

Міністерство освіти і науки, молоді та спорту України  
Національний університет “Львівська політехніка”

Кафедра ЕОМ



## Звіт

з домашнє завдання №22.2 з дисципліни:

“Алгоритми та моделі обчислень”

Варіант: № 24.

Виконав:

ст. групи КІ-203

Ширий Богдан Ігорович

Перевірів:

ст. викладач кафедри ЕОМ

Козак Назар Богданович

Львів – 2023

## ЗАВДАННЯ:

### УМОВА:

C++/CLI дозволяє писати програми, які водночас можуть містити і звичайний некерований код (*англ. unmanaged code*) написаний на C++ для компіляції безпосередньо в машинний код, і некерований код (*англ. managed code*) написаний на C++ для .NET.

Виконати домашнє завдання №22.1 повторно за допомогою C++/CLI.

### ВИБІР ВАРІАНТУ:

Перша літера мого прізвища латинськими літерами це S.

### ВИКОНАННЯ:

Склав C++/CLI десктопну програму та її роботу зобразив на рисунках 1 та 2.

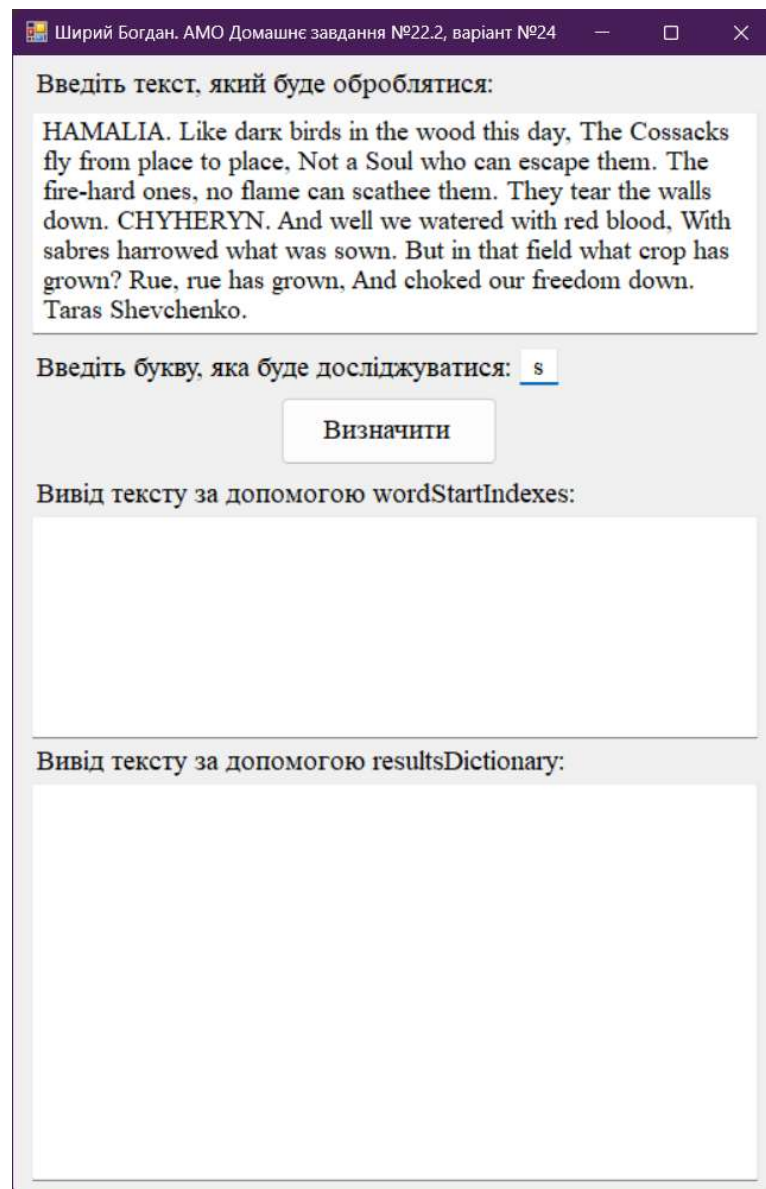


Рис. 1. Початкові дані програми.

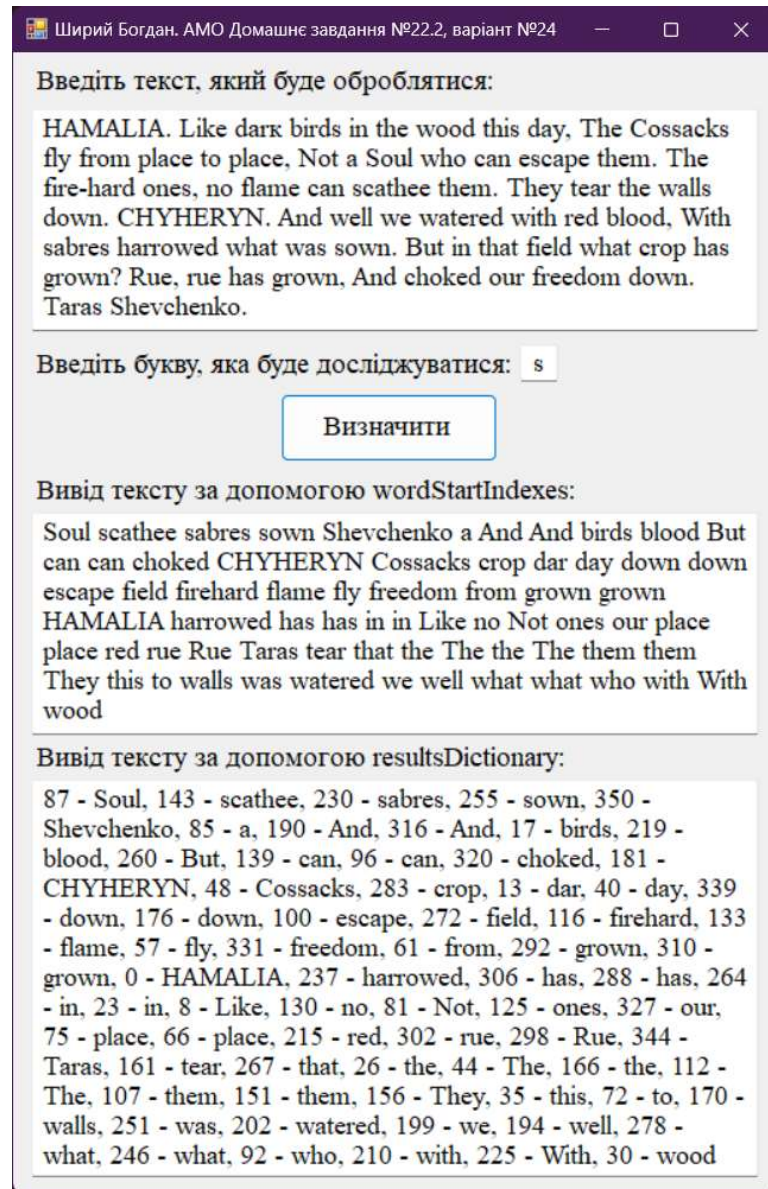


Рис. 2. Програма після натиску на кнопку.

Всю програму, що написав на C# у завданні №22.1, перевів у C++/CLI.

Відповідно, у лістингах навів код кожного файлу:

- ShyryiForm.cpp - лістинг 1,
- ShyryiForm.h - лістинг 2.

Лістинг 1. Код файлу MainForm.cpp.

```
#include "ShyryiForm.h"

using namespace System;
using namespace System::Windows::Forms;

[STAThreadAttribute]
int main() {
    Application::SetCompatibleTextRenderingDefault(false);
    Application::EnableVisualStyles();
    acmShyryiHW22_2::ShyryiForm form;
    Application::Run(% form);
}
```

## Лістинг 2. Код заголовкового файлу MainForm.h.

```
#pragma once
using namespace System;
using namespace System::Collections::Generic;
using namespace System::Text::RegularExpressions;

String^ GetWordFromIndex(String^ inputText, int startIndex)
{
    int endIndex = inputText->IndexOf(' ', startIndex);
    if (endIndex == -1)
        endIndex = inputText->Length;

    return inputText->Substring(startIndex, endIndex - startIndex);
}

ref class ComparisonHelper
{
private:
    String^ inputText;

public:
    ComparisonHelper(String^ text) : inputText(text) {}

    int Compare(int x, int y)
    {
        String^ wordX = GetWordFromIndex(inputText, x);
        String^ wordY = GetWordFromIndex(inputText, y);

        return String::Compare(wordX->ToLower(), wordY->ToLower());
    }
};

namespace acmShyryiHW22_2 {

    /// <summary>
    /// Summary for ShyryiForm
    /// </summary>
    public ref class ShyryiForm : public System::Windows::Forms::Form
    {
    public:
        ShyryiForm(void)
        {
            InitializeComponent();
            //
            //TODO: Add the constructor code here
            //
        }

    protected:
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        ~ShyryiForm()
        {
            if (components)
            {
                delete components;
            }
        }

    private: System::Windows::Forms::TextBox^ textInputTextBox;
    private: System::Windows::Forms::Label^ textInputLabel;
    private: System::Windows::Forms::Label^ letterInputLabel;
    private: System::Windows::Forms::TextBox^ letterInputTextBox;
    private: System::Windows::Forms::Button^ InvestigateButton;
    private: System::Windows::Forms::Label^ textOutputLabel;
    private: System::Windows::Forms::TextBox^ textOutputTextBox;
    private: System::Windows::Forms::Label^ dictionaryOutputLabel;
    private: System::Windows::Forms::TextBox^ dictionaryOutputTextBox;
    protected:

    protected:

    private:
        /// <summary>
        /// Required designer variable.
        /// </summary>
        System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        void InitializeComponent(void)
        {
            this->textInputTextBox = (gcnew System::Windows::Forms::TextBox());
            this->textInputLabel = (gcnew System::Windows::Forms::Label());
            this->letterInputLabel = (gcnew System::Windows::Forms::Label());
            this->letterInputTextBox = (gcnew System::Windows::Forms::TextBox());
            this->InvestigateButton = (gcnew System::Windows::Forms::Button());
            this->textOutputLabel = (gcnew System::Windows::Forms::Label());
            this->textOutputTextBox = (gcnew System::Windows::Forms::TextBox());
        }
    }
}
```

```

this->dictionaryOutputLabel = (gcnew System::Windows::Forms::Label());
this->dictionaryOutputTextBox = (gcnew System::Windows::Forms::TextBox());
this->SuspendLayout();
//
// textInputTextBox
//
this->textInputTextBox->Location = System::Drawing::Point(12, 38);
this->textInputTextBox->Multiline = true;
this->textInputTextBox->Name = L"textInputTextBox";
this->textInputTextBox->Size = System::Drawing::Size(490, 150);
this->textInputTextBox->TabIndex = 0;
//
// textInputLabel
//
this->textInputLabel->AutoSize = true;
this->textInputLabel->Location = System::Drawing::Point(12, 9);
this->textInputLabel->Name = L"textInputLabel";
this->textInputLabel->Size = System::Drawing::Size(382, 26);
this->textInputLabel->TabIndex = 1;
this->textInputLabel->Text = L"Введіть текст, який буде оброблятися.";
//
// letterInputLabel
//
this->letterInputLabel->AutoSize = true;
this->letterInputLabel->Location = System::Drawing::Point(12, 200);
this->letterInputLabel->Name = L"letterInputLabel";
this->letterInputLabel->Size = System::Drawing::Size(402, 26);
this->letterInputLabel->TabIndex = 2;
this->letterInputLabel->Text = L"Введіть букву, яка буде досліджуватися.";
//
// letterInputTextBox
//
this->letterInputTextBox->Location = System::Drawing::Point(341, 197);
this->letterInputTextBox->Multiline = true;
this->letterInputTextBox->Name = L"letterInputTextBox";
this->letterInputTextBox->Size = System::Drawing::Size(26, 25);
this->letterInputTextBox->TabIndex = 3;
this->letterInputTextBox->TextAlign =
    System::Windows::Forms::HorizontalAlignment::Center;
//
// InvestigateButton
//
this->InvestigateButton->Location = System::Drawing::Point(180, 230);
this->InvestigateButton->Name = L"InvestigateButton";
this->InvestigateButton->Size = System::Drawing::Size(146, 46);
this->InvestigateButton->TabIndex = 4;
this->InvestigateButton->Text = L"Визначити";
this->InvestigateButton->UseVisualStyleBackColor = true;
this->InvestigateButton->Click += gcnew System
    ::EventHandler(this, &ShyryiForm::InvestigateButton_Click);
//
// textOutputLabel
//
this->textOutputLabel->AutoSize = true;
this->textOutputLabel->Location = System::Drawing::Point(12, 285);
this->textOutputLabel->Name = L"textOutputLabel";
this->textOutputLabel->Size = System::Drawing::Size(382, 26);
this->textOutputLabel->TabIndex = 6;
this->textOutputLabel->Text = L"Вивід тексту за допомогою wordStartIndexes.";
//
// textOutputTextBox
//
this->textOutputTextBox->Location = System::Drawing::Point(12, 310);
this->textOutputTextBox->Multiline = true;
this->textOutputTextBox->Name = L"textOutputTextBox";
this->textOutputTextBox->Size = System::Drawing::Size(490, 150);
this->textOutputTextBox->TabIndex = 5;
//
// dictionaryOutputLabel
//
this->dictionaryOutputLabel->AutoSize = true;
this->dictionaryOutputLabel->Location = System::Drawing::Point(12, 465);
this->dictionaryOutputLabel->Name = L"dictionaryOutputLabel";
this->dictionaryOutputLabel->Size = System::Drawing::Size(382, 26);
this->dictionaryOutputLabel->TabIndex = 8;
this->dictionaryOutputLabel->Text = L"Вивід тексту за допомогою resultsDictionary.";
//
// dictionaryOutputTextBox
//
this->dictionaryOutputTextBox->Location = System::Drawing::Point(12, 490);
this->dictionaryOutputTextBox->Multiline = true;
this->dictionaryOutputTextBox->Name = L"dictionaryOutputTextBox";
this->dictionaryOutputTextBox->Size = System::Drawing::Size(490, 268);
this->dictionaryOutputTextBox->TabIndex = 7;
//
// ShyryiForm
//
this->AutoScaleDimensions = System::Drawing::SizeF(13, 26);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(514, 770);
this->Controls->Add(this->dictionaryOutputLabel);
this->Controls->Add(this->dictionaryOutputTextBox);

```

```

        this->Controls->Add(this->textOutputLabel);
        this->Controls->Add(this->textOutputTextBox);
        this->Controls->Add(this->InvestigateButton);
        this->Controls->Add(this->letterInputTextBox);
        this->Controls->Add(this->letterInputLabel);
        this->Controls->Add(this->textInputLabel);
        this->Controls->Add(this->textInputTextBox);
        this->Font = (gcnew System::Drawing::Font(L"Times New Roman", 13.8F));
        this->Margin = System::Windows::Forms::Padding(5);
        this->Name = L"ShyryiForm";
        this->Text = L"Ширій Богдан. АМО Домашнє завдання №22.2, варіант №24";
        this->ResumeLayout(false);
        this->PerformLayout();
    }

#pragma endregion
private: System::Void InvestigateButton_Click(System::Object^ sender, System::EventArgs^ e) {

    String^ inputText = Regex::Replace(textInputTextBox->Text, "[^a-zA-Z0-9\\s]", "");
    textOutputTextBox->Text = ""; dictionaryOutputTextBox->Text = "";

    List<int>^ wordStartIndexes = gcnew List<int>();
    Dictionary<int, String>^ resultsDictionary = gcnew Dictionary<int, String>();

    // Знаходимо індекси на початок слів
    for (int i = 0; i < inputText->Length; i++)
    {
        if (i == 0 && !Char::IsWhiteSpace(inputText[i]))
            wordStartIndexes->Add(i);
        else if (i > 0 && !Char::IsWhiteSpace(inputText[i])
            && Char::IsWhiteSpace(inputText[i - 1]))
            wordStartIndexes->Add(i);
    }

    // Виводимо слова, що починаються зі шуканої літери
    for each(int startIndex in wordStartIndexes)
    {
        String^ word = GetWordFromIndex(inputText, startIndex);
        if (Char::ToLower(word[0]) == Char::ToLower(letterInputTextBox->Text[0]))
        {
            textOutputTextBox->Text += word + " ";
            resultsDictionary[startIndex] = word;
        }
    }

    // Сортуємо решту слів в алфавітному порядку
    ComparisonHelper^ helper = gcnew ComparisonHelper(inputText);
    wordStartIndexes->Sort(gcnew Comparison<int>(helper, &ComparisonHelper::Compare));

    // Виводимо відсортовані слова
    for each(int startIndex in wordStartIndexes)
    {
        String^ word = GetWordFromIndex(inputText, startIndex);
        if (Char::ToLower(word[0]) != Char::ToLower(letterInputTextBox->Text[0]))
        {
            textOutputTextBox->Text += word + " ";
            resultsDictionary[startIndex] = word;
        }
    }

    // Виводимо результати збережені в словнику
    int counter = 0;
    for each(System::Collections::Generic::KeyValuePair<int, String> kvp in resultsDictionary)
    {
        dictionaryOutputTextBox->Text += kvp.Key + " - " + kvp.Value +
            (resultsDictionary->Count != ++counter ? ", " : " : ");
    }
}
};
}

```