджера процесів

1. Можливість запуску програм: Калькулятор, Microsoft Word та ще трьох на ваш вибір.
2. Перегляд список запущених процесів з інформацією про обсяг оперативної пам'яті, виділений під процес, часу, коли був запущений процес, пріоритет процесу, кількість потоків, які запущені процесом. Відобразити у вигляді таблиці.
3. Можливість зупинки вибраного процесу.
4. Можливість зміни пріоритету вибраного процесу

### Лістинг програми:

```
ProcessManager:
using System.Diagnostics;
using System.Diagnostics.PerformanceData;
```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – Лр3	Арк.
		Чижмоторя О.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

namespace ProcessManager
{
    public class Process
    {
        public string name;
        public float memory;
        public DateTime? start_time;
        public string priority;
        public int stream_count;
        private System.Diagnostics.Process proc;
        public Process(System.Diagnostics.Process proc)
        {
            this.proc = proc;
            this.name = proc.ProcessName;
            try
            {
                this.start_time = proc.StartTime;
            }
            catch
            {
                this.start_time = null;
            }

            PerformanceCounter performanceCounter = new PerformanceCounter();
            performanceCounter.CategoryName = "Process";
            performanceCounter.CounterName = "Working Set - Private";
            performanceCounter.InstanceName = proc.ProcessName;
            this.memory = Convert.ToInt32(performanceCounter.NextValue()) / (float)(1024
* 1024);

            performanceCounter.Close();
            performanceCounter.Dispose();

            try
            {
                this.priority = proc.PriorityClass.ToString();
            }
            catch
            {
                this.priority = "-";
            }
            this.stream_count = proc.Threads.Count;
        }
        public void Stop()
        {
            proc.Kill();
        }

        public void ChangePriority(string priority)
        {
            switch (priority)
            {
                case "RealTime":
                    this.proc.PriorityClass = ProcessPriorityClass.RealTime;
                    break;
                case "High":
                    this.proc.PriorityClass = ProcessPriorityClass.High;
                    break;
                case "AboveNormal":
                    this.proc.PriorityClass = ProcessPriorityClass.AboveNormal;
                    break;
                case "Normal":
                    this.proc.PriorityClass = ProcessPriorityClass.Normal;
                    break;
                case "BelowNormal":

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – Лр3	Арк.
		Чижмоторя О.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        this.proc.PriorityClass = ProcessPriorityClass.BelowNormal;
        break;
    case "Idle":
        this.proc.PriorityClass = ProcessPriorityClass.Idle;
        break;
    }

    this.priority = this.proc.PriorityClass.ToString();
}

public static int nameComp(Process mp1, Process mp2)
{
    return String.Compare(mp1.name, mp2.name,
StringComparison.OrdinalIgnoreCase);
}

}

public class ProcessManager
{
    Process[] processes;
    public int length;
    private int sort = 0;
    public ProcessManager()
    {
        refresh();
    }

    public void refresh()
    {
        System.Diagnostics.Process[] procs =
System.Diagnostics.Process.GetProcesses();
        processes = new Process[procs.Length];
        int i = 0;
        length = procs.Length;
        foreach (System.Diagnostics.Process proc in procs)
        {
            processes[i] = new Process(proc);
            i++;
        }
        Array.Sort(processes, Process.nameComp);
    }

    public Process this[int key]
    {
        get => processes[key];
    }
}
}

```

### Menager:

```

using System.Diagnostics;
using ProcessManager;

namespace Meneger
{
    public partial class TaskManeger : Form
    {
        ProcessManager.ProcessManager processManager = new
ProcessManager.ProcessManager();

        public TaskManeger()
        {
            InitializeComponent();
        }
    }
}

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        REFRESH();
    }

    void REFRESH()
    {
        processManager.refresh();
        ProcsList.Items.Clear();
        for (int i = 0; i < processManager.length; i++)
        {
            var row = new String[] { processManager[i].name,
                                    processManager[i].memory.ToString("F2"),
                                    processManager[i].priority,
                                    "-",
                                    processManager[i].stream_count.ToString()
                                };
            if (processManager[i].start_time != null)
                row[3] = processManager[i].start_time.ToString();
            var lvi = new ListViewItem(row);
            lvi.Tag = processManager[i];
            ProcsList.Items.Add(lvi);
        }
    }

    private void CalcButton_Click(object sender, EventArgs e)
    {
        System.Diagnostics.Process.Start("calc");
        REFRESH();
    }

    private void StopToolStripMenuItem_Click(object sender, EventArgs e)
    {
        var proc = (ProcessManager.Process)ProcsList.SelectedItems[0].Tag;
        proc.Stop();
        REFRESH();
    }

    public void ChangePriorityClick(object sender, EventArgs e)
    {
        var but = (ToolStripMenuItem) sender;
        var proc = (ProcessManager.Process)ProcsList.SelectedItems[0].Tag;
        proc.ChangePriority(but.Text);
        REFRESH();
    }

    private void WordButton_Click(object sender, EventArgs e)
    {
        System.Diagnostics.Process.Start(@"C:\Program Files\Microsoft
Office\root\Office16\WINWORD.EXE");
        REFRESH();
    }

    private void PhpStorm_Click(object sender, EventArgs e)
    {
        System.Diagnostics.Process.Start(@"C:\Program Files\JetBrains\PhpStorm
2022.2.1\bin\phpstorm64.exe");
        REFRESH();
    }

    private void WebStormButton_Click(object sender, EventArgs e)
    {
        System.Diagnostics.Process.Start(@"C:\Program Files\JetBrains\WebStorm
2021.3.2\bin\webstorm64.exe");
        REFRESH();
    }

    private void PyCharmButton_Click(object sender, EventArgs e)
    {

```

		Ігліньський І.Ю.			ДУ «Житомирська політехніка».22.121.04.000 – Лр3	Арк.
		Чижмоторя О.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		



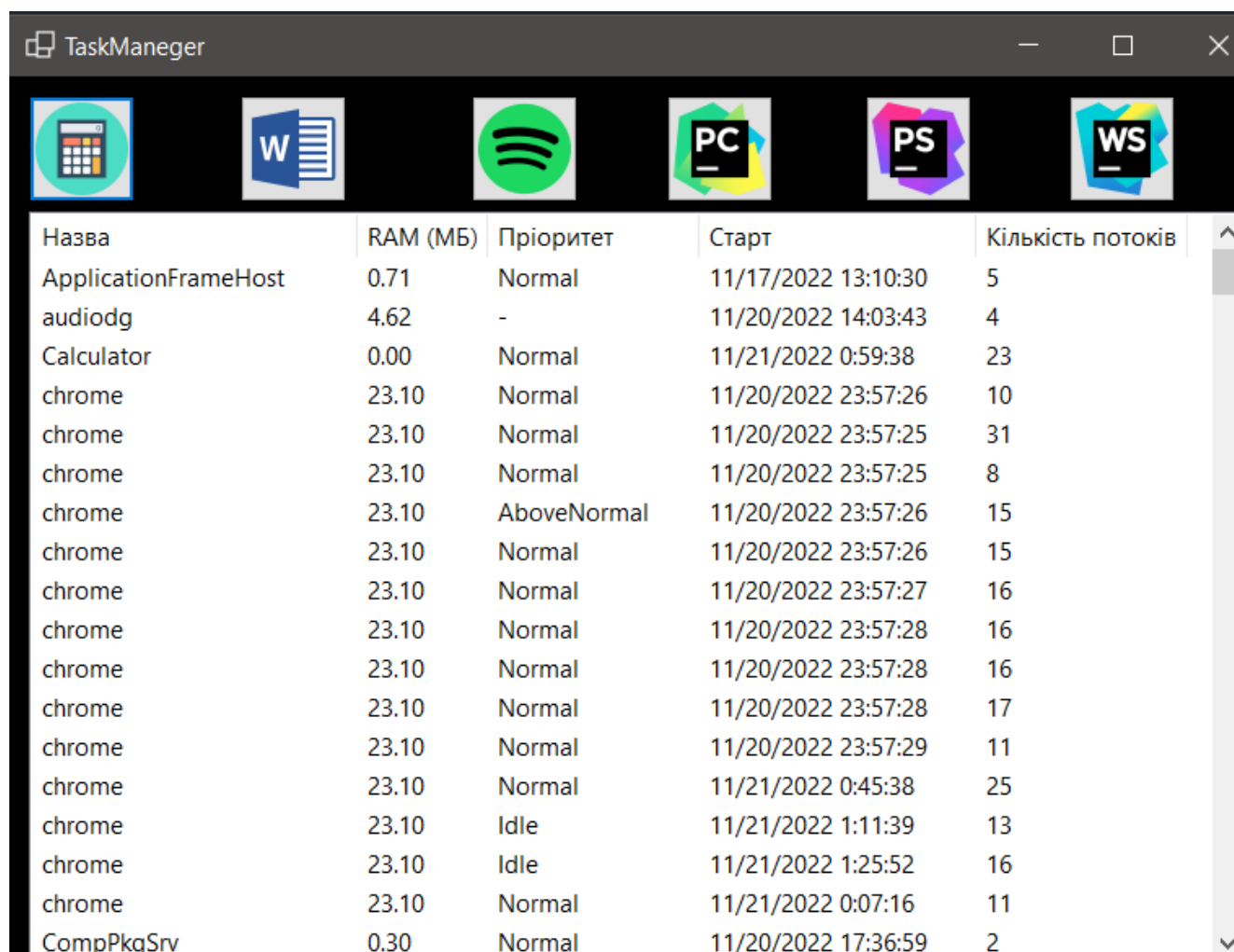
```

        System.Diagnostics.Process.Start(@"C:\Program Files\JetBrains\PyCharm
Community Edition 2022.1.4\bin\pycharm64.exe");
        REFRESH();
    }

    private void SpotifyButton_Click(object sender, EventArgs e)
    {
        System.Diagnostics.Process.Start(@"C:\Users\ivanm\AppData\Roaming\Spotify\Spotify.exe");
        REFRESH();
    }
}

```

### Результат виведення:



Назва	RAM (МБ)	Пріоритет	Старт	Кількість потоків
ApplicationFrameHost	0.71	Normal	11/17/2022 13:10:30	5
audiodg	4.62	-	11/20/2022 14:03:43	4
Calculator	0.00	Normal	11/21/2022 0:59:38	23
chrome	23.10	Normal	11/20/2022 23:57:26	10
chrome	23.10	Normal	11/20/2022 23:57:25	31
chrome	23.10	Normal	11/20/2022 23:57:25	8
chrome	23.10	AboveNormal	11/20/2022 23:57:26	15
chrome	23.10	Normal	11/20/2022 23:57:26	15
chrome	23.10	Normal	11/20/2022 23:57:27	16
chrome	23.10	Normal	11/20/2022 23:57:28	16
chrome	23.10	Normal	11/20/2022 23:57:28	16
chrome	23.10	Normal	11/20/2022 23:57:28	17
chrome	23.10	Normal	11/20/2022 23:57:29	11
chrome	23.10	Normal	11/21/2022 0:45:38	25
chrome	23.10	Idle	11/21/2022 1:11:39	13
chrome	23.10	Idle	11/21/2022 1:25:52	16
chrome	23.10	Normal	11/21/2022 0:07:16	11
CompPkgSrv	0.30	Normal	11/20/2022 17:36:59	2

Рис. 8 – Стартове меню

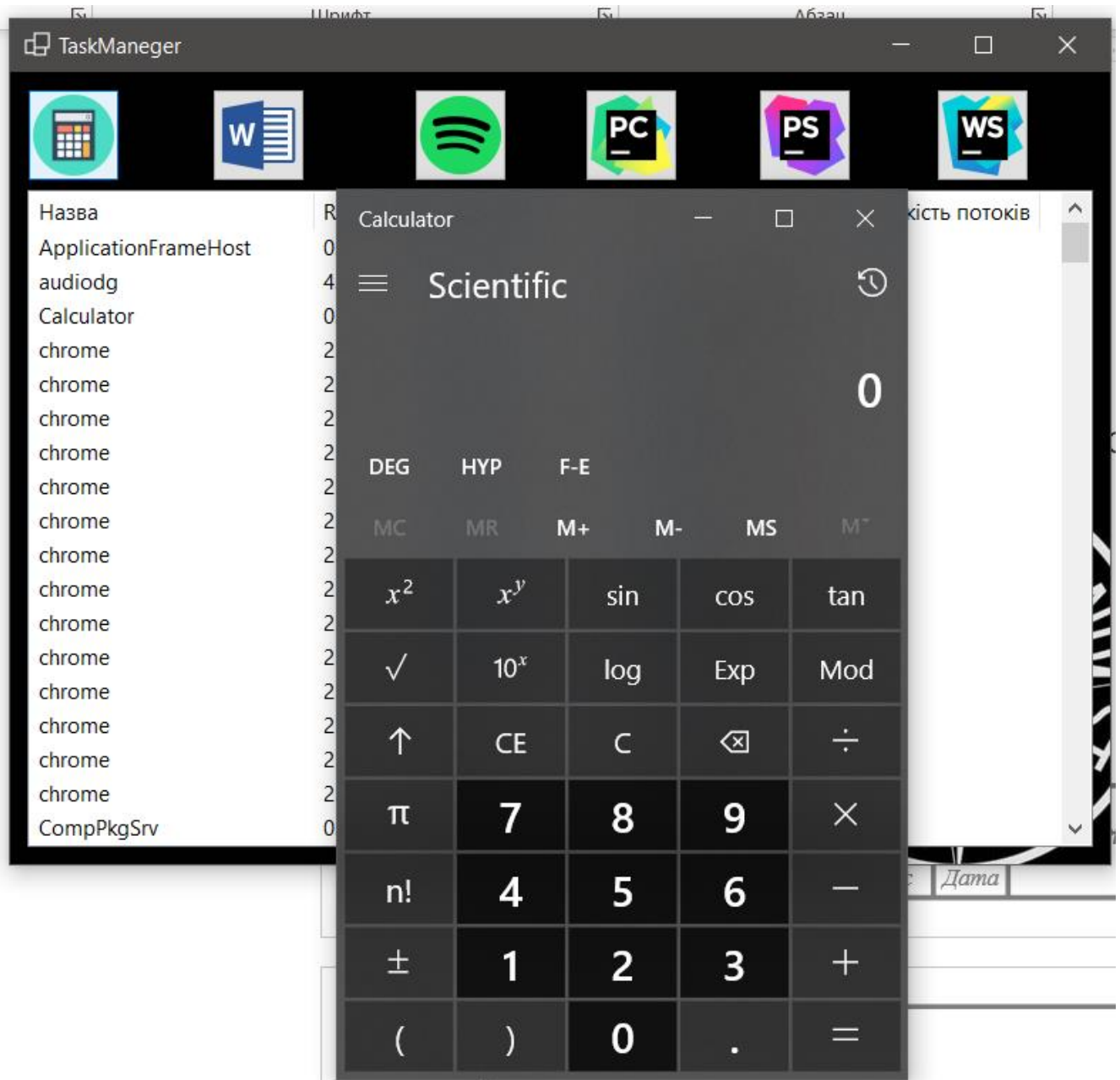


Рис. 9 – Відкриття калькулятора

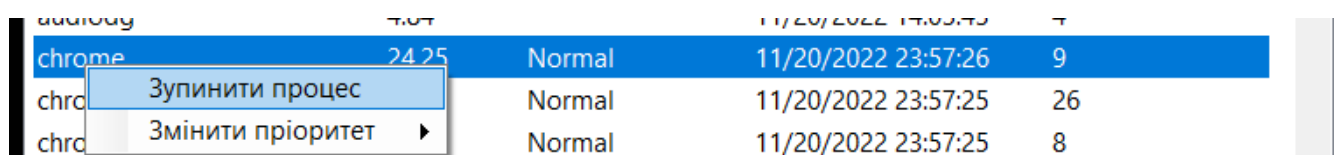


Рис. 10 – меню процесів

**Висновок:** в ході виконання лабораторної роботи було ознайомлено з потоками та процесами у мові C#.

**Посилання на Git-lab:** [https://gitlab.com/ivan\\_igllinskyi/dotnetlab/-/tree/main/Lab3](https://gitlab.com/ivan_igllinskyi/dotnetlab/-/tree/main/Lab3)