Datentypen: Array, Enum, Struct

Freitag, 13. Januar 2017

14:47

**Enumeration** (Aufzählung)

Beispiel:

public enum Karte{

Karo = 9,

Herz = 10,

Pik = 11,

Kreuz = 12

}

Eigenschaften

* gruppiert Konstanten die thematisch verwandt sind
* Konstanten sind (wenn nicht anders angegeben) vom Typ int
* ohne angabe von Werten werden automatisch Werte vergeben (0,…,n)
* ein Objekt von Typ enum hat genau einen Wert

Karte card1;

card1 = Karte.Herz;

if (card1 == Karte.Herz) Console.WriteLine("Sie haben Herz.");

Einen enum durchlaufen

foreach (Karte karte in Enum.GetValues(typeof(Karte)))

Console.WriteLine(karte);

**Struct**

* ähnlich wie class, aber Wertetyp (auf dem Stack abgelegt)
* Unterschiede zu class:
  + kann nicht abgeleitet werden (erbt nur von ValueType)
  + besitzt immer parameterlosen Konstruktor
  + Speicherallokation mit new nicht nötig, wenn auf initialisierung der Felder des struct verzichtet werden kann

Beispiele: alle primitiven Datentypen

**Array**

Arrays erben direkt von System.Object, sind also Referenztypen!

Was tun die folgenden Zeilen ?

|  |
| --- |
| int[] elements = new int[3];  elements = {23, 9, 7};      int[,] points = new int[4,3];  int[,] points = {{1,2,3},{4,5,6}};      int[][] myArray = new int[3][];  myArray[0] = new int[3];  myArray[1] = new int[4];  myArray[2] = new int[2]; |