

3 Enumeration

Eine Enumeration bezeichnet eine zusammengehörige Gruppe von Konstanten. Die Enumerationsmember zwischen den Anweisungen *Enum* und *End Enum* werden mit konstanten Werten initialisiert. Die definierten Werte können zur Laufzeit nicht geändert werden. Die Aufzählungen können einem bestimmten Typ zugeordnet werden. Innerhalb der Aufzählung können die Typen aber nicht unterschiedlich sein. Wird kein Typ explizit angegeben, wird ein *Integer* angenommen. Folgende Typen für Aufzählungen sind zulässig:

- Integer (Default)
- Long
- Short
- Byte

Wie im folgenden Beispiel dargestellt, können die Werte sowohl positive als auch negative Zahlen sein:

```
Enum Colors
    Red = 1
    Green = 2
    Blue = 4
    Yellow = 8
End Enum
```

Sämtliche Verweise auf Enumerationsmember müssen qualifiziert werden, entweder mit dem Namen einer Enumerationsvariablen oder dem Namen der Enumeration selbst. Im vorhergehenden Beispiel kann auf den ersten Member als *Genres.Blues* verwiesen werden, jedoch nicht als *Blues*

In diesem Beispiel wird die *Enum*-Anweisung zur Definition einer Gruppe von benannten Konstanten verwendet. In diesem Fall handelt es sich bei den Konstanten um Farben, die Ihnen für Dateneingabeformulare einer Datenbank zur Verfügung stehen

```
Public Enum Farben
    Rosa = &HE1E4FF&
    Grau = &H908070&
    Blau = &HFF901E&
    Gruen = &H7FFF00&
End Enum
```

Einsatz von Attributen in Enumerationen

Das Attribut <FlagsAttribute()> gibt an, dass eine Enumeration als Bitfeld, d. h. als Gruppe von Flags, behandelt werden kann.

```
<FlagsAttribute()> _
Enum Colors ' Mit Attribut
    Red = 1
    Green = 2
    Blue = 4
    Yellow = 8
End Enum

Enum Colors ' Ohne Attribut
    Red = 1
    Green = 2
    Blue = 4
    Yellow = 8
End Enum
```

Unterschiedliches Ergebnis mit <FlagsAttribute()> .

```
Aufzählung von Colors Enum ist:
Red
Green
Blue
Yellow

Die Variable myOrange hat einen Wert kombiniert mit Red, Yellow
Enter für weiter!
```

Unterschiedliches Ergebnis ohne <FlagsAttribute()> .

```
Aufzählung von Colors Enum ist:
Red
Green
Blue
Yellow

Die Variable myOrange hat einen Wert kombiniert mit 9
Enter für weiter!
```

Bitfelder können im Gegensatz zu Enumerationskonstanten mit der bitweisen OR-Operation kombiniert werden.

Bitfelder werden im Allgemeinen für Aufzählungen von Elementen verwendet, die zusammen auftreten können, während Enumerationskonstanten gewöhnlich für Aufzählungen von Elementen verwendet werden, die sich gegenseitig ausschließen. Bitfelder sind daher im Gegensatz zu Enumerationskonstanten dazu vorgesehen, zusammen mit der bitweisen OR-Kombination unbenannte Werte zu generieren. Bitfeldern werden in verschiedenen Programmiersprachen im Gegensatz zu Enumerationskonstanten unterschiedlich verwendet.

3.1 Arbeiten mit Enumerationen

Rezept	11.	Verschiedenes
Beispiel		Enumeration_01

```
Public Class ParseTest

    <FlagsAttribute()> _
    Enum Colors
        Red = 1
        Green = 2
        Blue = 4
        Yellow = 8
    End Enum

    Public Shared Sub Main()

        Console.WriteLine("Aufzählung von Colors Enum ist:")
        Dim s As String
        For Each s In [Enum].GetNames(GetType(Colors))
            Console.WriteLine(s)
        Next s

        Console.WriteLine()

        Dim myOrange As Colors = CType([Enum].Parse(GetType(Colors), "Red,
Yellow"), Colors)

        Console.WriteLine("Die Variable myOrange hat einen Wert kombiniert
mit {0}", myOrange)
    End Sub
End Class
```

Methoden zur Enumeration

[Enum].Parse

Konvertiert die Zeichenfolgenentsprechung des Namens oder des numerischen Wertes einer oder mehrerer Enumerationskonstanten in ein entsprechendes Enumerationsobjekt

[Enum].Format

Konvertiert den angegebenen Wert eines angegebenen Enumerationstyps unter Berücksichtigung des angegebenen Formats in die entsprechende Zeichenfolgenentsprechung.

[Enum].GetName

Ruft den Namen der Konstanten der angegebenen Enumeration mit dem angegebenen Wert ab.

[Enum].GetNames

Ruft ein Array mit den Namen der Konstanten einer angegebenen Enumeration ab.

[Enum].GetValues

Ruft ein Array mit den Werten der Konstanten einer angegebenen Enumeration ab.

[Enum].GetUnderlyingType

Gibt den zugrunde liegenden Typ der angegebenen Enumeration zurück.

Rezept	12.	Verschiedenes
Beispiel		Enumeration_02

```
Enum Colors
    Red = 1
    Green = 2
    Blue = 4
    Yellow = 8
End Enum 'Colors

' List all the values in Colors.
Dim names() As String = [Enum].GetNames(GetType(Colors))
Dim values As Array = [Enum].GetValues(GetType(Colors))
Dim i As Integer

For i = 0 To names.Length - 1
    Console.WriteLine("{0} = {1}", names(i), CInt(values.GetValue(i)))
Next
```

Ergebnis des Beispiels

Auflistung der Enum Colors

```
Red = 1
Green = 2
Blue = 4
Yellow = 8

Enter für weiter!
```