# ZAAWANSOWANE PROGRAMOWANIE LAB06: KOLEKCJE I OKNA DIALOGOWE

## Data: 7.11.2023

# WYKŁAD

Wykład III (kolekcje) I IV (operacje dyskowe):

http://www.cs.put.poznan.pl/mradom/teaching/lectures/ZaawProg/zp\_w03.zip oraz http://www.cs.put.poznan.pl/mradom/teaching/lectures/ZaawProg/zp\_w04.zip

# LABORATORIUM

Projekt dostępny na stronie pod adresem:

http://www.cs.put.poznan.pl/mradom/teaching/laboratories/WindowsFormsApp\_Collections.7z

Okno programu wygląda następująco:

🖳 Kolekcje i komunikaty		—	×
OpenFileDialog         Ile najczęstszych słów pokazać         Pobierz wartość           OpenFileDialog         Pokaż słów:         30           Click!	z klucza Klucz:	Value:	
Zawartość Statystyka			
			~
			~

Po wciśnięciu przycisku OpenFileDialog program wyświetli okno dialogowe wczytywania pliku – musimy wybrać plik tekstowy. Kod jest następujący:

```
OpenFileDialog openFileDialog1 = new OpenFileDialog();
```

```
openFileDialog1.InitialDirectory = Environment.CurrentDirectory;
```

DialogResult result = openFileDialog1.ShowDialog(); // Show the dialog.

Środkowa linijka powoduje, że domyślnie po otwarciu okna dialogowego "lądujemy" w katalogu, w którym jest plik wykonywalny programu (domyślnie /bin/debug w katalogu projektu). Jest tam plik eden.txt który proszę wybrać. Efektem będzie jego wyświetlenie się w głównym oknie programu, ale to akurat najmniej ciekawy efekt. W międzyczasie wydarzyło się bardzo wiele interesujących rzeczy.

Założenie wstępne: jedynym polem naszej klasy okna Form1 jest słownik, którym się teraz zainteresujemy. Chodzi o linijkę:

#### Dictionary<string, int> ds = new Dictionary<string, int>();

Umieszczoną przed konstruktorem klasy. Powoduje ona utworzenie pustego na początku słownika, w którym klucze (które muszą być unikalne) będą typu string, a dane związanego 1 do 1 z każdym kluczem to liczby – typ int.

Najpierw opiszmy kod czytania pliku do słownika.

```
string[] tablica = null;
if (result == DialogResult.OK) // Test result.
{
    string file = openFileDialog1.FileName;
    try {
        string text = File.ReadAllText(file);
        textBox1content.AppendText(text);
        text = text.Replace(System.Environment.NewLine, " ");
        tablica = text.Split(' ');
        przeczytaneSlowa = text.Length;
    } catch (IOException) {
    }
}
```

Linijka otwierająca okno dialogowe o której pisałem wcześniej to:

DialogResult result = openFileDialog1.ShowDialog(); // Show the dialog.

Powoduje ona, że wynik działania na tym oknie jest zapisywany w zmiennej result.

Pierwsza instrukcja if sprawdza właśnie, czy wcisnęliśmy tam przycisk OK po wybraniu pliku.

Nazwa pliku pobierana jest w następnej instrukcji (string file = openFileDialog1.FileName;)

Następnie jest on czytany jako całość (instrukcja: string text = File.ReadAllText(file);) a następnie dodawany do głównego textboxa - textBox1content okna.

Dodatkowo, cały tekst w zmiennej text podlega obróbce – najpierw wszystkie entery są zamieniane na spacje (text = text.Replace(System.Environment.NewLine, ""); ), a następnie ta spacja służy jako rozdzielnik przy podziale obiektu string text na słowa, które lądują w tablicy tablica.

Instrukcja (textBox1content.SelectionStart = 0;) powoduje, że po wczytaniu tekstu do textBoxa, przewijany jest on na sam początek automatycznie.

Kolejny blok instrukcji jest następujący:

```
foreach (string word in tablica) {
    string tmp = word.ToLower();
    if (ds.ContainsKey(tmp)) {
        ds[tmp] += 1;
    } else {
        ds.Add(tmp, 1);
    }
}
```

Powoduje on dodawanie słów do słownika w taki sposób, że:

- Jeśli słowo (czyli klucz) już jest w słowniku, to wartość z nim związana jest zwiększana o 1
- Jeśli go w słowniku jeszcze nie ma, jest tam umieszczane z wartością 1.

Magia dzieje się tutaj:

Pierwsza linia używa subjęzyka LINQ do stworzenia słownika-kolekcji items w której słowa (klucze) są posortowane w kolejności rosnących wartości. W skrócie: sortować bezpośrednio słownik można po kluczach, np. gdybyśmy chcieli kolejność alfabetyczną. Ale ponieważ chcemy wyświetlić je od najpopularniejszego w tekście do najrzadszego, potrzebna jest tego typu instrukcja.

Instrukcje dalsze wyświetlają domyślnie 30 najpopularniejszych słów w drugiej zakładce okna programu. Ile dokładnie ma się wyświetlić – za to odpowiada wartość w obiekcie numericUpDown1.

Pomocnicza funkcja NormalizeStr której kod jest zaraz niżej w projekcie odpowiada za dodanie spacji. Dodatkowo ustawiłem czcionkę tekstu textBox2 w zakładce drugiej na taką, która ma stałą szerokość, żeby cała ta sprawa ze spacjami miała sens (patrz obrazek niżej).

Scaling on your ma	in display is	set to 200%. Restart Vis	ual Studio with 100% scaling	Help me decide		×	© © ∯ 🛱 • ]o •	ち 🖒 🕫 🗟 💠 🎤 🗕		
							Search Solution Explorer (C	:trl+;)		
Kolekcje i komun OpenFileDialog	lle najcz Pokaż	restszych słów pokazać Czcionka	Pobierz wartość z klucza		×	Ш	Solution 'WindowsFor WindowsFormsA P Properties	msApp_ZaawProgLab07' (1 of 1 pr <b>pp_ZaawProgLab07</b>		
Zawartość Statystyka		Czcionka: Styl czcionki: Rozmii Lucida Console Lucida Console Lucida Fax Lucida Fax Lucida Fax Lucida Sans Unicode Malgun Gothic Mariett Mariett Przekreślenie Połkreślenie Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład Przykład		Rozmiar: 10 0 Ant 12 Ant 14 Ant 20 22 V	K		<ul> <li>▶ ■■ References</li> <li>♀ App.config</li> <li>■ Form1.cs</li> <li>▶ ↑ Form1.Designer.cs</li> <li>↑ Form1.resx</li> <li>▷ C■ Program.cs</li> </ul>			
			Środkowoeuropejski	~		10	Properties	<del>▼</del> ₽ 3		
							textBox2stats System.Windows.Forms.TextBox			
	l					- 11	1 💱 🖓 🗲 🎤			
						AutoCompleteCustomSourc (Collection)				
						- 11	AutoCompleteMode	None		
						- 18	AutoCompleteSource	None		
						- 11	BackColor	Window		
						- 28	BorderStyle	Fixed3D		
d					o	-	CausesValidation	True		
							CharacterCasing	Normal		
ror List							ContextMenuStrip	(none)		
ntire Solution	- 🛛 🖸	Errors 0 Warnings	1 0 of 5 Messages	Build + IntelliSense	-		Cursor	lBeam		
					_	0.0	Dock	Fill		
arch Error List						2.	Enabled	True		
Code Desc	ription	Project	File	Line Suppre	ession State		I Font	Lucida Console; 10,125pt		
							ForeColor	WindowText		

Przycisk drugi (**Click**) proszę przeanalizować we własnym zakresie. W ogólności pokazuje on jak pobrać wartość w konkretnym kluczu ze słownika oraz przede wszystkim jest przykładem na całą masę okienek dialogowych, bardzo przydatnych do wyświetlania krótkich komunikatów przez programy.

Trochę wartościowych tutoriali o okna dialogowych:

http://www.dotnetperls.com/messagebox-show

http://www.dotnetperls.com/openfiledialog

http://www.dotnetperls.com/savefiledialog