Platformy Technologiczne

ćw. 1

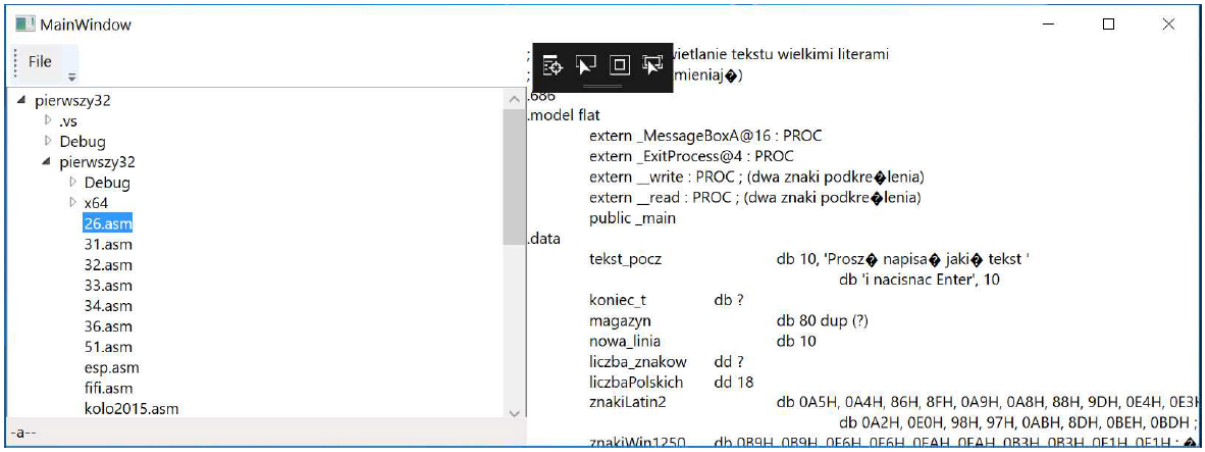
Windows Presentation Foundation

Celem poniższego laboratorium jest stworzenie aplikacji Windows Presentation Foundation (WPF) do przeglądania i modyfikowania struktury plików odczytanych z wybranego folderu na dysku. Aplikacja powinna posiadać rozwijane menu w górnej belce z dwoma opcjami:

- File->Open – wskazanie korzenia, od którego należy wczytać strukturę plików
- File->Exit – zamknięcie aplikacji

W centralnej części ekranu, z lewej strony, powinno być wyświetlane rozwijalne drzewo elementów (TreeView) pokazujące foldery i pliki. Element folderu powinien być rozwijalny wtedy i tylko wtedy, gdy folder zawiera w sobie jakieś elementy.

Na dole ekranu powinien być wyświetlany pasek statusu z atrybutami 'rash' wskazanego pliku lub folderu.



Każdy z wyświetlonych elementów posiadać powinien swoje własne menu kontekstowe pojawiające się po kliknięciu na nie prawym przyciskiem myszy. Powinno ono posiadać podstawowe opcje Delete dostępną dla każdego elementu drzewa. Pliki powinny być usuwane zarówno z widocznego w aplikacji drzewa jak i z dysku. W przypadku folderów ich zawartość także powinna zostać usunięta.

Dodatkowo foldery i pliki powinny mieć po jednej dedykowanej opcji w menu kontekstowym dostępnym tylko dla nich:

- Foldery powinny posiadać opcję Create umożliwiającą tworzenie nowych plików lub katalogów. Po kliknięciu w to pole powinno pojawiać się nowe okno z formularzem do wypełnienia.
- Pliki o rozszerzeniu ".txt" powinny posiadać opcję Open pozwalającą na pokazanie zawartości wskazanego pliku w prawym panelu obok drzewa folderów i plików.

W kolejnych 5 zadaniach podano rozwiązanie standardowe. Studenci zaawansowani mogą zrealizować rozwiązanie alternatywne opisane w dodatkowym zadaniu 6.

Zadanie 1. Stworzenie struktury aplikacji WPF

(1 pkt)

Instrukcję tworzenia prostej aplikacji WPF oraz języka XAML można znaleźć tutaj:

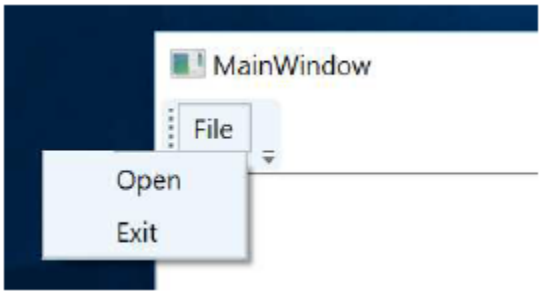
https://msdn.microsoft.com/pl-pl/library/jj153219.aspx

http://www.wpf-tutorial.com/xaml/what-is-xaml/

Do podziału obszaru okna na panele można użyć elementu typu Grid. Instrukcję na ten temat można znaleźć tutaj:

http://www.wpf-tutorial.com/panels/grid-rows-and-columns/

Najprostszym sposobem zaimplementowania menu aplikacji jest stworzenie elementu ToolBar w górnej części aplikacji. Element ToolBar będzie zawierał w sobie element Menu z dwoma elementami typu MenuItem.



Materiały instruktażowe nt. konstruowania menu można znaleźć tutaj:

http://www.wpf-tutorial.com/common-interface-controls/menu-control/

Zadanie 2. Wczytanie struktury plików i folderów

(1 pkt)

WPF nie posiada bezpośrednio dialogu do pobrania od użytkownika ścieżki do wybranego folderu. W tym celu należy skorzystać z klasy FolderBrowserDialog() udostępnionej w przestrzeni nazw System.Windows.Forms. Przykładowe użycie:

```
var dlg = new FolderBrowserDialog() { Description = "Select directory to open"};
dlg.ShowDialog();
```

Wczytane pliki i foldery powinny zostać zaprezentowane za pomocą elementu TreeView. Poszczególne elementy powinny zostać dodane do drzewa poprzez utworzenie struktur TreeViewItem i dodanie folderu źródłowego do pola Items elementu TreeView. Właściwość Header klasy TreeViewItem powinna zostać ustawiona na nazwę pliku, a właściwość Tag na jego ścieżkę. Poniżej pokazano przykładowy fragment kodu:

```
var item = new TreeViewItem
{
  Header = itemName,
  Tag = itemPath
};
root.Items.Add(item);
```

Zadanie 3. Pokazywanie treści pliku tekstowego

(1 pkt)

W WPF tworzenie menu kontekstowych polega na przypisaniu do własności ContextMenu danej kontrolki nowej instancji klasy ContextMenu z przestrzeni nazw System.Windows.Controls a następnie dodanie do tego menu elementów MenuItem.

Odczyt pliku tekstowego następuje za pomocą klasy TextReader. Warto otwarcie pliku umieścić w nagłówku instrukcji using, a odczyt zawartości wewnątrz tej instrukcji. To spowoduje zwolnienie zasobów systemowych (w tym uchwytu pliku) po zakończeniu instrukcji.

```
using (var textReader = System.IO.File.OpenText(filePath))
{
  string text =textReader.ReadToEnd();
  ...
}
```

Tekst trzeba wczytać do wielowierszowego elementu TextBlock umieszczonego w prawym panelu obok drzewa folderów i plików. By ułatwić odczyt większych plików element TextBlock powinien znajdować się w środku ScrollViewer.

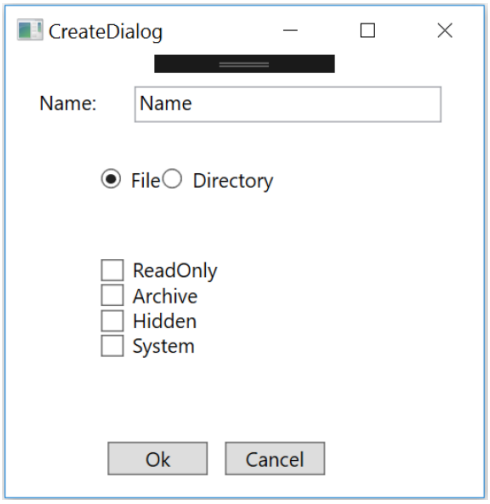
Zadanie 4. Usuwanie i tworzenie plików i folderów

(1 pkt)

Do usuwania folderu lub pliku z dysku posłużyć mogą funkcje Directory.Delete() oraz File.Delete(), natomiast w celu usunięcia elementu z wyświetlanego drzewa należy usunąć element z listy Items jego elementu nadrzędnego. Aby poprawnie wykrywać, który element ma zostać usunięty można założyć, że element ten został wcześniej zaznaczony pojedynczym kliknięciem, przez co jest on dostępny w własności SelectedItem elementu TreeView.

Menu kontekstowe folderów powinno umożliwiać stworzenie pliku lub folderu. Aby to zrobić, trzeba zaprojektować osobny formularz, który trzeba otwierać w menu kontekstowym w opcji Create.

Formularz ten powinien zawierać możliwość stworzenia pliku lub folderu, wpisania nazwy oraz wybrania atrybutów dla tworzonego elementu. Nowy element powinien być dzieckiem katalogu zaznaczonego w TreeView podczas klikania Create oraz pojawić się w drzewie po jego utworzeniu. Przykład formularza poniżej:



W przypadku błędu tworzenia użytkownik powinien otrzymać stosowny komunikat. W tym celu obsługa przycisku Ok powinna być ujęta w instrukcję try ... catch.

Zadanie 5. Wyświetlanie atrybutów pliku

(1 pkt)

Ostatnim zadaniem jest stworzenie pasku stanu, na którym wyświetlane będę atrybuty wskazanego pliku/katalogu. Atrybuty należy wyświetlić jako ciąg 'rash', gdzie każdy znak świadczy o posiadaniu konkretnego atrybutu. Do stworzenia takiego pasku stanu należy wykorzystać klasę StatusBar z elementem TextBlock.

Aby sprawdzić atrybuty danego pliku lub folderu można skorzystać z funkcji File.GetAttributes(path). Operacje na atrybutach plików w środowisku .Net wykonuje się na bazie nakładania odpowiednich masek operacjami and lub or.

Zadanie 6. (dodatkowe) Model MVVM

(+1 pkt)

Wykorzystać wiązanie danych (DataBinding) i hierarchiczne szablony danych (DataTemplate), aby wprowadzić model MVVM. W tym modelu o wyglądzie i zachowaniu elementu drzewa widoku decyduje obiekt przypisany do elementu TreeViewItem jako DataContext. Trzeba zdefiniować klasy FolderItem oraz FileItem, reprezentujące odpowiednio folder i plik. Klasa FolderItem powinna zawierać właściwość Items klasy ObservableCollection, której elementy są wczytywane dopiero po rozwinięciu elementu drzewa.

Materiały można znaleźć tutaj:

http://www.wpf-tutorial.com/treeview-control/treeview-data-binding-multiple-templates/

https://www.tutorialspoint.com/mvvm/index.htm

https://social.msdn.microsoft.com/Forums/vstudio/en-US/feac927d-b73c-4361-82f4-cd0a3f055ad0/treeview-binding-and-hierarchicaldatatemplate?forum=wpf