Grundlagen C#

Montag, 12. Dezember 2016

17:12

// The topmost class 
namespace System 
{ public class Object 
the 
.NET universe: 
System. Object 
public 
public 
public 
public 
public 
public 
Object(); Konstruktor, wird auch von Konstruktoren abgel. Klassen gerufen 
virtual Boolean Equals(Object obj); // Aktuelles Objekt und Objekt obj gleich '? 
static Boolean Equals(Object 01, Object 02); // Objekte 01 und 02 gleich '? 
virtual int GetHashCode(); // Eindeutiger Wert, override seitens abgel. Klasse 
Type GetType()•, // 
liefert exakten Typ des Objektes zur Laufzeit 
virtual String ToString(); // gibt String zurück, der das aktuelle Objekt darstellt 
protected virtual void Finalize(); //Gibt einem Objekt Gelegenheit zum Versuch, Ressourcen 
freizugeben, bevor Objekt vom Garbage Collector freigegeben wird 
protected Object MemberwiseClone(); // Erstellt eine flache Copy des aktuellen Objektes 
public static bool ReferenceEquals(object objA, object objB); //Stellt fest, ob die ange- 
gebenen Objekte identischen Speicherplatz belegen, d.h. nicht nur gleiche Werte besitzen. 
Aus ReferenceEquals(ol,02) 
true folgt immer Equals(ol, 02) 
true bzw. 
01. Equals(02) 
true. Umgekehrt gilt diese Aussage jedoch nicht ! 

**intrinsische Datentypen**

CLS 
System Type 
Shorthand 
Compliant? 
bool 
System. Boolean 
sbyte 
System. SByte 
byte 
short 
us hort 
int 
uint 
long 
System. Int64 
ulong 
Yes 
No 
Yes 
Yes 
No 
Yes 
No 
Yes 
No 
System. Byte 
System. 
Int16 
System 
. Ulnt16 
System. 
Int32 
System. 
Ulnt32 
System 
. Ulnt64 
Range 
true or false 
-128 to 127 
O to 255 
-32,768 to 32,767 
O to 65,535 
-2, to 
2,147,483,647 
O to ,295 
-9,223,372,036,854,775,808 
to 
9,223,372,036,854,775,807 
O to 18,446,744,073,709,551, 
615 
Meaning in Life 
Represents truth or 
falsity 
Signed 8-bit number 
Unsigned 8-bit 
number 
Signed 16-bit 
number 
Unsigned 16-bit 
number 
Signed 32-bit 
number 
Unsigned 32-bit 
number 
Signed 64-bit 
number 
Unsigned 64-bit 
number char 
float 
System. Single 
double 
System. Double 
decimal 
string 
System. String 
Object 
System. Object 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
System. Char 
System. Decimal 
U+OOOO to U+ffff 
-3.4 1038 to +3.4 1038 
±5.0 10-324 to ± 1.7 10308 
(-7.9 x 1028 to 7.9 x 1028) / (100 
to 28) 
Limited by system memory 
Can store any data type in 
an object variable 
Single 16-bit Unicode 
character 
32-bit floating-point 
number 
64-bit floating-point 
number 
128-bit signed 
number 
Represents a set of 
Unicode characters 
The base class of all 
types in the .NET 
universe 

Object 
Type 
String 
Array 
Exception 
Delegate 
MulticastDelegate 
ValueType 
Any type 
that derives 
from 
ValueType 
is a 
structure 
or enum 
not a class. 
Enumerations and Structures 
Boolean 
Uint16 
Byte 
Uint32 
Char 
Uint64 
Decimal 
Void 
Double 
DateTime 
Int16 
Guid 
Int32 
TimeSpan 
Int64 
Single 
SByte 

**Boxing / Unboxing**

boxing: Konvertierung (Einschachteln) eines Wertetyps in ein object (implizit)

unboxing: Extraktion eines Wertetyps aus einem object (explizit)

Beispiel boxing

boxing:

Console.WriteLine("12.Equals(23) = {0}", 12.Equals(23));

Console.WriteLine("12.ToString() = {0}", 12.ToString());

Console.WriteLine("12.GetType() = {0}", 12.GetType());

Ausgabe:

12.Equals(23) =

12.ToString() =

12.GetType() =

unboxing:

object o = 123;

int i = (int) o;

**Member der intrinsischen Datentypen**

int.MaxValue // liefert größten int

int.MinValue // liefert kleinsten int

double.Epsilon // liefert kleinste darstellbare Zahl

char myChar = 'a';

char.isDigit(myChar) // liefert True wenn myChar eine Zahl ist

char.isLetter(myChar) // liefert True wenn myChar eine Zahl ist

char.isWhiteSpace('Hallo du',5) // steht an 5-ter Stelle ein Leerzeichen ?

Konvertierung von Strings

bool b = bool.Parse(inputString); // liefert True wenn inputString = "True"

double d = double.Parse(inputString); // convertiert inputString in double

int i = int.Parse(inputString); // convertiert inputString in int

**Arbeiten mit Strings**

String Member 
Length 
Compare() 
Contains() 
Equals() 
Format() 
Insert() 
Meaning in Life 
This property returns the length of the current string. 
This static method compares two strings. 
This method determines whether a string contains a specific substring. 
This method tests whether two string objects contain identical character 
data. 
This static method formats a string using other primitives (e.g., numerical 
data, other strings) and the {0} notation examined earlier in this chapter. 
This method inserts a string within a given string. Pad Left() 
PadRight() 
Remove ( ) 
Replace() 
Split() 
Trim() 
ToUpper() 
To Lower() 
These methods are used to pad a string with some characters. 
Use these methods to receive a copy of a string with modifications 
(characters removed or replaced). 
This method returns a String array containing the substrings in this 
instance that are delimited by elements of a specified char array or string 
array. 
This method removes all occurrences of a set of specified characters from 
the beginning and end of the current string. 
These methods create a copy of the current string in uppercase or 
lowercase format, respectively. 

Beispiel

string s = Console.ReadLine(); // z.B. "Hallo Welt"

s = s.Replace("Hallo", "Tschüss");

Console.WriteLine("Modified string: {0}", s); // Tschüss Welt

s = s + " ..."; // string concatenation

Console.WriteLine("Modified string: {0}", s); // Tschüss Welt ...

**Textformatierung**

Console.WrlteLlne( I 
• = {O:F8}", I) 
allgemein: 
I ex 
I e :formatierung 
Lfd. Nr. des 
Ausdrucks 
(von 0 an 
0(1) 
ormatierun 
Zeichen 
Anzahl Zeichen für 
Ergebnis der Formatierung, 
Rechtsbündig 
Linksbündig 
wenn fehlt: kürzeste Ganzzahl (Länge) 
Beschreibunq 
Währung 
dezimal 
exponentiell 
Festkomma 
exponentiell 
Nummer mit 
Trennzeichen 
hexa, ganz 
Defaulterqebnis 
€,$ xx,xx.xx 
[-] xxxxxxx 
[-] x.xxxxxxE±xxx 
[-] xxxxxxx.xx 
Variabel 
[-] xx,xxx.xx 
variabel 

**Escape Characters**

Character 
\a 
\r 
\t 
Meaning in Life 
Inserts a single quote into a string literal. 
Inserts a double quote into a string literal. 
Inserts a backslash into a string literal. This can be quite helpful when defining file or 
network paths. 
Triggers a system alert (beep). For console programs, this can be an audio clue to the 
user. 
Inserts a new line (on Windows platforms). 
Inserts a carriage return. 
Inserts a horizontal tab into the string literal. 

**Widening / Narrowing**

Im folgenden Code nimmt die Add Methode nur Argumente vom Typ int,

bekommt aber short übergeben. Was passiert ? Welcher Teil des Codes könnte Probleme geben ?

public static void Main(string[] args)

{

short a = 10, b = 10;

short c = Add(a, b);

Console.WriteLine("{0} + {1} = {2}", a, b, c);

}

public static int Add(int a, int b)

{

return a + b;

}

**Var Keyword**

Typen werden automatisch aus der anfänglichen Zuweisungen geschlussfolgert.

-> implizite Typisierung

Beispiel:

var myInt = 0;

var myBool = true;

var myString = "Time, marches on ... ";

Console.WriteLine("myInt is a: {0}", myInt.GetType().Name);

Console.WriteLine("myBool is a: {0}", myBool.GetType().Name);

Console.WriteLine("myString is a: {0}", myString.GetType().Name);

**Schleifen**

wie in C# gibt es die folgenden Schleifen:

* do {…} while (…)
* while (…) {…}
* for (…; …; …) {…}

zusätzlich gibt es die foreach Schleife:

* foreach (Typ element in Menge)
* jedes Element einer Menge wird genau 1 mal durchlaufen
* Die Menge kann alles sein was das Ienumerable Interface implementiert (Array, Liste, …)
* entspricht einem Iterator in C++
* durchlaufene Elemente können nicht verändert werden
* Beispiel:

string[] carTypes = {"Ford", "BMW", "Trabbi", "Honda" };

foreach (string c in carTypes)

Console.WriteLine(c);

C:\AE5BEB05\807779E6-776F-4DB8-ACAE-B98806A5D697-Dateien\image009.png

**Garbage Collector**

* Objekte benötigen Speicher im Heap (Speicherallokation mit new)
* werden Objekte nicht mehr benötigt sollte der Speicher wieder freigegeben werden
* Bsp:

* Speicherfreigabe ist in C# nicht explizit möglich, sondern geschieht durch den Garbage Collector

Funktionsweise:

Circle kreis1 = new Circle();

Circle kreis2 = new Circle();

[...]

kreis1 = null2;

|  |
| --- |
| A simple approximation how the GC works |

Garbage Collector wird ausgelöst:

1. wenn Programm wenig ausgelastet
2. wenn Speicherressurcen knapp
3. bei Programmende
4. manuell mit GC.collect();