﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

/\*! \mainpage Programová dokumentace

\*

\* \section intro Úvod

\*

\* Vítejne na hlavní stránce programové dokumentace aplikace na převod textu do Morseovky a naopak.

\*

\* \section motiv Smysl projektu

\*

\* Cílem této dokumentace je představení kódu, který je schopen přepsat text do morseovy abecedy a zpět.

\* - https://github.com/filipvabrousek/Morse

\*

\*/

/\*\*

\* \file Program.cs

\* \brief C# kód s komentáři

\* \author Filip Vabroušek, Jiří Zenzinger, Daniel Mareček, Erik Faltynek

\* \date Prosinec 2019

\* \details Konzolová aplikace - vlastnosti:

\* - Uživatel může přeložit text do morseovky

\* - Uživatel může přeložit morseovku do textu

\*/

namespace ConsoleApp3

{

/\*! <b>GreetUser - uvítací funkce, která uživatele seznámí s programem</b>

\*/

/\*! <b>METODA Morse2Text - převádí morseovku do textu</b>

\* \return - morseovka v textu

\*/

/\*! <b>METODA ConvertMorse2Text - hledá a nahrazuje odpovídající znaky v poli abeceda v textu</b>

\* \return - morseovka v textu

\*/

/\*! <b>METODA Text2Morse - převádí text do morseovky</b>

\* \return - text v morseovce

\*/

/\*! <b>METODA ConvertText2Morse - hledá a nahrazuje odpovídající znaky v poli abeceda v textu</b>

\* \return - text v morseovce

\*/

/\*! <b>METODA ShowResult - vypíše výstup programu uživateli na konzoli</b>

\* \return - výpis na konzoli

\*/

/\*! <b>TEST - zkouška funkcionality programu</b>

\* \return - očekávaný výstup

\*/

class Program

{

static void Main(string[] args)

{

// https://supermartas.cz/aplikace/online/prekladac-morseovky/

string[] alphabet = { "A", "B", "C", "D", "E", "F", "G", "H", "CH", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", " ", "." };

string[] morse = { ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "----", "..", ".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", "||", ".-.-.-." };

string name = "";

string text = "";

int opt = 0;

GreetUser();

opt = int.Parse(Console.ReadLine());

// convert text 2 morse

if (opt == 1)

{

text2Morse(alphabet, morse);

}

if (opt == 2)

{

morse2Text(alphabet, morse);

}

}

/\*! <b> Metoda Morse2Text převádí morseovku do textu</b>

\* \return - morseovka v textu

\*/

private static void morse2Text(string[] alphabet, string[] morse)

{

string res = "";

Console.WriteLine("Zadejte morseovku a slova oddělujte / pro převedení na text.");

res = Console.ReadLine();

char[] reschar = res.ToCharArray();

var everystr = "";

for (var i = 0; i < reschar.Length; i++) // ..-.|..|.-..|..|.--.|

{

var ch = reschar[i].ToString();

everystr += ch;

}

var resa = "";

// do not extract

if (everystr.Contains(".-.-.-."))

{

everystr.Replace(".-.-.-.", "Γ"); // .

}

// ../.-/--//

if (everystr.Contains("/"))

{

everystr.Replace("/", "Δ");

}

// ../.-/--//

Console.WriteLine(everystr);

string[] split = everystr.Split('/'); // Single quotes character

resa = convertMorse2Text(alphabet, morse, resa, split);

showResult(everystr, resa, false);

}

/\*! <b>Metoda ConvertMorse2Text hledá a nahrazuje odpovídající znaky v poli abeceda v textu</b>

\* \return - morseovka v textu

\*/

private static string convertMorse2Text(string[] alphabet, string[] morse, string resa, string[] split)

{

for (var i = 0; i < split.Length; i++)

{

var idx = Array.IndexOf(morse, split[i]);

if (idx != -1)

{

resa += alphabet[idx];

}

else

{

if (split[i] == "Γ")

{

resa += ".";

}

else if (split[i] == "Δ")

{

resa += " ";

}

}

}

return resa;

}

/\*! <b>Metoda Text2Morse převádí text do morseovky</b>

\* \return - text v morseovce

\*/

private static void text2Morse(string[] alphabet, string[] morse)

{

string text;

Console.WriteLine("Zadejte text, který chcete převést do morseovky.");

text = Console.ReadLine();

string res = "";

char[] characters = text.ToCharArray();

var stro = new String(characters);

if (stro.Contains("."))

{

stro.Replace(".", "Γ"); // .

}

if (stro.Contains(" "))

{

stro.Replace(" ", "Δ"); //

}

res = convertText2Morse(alphabet, morse, res, stro);

showResult(text, res, true);

// return text;

}

/\*! <b>Metoda ConvertText2Morse hledá a nahrazuje odpovídající znaky v poli abeceda v textu</b>

\* \return - text v morseovce

\*/

private static string convertText2Morse(string[] alphabet, string[] morse, string res, string stro)

{

for (var i = 0; i < stro.Length; i++)

{

var ch = stro[i].ToString().ToUpper();

// get index of letter in Alphabet field

// add letter from morse array in morse field

if (alphabet.Length == morse.Length)

{

Console.WriteLine("Char " + ch);

var idx = Array.IndexOf(alphabet, ch);

// Console.WriteLine("Character is " + ch + " idx is " + idx);

if (idx > -1)

{

res += morse[idx] + "/";

}

else

{

if (ch == "Γ")

{

res += ".-.-.-.";

}

else if (ch == "Δ")

{

res += "||"; // 237 is space two ||

}

else

{

Console.WriteLine("SQUARE");

Console.WriteLine(ch);

res += "⬛";

}

//res += "■";

}

}

else

{

Console.WriteLine("Length is invalid");

}

}

return res;

}

/\*! <b>Metoda ShowResult - vypíše výstup programu uživateli na konzoli</b>

\* \return - výpis na konzoli

\*/

private static void showResult(string text, string res, bool toMorse)

{

Console.ForegroundColor = ConsoleColor.Green;

if (toMorse)

{

Console.WriteLine(text + " v morseovce " + res);

}

else

{

Console.WriteLine(text + " v textu " + res);

}

Console.ForegroundColor = ConsoleColor.White;

Console.ReadKey();

}

/\*! <b>GreetUser - uvítací funkce, která uživatele seznámí s programem</b>

\* \return

\*/

private static void GreetUser()

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Volby:");

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("Zadejte 1 pro převod textu do morseovky.");

Console.WriteLine("Zadejte 2 pro převod morseovky do textu.");

}

/\*! <b>Test - Test, který vyzkouší funkcionalitu programu</b>

\* \return - očekávaný výstup

\*/

private static void test()

{

string[] alphabet = { "A", "B", "C", "D", "E", "F", "G", "H", "CH", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", " ", "." };

string[] morse = { ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "----", "..", ".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", "||", ".-.-.-." };

string res = "";

string text = "TEST MORSEOVKY";

char[] characters = text.ToCharArray();

var stro = new String(characters);

if (stro.Contains("."))

{

stro.Replace(".", "Γ"); // .

}

if (stro.Contains(" "))

{

stro.Replace(" ", "Δ"); //

}

res = convertText2Morse(alphabet, morse, res, stro);

// showResult(text, res, true);

Console.WriteLine("Testuji zda se vstup rovná očekávanému výstupu.");

if (res == "-/./.../-/||/--/---/.-./..././---/...-/-.-/-.--/")

{

Console.WriteLine("TEST PROŠEL.");

}

}

}

}