





ALLGEMEIN

1. Für Anfänger leicht zu erlernen
2. Universell einsetzbar
3. Erweiterbarkeit der Sprache für Experten
4. Interaktivität
5. Klare Fehlermeldungen
6. Kurze Antwortzeiten
7. Hardwareunabhängigkeit
8. Betriebssystemunabhängigkeit



GESCHICHTE

- 1963 von John George Kemeny und Thomas Eugene Kurtz am Dartmouth College entwickelt, um den Elektrotechnikstudenten den Einstieg in die Programmierung zu erleichtern
- BASIC war entsprechend dem Wunsch seiner „Väter“ für die Schulen kostenlos
- Seinen Höhepunkt erlebte BASIC mit den aus den Bastelcomputern hervorgehenden ersten Heimcomputern Ende der 1970er- und Anfang der 1980er-Jahre, die nahezu alle als Benutzeroberfläche und Programmierumgebung einen BASIC-Interpreter besaßen
- Prominente Beispiele sind der Apple II oder der meistverkaufte Heimcomputer aller Zeiten, der Commodore 64
- Praktisch alle Besitzer von Heimcomputern hatten damals zumindest Grundkenntnisse in BASIC, da die meisten Rechner beim Einschalten den Basic-Interpreter starteten, welcher das Laden weiterer Programme unter Verwendung von BASIC-Befehlen erlaubte





GESCHICHTE 2

- Das seit 1981 verbreitete MS-DOS enthielt ebenfalls einen BASIC-Interpreter – zunächst BASICA bzw. GW-BASIC, später QBasic – der in Deutschland an vielen Schulen eine Rolle im Unterricht der Informatik spielte
- Zu dieser Zeit setzte aber ein Wandel ein, weil andere Hochsprachen wie beispielsweise C
- Mit Einführung von Visual Basic ab Version 5 erzielte Microsoft deutliche Erfolge in Bezug auf die Verarbeitungsgeschwindigkeit, die aber den zuvor verlorenen Boden nicht wieder rückgewinnen konnten
- Mit der Zeit nahm der Anteil der Menschen zu, die einen Computer nur bedienen, aber nicht programmieren konnten → Markt von Windows PC dominiert



EINSTIEG IN BASIC

1. Variablen
2. Funktionen & Sub
3. Kontrollstrukturen
4. Objektorientierung

1. VARIABLEN → DIM VARIABLE AS DATENTYP

Datentyp	Wertebereich	Beschreibung	Standardwert
Vorzeichenbehaftete Ganzzahlen			
SByte	-128 bis 127	8-Bit-Ganzzahl	0
Short	-32.768 bis 32.767	16-Bit-Ganzzahl	0
Integer	-2.147.483.648 bis 2.147.483.647	32-Bit-Ganzzahl	0
Long	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807	64-Bit-Ganzzahl	0
Vorzeichenlose Ganzzahlen			
Byte	0 bis 255	vorzeichenlose 8-Bit-Ganzzahl	0
UShort	0 bis 65.535	vorzeichenlose 16-Bit-Ganzzahl	0
UInteger	0 bis 4.294.967.296	vorzeichenlose 32-Bit-Ganzzahl	0
ULong	0 bis 18.446.744.073.709.551.616	vorzeichenlose 64-Bit-Ganzzahl	0
Gleitkommazahlen			
Single	-3,402823E38 bis 3,402823E38	Gleitkommazahl einfacher Genauigkeit	0.0
Double	-1,79769313486231E208 bis 1,79769313486231E208	Gleitkommazahl doppelter Genauigkeit	0.0
Zeichen und Zeichenketten			
Char	Unicode	einzelnes Zeichen	 Nullzeichen
String	Unicode	Zeichenkette (bis zu 2 Mrd. Zeichen)	leere Zeichenkette
Andere Datentypen			
Boolean	True (wahr) oder False (falsch)	Wahrheitswerte	False
Date	1. Januar 1 bis 31. Dezember 9999	Datum- und Uhrzeitangabe in einem	1. Januar 1, Mitternacht

1. VARIABLEN → DIM VARIABLE AS DATENTYP

Datentyp	Bereich
BOOLEAN	Wahr oder Falsch
INTEGER	- 32.768 bis 32.767
LONGINTER	- 2.147.483.648 bis + -3,402823 · 10 ³⁸ bis 3,402823 · 10 ⁻⁴⁵
SINGLE	1,401298 · 10 ⁻⁴⁵ Bis 3,402823 · 10 ³⁸
DOUBLE	-1,79769313486232 · 10 ³⁰⁸ Bis -4,94065645841247 · 10 ⁻³²⁴ 4,94065645841247 · 10 ⁻³²⁴ Bis 1,79769313486232 · 10 ³⁰⁸
STRING	Kann beliebige Zeichen enthalten
ARRAY	Reihung der anderen Datentypen Dim var(n to m) As Datentyp

1. VARIABLEN → VARIABLENNAMEN

- Der Variablenname muss mindestens ein Zeichen lang sein und mit einem Buchstaben (a-z) beginnen
- zulässige Zeichen in Variablennamen sind alle Buchstaben (a..z) und Zahlen (0..9)

2. FUNKTIONEN & SUB

Der gesamte ausführbare Code muss in einer Prozeduren enthalten sein. Verwenden Sie eine **Sub**-Prozedur, wenn Sie keinen Wert an den aufrufenden Code zurückgeben müssen. Verwenden Sie eine **Function**-Prozedur, wenn Sie einen Wert zurückgeben müssen. Die **Function**-Anweisung kann den Datentyp des Werts deklarieren, den sie zurückgibt.

```
Sub Main()  
    MsgBox „Hallo Welt!“  
End Sub
```

```
Function myFunction(ByVal j As Integer) As Double  
    Return 3.87 * j  
End Function
```

3.KONTROLLSTRUKTUREN

Schleifen

- For...Next
- Do...Loop
- While...Wend

Bedingungen

- If...Then...Else...End If



```
For i = 0 To 100 [Step x]
  Anweisung
Next i
```

```
Do
  Anweisung
Loop [While Bedingung]
```

```
Do
  Anweisung
Loop [Until Bedingung]
```

```
Do [While Bedingung]
  Anweisung
Loop
```

```
Do [Until Bedingung]
  Anweisung
Loop
```

```
If Bedingung1 Then
  Anweisung1
[Elseif Bedingung2 Then
  Anweisung2]
Else
  Anweisung3
End IF
```

```
While Bedingung
  Anweisung
Wend
```

Until bedeutet *Bis etwas ist* und While *so lange etwas ist*



4. OBJEKTORIENTIERUNG (OO)

- Basic unterstützt jegliche Aspekte der Objektorientierung, d.h.
 1. Vererbung
 2. Polymorphie
 3. Kapselung



http://openbook.galileocomputing.de/einstieg_vb_2008/index.htm#_top

