Полынь,, пятнадцатая

using System.Diagnostics.Metrics;

using System.Text;

using System.IO;

using System.Drawing;

using System;

using System.Net;

using System.Xml.Linq;

using System.Diagnostics;

namespace TextAnalysis;

public class Program

{

public static void Main(string[] args)

{

var name = Environment.UserName;

string text = "";

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Начнём??\t\t");

Console.ForegroundColor = ConsoleColor.DarkGray;

TypeWriter("Если хотите узнать о программе,, введите \"справка\"");

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write("\rНачнём?? ");

Console.ForegroundColor = ConsoleColor.Yellow;

var start = Console.ReadLine();

while (start == "справка" || start == "Справка" || start == "?" || start == "??" || start == "???" || start == "????" || start == "?????" || start == "??????")

{

Help();

Console.ForegroundColor = ConsoleColor.Yellow;

start = Console.ReadLine();

}

if (Stop(start))

{

bool check = true;

var count = 1;

Console.ForegroundColor = ConsoleColor.Green;

TypeWriter("Сколько файлов хотите открыть?? ", 10);

while (check)

{

Console.ForegroundColor = ConsoleColor.Cyan;

var stringCount = Console.ReadLine();

if (int.TryParse(stringCount, out count))

check = false;

else if (Stop(stringCount))

{

check = false;

count = 0;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkRed;

TypeWriter("Введите целое число (например,, 7) или \"стоп\": ", 10);

}

}

if (count == 0)

text = File.ReadAllText("C:\\Users\\" + name + "\\Downloads\\Dom.txt");

for (int i = 0; i < count; i++)

{

check = true;

TypeWriter("Введите путь к файлу: ");

while (check)

{

Console.ForegroundColor = ConsoleColor.Cyan;

var path = Console.ReadLine();

if (File.Exists(path))

{

text = text + File.ReadAllText(path);

check = false;

}

else if (Stop(path) && text != "")

{

i = count;

check = false;

}

else if (Stop(path))

{

text = File.ReadAllText("C:\\Users\\" + name + "\\Downloads\\Dom.txt");

i = count;

check = false;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkRed;

TypeWriter("Введите верный путь к файлу или \"стоп\": ", 10);

}

}

}

}

else

{

if (File.Exists("C:\\Users\\" + name + "\\Downloads\\Dom.txt"))

text = File.ReadAllText("C:\\Users\\" + name + "\\Downloads\\Dom.txt");

else

{

Console.ForegroundColor = ConsoleColor.DarkRed;

Console.WriteLine("Файл отсутствует.. Пожалуйста,, запустите заново и прочитайте справку..");

Console.ForegroundColor = ConsoleColor.Cyan;

return;

}

}

Console.Write("\n");

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Пожалуйста,, подождите....\t");

Console.ForegroundColor = ConsoleColor.White;

TypeWriter("-| ЗАПУСК |-", 5);

var sentences = SentencesParserTask.ParseSentences(text);

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write("\rПожалуйста,, подождите....\t");

Console.ForegroundColor = ConsoleColor.White;

Console.BackgroundColor = ConsoleColor.DarkBlue;

TypeWriter("-| ", 25);

var frequency = RandomNext.Frequency(sentences);

TypeWriter(" З", 25);

var ends = NewEnds.Ends(sentences);

TypeWriter("АПУСК ", 25);

var nextSent = NextSentence.FirstWord(sentences);

TypeWriter(" |-", 25);

Console.ForegroundColor = ConsoleColor.Cyan;

Console.BackgroundColor = ConsoleColor.Black;

Console.Write("\rПожалуйста,, подождите....\t");

Console.BackgroundColor = ConsoleColor.Green;

Console.ForegroundColor = ConsoleColor.White;

TypeWriter("-| ГОТОВО |-", 5);

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Black;

Console.WriteLine("LOTT \n");

System.Threading.Thread.Sleep(42);

while (true)

{

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("\rВведите первые слова (например,, \"Мир\" или \"Было время\"): ", 10);

Console.ForegroundColor = ConsoleColor.Yellow;

var beginning = Console.ReadLine();

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Введите количество новых слов (например,, 7): ", 10);

var check = true;

var count = 0;

while (check)

{

Console.ForegroundColor = ConsoleColor.Yellow;

var stringCount = Console.ReadLine();

if (int.TryParse(stringCount, out count) || Stop(stringCount))

check = false;

else

{

Console.ForegroundColor = ConsoleColor.DarkRed;

TypeWriter("Введите целое число (например,, 7) или \"стоп\": ", 10);

}

}

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("\n");

if (!string.IsNullOrEmpty(beginning))

{

var phrase = RandomGenerator.RandomContinue(frequency, ends, nextSent, sentences, beginning, count);

if (phrase.Length > 500)

{

Console.Write(phrase);

System.Threading.Thread.Sleep(2000);

}

else

TypeWriter(phrase);

Console.Write("\n\n");

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGray;

TypeWriter("Вы не ввели первые слова");

Console.Write("\n\n");

}

}

}

public static void TypeWriter(string text)

{

foreach (char letter in text)

{

Console.Write(letter);

System.Threading.Thread.Sleep(22);

}

}

public static void TypeWriter(string text, int pause)

{

foreach (char letter in text)

{

Console.Write(letter);

System.Threading.Thread.Sleep(pause);

}

}

public static bool Stop(string word)

{

return (word == "Stop" || word == "stop" || word == "Стоп" || word == "стоп");

}

public static void Help()

{

Console.Write("\n");

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Привет,, звёздочки и созвездия");

Console.ForegroundColor = ConsoleColor.Green;

Console.Write("\n\t");

TypeWriter("Меня зовут Марон Глясе,,");

Console.Write("\n\t");

TypeWriter("и я хочу познакомить вас с моей программой,,");

Console.Write("\n\t");

TypeWriter("но для начала скачайте в папку \"Загрузки\" файл отсюда:");

Console.Write("\n\t");

Console.ForegroundColor = ConsoleColor.DarkRed;

TypeWriter("https://www.dropbox.com/scl/fi/sr2wpf88ixnj3lxwcxm7z/Dom.txt?rlkey=675amkt0vguc4z5bltqw2vncy&st=n0kpmwwc&dl=0");

Console.Write("\n\n");

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Здесь вы можете выбрать ваши тексты и на их основе получить новые предложения");

Console.ForegroundColor = ConsoleColor.Green;

Console.Write("\n\t");

TypeWriter("Программа анализирует полученные тексты и случайным образом");

Console.Write("\n\t");

TypeWriter("продолжает ваши слова");

Console.Write("\n\t");

TypeWriter("Она завершает предложения,, и старается закончить текст");

Console.Write("\n\t");

TypeWriter("ровно за запрошенное количество слов");

Console.Write("\n\n");

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("После первого вопроса вы можете ввести \"стоп\"");

Console.Write("\n\t");

Console.ForegroundColor = ConsoleColor.Green;

TypeWriter("Вас попросят ввести число файлов,, которые вы хотите открыть");

Console.Write("\n\t");

TypeWriter("После ввода числа вы сможете по очереди добавлять файлы,,");

Console.Write("\n\t");

TypeWriter("вставляя прямые пути к .txt файлам");

Console.Write("\n\t");

TypeWriter("В любой момент вы можете остановить ввод новых файлов,,");

Console.Write("\n\t");

TypeWriter("написав \"стоп\"");

Console.Write("\n\n");

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Если вы не остановили программу,, в неё будет загружен \"Дом в котором...\"");

Console.Write("\n\t");

Console.ForegroundColor = ConsoleColor.Green;

TypeWriter("В любом случае,, тексты будут проанализированы");

Console.Write("\n\t");

TypeWriter("После этого программа начнёт свою работу");

Console.Write("\n\n");

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Работа программы");

Console.Write("\n\t");

Console.ForegroundColor = ConsoleColor.Green;

TypeWriter("Вас попросят ввести одно или несколько слов");

Console.Write("\n\t");

TypeWriter("С этих слов начнётся текст,, который выведет программа");

Console.Write("\n\t");

TypeWriter("После этого вас попросят ввести число");

Console.Write("\n\t");

TypeWriter("Это примерное число слов будущего текста без учёта вашего начала");

Console.Write("\n\t");

TypeWriter("Итоговое число слов вывода может немного отличаться");

Console.Write("\n\t");

TypeWriter("Если вы не введёте слова,, программа выдаст лишь комментарий");

Console.Write("\n\t");

TypeWriter("Если вы в место числа введёте 0 или \"стоп\",, программа выведет ваши слова");

Console.Write("\n\t");

TypeWriter("Один и тот же запрос может выдавать разные ответы программы");

Console.Write("\n\n");

Console.ForegroundColor = ConsoleColor.DarkCyan;

TypeWriter("Приятного пользования \* <3 \*");

Console.WriteLine("\n");

Console.ForegroundColor = ConsoleColor.Cyan;

TypeWriter("Теперь начнём?? ");

}

}

static class SentencesParserTask

{

private static char[] signs = { '.', ';', '!', '?', '(', ')' };

public static List<List<string>> ParseSentences(string text)

{

var sentencesList = new List<List<string>>();

foreach (var sentence in text.Split(signs))

ParseSentence(sentencesList, sentence);

return sentencesList;

}

private static void ParseSentence(List<List<string>> sentencesList, string sentence)

{

var word = new StringBuilder();

var words = new List<string>();

foreach (var symbol in sentence)

{

if (char.IsLetter(symbol) || symbol == '\'')

{

var lowerSymbol = char.ToLower(symbol);

word.Append(lowerSymbol);

}

else if (word.Length > 0)

{

words.Add(word.ToString());

word.Clear();

}

}

if (word.Length > 0)

words.Add(word.ToString());

if (words.Count > 0)

sentencesList.Add(words);

}

}

internal class RandomNext

{

public static Dictionary<string, List<string>> Frequency(List<List<string>> text)

{

var result = new Dictionary<string, List<string>>();

for (int i = 0; i < text.Count; i++)

{

if (text[i].Count > 1)

{

for (int j = 0; j < text[i].Count - 1; j++)

{

if (!result.ContainsKey(text[i][j]))

result[text[i][j]] = new List<string>();

result[text[i][j]].Add(text[i][j + 1]);

}

}

}

for (int i = 0; i < text.Count; i++)

{

if (text[i].Count > 2)

{

for (int j = 0; j < text[i].Count - 2; j++)

{

string couple = text[i][j] + " " + text[i][j + 1];

if (!result.ContainsKey(couple))

result[couple] = new List<string>();

result[couple].Add(text[i][j + 2]); ;

}

}

}

return result;

}

}

internal class NewEnds

{

public static Dictionary<string, string[]> Ends(List<List<string>> text)

{

var result = new Dictionary<string, string[]>();

foreach (var words in text)

{

string prequel = "";

if (words.Count > 1)

prequel = words[words.Count - 2];

if (!result.ContainsKey(prequel))

result[prequel] = new string[] { words[words.Count - 1] };

else

{

string[] moment = new string[result[prequel].Length + 1];

for (var i = 0; i < result[prequel].Length; i++)

moment[i] = result[prequel][i];

moment[moment.Length - 1] = words[words.Count - 1];

result[prequel] = moment;

}

}

return result;

}

}

internal class NextSentence

{

public static Dictionary<string, string[]> FirstWord(List<List<string>> text)

{

var result = new Dictionary<string, string[]>();

for (int i = 0; i < text.Count - 1; i++)

{

var end = text[i][text[i].Count - 1];

var beg = text[i + 1][0];

if (result.ContainsKey(end))

{

string[] sent = new string[result[end].Length + 1];

for (int j = 0; j < sent.Length - 1; j++)

sent[j] = result[end][j];

sent[sent.Length - 1] = beg;

result[end] = sent;

}

else

result[end] = new string[] { beg };

}

return result;

}

}

internal class RandomGenerator

{

public static string RandomContinue(

Dictionary<string, List<string>> nextWords,

Dictionary<string, string[]> lastWords,

Dictionary<string, string[]> nextSent,

List<List<string>> text,

string phraseBeginning,

int wordsCount)

{

List<string> phrase = new List<string>();

phraseBeginning = phraseBeginning.ToLower();

foreach (string word in phraseBeginning.Split(' '))

phrase.Add(word);

var count = wordsCount + phrase.Count;

var rnd = new Random();

while (phrase.Count < count)

{

var prequel = "";

if (phrase.Count > 1)

prequel = phrase[phrase.Count - 2];

var moment = phrase[phrase.Count - 1];

int newRnd = rnd.Next(1, 11);

if (phrase.Count == count - 1)

{

if (lastWords.ContainsKey(moment))

{

int forPoint = rnd.Next(lastWords[moment].Length);

phrase.Add(lastWords[moment][forPoint] + "..");

}

else

{

if (nextWords.ContainsKey(prequel + " " + moment))

{

newRnd = rnd.Next(nextWords[prequel + " " + moment].Count);

phrase.Add(nextWords[prequel + " " + moment][newRnd]);

count++;

}

else if (nextWords.ContainsKey(moment))

{

newRnd = rnd.Next(nextWords[moment].Count);

phrase.Add(nextWords[moment][newRnd]);

count++;

}

else

{

phrase[phrase.Count - 1] = moment + "..";

count--;

}

}

}

else if ((newRnd == 1 || newRnd == 2 || newRnd == 3 || newRnd == 4) && lastWords.ContainsKey(prequel) && Contains(lastWords[prequel], moment))

{

phrase[phrase.Count - 1] = moment + "..";

if (nextSent.ContainsKey(moment))

{

newRnd = rnd.Next(nextSent[moment].Length);

var begining = nextSent[moment][newRnd];

phrase.Add(begining);

}

else

{

var begining = RandFirstWord(text);

phrase.Add(begining);

}

}

else if (nextWords.ContainsKey(prequel + " " + moment))

{

newRnd = rnd.Next(nextWords[prequel + " " + moment].Count);

phrase.Add(nextWords[prequel + " " + moment][newRnd]);

}

else if (nextWords.ContainsKey(moment))

{

newRnd = rnd.Next(nextWords[moment].Count);

phrase.Add(nextWords[moment][newRnd]);

}

else

{

phrase[phrase.Count - 1] = moment + "..";

if (nextSent.ContainsKey(moment))

{

newRnd = rnd.Next(nextSent[moment].Length);

var begining = nextSent[moment][newRnd];

phrase.Add(begining);

}

else

{

var begining = RandFirstWord(text);

phrase.Add(begining);

}

}

}

StringBuilder result = new StringBuilder();

foreach (string word in phrase)

{

result.Append(word);

result.Append(" ");

}

return result.ToString().Trim();

}

public static bool Contains(string[] array, string value)

{

foreach (string word in array)

{

if (word == value) return true;

}

return false;

}

public static string RandFirstWord(List<List<string>> text)

{

var rnd = new Random();

int r = rnd.Next(0, text.Count);

if (text.Count != 0)

return text[r][0];

else

return null;

}

}