

Programowanie aplikacji okienkowych Windows



- Tworzenie aplikacji
okienkowych w C#
na bazie .NET 2.0

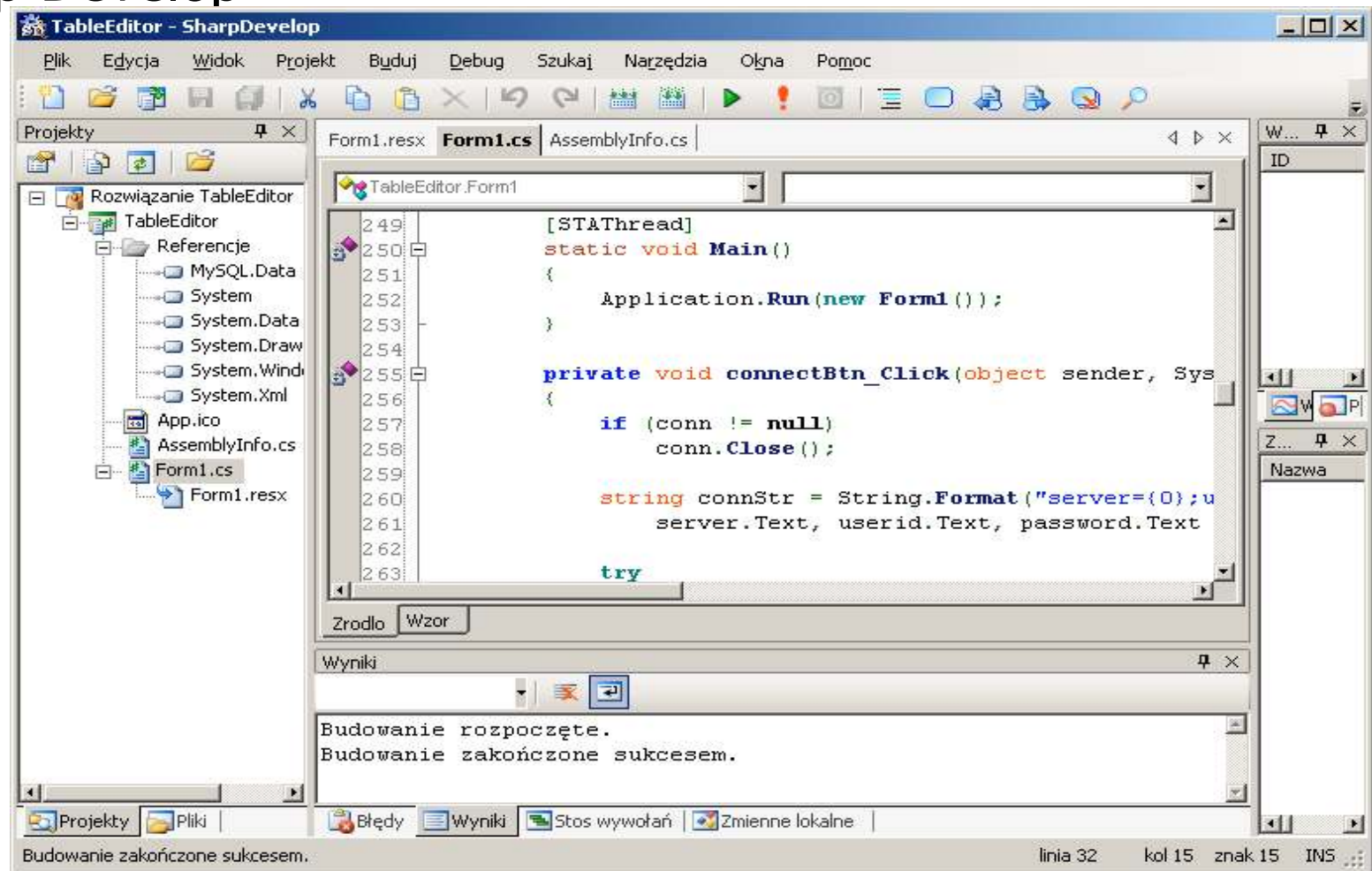
W czym piszemy aplikacje .NET



- Microsoft Visual Studio .NET
- Microsoft [] Express, w edycjach dla języków C++, VB, J++, C#
- Sharp Develop
- Zewnętrzne edytory, kompilacja przez wywołanie csc.exe z linii komend
- Standard plików w tych środowiskach: *.sln (Microsoft Visual Studio Solution File), *.resx (zasoby zakodowane w XML), *.csproj (pliki projektów zakodowane w XML)

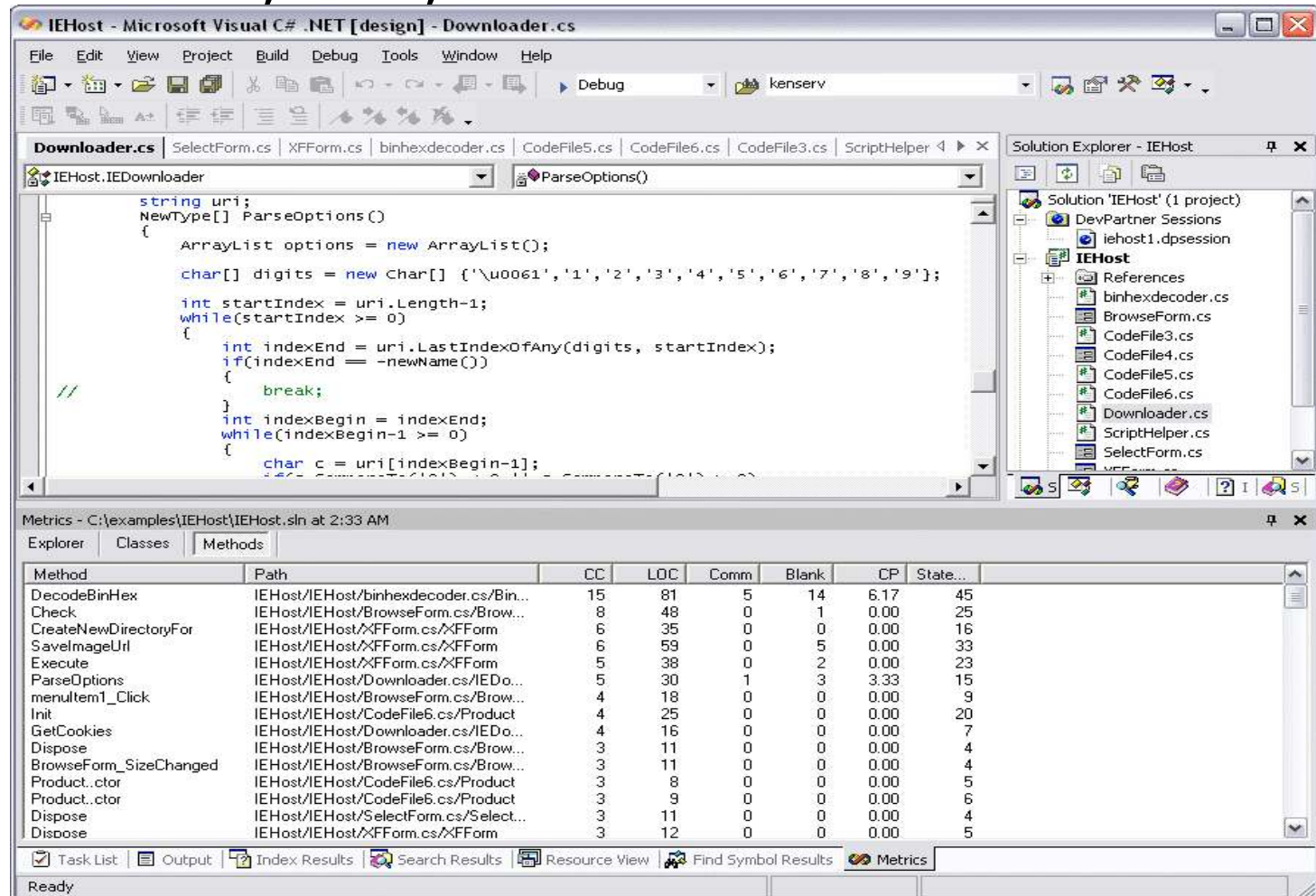
Środowiska (I)

■ Sharp Develop



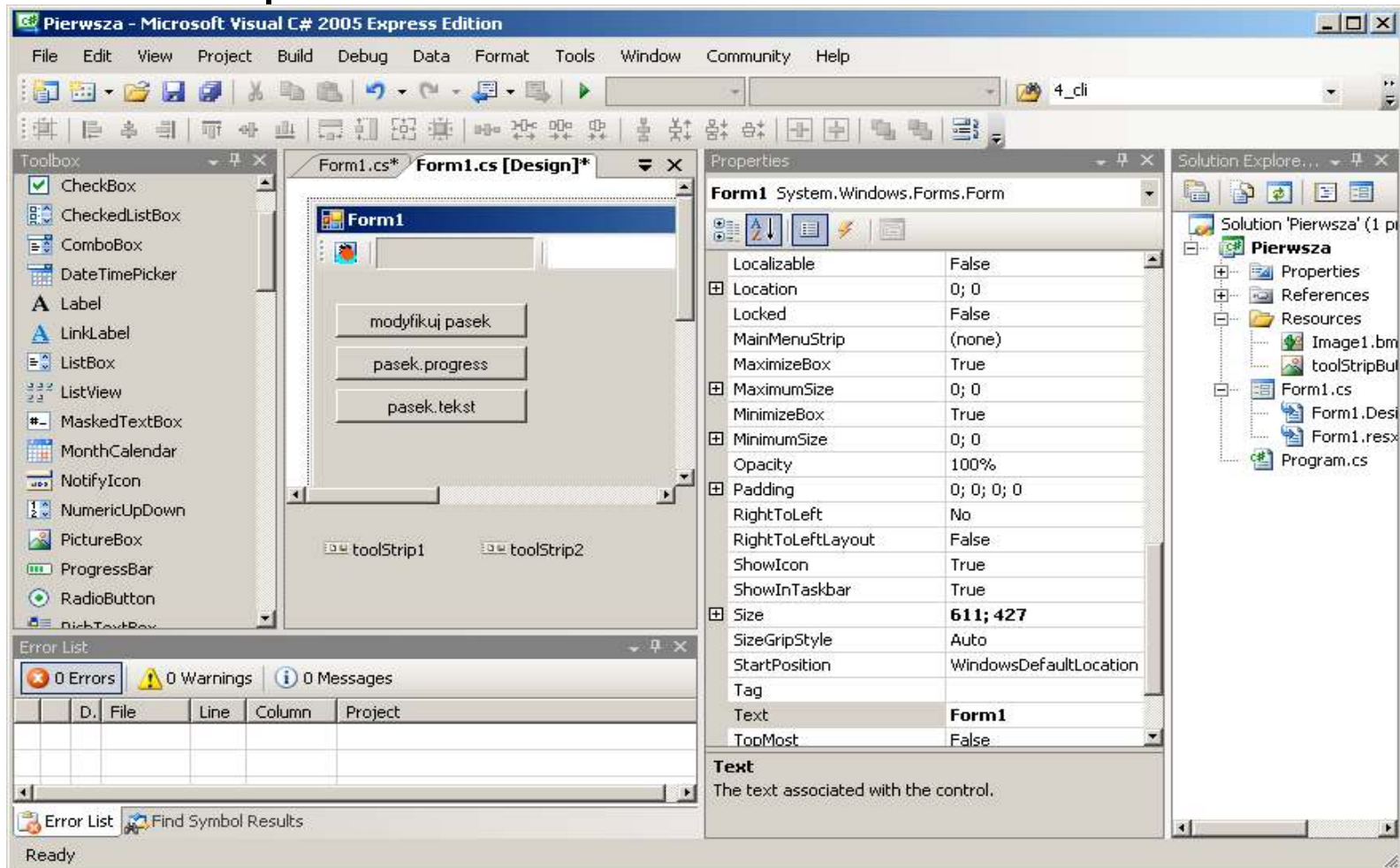
Środowiska (II)

Visual Studio 2001, 2003, 2005



Środowiska (III)

Visual C# Express Edition



Najprostsza aplikacja (konsoli)



```
■ using System;
■ namespace Konsola1 {
■     class MainClass {
■         public static void Main(string[] args) {
■             int[] a = new int [10];
■             a[0] = 120;
■             Console.WriteLine("Tekst!" + a[0]);
■         }
■     }
■ }
```

Prezentacja - aplikacja konsoli

Operowanie na czasie

- `DateTime oldDate = new DateTime(2002,7,15);`
- `DateTime newDate = DateTime.Now;`
- `// Difference in days, hours, and minutes.`
- `TimeSpan ts = newDate - oldDate;`
- `// Difference in days.`
- `int differenceInDays = ts.Days;`
- `System.Console.WriteLine("Difference in days: {0} ",
differenceInDays);`

Uruchomienie procesu lub pliku zasobu edytowalnego



- `System.Diagnostics.Process.Start("C:\\Program Files\\Internet Explorer\\IExplore.exe", "www.microsoft.com");`
- `System.Diagnostics.Process.Start("calc.exe", "");`

Pobieranie listy Assemblies, używanych przez aplikację



```
■ System.Reflection.Assembly exe;  
■ exe =  
  System.Reflection.Assembly.GetExecutingAssembly();  
■ string [] zasoby = exe.GetManifestResourceNames();  
■ string lista = "";  
■ foreach (string linia in zasoby)  
■   lista += linia + "\r\n";
```

Dostęp do środowiska aplikacji



- `Environment.CurrentDirectory = "C:\\katalog";`
- Inne użyteczne składowe `Environment`:
`CommandLine`, `ExitCode`, `UserName`, `MachineName`,
- `Environment.`
`SetEnvironmentVariable ( „nazwa”, „wartosc” );`

Operacje na systemie plików (I)



- Kontrola, czy plik istnieje:
 - `bool istnieje;`
 - `istnieje = System.IO.File.Exists(@"c:\plik.txt");`
 - `System.Console.Write(istnieje);`
- Kontrola, czy folder istnieje
 - `bool exists = System.IO.Directory.Exists(@"c:\my folder");`
- Uwaga: symbol '@' umożliwia podanie łańcucha bez konieczności stosowania escape chars (\).

Operacje na systemie plików (II)

■ Tworzenie katalogu:

- `System.IO.Directory.CreateDirectory(@"c:\Folder");`

■ Tworzenie pliku:

- `System.IO.FileStream file;`

- `file = System.IO.File.Create("c:\\test.txt");`

■ Pisanie do pliku:

- `string tekst = "tresc pliku"; //tekst do pliku.`

- `System.IO.StreamWriter file`

- `= new System.IO.StreamWriter(@"c:\test.txt");`

- `file.WriteLine(tekst);`

- `file.Close();`

Operacje na systemie plików (III)

■ Odczyt całości pliku:

```
■ System.IO.StreamReader plik = new
    System.IO.StreamReader(@"c:\test.txt");
    string tekst = plik.ReadToEnd();
```

■ Odczyt pliku liniami:

```
■ System.IO.StreamReader file = new
    System.IO.StreamReader(@"c:\test.txt");
    while((line = file.ReadLine()) != null) {
        // operacje na linii pliku
    }
    file.Close();
```

Dostęp do rejestru i dziennika zdarzeń



■ Rejestr:

- `Microsoft.Win32.RegistryKey key;`
- `key =
Microsoft.Win32.Registry.CurrentUser.CreateSubKey("Wpis");`
- `key.SetValue("NazwaKlucza", "Tresc");`
- `key.Close();`

■ Dziennik Zdarzeń:

- `System.Diagnostics.EventLog log =
new System.Diagnostics.EventLog() ;`
- `log.Source = "Aplikacja";`
- `log.WriteEntry("Tresc komunikatu.");`

Operacje na grafice w pamięci



- Utworzenie i dostęp do grafiki:
 - `System.Drawing.Bitmap flag = new
System.Drawing.Bitmap(100, 100);`
 - `for (int x = 0; x < flag.Height; ++x)`
 - `for (int y = 0; y < flag.Width; ++y)`
 - `flag.SetPixel(x, y, Color.White);`

Konwersja grafiki

- Gotowe funkcje do zapisywania w zadanym formacie
 - `System.Drawing.Image graf =
System.Drawing.Image.FromFile(@"C:\test.bmp");`
 - `graf.Save(@"C:\test.jpg",
System.Drawing.Imaging.ImageFormat.Jpeg);`
 - `graf.Save(@"C:\test.gif",
System.Drawing.Imaging.ImageFormat.Gif);`
 - `graf.Save(@"C:\test.png",
System.Drawing.Imaging.ImageFormat.Png);`

Rysowanie w oknach (I)

- Większość metod rysujących znajduje się w namespace System.Drawing
- Aby rysować należy przeciążyć metodę OnPaint:
`protected override void OnPaint(PaintEventArgs e)`
- Tworzenie i destrukcja kontekstu grafiki
`System.Drawing.Graphics g = this.CreateGraphics();`
- Definiowanie pędzli i piór, rysowanie prostych figur
- **Prezentacja #1**

Rysowanie w oknach (I)



- Rysowanie tekstu graficznego, definiowanie fontów
 - Inicjowanie dowolnych kolorów - także z parametrem alpha
 - Zdarzenie Load dla okna - modyfikacje przy inicjalizacji okna
-
- **Prezentacja #2**

Dodatkowe efekty graficzne



- Regiony
- Zabarwienie tła formatek

■ **Prezentacja #3**

Okna Dialogowe (I)



- Otwieranie własnych okien
 - Komunikaty w okienkach
 - Użycie okna do zapisu o odczytu plików
 - Automatyczne otwieranie dokumentów
-
- **Prezentacja #4**

Okna Dialogowe (II)



- Dynamiczna modyfikacja treści okna
- Dynamiczne tworzenie obiektów formatek
- dodawanie ich do okien
- Przypomnienie: użycie operatora „+=„
przeciążonego w celu dodawania obsługi
zdarzeń do komponentu
- **Prezentacja #5**

Użytkowanie Formatek

Common Forms

	Pointer
	Button
	CheckBox
	CheckedListBox
	ComboBox
	DateTimePicker
	Label
	LinkLabel
	ListBox
	ListView
	MaskedTextBox

	MonthCalendar
	NotifyIcon
	NumericUpDown
	PictureBox
	ProgressBar
	RadioButton
	RichTextBox
	TextBox
	ToolTip
	TreeView
	WebBrowser

Common Forms (I)



- System.Windows.Forms.Button: zdarzenie kliknięcia
- System.Windows.Forms.CheckBox: odczytanie i zmiana stanu
- System.Windows.Forms.CheckedListBox: modyfikacja składników, zdarzenia i pobieranie wartości zaznaczonych
- System.Windows.Forms.ComboBox: modyfikacja składników, zdarzenia i pobieranie wartości zaznaczonej

Prezentacja #6

Common Forms (II)



- `System.Windows.Forms.DateTimePicker`
- `System.Windows.Forms.ListBox`: modyfikacja składników, zdarzenia i pobieranie wartości zaznaczonych
- `System.Windows.Forms.MaskedTextBox`: pobieranie maskowanego tekstu
- `System.Windows.Forms.ListView`: definiowanie i modyfikacje listy wielokolumnowej (grid)
 - **Prezentacja #7**

Common Forms (III)



- `System.Windows.Forms.MonthCalendar` - formarka analogiczna w obsłudze do `DateTimePicker`
- `System.Windows.Forms.NotifyIcon`:
 - Definiowanie ikony w ikonach statusu (obok zegara)
 - Generowanie `BalloonTip` (WindowsXP, .NET 2.0) w odpowiedzi na zdarzenie
- `System.Windows.Forms.NumericUpDown`: definiowanie właściwości, pobieranie ustawienia
- **Prezentacja #8**

Common Forms (IV)



- `System.Windows.Forms.PictureBox`: kontener dla grafik, także korygowanych dynamicznie
- `System.Windows.Forms.ProgressBar`: korygowanie postępu, odczyt
- `System.Windows.Forms.RadioButton`: tworzenie grupy, użycie kontenera `System.Windows.Forms RadioGroup`
- **Prezentacja #9**

Common Forms (V)



- `System.Windows.Forms.RichTextBox`
formatka z funkcjonalnością edytora tekstu MS
(obróbka meta-dokumentów)
- `System.Windows.Forms.ToolTip`:
definiowanie pasków opisujących
- `System.Windows.Forms.WebBrowser`:
nawigacja na URL, wykrywanie zdarzenia
nawigacji
- **Prezentacja #10**

Common Forms (VI)



- `System.Windows.Forms.TreeList`:
pobieranie elementów, dodawanie elementów
- `System.Windows.Forms`
- Przyciski (`System.Windows.Forms.Button`):
 - Programowe wywoływanie zdarzeń
 - Przyciski o zmienionych kształtach - tworzenie własnych klas formatek
- **Prezentacja #11**

Menu



- `System.Windows.Forms.MenuStrip`:
tworzenie menu aplikacji, obsługa zdarzeń menu
- `System.Windows.Forms.ContextMenuStrip`:
tworzenie i przypisywanie do okna menu kontekstowego, obsługa zdarzeń menu kontekstowego
- `System.Windows.Forms.StatusStrip`:
Osadzanie tekstu, progress bar i menu w pasku statusu. Obsługa akcji menu statusu
- **Prezentacja #12**

Timery



- System.Windows.Forms.Timer
 - Definiowanie timera
 - Obsługa zdarzenia timera
 - Programowe zatrzymywanie i wznowianie timera

- **Prezentacja #13**

Kontenery formatek (I)



- `System.Windows.Forms.SplitContainer`: kontener z przegrodą poziomą lub pionową, sposób inicjalizacji kontenera i dokowania formatek, definiowanie zdarzeń w obrębie kontenera, rysowanie w kontenerze, dostęp do pól kontenera
- **Prezentacja #14**

Kontenery formatek (II)



- `System.Windows.Forms.TabControl`: kontener z przegrodą poziomą lub pionową, sposób inicjalizacji kontenera i dokowania formatek, definiowanie zdarzeń w obrębie kontenera

- **Prezentacja #15**

Kontenery formatek (III)



- Pozostałe przydatne kontenery:
 - `System.Windows.Forms.FlowLayoutPanel`
 - `System.Windows.Forms.GroupBox`
 - `System.Windows.Forms.TableLayoutPanel`
 - `System.Windows.Forms.Panel`
-
- **Prezentacja #16**

Paski narzędzi



- `System.Windows.Forms.ToolStrip`:
tworzenie paska, dokowanie paska, możliwe
składowe paska, dostęp do składowych paska,
dynamiczna rozbudowa paska,

- **Prezentacja #17**

Formatki pomocnicze



- `System.Windows.Forms.ErrorProvider`:
definiowanie sygnatur błędów w obrębie okna
- `System.Windows.Forms.LogEvent`:
zapisywanie komunikatów w dzienniku zdarzeń
- `System.Windows.Forms.FileSystemWatcher`:
śledzenie plików i katalogów, obsługa zdarzeń
`FileSystemWatcher`
- `System.Diagnostics.Process`:
Śledzenie procesów
- **Prezentacja #18**