Wprowadzenie

Celem tych zajęć jest utworzenie aplikacji, która wykreśla wykresy. Program umożliwiać będzie dodawanie nowych wykresów. Wykresami będą różnego rodzaju przebiegi (przebieg prostokątny, sinus, itd.). Każda taka funkcja **f(x, z)** przyjmuje parametr określający kąt (**x**) oraz parametr (**z**) określający dodatkowe opcje, np. wypełnienie w przypadku funkcji prostokątnej. Każdy przebieg będzie obiektem klasy **mojPrzebieg**. W klasie tej zapisane będą pola określające amplitudę, okres, kąt, dodatkowy parametr oraz obiekt delegata wskazujący na metodę jaka zostanie uruchomiona by wyliczyć wartość. Wykresy będa rysowane za pomocą biblioteki Zedgraph, która jest osobną biblioteką o otwartym kodzie, napisaną przez osoby trzecie.

Zadanie 1

Sprawdź, czy na stanowisku zainstalowano bibliotekę ZedGraph. Jeśli jest ona zainstalowana poprawnie, to powinna być widoczna nowa ikona na pasku kontrolek w Visual Studio. By to sprawdzić, należy utworzyć pustą okienkową aplikację o nazwie **windowsFormWykres** i sprawdzić czy jest kontrolka ZedGraph.



Jeśli ikona jest niewidoczna, to do Visual Studio nie została dodana kontrolka (biblioteka) do rysowania wykresów. Bibliotekę można pobrać pod następującym adresem: http://sourceforge.net/projects/zedgraph/files/zedgraph%20dll%20only/5.1.5/zedgraph_dll http://sourceforge.net/projects/zedgraph/files/zedgraph%20dll%20only/5.1.5/zedgraph_dll http://sourceforge.net/projects/zedgraph/files/zedgraph%20dll%20only/5.1.5/zedgraph_dll

- 1. umieszczeniu pliku dll w katalogu, np C:\Program files\Microsoft Visual Studio...,
- 2. dodaniu kontrolki do paska narzędzi. W tym celu należy otworzyć pasek toolbox z kontrolkami, a następnie rozwinąć na nim menu wciskając prawy przycisk myszy.

1	Start Page - N	Aicros	oft Visual Studio	
Fi	le Edit Vie	ew 1	ools Window Help	
16] • 🛅 • 		9 × 12 12 19 - 19	• 🗏 • 🖳 🕨 🦳
X	Toolbox			
Toolbox	🗆 General			
	There are r Drag an ite	no usa	ble controls in this group. This text to add it to the Paste	C 2008 MSDN: Visual C# Heat Solving Combinatory Thu, 08 Apr 2010 22:50 Optional Parameters Wed, 31 Mar 2010 01:0
		 Image: A start of the start of	List View Show All	
			Choose Items	
			Sort Items Alphabetically	
			Reset Toolbox	
			Add Tab	CSharp4Expressions -
			Delete Tab	Mon, 29 Mar 2010 23:0 control flow in express
			Rename Tab	Videos of the Window
			Move Up	Fri, 19 Mar 2010 21:55:
			Move Down	Just Announced – OD Fri, 12 Mar 2010 23:20:1

- 3. Odczekaniu na reakcję komputera i wzkazaniu pliku dll
- 4. Zaznaczeniu biblioteki ZedGraph.

	COM Components WPP Co	mponents Activities			
Name	Namespace	Assembly Name	Directory		
AccessDataSource	System.Web.UI.WebControls	System.Web (2.0.0.0)	Global Asse		
ActionsPane	Microsoft.Office.Tools	Microsoft.Office.Tool	Global Asse		
ActionsPane	Microsoft.Office.Tools	Microsoft.Office.Tool	Global Asse		
Activity	System.Workflow.Compone	System.Workflow.Co	Global Asse		
ADODC	Microsoft.VisualBasic.Compa	Microsoft.VisualBasic	Global Asse		
ADODCArray	Microsoft.VisualBasic.Compa	Microsoft.VisualBasic	Global Asse		
AdomdCommand	Microsoft.AnalysisServices.A	Microsoft.AnalysisSer Global Asse			
AdomdCommand	Microsoft.AnalysisServices.A	Microsoft.AnalysisSer Microsoft.AnalysisSer	Global Asse Global Asse Global Asse		
AdomdConnection	Microsoft.AnalysisServices.A				
AdomdConnection	Microsoft.AnalysisServices.A	Microsoft.AnalysisSer			
<u>F</u> ilter:			Clear		
AccessDataSource					
Language: I	Language: Invariant Language (Invariant Country)				
Version: 2	2.0.0.0				
			Concel Devet		

5. Dodaniu using ZedGraph.

Zadanie 2: aplikacja okienkowa z wykresem

Utwórz pustą okienkową aplikację. Na początku nasz program będzie rysował wykres w przedziale od 0 do 1000.

Dodaj do niej następujące elementy (kontrolki):

- Wykres zedGraphControl i nazwij go zedGraphControlWykres,
- kontrolkę NumericUpdown i nazwij ją numericUpdownAmplituda,
- kontrolkę NumericUpDown i nazwij ją numericUpDownOkres,
- przycisk i nazwij go buttonNowyPrzebieg,
- przycisk i nazwij go buttonCzysc,
- etykiety opisujące od czego służą kontrolki numericUpDown.

Nazywanie w ten sposób obiektów (instancji) kontrolek jest dobrym nawykiem. Jeśli chcemy zmodyfikować właściwości przycisku, to wiemy wtedy, że nazwa obiektu (który chcemy zmodyfikować) rozpocznie się od button. Następnie można skorzystać z podpowiedzi edytora. Zatem staranne nazywanie obiektów przyspiesza pracę.

Proszę zwrócić uwagę na fakt, że kontrolki mają często 2 pola do edycji: text (to co jest na niej napisane) i name. To właśnie name określa nazwę obiektu (tak naprawdę to instancja obiektu może mieć inną nazwę niż właściwość name wewnątrz niego, jeśli korzystamy z kreatora to obie nazwy są identyczne).



Dodaj obsługę zdarzenia **buttonNowyPrzebieg_Click**. Spowoduje ono wykreślenie na wykresie nowego przebiegu o wybranej wcześniej amplitudzie i okresie. Do określenia tych parametrów służą pola kontrolki NumericUpDown. W tym celu skorzystaj z kreatora.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private void buttonNowyPrzebieg_Click(object sender, EventArgs e)
    {
        }
    }
}
```

W metodzie tej należy:

• Odczytać wartość amplitudy i okresu z kontrolek numericUpDown. Fragment kodu:

double aumlituda = (double) (numericUpDownAmplituda.Value);

• Utworzyć nową listę z punktami wykresu:

ZedGraph.PointPairList przebieg = new PointPairList();

• W pętli (np for) wypełnij przebieg. Niech naszym przebiegiem jest np. funkcja sinus. Jest ona dostępna w bibliotece **Math**,

```
for (double x ...
{
    przebieg.Add(x, aumlituda * Math.Sin( ...
}
```

• Dodaj nowy przebieg. Najpierw trzeba go utworzyć. W tym celu tworzymy obiekt Lineltem. Obiekt ten będzie zawierał listę wartości x,y oraz informacje o nazwie funkcji, kolorze i grubości linii oraz znaczniku jaki będzie rysowany w każdym punkcie przebiegu.

```
LineItem funkcja = new LineItem(string.Format("Funkcja {0}",
nrPrzebiegu), przebieg, Color.Red, SymbolType.Plus, 1);
```

• Dodaj nowy przebieg do wykresu.

zedGraphControlWykres.GraphPane.CurveList.Add(funkcja);

• Wymuś przeskalowanie i przerysowanie wykresu.

```
zedGraphControlWykres.GraphPane.AxisChange();
Refresh();
```

Zadanie 3

Napisz obsługę przycisku czyść. Skorzystaj z odpowiedniej metody obiektu zedGraphControlWykres.GraphPane.CurveList.

Na podstawie tutoriala opisz osie. W kontrolkach numericUpDown ustaw pewne początkowe wartości dla amplitudy i okresu.

Zadanie 4

Dodaj kontrolki, które pozwalają na określenie początkowej i końcowej wartości x dla f(x).

Zadanie 5

Zrób wykres z przedziałami błędów. Niech dolną wartością błędu jest 0.9*wartość funkcji, a górną 1.1*wartość funkcji. Przykład:

```
// Call this method from the Form Load method, passing your ZedGraphControl
public void CreateChart( ZedGraphControl zgc )
{
 GraphPane myPane = zgc.GraphPane;
  // Set the titles and axis labels
 myPane.Title.Text = "Error Bar Demo Chart";
  myPane.XAxis.Title.Text = "Label";
  myPane.YAxis.Title.Text = "My Y Axis";
  // Make up some data points based on the Sine function
  PointPairList list = new PointPairList();
  for (double x = 0.0; x < 10; x + = 0.01)
  {
    double y = Math.Sin( x * Math.PI );
    list.Add( x, 0.9*y, 1.1*y );
  }
  // Generate a red bar with "Curve 1" in the legend
  ErrorBarItem myCurve = myPane.AddErrorBar( "Curve 1", list, Color.Red );
  // Make the X axis the base for this curve (this is the default)
  myPane.BarSettings.Base = BarBase.X;
  myCurve.Bar.PenWidth = 1f;
  // Use the HDash symbol so that the error bars look like I-beams
  myCurve.Bar.Symbol.Type = SymbolType.HDash;
  myCurve.Bar.Symbol.Border.Width = .1f;
  myCurve.Bar.Symbol.IsVisible = true;
  myCurve.Bar.Symbol.Size = 4;
  // Fill the axis background with a color gradient
  myPane.Chart.Fill = new Fill( Color.White,
     Color.LightGoldenrodYellow, 45.0F );
  // Calculate the Axis Scale Ranges
  zgc.AxisChange();
}
```