


Informationsblatt	System und- Anwendungssoftware	
Datentypen/Konvertierung		

Datentypen

Variable können über den Typbezeichner der .NET-Klassenbibliothek sowie der C#-Alias Deklariert werden. Entspricht der Typ nicht den Vorgaben der Common Language Specification (CLS), existieren ggf. keine Operatoren für diesen Datentypen.

.NET-Laufzeittyp	C#-Alias	CLS-Konform	Größe
Byte	byte	ja	Ganze 8-Bit Zahl mit Vorzeichen
SByte	sbyte	nein	8-Bit-Ganzzahl ohne Vorzeichen
Int16	short	ja	Ganze 16-Bit-Zahl mit Vorzeichen
UInt16	ushort	nein	16-Bit-Ganzzahl mit Vorzeichen
Int32	int	ja	32-Bit-Ganzzahl mit Vorzeichen
UInt32	uint	nein	32-Bit-Ganzzahl mit Vorzeichen
Int64	long	ja	64-Bit-Ganzzahl mit Vorzeichen
UInt64	ulong	nein	64-Bit-Ganzzahl ohne Vorzeichen
Single	float	ja	32-Bit-Dezimalzahl mit Vorzeichen
Double	double	ja	64-Bit-Dezimalzahl mit Vorzeichen
Decimal	decimal	ja	128-Bit-Dezimalzahl mit Vorzeichen

Ganze Zahlen

Short: Ein ,short' hat eine Größe von 16-Bit.	<code>int value_1 = 3;</code>
Int: Ein ,int' hat eine Größe von 32-Bit.	<code>short value_2 = 1;</code>
Long: Ein ,long' hat eine Größe von 64-Bit.	<code>long value_3 = 12;</code>


Dezimalzahlen

Literale (Zeichenfolge zur direkten Darstellung der Werte), die eine Dezimalzahl beschreiben werden von der .NET-Laufzeitumgebung als Double-Typ angesehen.

Double: Ein double hat die doppelte Genauigkeit eines float (Genauigkeit etwa 16 Stellen).	<code>double value_4 = 3.141;</code>
Float: Der Buchstabe ,F' zwingt das Literal in den Typ float (Genauigkeit etwa 16 Stellen).	<code>float value_5 = 3.141F;</code>
Decimal: Der Buchstabe ,M' zwingt das Literal in den Typ decimal (Genauigkeit etwa 16-25 Stellen).	<code>decimal value_6 = 3.141M;</code>

String und Char

Char: Variablen vom Typ ,char' können Zeichen des Unicode-Zeichensatzes aufnehmen (65.536 Zeichen). Das Literal wird in einfache Anführungsstriche gesetzt.	<code>char letter = 'A';</code>
String: Ein ,char' beschränkt sich auf ein Zeichen. Um eine Zeichenkette zu deklarieren, wird ,string' verwendet.	<code>string text = "Hello";</code>

Informationsblatt	System und- Anwendungssoftware	
Datentypen/Konvertierung		

Objekte

Ein Objekt-Datentyp beschreibt einen vier Byte Zeiger, der auf die Speicheradresse eines beliebigen Wertetyps verweist. Verwenden Sie den Objekt-Datentyp nur wenn sie zum Kompilieren nicht wissen, auf welchen Datentyp die Variable möglicherweise verweist.

Objekt: Eine Variable diesen Typs kann jeden beliebigen anderen Datentyp beschreiben	<code>object input_z1 = "Test";</code> <code>input_z1 = 5;</code>
---	--

Eigenschaften

Statische Eigenschaften: Alle Datentypen sind Klassen, hinter denen sich Möglichkeiten verbergen.	<code>Console.WriteLine</code> <code>("Int(min): {int.MinValue}");</code> <code>//Int(min): -2147483648</code> <code>Console.WriteLine</code> <code>("Int(max): {int.MaxValue}");</code> <code>//Int(max): 2147483647</code>
Eigenschaften: Alle Variablen sind Instanzen von Klassen, hinter deren Variablennamen sich Möglichkeiten verbergen.	<code>int a = 10;</code> <code>string zehn = a.ToString(); //zehn="10"</code> <code>int len_str = zehn.Length; //len_str=2</code>

Konvertierung

Implizite Konvertierung: Der Inhalt der Variable mit dem Typ ,int' wird in einer Variable vom Typ ,long' gespeichert.	<code>int value_7 = 120;</code> <code>long value_8 = value_7;</code>
Explizite Konvertierung: Der größere Wertebereich von ,long' wird in ,int' nur explizit umgewandelt, da der Datenverlust bewusst eingegangen werden muss.	<code>long value_9 = 123;</code> <code>int value_10 = (int)value_9;</code>
Einige Datentypen können mit dem Konvertierungsoperator ,() ' nicht konvertiert werden (Beispielsweise: ,string', ,bool'...). Um dennoch diese Typen zu konvertieren, können Methoden der ,Convert'-Klasse aufgerufen werden.	<code>int zahl_int = Convert.ToInt32("10");</code> <code>string zahl_str =</code> <code>Convert.ToString(zahl_int);</code> <code>bool flag = Convert.ToBoolean("true");</code>

Convert-Klasse

ToByte(Ausdruck)	Konvertiert den Ausdruck in einen 'byte'-Typ.
ToBoolean(Ausdruck)	Konvertiert den Ausdruck in einen 'bool'-Typ.
ToChar(Ausdruck)	Konvertiert den Ausdruck in einen 'char'-Typ.
ToDecimal(Ausdruck)	Konvertiert den Ausdruck in einen 'decimal'-Typ.
ToDouble(Ausdruck)	Konvertiert den Ausdruck in einen 'double'-Typ.
ToInt16(Ausdruck)	Konvertiert den Ausdruck in einen 'short'-Typ. */