

SWPSD - 2023/2024

Budowa aplikacji głosowych na bazie Microsoft Speech Platform 11

Czynności wstępne:

Zainstalować Microsoft Speech Platform SDK 11 oraz dodatki, które umożliwiają syntezę głosu i rozpoznawanie mowy w języku polskim. Utworzyć puste rozwiązanie (nie zawierające żadnych projektów) o nazwie: SWPSD.

Zadanie do wykonania:

- 1) Do rozwiązania SWPSD dodaj projekt programu konsolowego (aplikacja konsoli .NET Framework) w języku C#. Projekt nazwij: Lab1;
- 2) Do projektu Lab1 dodaj bibliotekę: Microsoft.Speech.dll, która znajduje się w katalogu: C:\Program Files\Microsoft SDKs\Speech\v11.0\Assembly;
- 3) W menadżerze konfiguracji (prawy przycisk myszy na: Rozwiązanie "SWPSD") zmień aktywną konfigurację Any CPU na konfigurację x64 (należy utworzyć nową konfigurację x64 na bazie Any CPU);
- 4) W pliku: **Program.cs** dodaj możliwość korzystania z następujących przestrzeni nazw:

```
using Microsoft.Speech.Recognition;
```

- using Microsoft.Speech.Synthesis;
- using System.Globalization;
- 5) Dodaj do klasy **Program** (plik: **Program.cs**) następujące składowe:

```
static bool speechOn = true;
```

- static SpeechSynthesizer pTTS = new SpeechSynthesizer();
- static SpeechRecognitionEngine pSRE;
- 6) Zmień metodę Main następująco:

{

```
try
    pTTS.SetOutputToDefaultAudioDevice();
    pTTS.Speak("Witam w kalkulatorze");
    // Ustawienie języka rozpoznawania:
    CultureInfo ci = new CultureInfo("pl-PL");
    // Utworzenie "silnika" rozpoznawania:
    pSRE = new SpeechRecognitionEngine(ci);
    // Ustawienie domyślnego urządzenia wejściowego:
    pSRE.SetInputToDefaultAudioDevice();
    // Przypisanie obsługi zdarzenia realizowanego po rozpoznaniu wypowiedzi
                                                             zgodnej z gramatyka:
    pSRE.SpeechRecognized += PSRE_SpeechRecognized;
    // Budowa gramatyki numer 1 - POLECENIA SYSTEMOWE
    // Budowa gramatyki numer 1 - określenie komend:
    Choices stopChoice = new Choices();
    stopChoice.Add("Stop");
    stopChoice.Add("Pomoc");
```

```
// Budowa gramatyki numer 1 - definiowanie składni gramatyki:
            GrammarBuilder buildGrammarSystem = new GrammarBuilder();
            buildGrammarSystem.Append(stopChoice);
            // Budowa gramatyki numer 1 - utworzenie gramatyki:
            Grammar grammarSystem = new Grammar(buildGrammarSystem); //
            // Budowa gramatyki numer 2 - POLECENIA DLA PROGRAMU
            // Budowa gramatyki numer 2 - określenie komend:
            Choices chNumbers = new Choices(); //możliwy wybór słów
            string[] numbers = new string[] { "0", "1", "2", "3", "4", "5", "6", "7",
            chNumbers.Add(numbers);
            // Budowa gramatyki numer 2 - definiowanie składni gramatyki:
            GrammarBuilder grammarProgram = new GrammarBuilder();
            grammarProgram.Append("Oblicz");
            grammarProgram.Append(chNumbers);
            grammarProgram.Append("plus");
            grammarProgram.Append(chNumbers);
            // Budowa gramatyki numer 2 - utworzenie gramatyki:
           Grammar g_WhatIsXplusY = new Grammar(grammarProgram); //gramatyka
            // -----
            // Załadowanie gramatyk:
            pSRE.LoadGrammarAsync(g_WhatIsXplusY);
            pSRE.LoadGrammarAsync(grammarSystem);
            // Ustaw rozpoznawanie przy wykorzystaniu wielu gramatyk:
            pSRE.RecognizeAsync(RecognizeMode.Multiple);
            // -----
            Console.WriteLine("\nAby zakonczyć działanie programu powiedz 'STOP'\n");
           while (speechOn == true) {; } //petla w celu unikniecia zamkniecia programu
            Console.WriteLine("\tWCIŚNIJ <ENTER> aby wyjść z programu\n");
            Console.ReadLine();
        catch (Exception ex)
            Console.WriteLine(ex.Message);
           Console.ReadLine();
7) Dodaj do metody PSRE_SpeechRecognized następujący kod:
        string txt = e.Result.Text;
        string comments;
        float confidence = e.Result.Confidence;
        comments = String.Format("ROZPOZNANO (wiarygodność: {0:0.000}): '{1}'",
                                                           e.Result.Confidence, txt);
        Console.WriteLine(comments);
        if (confidence > 0.60)
        {
            if (txt.IndexOf("Stop") >= 0)
               speechOn = false;
            else if (txt.IndexOf("Pomoc") >= 0)
               pTTS.SpeakAsync("Składnia polecenia: Oblicz liczba plus liczba. Na
                                                           przykład: dwa plus trzy");
```

```
else if ((txt.IndexOf("Oblicz") >= 0) && (txt.IndexOf("plus") >= 0) &&
                                                                      (speechOn == true))
                string[] words = txt.Split(' ');
                int liczba1 = int.Parse(words[1]);
                int liczba2 = int.Parse(words[3]);
                int suma = liczba1 + liczba2;
                comments = String.Format("\tOBLICZONO: {0} + {1} = {2}",
                                                                 liczba1, liczba2, suma);
                Console.WriteLine(comments);
                pTTS.SpeakAsync("Wynik działania to: " + suma);
            }
        }
        else
        {
            comments = String.Format("\tNISKI WSPÓŁCZYNNIK WIARYGODNOŚCI
                                                                   powtórz polecenie");
            Console.WriteLine(comments);
            pTTS.SpeakAsync("Proszę powtórzyć");
        }
8) Przetestuj aplikację;
9) Zaproponuj zmiany, które umożliwią wykonywanie takich działań jak: dodawanie,
```

odejmowanie, mnożenie i dzielenie.