

Ein- und Ausgabe

Ein- und Ausgabe

- ◆ Wie werden Schriftzeichen kodiert?
 - ◆ ASCII
 - ◆ Unicode
- ◆ ASCII-Codes in Prolog
- ◆ Prolog-Befehle zur Ein- und Ausgabe

Ziel

- ◆ Kennen der Ein- und Ausgabebefehle in Prolog
- ◆ name/2
- ◆ *nicht nötig*: Auswendiglernen von Codetabellen

Buchstaben als Zahlen: Kodierung

Buchstaben können als Zahlen angesehen werden.

- ◆ Eine *Kodierung* legt fest, welcher Buchstabe mit welcher Zahl gemeint ist.
 - ◆ Willkürliches Beispiel:
 - ◆ 1 = »A«
 - ◆ 2 = »B«
 - ◆ ...
 - ◆ Anderes, ebenso willkürliches Beispiel:
 - ◆ 65 = »A«
 - ◆ 66 = »B«
 - ◆ ...

Wie werden Schriftzeichen kodiert?

Verschiedene Normen/Standards:

- ◆ American Standard Code for Information Interchange (ASCII)
 - anderer Name: International Alphabet 5 (IA5) —
 - regelt Codes für englisches Alphabet (A – Z; a – z; 0 – 9) und einige Sonderzeichen wie @, {, /, (, %.
- ◆ ISO 8859-1 regelt Codes für die Alphabete der meisten westeuropäischen Sprachen, z.B. ä, ß, É, Ò
- ◆ dutzende andere Standards

Probleme:

- ◆ Manche Standards widersprechen sich
- ◆ Nur ASCII ist wirklich weit verbreitet, umfasst aber wenig Zeichen

Wie werden Schriftzeichen kodiert?

ASCII-Codetabelle

10/13 neue Zeile	47 /	63 ?	79 O	95 _	111 o
32 Leerschlag	48 0	64 @	80 P	96 `	112 p
33 !	49 1	65 A	81 Q	97 a	113 q
34 "	50 2	66 B	82 R	98 b	114 r
35 #	51 3	67 C	83 S	99 c	115 s
36 \$	52 4	68 D	84 T	100 d	116 t
37 %	53 5	69 E	85 U	101 e	117 u
38 &	54 6	70 F	86 V	102 f	118 v
39 '	55 7	71 G	87 W	103 g	119 w
40 (56 8	72 H	88 X	104 h	120 x
41)	57 9	73 I	89 Y	105 i	121 y
42 *	58 :	74 J	90 Z	106 j	122 z
43 +	59 ;	75 K	91 [107 k	123 {
44 ,	60 <	76 L	92 \	108 l	124
45 -	61 =	77 M	93]	109 m	125 }
46 .	62 >	78 N	94 ^	110 n	126 ~

Wie werden Schriftzeichen kodiert?

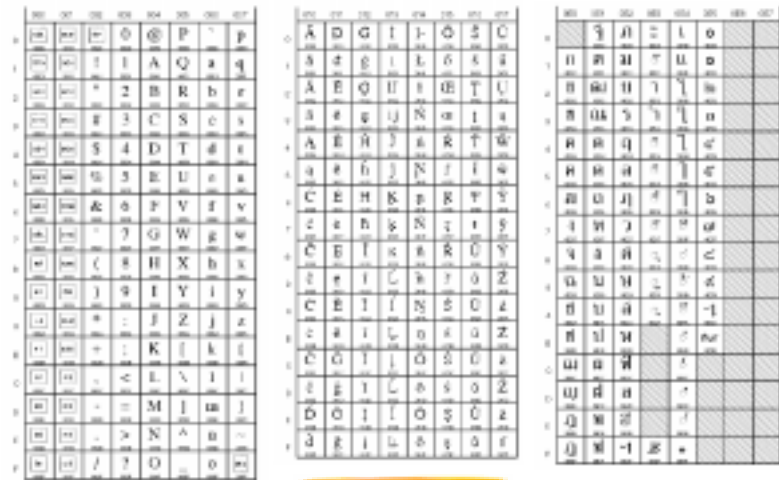
Ein Versuch zur Vereinheitlichung: Unicode

- ◆ Codes für alle gegenwärtig verwendeten Schriftzeichen und Symbole in (fast) allen Sprachen der Welt
- ◆ Codes für Zeichen einiger ausgestorbene Sprachen
- ◆ Kodierung mit 16stelligen 0/1-Folgen (entspricht Zahlen zwischen 0 und 65535 ⇒ maximal 65536 verschiedene Zeichen)
- ◆ Ursprünglich von Xerox/Apple/Sun entwickelt, später von vielen anderen Firmen übernommen
- ◆ Normierung durch Teil von ISO 10646
- ◆ Heute: Unterstützung durch Java, Windows NT, MacOS 8, ...
- ◆ Codetabellen auf <http://www.unicode.org/>

Ein- und Ausgabe – 5

Wie werden Schriftzeichen kodiert?

Ein Versuch zur Vereinheitlichung: Unicode



Ein- und Ausgabe – 6

ASCII-Codes in Prolog

`put/1` gibt ein einzelnes Zeichen aus. Das Argument ist der ASCII-Code des Zeichens.

```
?- put(72), put(97), put(108), put(108), put(111).  
Hallo
```

Allerdings definiert ASCII keine Codes für die Zeichen exotischer Sprachen.

- ◆ Deutsch ist wegen Ä, Ö, Ü und ß eine »exotische« Sprache
- ◆ Ergebnis von `put(138)` oder `put(5000)`: nicht normiert

Ein- und Ausgabe – 7

ASCII-Codes in Prolog

`get0/1` wartet, bis der Benutzer ein einzelnes Zeichen auf der Tastatur eingibt.

Danach wird X an den ASCII-Code des eingegebenen Zeichens gebunden.

```
?- get0(X).  
X = 65
```

Benutzer hat A gedrückt

63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H

Ein- und Ausgabe – 8

ASCII-Codes in Prolog

Eine beliebige Zeichenkette, zwischen zwei " eingeschlossen, wird als Liste der ASCII-Codes dargestellt.

```
?- Kette = "Hallo Du".  
Kette = [72,97,108,108,111,32,68,117].
```

32	Leerschlag	108	l
68	D	111	o
72	H	117	u
97	a		

Ein- und Ausgabe - 9

ASCII-Codes in Prolog

name(Atom, Liste)

- ♦ gibt den Namen von Atom (darf auch eine Zahl sein) als Liste von ASCII-Codes zurück
- ♦ oder erzeugt umgekehrt einen atomaren Term aus einer Liste von ASCII-Codes

```
?- name(bla, L).  
L = [98, 108, 97]
```

```
?- name(27, L).  
L = [50, 55]
```

```
?- name(A, [98, 108, 97]).  
A = bla
```

50	2
51	3
52	4
53	5
54	6
55	7

Ein- und Ausgabe - 10

Schreiben in eine Datei

tell/1 leitet die Ausgabe der Prädikate

- ♦ write/1
- ♦ nl/0
- ♦ put/1
- ♦ display/1

in eine Datei um.

told/0

- ♦ beendet die Umleitung
- ♦ sorgt dafür, dass zukünftige Ausgaben wieder auf dem Bildschirm erscheinen

Ein- und Ausgabe - 11

Schreiben in eine Datei

```
?- tell(hans),  
write(hallo), nl,  
write(du), nl,  
told.
```



Ein- und Ausgabe - 12

Schreiben in eine Datei

telling/1 gibt an, in welche Datei die Ausgabe zur Zeit gerade geleitet wird.

user steht für den Bildschirm.

```
?- telling(Zuerst),
    tell(hans), write(hallo),
    telling(Mitte),
    told,
    telling(Zuletzt).
```

```
Zuerst = user,
Mitte = hans,
Zuletzt = user
```

Lesen aus einer Datei

see/1 nimmt die Eingabe für die Prädikate

- ◆ read/1
- ◆ get0/1
- ◆ ...

aus einer Datei.

seen/0

- ◆ beendet die Umleitung
- ◆ sorgt dafür, dass zukünftige Eingaben wieder vom Benutzer abgefragt werden

Lesen aus einer Datei



```
?- see(hans),
    get0(X), put(X), nl,
    get0(Y), put(Y), nl,
    seen.
```

h
a

Lesen aus einer Datei

seeing/1 gibt an, aus welcher Datei die Eingabe zur Zeit gerade genommen wird.

user steht für die Tastatur.

```
?- seeing(Zuerst),
    see(hans),
    seeing(Mitte),
    seen,
    seeing(Zuletzt).
```

```
Zuerst = user,
Mitte = hans,
Zuletzt = user
```