

Programmieren 1



Wintersemester 2016/2017

Marcus Riemer, B.Sc.

Basierend auf den Unterlagen „Programmstrukturen 1“
von Prof. Dr. Andreas Häuslein

Computer speichert Daten im Binärsystem

- Umwandlung in das Dezimalsystem durch Addition aller Stellenwertigkeiten:

64	32	16	8	4	2	1	Dezimal
0	1	0	0	0	0	1	
0	1	0	1	0	1	0	
1	0	0	1	0	1	1	

- Umwandlung in das Binärsystem durch wiederholten Divisionsrest:

33 "durch" 2 = 16 Rest 1
16 "durch" 2 = 8 Rest 0
8 "durch" 2 = 4 Rest 0
4 "durch" 2 = 2 Rest 0
2 "durch" 2 = 1 Rest 0
1 "durch" 2 = 0 Rest 1

42 "durch" 2 = 21 Rest 0
21 "durch" 2 = 10 Rest 1
10 "durch" 2 = 5 Rest 0
5 "durch" 2 = 2 Rest 1
2 "durch" 2 = 1 Rest 0
1 "durch" 2 = 0 Rest 1

↑
101010₂ = 42

- Kleinste adressierbare Einheit sind 8 Bit = 1 Byte.
- Datentypen ordnen vielfache einem Byte Bezeichner zu.

Datentyp (Bezeichner)	Wertebereich	Speicherung
Shortint	-128 .. 127	8 Bit mit Vorz.
Byte	0 .. 255	8 Bit ohne Vorz.
Smallint	-32768 .. 32767	16 Bit mit Vorz.
Word	0 .. 65535	16 Bit ohne Vorz.
Integer	-2147483648 .. 2147483647	32 Bit mit Vorz.
Longint	-2147483648 .. 2147483647	32 Bit mit Vorz.
Cardinal	0 .. 4294967295	32 Bit ohne Vorz.
Longword	0 .. 4294967295	32 Bit ohne Vorz.
Int64	$-2^{63} .. 2^{63}-1$	64 Bit mit Vorz.
UInt64	$0 .. 2^{64}-1$	64 Bit ohne Vorz.

- Länge der generischen Datentypen Integer und Cardinal hängt ab von der voreingestellten Speicherbreite (typisch: 32 oder 64 Bit).
- Auswirkungen auf maximal adressierbare Speichermengen
Maximaler Wert * Blockgröße = Maximal adressierbarer Speicher
 - Arbeitsspeicher hat eine Blockgröße von 8 Bit
 - Festplatten haben variable Blockgrößen, häufig aber 4 Kilobyte

Max Breite	Max Wert	Max RAM	Max HDD
8 Bit	255	0,256 kB	1,024 MB
16 Bit	65.535	65,54 kB	262,1 MB
32 Bit	4.294.967.295	4,295 GB	17.18 TB
64 Bit	18.446.744.073.709.551.615	18,45 EB	~ 70x Internet-Traffic 2015

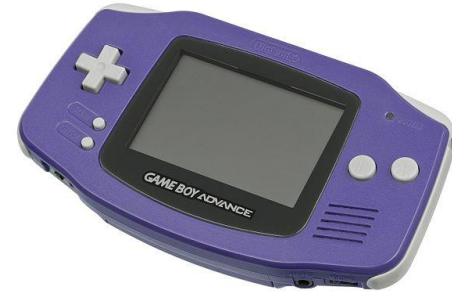
6 Architekturen unterschiedlicher Bit-Breiten



GameBoy
1989



GameBoy Color
1998



GameBoy Advance
2001



SNES
1991



Playstation
1994



N64
1997

6 Auch Buchstaben sind Zahlen



Der Datentyp Char ist ein Hinweis an den Compiler, dass statt der Ziffernrepräsentation der codierten Zahl ein Zeichen ausgegeben werden soll.

- Umrechnungstabelle von Zahl auf Zeichen nennt man „Codepage“
- Heute relevante Codepages:
 - ASCII (0 – 127)
 - Unicode (128 – 2³²)
- Nicht alle diese Zeichen entsprechen exakt einem Buchstaben.
 - Kontrollzeichen
 - Kombinerende Zeichen

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0 NUL	1 SOH	2 STX	3 ETX	4 EOT	5 ENQ	6 ACK	7 BEL	8 BS	9 HT	10 LF	11 VT	12 FF	13 CR	14 SO	15 SI
1	16 DLE	17 DC1	18 DC2	19 DC3	20 DC4	21 NAK	22 SYN	23 ETB	24 CAN	25 EM	26 SUB	27 ESC	28 FS	29 GS	30 RS	31 US
2	32 SPACE	33 EXCLAM. MARK	34 QUOT. MARK	35 NUMB. SIGN	36 DOLLAR SIGN	37 PERCENT SIGN	38 AMPERSAND	39 APOSTROPHE	40 LEFT PAREN.	41 RIGHT PAREN.	42 ASTERISK	43 PLUS SIGN	44 COMMA	45 HYPHEN-MINUS	46 FULL STOP	47 SOLIDUS
3	48 DIGIT ZERO	49 DIGIT ONE	50 DIGIT TWO	51 DIGIT THREE	52 DIGIT FOUR	53 DIGIT FIVE	54 DIGIT SIX	55 DIGIT SEVEN	56 DIGIT EIGHT	57 DIGIT NINE	58 COLON	59 SEMI-COLON	60 LESS-THAN SIGN	61 EQUALS SIGN	62 GREATER-THAN SIGN	63 QUESTION MARK
4	64 COMMERCIAL AT	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
5	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 LEFT SQUARE BRACKET	92 REVERSE SOLIDUS	93 RIGHT SQUARE BRACKET	94 CIRCUMFLEX ACCENT	95 LOW LINE
6	96 GRAVE ACCENT	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
7	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 LEFT CURLY BRACKET	124 VERTICAL LINE	125 RIGHT CURLY BRACKET	126 TILDE	127 DELETE

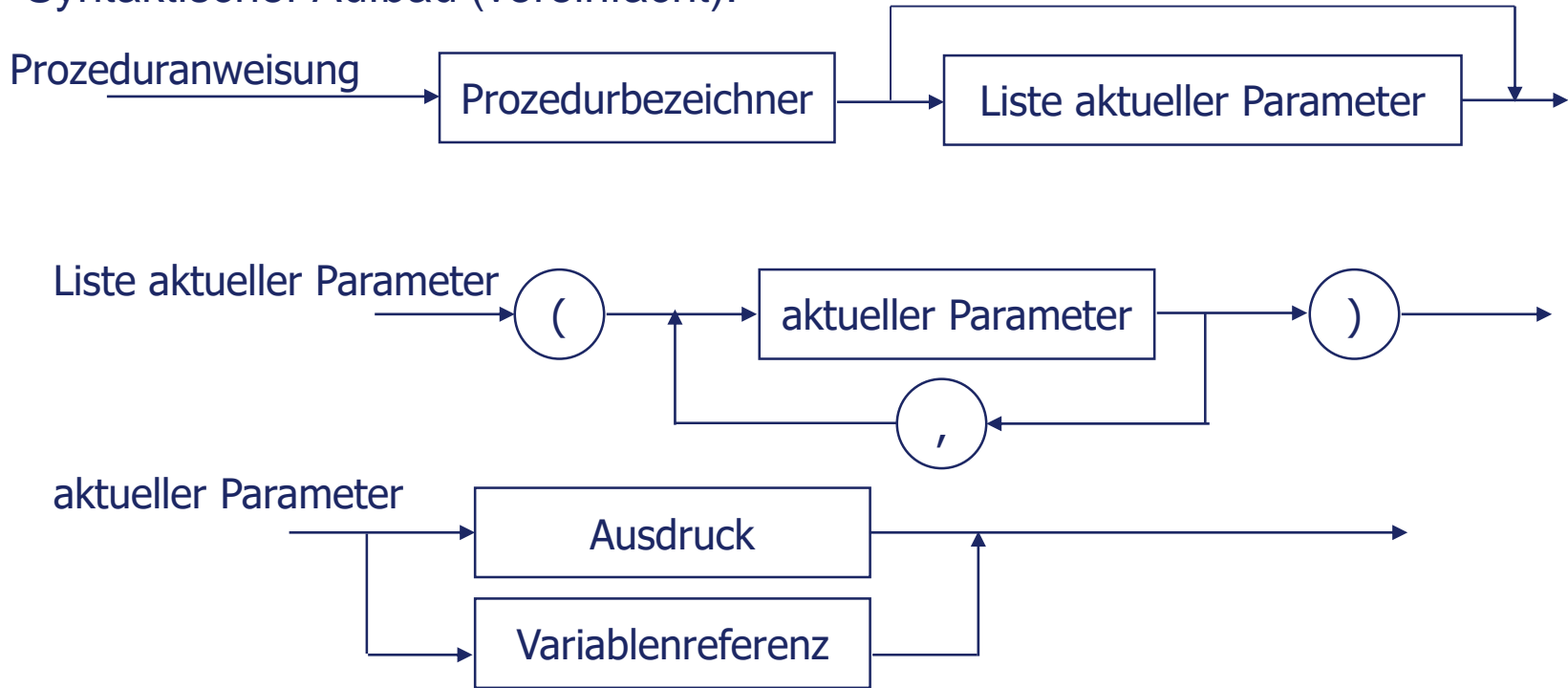
ASCII code table including entity references, control codes and Unicode names (1.1)

Tom Gibara July 2014



- Prozedur: Unter einem Namen (Prozedurbezeichner) zusammengefasste Anweisungsfolge
- Unterscheidung:
 - Vordefinierte Prozeduren: Sind unmittelbarer Bestandteil der Programmiersprache (z.B. `writeln` oder `readln`)
 - Benutzerdefinierte Prozeduren: Anweisungsfolge und ihr Name werden durch die Programmierer festgelegt
- Durch Prozeduranweisung ("Prozeduraufruf") kann die unter dem Namen zusammengefasste Anweisungsfolge zur Ausführung gebracht werden

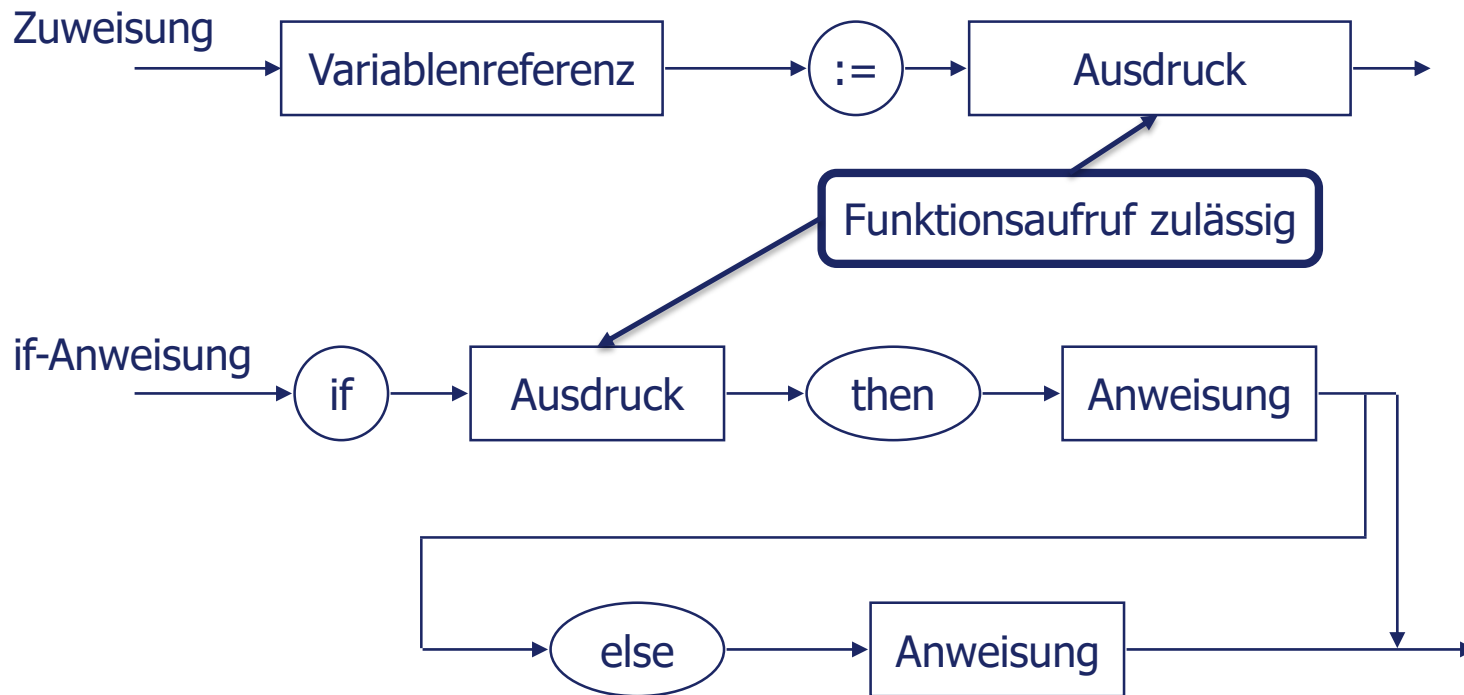
Syntaktischer Aufbau (vereinfacht):



- Die *aktuellen Parameter* versorgen die Prozedur mit Werten, auf denen sie arbeiten kann, und liefern evtl. Ergebniswerte zurück
- Beispiele für Prozeduranweisungen:

```
writeln ('Der Gesamtbetrag ist: ', Gesamtbetrag);  
Bezeichner      aktuelle Parameter  
readln (Menge);
```


- Syntaktischer Aufbau entspricht den Prozeduren, allerdings liefern Funktionen einen Ergebniswert zurück
- Funktionen lassen sich überall dort aufrufen, wo ein Ausdruck erwartet wird.



Und viele weitere Orte

- Für ordinale Datentypen
 - **Ord** liefert die Ordinalzahl des betreffenden Wertes zurück
 - **Pred** liefert den Vorgängerwert ausgehend von einem vorgegebenen Wert
 - **Succ** liefert den Nachfolgerwert ausgehend von einem vorgegebenen Wert
 - **Low** liefert den kleinsten Wert eines ordinalen Datentyps
 - **High** liefert den größten Wert eines ordinalen Datentyps

Zur Erinnerung: Ordinale Datentypen sind ganze Zahlen, Wahrheitswerte und einzelne Zeichen.



- Vordefinierte Funktionen für einzelne Zeichen
 - `UpCase(x)` Umwandlung in Großbuchstabe
- Vordefinierte arithmetische Funktionen für Gleitkommadaten (u.a.):
 - `abs(x)` Betrag von x
 - `sqr(x)` Quadrat von x
 - `sqrt(x)` Quadratwurzel von x
 - `sin(x)` Sinus von x (x im Bogenmaß)
 - `cos(x)` Cosinus von x
 - `arctan(x)` Arcustangens von x
 - `exp(x)` e^x
 - `power(x,y)` x^y
 - `ln(x)` natürlicher Logarithmus von x
 - `int(x)` Vorkommastellen von x
 - `frac(x)` Nachkommastellen von x

Vordefinierte Funktionen zur Umwandlung von Gleitkommazahlen in ganze Zahlen

- Funktion **Round**
 - Rundet Gleitkommawerte zu ganzzahligen Werten
 - Ergebnis ist die nächstliegende ganze Zahl zu einer reellen Zahl
 - Liegt die Gleitkommazahl genau in der Mitte zwischen zwei ganzen Zahlen wird die Zahl mit dem größeren Absolutwert zurückgegeben
- Funktion **Trunc**
 - Verkürzt Gleitkommawerte zu ganzzahligen Werten (Abschneiden der Nachkommastellen)
 - **Trunc** gibt einen Wert zurück, der den gegen Null gerundeten Wert einer Gleitkommazahl darstellt



Aktivierte Überlaufprüfung (Overflow Checking, {\$Q+})

- Bricht die Programmausführung ab, wenn bei einer der ganzzahligen Operationen (+, -, *, Abs, Sqr, Succ, Pred, Inc, Dec) ein Wert außerhalb des zulässigen Wertebereichs auftritt.

Aktivierte Bereichsüberprüfungen (Range Checking, {\$R+})

- Diese Option stellt sicher, dass keiner Variablen ein Wert außerhalb des Wertebereichs zugewiesen wird.
- Ohne diese Option würden Werte vor der Zuweisung (sinngemäß) mit `mod high(Typ)` in den Wertebereich umgerechnet werden.



```
program range_check;
{$R+}
{$Q+}
var
    v1, v2: Int64;
    v3: byte;
begin
    v1 := high(Int64);
    // Folgende Zeile wird durch $R+ zu einem Absturz führen,
    // wenn der Benutzer nicht 0 eingibt;
    readln(v2);
    writeln(v1 + v2);
    // Folgende Zeile wird durch $Q+ zu einem Absturz führen,
    // wenn der Benutzer einen zu großen Wert eingibt.
    readln(v1);
    v3 := v1;
end.
```



Deaktivierte Erweiterte Syntax (Extended Syntax, {\$X-})

- Erlaubt die Rückgabewerte von Funktionen zu ignorieren.
- Führt in eigens definierten Funktionen eine künstliche Variable `result` ein, an welche das Ergebnis zugewiesen werden kann.
- Erlaubt `null`-terminierte Zeichenketten.
- Nimmt automatisch Dereferenzierungen von Zeigertypen vor.



Auch in dieser Übung sollen Eigenschaften von Zahlen errechnet werden. Dieses Mal allerdings nur unter Zuhilfenahme von eingebauten Prozeduren und Funktionen.

- Fragt vom Benutzer zunächst eine ganze Zahl ab und gibt dann die folgenden Eigenschaften aus:
 - Den Vorgänger und den Nachfolger der Zahl, sofern die eingegebene Zahl nicht selbst der maximale oder minimale Wert ist.
 - Die Anzahl der Ziffern in der Dezimaldarstellung. Sie sollen diese nicht mit einer Schleife berechnen, die Funktion `log10` kann diesen Wert für sie errechnen. Sie müssen zur Darstellung dann allerdings noch geeignet runden.
 - Die Anzahl der nötigen Bits für die Binärdarstellung. Hier gelten die gleichen Hinweise wie für die Darstellung im Dezimalsystem.



- Fragt den Benutzer dann, ob er eine weitere Zahl eingeben möchte. Antwortet er mit 'J' oder 'j' soll mit den Anweisungen für die zweite Zahl fortgefahren werden. Andernfalls muss der Benutzer keine Eingaben mehr vornehmen.
- Fragt danach vom Benutzer eine reelle Zahl ab und gibt dann die folgenden Eigenschaften aus:
 - Die Stellen vor dem Komma
 - Die Stellen nach dem Komma
 - Die auf eine Ganzzahl gerundete reelle Zahl
 - Die Quadratwurzel der Zahl
 - Den absoluten Wert (Betrag) der Zahl
 - Die ersten zehn Potenzen der Zahl