

Tematyka ćwiczeń

- różnice między Grid, StackPanel, DockPanel, WrapPanel.
- jak działa ComboBox (lista rozwijana) w WPF?
- WPF a „obiektywka” - jak to połączyć?
- ukrywanie/odkrywanie kontrolek
- praktyczne wykorzystanie „sender”, przypisanie tej samej metody po kliknięciu w różne buttony

1. Stwórz nowy projekt dla WPF.
 2. Za pomocą designera lub ręcznie kodem usuń grid w MainWindow.
 3. Na window za pomocą designera upuść StackPanel.
 4. Na StackPanel upuść 5 przycisków. Śledź zmiany w kodzie. Jakie parametry możesz pozmienić w designer? jakie w Properties? a jakie ręcznie w kodzie?
 - a. czy możesz w dowolny sposób umieścić buttony?
 - b. dla każdego z buttonów poustawiaj parametry takie jak VerticalAlignment, HorizontalAlignment, Margin (tu zwróć uwagę co znaczy podanie jednego, dwóch lub czterech parametrów), VerticalContentAlignment, HorizontalContentAlignment, Padding
 - c. spójrz co możesz zrobić w oknie Design, co w Properties, co pod prawym klawiszem myszy, a co tylko w kodzie.
 5. Ręcznie w kodzie zmień StackPanel na WrapPanel. Co się zmieniło?
 6. Teraz WrapPanel zamień na DockPanel (naciskając prawym klawisz myszy na WrapPanel, potem Change Layout Type). Sprawdź zmiany.
 7. Zmień DockPanel na Grid.
 8. Za pomocą designera stwórz kilka rzędów i kolumn. Spójrz na kod.
 9. Za pomocą designera i bezpośrednio w kodzie zrób tak, aby każdy wcześniej dodany button był w oddzielnym elemencie w siatce.
 10. Upewnij się że w kodzie do każdego button jest przypisany rząd i kolumna. Następnie za pomocą designera zmień szerokość niektórych rzędów i kolumn.
-
11. Stwórz nowy projekt. W gridzie umieść ComboBox.
 12. Za pomocą PPM dodaj kilka itemów na listę rozwijaną.

13. Dwukrotnie kliknij na listę i dodaj eventy po wybraniu odpowiedniej opcji.

14. Stwórz nowy projekt. Stworzymy to samo bez designera.

15. Wewnątrz grid dodaj kod:

```
<ComboBox  
HorizontalAlignment="Center"  
Margin="10,10,0,0"  
VerticalAlignment="Top"  
Width="200"  
Loaded="ComboBox_Loaded"  
SelectionChanged="ComboBox_SelectionChanged"/>
```

16. Ustaw kursor na Combox_Loaded i wciśnij F12. VS utworzy automatycznie odpowiednią metodę. We wnętrzu wpisz:

```
// ... lista  
List<string> data = new List<string>();  
data.Add("Book");  
data.Add("Computer");  
data.Add("Chair");  
data.Add("Mug");  
  
// ... ustawienie referencji  
var comboBox = sender as ComboBox;  
  
// ... ustawienie źródła  
comboBox.ItemsSource = data;  
  
// ... czy cos domyslnie ma byc wybrane  
comboBox.SelectedIndex = 0;
```

17. zrób ja aby utworzyć metodę ComboBox_SelectionChanged, we wnętrzu wpisz:

```
// ... dostep do combo.  
var comboBox = sender as ComboBox;  
  
// ... ustawimy wybraną opcję jako tytuł okna  
string value = comboBox.SelectedItem as string;  
this.Title = "Selected: " + value;
```

18. Która metoda jest dla Ciebie lepsza, wygodniejsza, bardziej elastyczna?

19. Stwórz nowy projekt WPF. Dodaj w nim klasę Osoba z polami imie(string), nazwisko(string), dataUrodzenia(DateTime) -mod private.

Dodaj w tej klasie konstruktor domyślny (pusty).

20. W klasie Osoba dodaj metodę na ustawienie jednocześnie imienia, nazwiska i daty urodzenia.

21. W klasie Osoba dodaj metodę ObliczWiek() zwracającą typ int, który z daty urodzenia oblicza wiek.

22. W klasie Osoba przestroń metodę ToString() aby zwracała imie, nazwisko, datę urodzenia oraz wiek w kolejnych wierszach (z użyciem metody oblicz wiek).

23. W MainWindow dodaj 3 etykiety (Label), 3 pola tekstowe (TextBox), 1 przycisk (buton), 1 blok tekstowy(TextBlock). Przykładowo:

Imie

Nazwisko

Data ur. (format DD-MM-YYYY)

Opis osoby

24. W klasie MainWindow dodaj prywatne pole osoba typu Osoba i zainicjuj obiekt. np.

```
private Osoba osoba = new Osoba();
```

(kod najlepiej dodać we wnętrzu klasy MainWindow a przed konstruktorem w tej klasie).

25. Dodaj event po kliknięciu, metoda ma wykonywać następujące czynności:

1) ustawić parametry obiektu Osoba z pobranych wartości z pól tekstowych

2) należy zamienić string z Textboxa od daty na typ DateTime (w przypadku błędnego formatu ma być komunikat dla użytkownika np. za pomocą MessageBox) - odpowiedź: możesz użyć Convert.ToDateTime,

DateTime.Parse lub DateTime.ParseExact

3) We właściwości Text kontrolki TextBlock podstawić stringa, który będzie zwrócony po wywołaniu dla obiektu osoba metody ToString().

26. Stwórz nowy projekt. Stworzymy program z mechanizmem chowania i pokazywania kontrolek. Na okno designera upuść dwa butony i nadaj im nazwy Przycisk1 i Przycisk2.

27. We wnętrzu metody Przycisk1_Click dodaj następująco kod:

```
Przycisk1.Visibility = Visibility.Hidden;
```

```
Przycisk2.Visibility = Visibility.Visible;
```

We wnętrzu metody Przycisk2_Click dodaj następująco kod:

```
Przycisk2.Visibility = Visibility.Hidden;
```

```
Przycisk1.Visibility = Visibility.Visible;
```

zaś w konstruktorze MainWindow dodaj linijkę:

```
Przycisk2.Visibility = Visibility.Hidden;
```

28. Stwórz nowy projekt. Z pomocą designera upuść dwa przyciski. W oknie Properties naciśnij „piorunek” i w polu Click wpisz Button_Click (to samo dla obu przycisków).

We wnętrzu metody Button_Click wpisz przykładowy kod:

```
Button btn = sender as Button;
```

```
btn.Content += "x";
```