ВСТУП

На даний час існує дуже багато прототипів програм, прямим призначенням яких є переклад текстів із можливістю вибору десятків найпоширеніших мов світу. Деякі з перекладачів мають можливість виконувати свою дію у режимі онлайн, інші не вимагають підключення до мережі.

Системи-перекладачі являються корисними та затребуваними програмами, які можуть застосовуватись у багатьох сферах людської життєдіяльності. Програми-перекладачі ніколи не втратять своєї актуальності, тому що мовний бар'єр існуватиме завжди, і саме тому розробка таких систем має доволі високу пріоритетність.

З часом програми-перекладачі почали все більше й більше удосконалюватись, інтегровуватись у середовища розробки та браузери і подолання мовного бар'єру стало набагато успішнішим. Але, попри те, і звичайні стандартні перекладачі не втратили свою затребуваність. Можливість швидкого наочного перекладу документованої електронної інформації являється основним пріоритетом багатьох компаній на корпорацій, в основі роботи яких лежить співпраця із іноземними колегами. Крім того перекладачі можуть використовуватись і для банального спілкування між окремими особами, тобто всюди, де вимагається швидкий та ефективний переклад інформації з однієї мови на іншу.

ВИСНОВКИ

Під час виконання даного проекту було розроблено продукт, призначений для якісного та повноцінного перекладу текстів, з можливістю подальшого зчитування та зберігання інформації у декількох форматах, також у програмі була передбачена можливість редагування вхідного тексту шляхом зміни кольору, типу та розміру шрифту. Також є можливість введення інформації стандартним чином, тобто із клавіатури. Крім того, програма також передбачає подальше своє удосконалення шляхом розширення бази даних, яка являється словником. Користувач у будь-яку мить та у будь-якому обсязі може добавляти нову інформацію до бази даних, залежно від потреб, тим самим розширюючи потенціал програми у тому напрямку, який справді необхідний.

Даний продукт було створено для використання з ціллю подолання мовного бар'єру в розмовній чи задокументованій формі, і призначений для застосування як окремою особою, так і цілою корпорацією. Не потребує підключення до Інтернету, маючи власний інтегрований локальний словник.

Розроблену програму було протестовано та проведено порівняння із аналогічними програмами, що дозволило виявити переваги та недоліки як і даної розробленої системи-перекладача, так і її аналогів. Дослідження та аналіз даного програмного продукту проводились на основі порівняння швидкості перекладу шаблонного тексту і якості самого отриманого вихідного перекладу, та порівнювались із результатами дослідження програм-аналогів.

Скриншоти програми

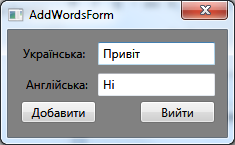


Рисунок 1.-Добавляємо слово у словник

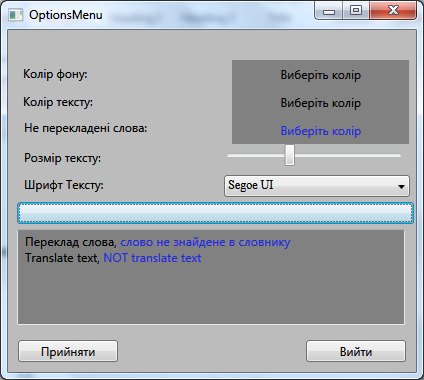


Рисунок 2. - Зміна налаштування програми

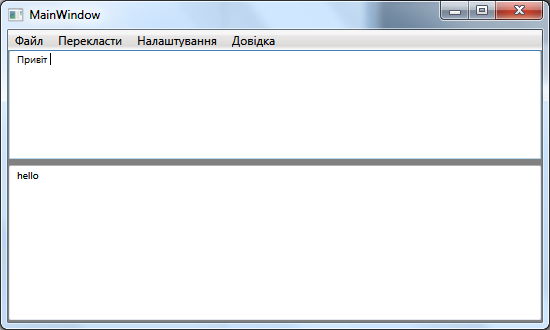


Рисунок 3. - Робота перекладача

Список використаної літератури

1. [Електронний ресурс]. – Режим доступу: http://www.msdn.com.
2. [Електронний ресурс]. – Режим доступу: http://www.codenet.ru
3. [Електронний ресурс]. – Режим доступу: http://professorweb.ru/
4. [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org/
5. С#: Пер. С англ./Дейтел Х.,Дейтел П., Листфилд Дж., Нието Т., Йегер Ш., Златкина М. – СПб.: БХВ-Петербург, 2006. – 1056 с.:ил.
6. Джеффри Рихтер - CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке С#. Мастер-класс./Пер. с англ. — М.: Издательство «Русская Редакция»; СПб.: Питер, 2007. — 656 стр. : ил.

ДОДАТОК А

Код програми

Код класу Word:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace UniversiumTranslator

{

class Word

{

public string UaWord { get; set; }

public string EngWord { get; set; }

public static List<Word> LoadWords()

{

List<Word> returnWords = new List<Word>();

StreamReader Sr = new StreamReader(Options.PathDataWords);

string[] AllWords = Sr.ReadToEnd().Split(Options.NewWords);

for (int Index = 0; Index < AllWords.Length-1; Index++)

{

string[] Word = AllWords[Index].Split(Options.SaparatorWords);

returnWords.Add(new Word() { EngWord = (Word[0].Replace(Environment.NewLine,string.Empty)), UaWord = (Word[1].Replace(Environment.NewLine,string.Empty)) });

}

return returnWords;

}

}

}

Код класу Option:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.IO;

namespace UniversiumTranslator

{

public static class Options

{

public const char SaparatorWords = '=';

public const char NewWords = ';';

private const string pathOptions = @"Options.txt";

private const char SaparatorOptions = '=';

public static string PathDataWords { get; set; }

public static Color BackgroundColor { get; set; }

public static Color ForegroundColor { get; set; }

public static Color NotTanslateWordColor { get; set; }

public static FontFamily FontFamily { get; set; }

public static UInt16 SizeFont { get; set; }

public static void ReadOptions()

{

ChangeDefultOption();

try

{

StreamReader SR = new StreamReader(pathOptions);

string allOption = SR.ReadToEnd();

string[] allLineOption = allOption.Split('\n');

for(int Index = 0; Index<allLineOption.Length; Index++)

{

string[] options = allLineOption[Index].Split(SaparatorOptions);

switch(options[0])

{

case "DBWords":

Options.PathDataWords = options[1];

break;

case "FontFamily":

Options.FontFamily = new System.Windows.Media.FontFamily(options[1]);

break;

case "ForegroundColor":

Options.ForegroundColor = (Color)ColorConverter.ConvertFromString(options[1]);

break;

case "NotTanslateWordColor":

Options.NotTanslateWordColor = (Color)ColorConverter.ConvertFromString(options[1]);

break;

case "SizeFont":

Options.SizeFont = Convert.ToUInt16(options[1]);

break;

case "BackgroundColor":

Options.BackgroundColor = (Color)ColorConverter.ConvertFromString(options[1]);

break;

}

}

SR.Close();

}

catch (FileNotFoundException)

{

System.Windows.MessageBox.Show("Не можливо завантажити налаштування.");

SaveOptions();

}

}

private static void ChangeDefultOption()

{

Options.PathDataWords = @"DataBase.TXT";

Options.FontFamily = new System.Windows.Media.FontFamily("Showcard Gothic");

Options.ForegroundColor = Colors.Black;

Options.NotTanslateWordColor = Colors.Red;

Options.BackgroundColor = Colors.Gray;

Options.SizeFont = 9;

}

public static void setOptions(System.Windows.Window updateWindow)

{

try

{

updateWindow.Background = new SolidColorBrush(Options.BackgroundColor);

updateWindow.FontFamily = Options.FontFamily;

updateWindow.Foreground = new SolidColorBrush(Options.ForegroundColor);

updateWindow.FontSize = Options.SizeFont;

}

catch (Exception) {

System.Windows.MessageBox.Show("не можливо премінити налаштування!\n Перевантажте програму!");

}

}

public static void SaveOptions()

{

StreamWriter SW = new StreamWriter(pathOptions);

SW.Write("BackgroundColor" + Options.SaparatorOptions + Options.BackgroundColor);

SW.WriteLine();

SW.Write("FontFamily" + Options.SaparatorOptions + Options.FontFamily);

SW.WriteLine();

SW.Write("ForegroundColor" + Options.SaparatorOptions + Options.ForegroundColor);

SW.WriteLine();

SW.Write("NotTanslateWordColor" + Options.SaparatorOptions + Options.NotTanslateWordColor);

SW.WriteLine();

SW.Write("SizeFont" + Options.SaparatorOptions + Options.SizeFont);

SW.WriteLine();

SW.Write("DBWords" + Options.SaparatorOptions + Options.PathDataWords);

SW.Close();

}

}

}

Код класу MainWindow:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.IO;

namespace UniversiumTranslator

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

char[] splitChar = new char[] { ' ', '\n', '.', ',' };

List<Word> allWords = new List<Word>();

public MainWindow()

{

InitializeComponent();

Options.ReadOptions();

Options.setOptions(this);

allWords = Word.LoadWords();

}

//Переклади текстів

private void TranslateUa\_Eng()

{

string[] toBeTranslate = new TextRange(ToBeTranslated.Document.ContentStart, ToBeTranslated.Document.ContentEnd).Text.Split(splitChar);

Paragraph additingParagraf = new Paragraph();

for (int IndexText = 0; IndexText < toBeTranslate.Length; IndexText++)

{

int IndexWord = 0;

bool isSarchingWords = false;

for (; IndexWord < allWords.Count; IndexWord++)

if (allWords[IndexWord].EngWord == toBeTranslate[IndexText])

{

isSarchingWords = true;

additingParagraf.Inlines.Add(new Run() { Text = allWords[IndexWord].UaWord + " ", Foreground = new SolidColorBrush(Options.ForegroundColor) });

break;

}

if (!isSarchingWords)

additingParagraf.Inlines.Add(new Run() { Text = toBeTranslate[IndexText] + " " , Foreground = new SolidColorBrush(Options.NotTanslateWordColor)});

}

TranslatedText.Document.Blocks.Clear();

TranslatedText.Document.Blocks.Add(additingParagraf);

}

private void TranslateEng\_Ua()

{

string[] toBeTranslate = new TextRange(ToBeTranslated.Document.ContentStart, ToBeTranslated.Document.ContentEnd).Text.Split(splitChar);

Paragraph additingParagraf = new Paragraph();

for (int IndexText = 0; IndexText < toBeTranslate.Length; IndexText++)

{

int IndexWord = 0;

bool isSarchingWords = false;

for (; IndexWord < allWords.Count; IndexWord++)

if (allWords[IndexWord].UaWord == toBeTranslate[IndexText])

{

isSarchingWords = true;

additingParagraf.Inlines.Add(new Run() { Text = allWords[IndexWord].EngWord + " ", Foreground = new SolidColorBrush(Options.ForegroundColor) });

break;

}

if (!isSarchingWords)

additingParagraf.Inlines.Add(new Run() { Text = toBeTranslate[IndexText] + " ", Foreground = new SolidColorBrush(Options.NotTanslateWordColor) });

}

TranslatedText.Document.Blocks.Clear();

TranslatedText.Document.Blocks.Add(additingParagraf);

}

private void AddWord\_Click(object sender, RoutedEventArgs e)

{

new AddWordsForm().ShowDialog();

}

//Перекладаємо текст

private void Translate\_Click(object sender, RoutedEventArgs e)

{

if (Ua\_Eng.IsChecked == true)

TranslateUa\_Eng();

else if (Eng\_Ua.IsChecked == true)

TranslateEng\_Ua();

else MessageBox.Show("Виберіть переклад.");

}

//Збереження файлу

private void SaveOn\_Click(object sender, RoutedEventArgs e)

{

System.Windows.Forms.SaveFileDialog SFD = new System.Windows.Forms.SaveFileDialog();

SFD.Filter = "Txt Files(\*.txt)|\*.txt|Doc files|\*.Doc";

if (SFD.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

TextRange range = new TextRange(TranslatedText.Document.ContentStart, TranslatedText.Document.ContentEnd);

StreamWriter SW = new StreamWriter(SFD.FileName);

SW.Write(range.Text);

SW.Close();

}

}

//Відкриття файлу

private void OpenFile\_Click(object sender, RoutedEventArgs e)

{

System.Windows.Forms.OpenFileDialog Ofd = new System.Windows.Forms.OpenFileDialog();

Ofd.Filter = "TxtFiles(\*.txt)|\*txt";

if (Ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

StreamReader SR = new StreamReader(Ofd.FileName, Encoding.Default);

ToBeTranslated.AppendText(SR.ReadToEnd());

SR.Close();

}

}

//Відкриваємо інформацію про автора

private void AutorInfo\_Click(object sender, RoutedEventArgs e)

{

new Autorinfo().ShowDialog();

}

//Виходимо з програми

private void Exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

private void Option\_Click(object sender, RoutedEventArgs e)

{

new OptionsMenu().ShowDialog();

} } }