

Praktikum 7./8. Juni 2011

Jörn Loviscach

Versionsstand: 6. Juni 2011, 21:40



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

In diesem Praktikum soll mit einer Treemap visualisiert werden, wie der Speicherplatz in einem Verzeichnis verwendet wird.



Ergänzen Sie das Hauptfenster um einen Button und um eine Canvas, auf der nachher das Ergebnis erscheint. Der Button benutzt einen `System.Windows.Forms.FolderBrowserDialog`, um ein Verzeichnis vom Benutzer abzufragen. (Im Projekt ist für diese Klasse ein Verweis auf `System.Windows.Forms` hinzuzufügen.) Daraufhin soll die Datenstruktur aufgebaut werden und als Sammlung von Zeichenobjekten in die Canvas gegeben werden.

Der Kern des Programms ist das Composite-Pattern. Dateien und Verzeichnisse werden als Leaf bzw. Composite behandelt. Sie sind konkrete Ausprägungen einer abstrakten Klasse `TreemapComponent`. Diese Klasse verfügt über ein Attribut vom Typ `long`, das das Datenvolumen in Bytes angibt. Sie verfügt außerdem über folgende abstrakte Methode:

```
public abstract void AddToCanvas
    (System.Windows.Controls.Canvas canvas,
     System.Windows.Rect rect, bool vertical);
```

Diese Methode soll das aktuelle Element und seine Unterelemente in einem Ausschnitt `rect` der `canvas` platzieren. Die boolesche Variable `vertical` gibt an, ob die Elemente auf der aktuellen Ebene von links nach rechts oder aber von oben

nach unten zu stapeln sind. Die Anordnung eine Ebene darunter ist dann in der jeweils anderen Richtung.

Der Konstruktor für die Composite-Klasse – sie stellt Verzeichnisse dar – nimmt ein `System.IO.DirectoryInfo` und eine Ganzzahl `levels` entgegen. Er geht die Dateien und die Verzeichnisse des `System.IO.DirectoryInfo` durch, ruft die entsprechenden Konstruktoren auf und merkt sich die so entstehenden Objekte und ihre Gesamtzahl an Bytes. Der Wert von `levels` dient als Abbruchbedingung für die so entstehende Rekursion.

Die Methode `AddToCanvas` der Composite-Klasse geht die Unterelemente durch, berechnet ein passendes `System.Windows.Rect` für jedes Unterelement und ruft dann die Methode `AddToCanvas` des Unterelements auf. Anschließend erzeugt sie mit einem `System.Windows.Shapes.Rectangle` einen Rahmen.

Die Methode `AddToCanvas` der Leaf-Klasse erzeugt ein gefülltes `System.Windows.Shapes.Rectangle` in der angegebenen Größe. Des- sen Farbe vom Typ `System.Windows.Media.Color` wird mit Hilfe von `Random` zufällig gewählt. Setzen Sie dazu die Property `Fill` des Rechtecks auf eine neu konstruierte `System.Windows.Media.SolidColorBrush`.

Zusatzaufgabe: Zeigen Sie – wie im Screenshot auf der ersten Seite – in einem `Label` an, welcher Dateiname zu dem Rechteck gehört, über dem der Mauszeiger gerade schwebt. Benutzen Sie dazu das Ereignis `MouseMove` der `Canvas` und erweitern Sie die Klassen um passende Attribute, Properties und Methoden.