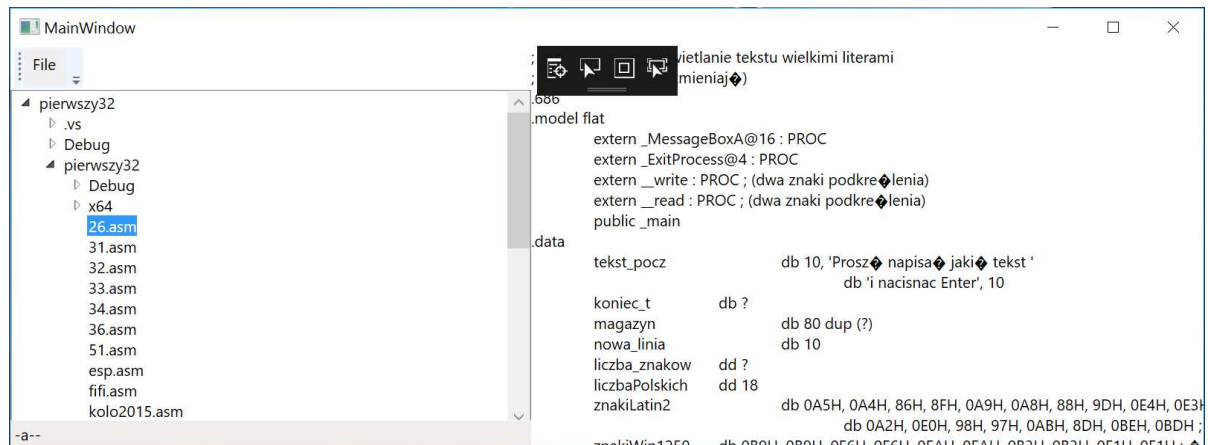


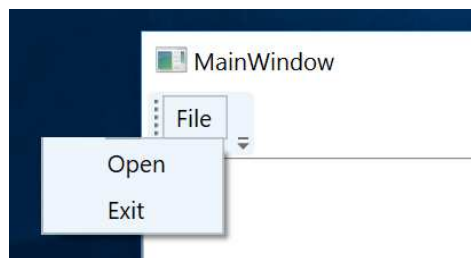
Celem poniższego laboratorium jest stworzenie aplikacji Windows Presentation Foundation (WPF) do przeglądania i modyfikowania struktury plików odczytanych z wybranego folderu na dysku.



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

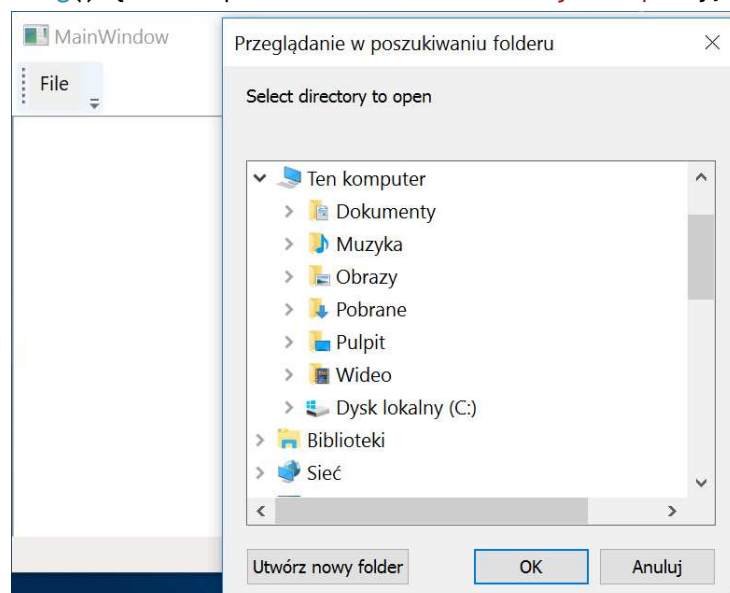
Najprostszym sposobem zaimplementowanie takiego menu jest stworzenie elementu ToolBar umieszczonego w górnej części aplikacji zawierającego w sobie element Menu z dwoma Menuitemem.



WPF nie posiada bezpośrednio dialogu do pobrania od użytkownika ścieżki do wybranego folderu. W tym celu należy skorzystać z `FolderBrowserDialog()` udostępnionego w `System.Windows.Forms`

Przykładowe użycie `FolderBrowserDialog()`:

```
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`. Parametr `Header` klasy `TreeViewItem` powinien zawierać nazwę pliku, a parametr `Tag` jego ścieżkę.

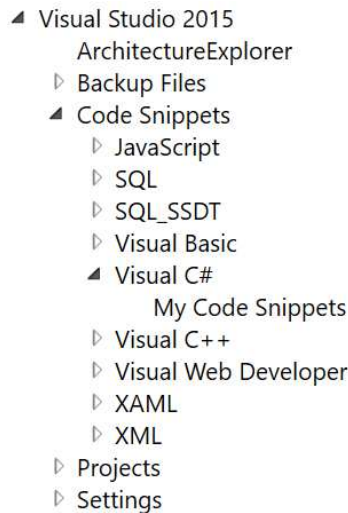
```
var root = new TreeViewItem
```

```
{
    Header = rootName,
    Tag = rootPath
};
```

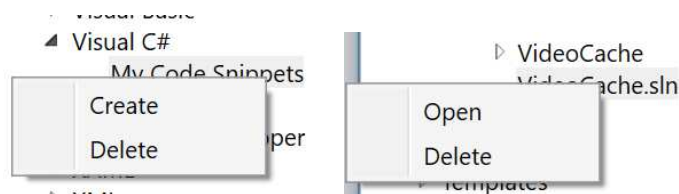
```
var item = new TreeViewItem
```

```
{
    Header = itemName,
    Tag = itemPath
};
```

```
root.Items.Add(item);
```



W WPF'ie tworzenie menu kontekstowych polega na przypisaniu do własności `ContextMenu` danej kontrolki nowej instancji klasy `ContextMenu` z namespace'u `System.Windows.Controls` a następnie dodanie do menu elementów `MenuItem`.



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. Jeśli element posiada parametr `ReadOnly` należy go zdjąć przed usunięciem.

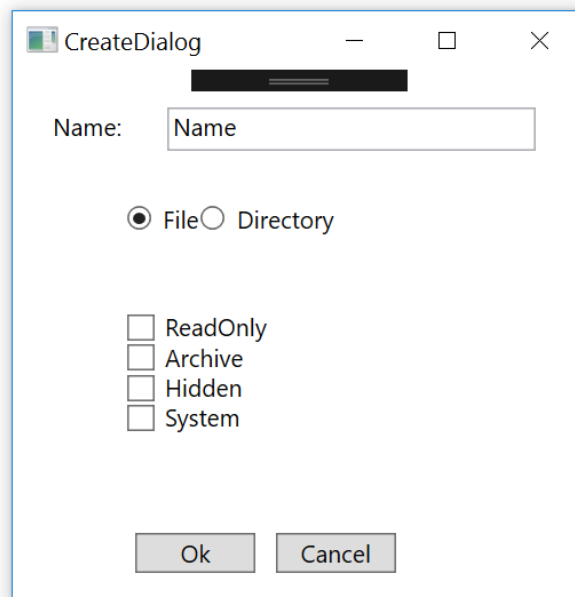
Aby sprawdzić atrybuty danego pliku lub folderu można skorzystać z funkcji `File.GetAttributes(path)`. Operacje na atrybutach plików w środowisku .Net wykonuje się na bazie nakładania odpowiednich masek operacjami `and` lub `or`, a następnie używając analogicznej funkcji ustawiającej atrybuty dla danego pliku - `File.SetAttributes(path, attributes)`.

Do usuwania 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 parenta.

Aby poprawnie wykrywać, który element ma zostać usunięty można założyć, że element został wcześniej zaznaczony pojedynczym kliknięciem przez co jest on dostępny w własności `SelectedItem` elementu `TreeView`.

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. 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.



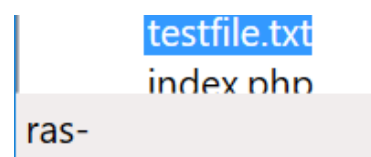
Nazwa tworzonego pliku powinna zostać sprawdzona przy użyciu `Regex.IsMatch(string input, string pattern)`, by spełniać poniższe standardy:

- nazwa pliku składa się z *basename* oraz *extension* oddzielonych kropką;
- *basename* stanowi od 1 do 8 znaków (litera, cyfry, podkreślenie, tylda, minus);
- dozwolone *extension* to *txt*, *php* i *html*.

W przypadku podania błędnej nazwy użytkownik powinien otrzymać stosowny komunikat.

Pojedyncze pliki niezależnie od formatu powinny posiadać opcję Open pozwalającą na wczytanie zawartości pliku wskazanego w `TreeView` do wielo-wierszowego `TextBlock` wyświetlającego się obok drzewa lub w nowym oknie. By ułatwić odczyt większych plików element `TextBlock` powinien znajdować się w środku `ScrollView`.

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



Zadanie		Punktacja	
Stworzenie menu z działającymi funkcjami Open i Exit		0,5 pkt	
Zaimplementowanie możliwości otwierania plików		1 pkt	
Zaimplementowanie możliwości usuwania elementów		0,5 pkt	
Zaimplementowanie możliwości tworzenia elementów	Wyświetlanie formularza	0,5 pkt	1,5 pkt
	Poprawne utworzenie elementu	0,5 pkt	
	Ustawienie atrybutów DOS'owych	0,5 pkt	
Wyświetlanie atrybutów na pasku stanu		0,5 pkt	