

Metodyka programowania

Komponent Chart

Komponent SplitContainer

SplitContainer dzieli okno na dwie części;

- Zależnie od potrzeb umożliwia zmianę szerokości/wysokości części albo utrzymuje rozmiar jednej z części stały (przy zmianie wielkości okna)



Komponent SplitContainer

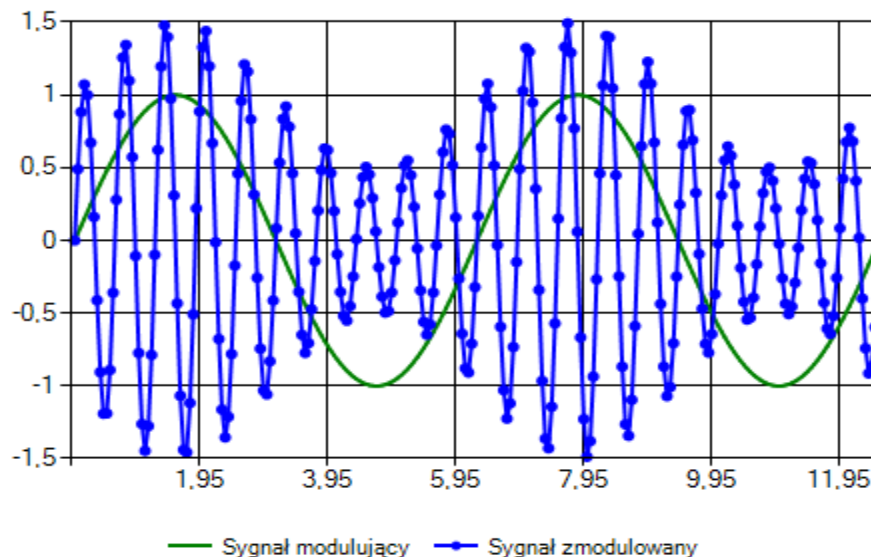
Właściwości SplitContainer

- Dock – automatyczne dostosowanie wielkości i położenia, domyślnie Fill
- Orientation – podział okna w poziomie lub pionie, domyślnie Vertical
- FixedPanel – panel zablokowany (bez możliwości zmiany rozmiaru) – None/Panel1/Panel2
- SplitterDistance – odległość krawędzi podziału od krawędzi komponentu (lewej lub górnej, zależnie od Orientation)

Komponent Chart

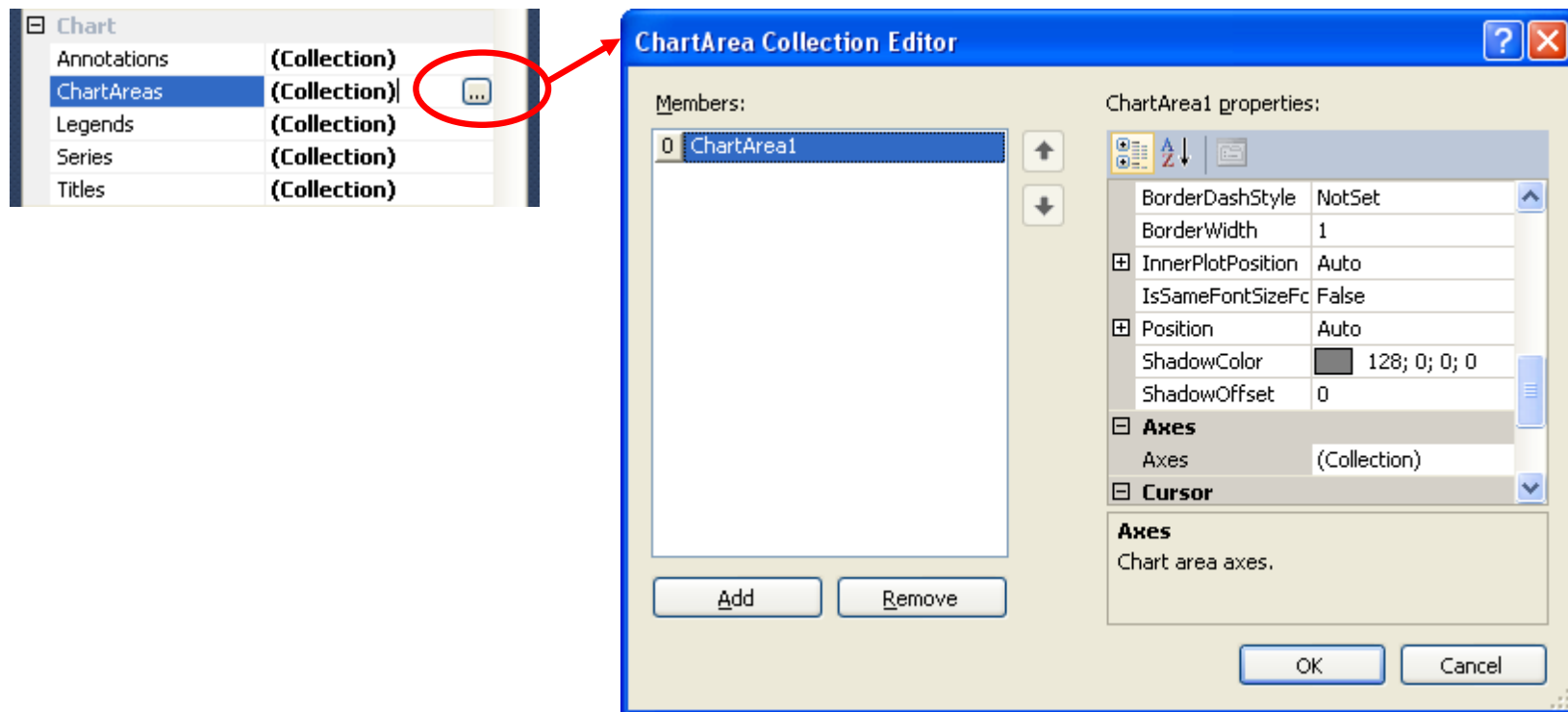
Komponent *Chart* pozwala tworzyć efektowne i zróżnicowane wykresy. Na wykres składa się:

- Jeden lub więcej *ChartArea* (obszar wykresu)
- Jeden lub więcej *Series* (zbiór danych) na każdym z obszarów
- Jedna lub więcej *Legend* (legenda)
- Jeden lub więcej *Title* (tytuł)



Komponent Chart

Poszczególne elementy wykresu występują jako kolekcje, z których każda ma odrębny edytor właściwości.



Komponent Chart

Istnieje też możliwość programowego modyfikowania wszystkich elementów wykresu, ze wszystkich kolekcji:

```
Series s;  
s = chart1.Series[0];  
s.ChartType = SeriesChartType.Line;  
ChartArea a;  
a = chart1.ChartAreas[0];  
a.AxisX.Minimum = 0.0;
```

Uwaga!

- Elementy kolekcji są indeksowane od 0, a nie od 1.
- Należy dodać klauzulę using:

```
using System.Windows.Forms.DataVisualization.Charting;
```

(pozwala to stosować nazwy klas związanych z wykresem, np. Series, ChartArea itp.)

Komponent Chart

Właściwości

- BackColor – kolor tła
inne elementy wykresu (np. poszczególne obszary) mogą mieć własne tło lub mogą być przezroczyste;
- BackSecondaryColor, BackGradientStyle – tło w postaci gradientu (płynnego przejścia między kolorami)
- BackImage, BackImageWrapMode – obrazek jako tło
- Palette – paleta kolorów do prezentacji serii danych
- PaletteCustomColors – własna paleta kolorów (będzie użyta, jeżeli ustawić Palette = None)

ChartArea

Obszar wykresu, mogący prezentować kilka serii danych

Właściwości

- BackColor, BackImage itp. – tło
- Axes – kolekcja osi wykresu
(w praktyce lepiej odwoływać się bezpośrednio do poszczególnych osi)
- AxisX, AxisX2 – główna i pomocnicza oś X
- AxisY, AxisY2 – główna i pomocnicza oś Y

Axis

Oś liczbowa wykresu.

- Każda seria danych przypisana do obszaru wykresu, może być przypisana do głównej lub pomocniczej osi X i osi Y
- Każda oś ma kilkadziesiąt (!) właściwości określających jej wygląd (kolor, podział na zakresy, etykiety, tytuł, ...)

Wybrane właściwości

- Minimum, Maximum – domyślnie "*Auto*", ale można podać
- IsLogarithmic – skala logarytmiczna

Series

Seria (zbiór) danych prezentowanych na wykresie.

- Każda seria jest przypisana do jednego z obszarów danych, a także do głównej lub pomocniczej osi X i osi Y tego obszaru
- Każda seria może mieć inny typ (kolumny, linie, punkty, ...), jednak nie wszystkie kombinacje są możliwe

Wybrane właściwości

- ChartType – rodzaj wykresu (np. Line, Point, Column, ...)
- ChartArea – obszar, na którym seria jest wyświetlana
- XAxisType, YAxisType – przypisanie do osi *Primary* / *Secondary*
- LegendText – opis wyświetlany w legendzie
- BorderWidth – dla typu "Line" – grubość linii
- MarkerStyle – rodzaj znacznika (np. Square, Circle, ...)

Legend

Legenda wykresu

- Może być wspólna dla wszystkich obszarów albo każdy obszar może mieć oddzielną; Również serie danych można przypisywać do wybranej legendy

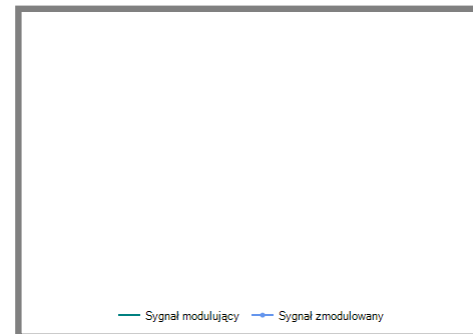
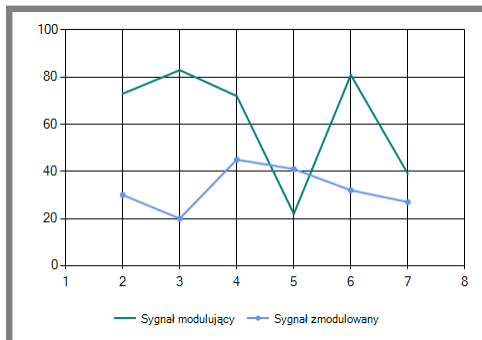
Wybrane właściwości

- `DockedToChartArea` – przypisanie do obszaru;
- `IsDockedInsideChartArea` – czy legenda ma być wewnątrz obszaru, czy obojętnie
- `Docking` – po której stronie obszaru (albo całego komponentu) ma być legenda (Top, Right, Left, Bottom)
- `Alignment` – sposób wyrównania (Near, Center, Far)

Komponent Chart

Po umieszczeniu komponentu na formularzu można go dostosować do potrzeb aplikacji – dodać i dostosować obszary, serie danych, osie, legendę itd. W tym czasie na wykresie są przedstawione przykładowe dane, jednak po uruchomieniu programu widać tylko legendę.

Dzieje się tak, ponieważ serie nie zawierają żadnych danych;
dopiero po dodaniu danych pojawią się one na wykresie;



Komponent Chart

Można rozważyć dwie strategie prezentacji danych:

- (1) Na etapie projektowania GUI są utworzone i odpowiednio dostosowane wszystkie potrzebne serie;
aplikacja wyznacza wartości, które mają być prezentowane na wykresie, po czym dodaje je do odpowiednich serii
- (2) Na etapie projektowania GUI są dostosowane właściwości obszaru wykresu i legendy, ale nie serii danych;
aplikacja dynamicznie dodaje serie danych i dane
(należy wówczas programowo zmienić właściwości serii)

Komponent Chart

Strategia 1

- Dla skrócenia zapisu warto odczytać referencje do serii danych:

```
Series s1, s2;  
s1 = chart1.Series[0];  
s2 = chart1.Series[1];
```

- Usunięcie uprzednio wyświetlanych danych – należy odwołać się do kolekcji punktów wykresu i użyć metody *Clear*

```
s1.Points.Clear();
```

- Umieszczenie na wykresie nowych danych – najwygodniej użyć pętli for oraz metody *AddXY*, np.:

```
for(x=0.0; x<=10.0; x+=0.25)  
    s1.Points.AddXY(x, Math.Sqrt(x));
```

Komponent Chart

Strategia 2

- Należy zacząć od utworzenia serii danych, dostosowania jej właściwości i dodania do wykresu:

```
Series s;  
s = new Series();  
s.LegendText = "funkcja kwadratowa";  
s.ChartType = SeriesChartType.Line;  
s.MarkerStyle = MarkerStyle.Square;  
chart1.Series.Add (s);
```

Umieszczenie na wykresie danych, np.:

```
for (x=0.0; x<=10.0; x+=0.25)  
    s.Points.AddXY (x, Math.Sqrt (x));
```

- Aby usunąć wszystkie serie danych z wykresu:

```
chart1.Series.Clear();
```

Komponent Chart

Programowa zmiana właściwości:

- Właściwości konkretnych, pojedynczych elementów, które nie są tworzone dynamicznie – przez ich indeks:

```
Series s;  
s = chart1.Series[0];  
s.ChartType = SeriesChartType.Line;  
s.MarkerStyle = MarkerStyle.None;
```

```
ChartArea a = chart1.ChartAreas[0];  
a.AxisX.Minimum = 0;  
a.AxisX.Maximum = 5;
```

Komponent Chart

Programowa zmiana właściwości:

- Właściwości wszystkich elementów określonej kategorii utworzonych dynamicznie – przez pętlę foreach

```
foreach (Series s in chart1.Series)
{
    s.ChartType = SeriesChartType.Line;
    s.MarkerStyle = MarkerStyle.Circle;
}
```

Komponent Chart - zadania

Program ilustrujący wykorzystanie komponentu *Chart*:

- Elementy obowiązkowe
 - wybranie dowolnej funkcji (np. $a \cdot \sin(b \cdot x)$, $a \cdot x^b$, $a \cdot x^2 - b$ itp.)
 - odczytanie parametrów (a i b) funkcji z pól tekstowych
 - wyświetlenie wartości funkcji na wykresie, przy stałym zakresie wartości i kroku (np. 0 .. 5, co 0,1)
- Elementy opcjonalne
 - wybór zakresu wartości i kroku przez użytkownika
 - dynamiczne dodawanie serii danych
 - zmiana wybranych parametrów wykresu (np. wyświetlanie legendy) oraz wszystkich serii danych (np. zmiana typu wykresu, włączenie lub wyłączenie wyświetlania markerów
 - przy użyciu pętli foreach)