

# PGdP Tutorium: Vierte Stunde

Benedikt Werner

München, 14. November 2017



# Organisatorisches

- Bei Problemen mit der Korrektur:
  - Nach dem Tutorium fragen
  - E-Mail: [benedikt.werner@tum.de](mailto:benedikt.werner@tum.de)
  - Telegram: <https://t.me/benediktwner>
- Erinnerung: Ab Blatt 4 gilt der Google Style Guide
  - Anleitung für Autoformatierung auf der Website
  - [home.in.tum.de/~wernerbe/pgdp](http://home.in.tum.de/~wernerbe/pgdp)
- Vor der Abgabe die Info auf dem Blatt lesen!
- **Nur auf Moodle** und nur **.java** und **.pdf** Dateien abgeben!

# Zahlenbasen

- Zahlen können auch in anderen Basen als 10 dargestellt werden

## Basis 10

$$\begin{array}{r}
 1 \ 4 \ 2 \ 0 \\
 | \quad | \quad | \quad | \\
 1 * 10^4 \\
 4 * 10^3 \\
 2 * 10^1 \\
 0 * 10^0
 \end{array}$$

## Basis 16

$$\begin{array}{r}
 5 \ 8 \ C_{16} \\
 | \quad | \quad | \\
 5 * 16^3 \\
 8 * 16^1 \\
 12 * 16^0 = 12 * 1 = 12 \\
 = 8 * 16 = 128 \\
 = 5 * 256 = 1280 \\
 \hline
 = 1420
 \end{array}$$

## Basis 2

$$\begin{array}{r}
 \dots 1 \ 0 \ 0 \ 1 \ 1_2 \\
 | \quad | \quad | \quad | \quad | \\
 1 * 2^5 \\
 0 * 2^4 \\
 0 * 2^3 \\
 1 * 2^1 = 1 * 2 = 2 \\
 1 * 2^0 = 1 * 1 = 1 \\
 \vdots
 \end{array}$$

## Extra Symbole für Basis 16

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

# Addition in anderen Basen

- Funktioniert genauso wie in der Basis 10

**Basis 10**

$$\begin{array}{r} 1783 \\ + 274 \\ \hline 11 \\ \hline 2057 \end{array}$$

**Basis 16**

$$\begin{array}{r} 18CB \\ + FF1 \\ \hline 11 \\ \hline 28BC \end{array}$$

**Basis 2**

$$\begin{array}{r} 10111 \\ + 10011 \\ \hline 1 \ 111 \\ \hline 101010 \end{array}$$

# Multiplikation in anderen Basen

- Funktioniert genauso wie in der Basis 10

**Basis 10**

12 · 321

3600

240

+ 12

3852

**Basis 16**

7 · BF

4D0

+ 69

1

539

**Basis 2**

10 · 1101

10000

1000

000

+ 10

11010

# Multiplikationstabelle für die Basis 16

	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

# Basisumwandlung

## Basis 2 zu 10

1 0 1 0 1 0<sub>2</sub>

						$0 * 2^0 = 0 * 1 = 0$
						$1 * 2^1 = 1 * 2 = 2$
						$0 * 2^2 = 0 * 4 = 0$
						$1 * 2^3 = 1 * 8 = 8$
						$0 * 2^4 = 0 * 16 = 0$
						$1 * 2^5 = 1 * 32 = 32$
						$1 * 2^6 = 1 * 64 = 64$

$$1 + 2 + 16 + 32 = 51$$

## Basis 2 zu 16

0010 1111 1010 0110<sub>2</sub>

2	15	10	6
2	F	A	6

# Basisumwandlung

## Basis 10 zu 2

$$24 : 2 = 12 \text{ Rest } 0$$

$$12 : 2 = 6 \text{ Rest } 0$$

$$6 : 2 = 3 \text{ Rest } 0$$

$$3 : 2 = 1 \text{ Rest } 1$$

$$1 : 2 = 0 \text{ Rest } 1$$

$$24 = 11000_2$$



# Strings und Characters

```
String myString = "Hello World!";  
char myChar = 'c';
```

Characters sind im Prinzip Zahlen:

```
char c = 'A';
```

```
char c = (char) 65;
```

```
char c = 65;
```

```
int number = (int) c;
```

```
number = 'B';
```

```
c += 3;
```

```
'B' < 'C'
```

# ASCII Tabelle

Scan-code	ASCII hex dez	Zeichen	Scan-code	ASCII hex dez	Zch.	Scan-code	ASCII hex dez	Zch.	Scan-code	ASCII hex dez	Zch.
	00 0	NUL ^@		20 32	SP		40 64	@	0D	60 96	`
	01 1	SOH ^A	02	21 33	!	1E	41 65	A	1E	61 97	a
	02 2	STX ^B	03	22 34	"	30	42 66	B	30	62 98	b
	03 3	ETX ^C	29	23 35	#	2E	43 67	C	2E	63 99	c
	04 4	EOT ^D	05	24 36	\$	20	44 68	D	20	64 100	d
	05 5	ENQ ^E	06	25 37	%	12	45 69	E	12	65 101	e
	06 6	ACK ^F	07	26 38	&	21	46 70	F	21	66 102	f
	07 7	BEL ^G	0D	27 39	'	22	47 71	G	22	67 103	g
0E	08 8	BS ^H	09	28 40	(	23	48 72	H	23	68 104	h
0F	09 9	TAB ^I	0A	29 41	)	17	49 73	I	17	69 105	i
	0A 10	LF ^J	1B	2A 42	*	24	4A 74	J	24	6A 106	j
	0B 11	VT ^K	1B	2B 43	+	25	4B 75	K	25	6B 107	k
	0C 12	FF ^L	33	2C 44	,	26	4C 76	L	26	6C 108	l
1C	0D 13	CR ^M	35	2D 45	-	32	4D 77	M	32	6D 109	m
	0E 14	SO ^N	34	2E 46	.	31	4E 78	N	31	6E 110	n
0F	15 15	SI ^O	08	2F 47	/	18	4F 79	O	18	6F 111	o
	10 16	DLE ^P	0B	30 48	0	19	50 80	P	19	70 112	p
	11 17	DC1 ^Q	02	31 49	1	10	51 81	Q	10	71 113	q
	12 18	DC2 ^R	03	32 50	2	13	52 82	R	13	72 114	r
	13 19	DC3 ^S	04	33 51	3	1F	53 83	S	1F	73 115	s
	14 20	DC4 ^T	05	34 52	4	14	54 84	T	14	74 116	t
	15 21	NAK ^U	06	35 53	5	16	55 85	U	16	75 117	u
	16 22	SYN ^V	07	36 54	6	2F	56 86	V	2F	76 118	v
	17 23	ETB ^W	08	37 55	7	11	57 87	W	11	77 119	w
	18 24	CAN ^X	09	38 56	8	2D	58 88	X	2D	78 120	x
	19 25	EM ^Y	0A	39 57	9	2C	59 89	Y	2C	79 121	y
	1A 26	SUB ^Z	34	3A 58	:	15	5A 90	Z	15	7A 122	z
01	1B 27	Esc ^[	33	3B 59	;		5B 91	[		7B 123	{
	1C 28	FS ^\	2B	3C 60	<		5C 92	\		7C 124	
	1D 29	GS ^]	0B	3D 61	=		5D 93	]		7D 125	}
	1E 30	RS ^^	2B	3E 62	>	29	5E 94	^		7E 126	~
	1F 31	US ^_	0C	3F 63	?	35	5F 95	_	53	7F 127	DEL

# String-Methoden

Methode: `char charAt(int i)`

```
char c = "Hello World!".charAt(2); // c = l
```


Methode: `int length()`

```
int len = "Hello World!".length(); // 12
```

Anders als bei Arrays: `myArray.length;`  
`myString.length();`

## Aufgabe 4.2 Cäsar lebt

A	B	C	D	E	F	...	X	Y	Z
C	D	E	F	G	H		Z	A	B



- (a) Lesen Sie einen `String` mittels der MiniJava-Methode `readString()` ein, der später ver-/entschlüsselt werden soll.
- (b) Lesen Sie einen `Integer` mittels der MiniJava-Methode `read()` ein, der als zyklischer Shift verwendet werden soll. Beachten Sie, dass der `Integer` sowohl negativ als auch betragsmäßig größer als 26 sein kann!
- (c) Implementieren Sie nun den Algorithmus und ver-/entschlüsseln Sie den `String` aus (a) anhand des Schlüssels aus (b).
- (d) Beachten sie Groß- und Kleinschreibung im Klartext wie auch Geheimtext.
- (e) Geben Sie den verschlüsselten `String` mittels `write(String s)` aus.

## Aufgabe 4.3 Vokalersetzung

Schreiben Sie ein Programm `Vokalersetzung`, das im unten gegebenen Text alle Vokale durch einen vom Benutzer eingegebenen Vokal ersetzt. Achten Sie darauf, dass die Groß- und Kleinschreibung nach der Ersetzung mit der vor der Ersetzung übereinstimmt. Umlaute (Ä/ä, Ö/ö, usw.) sollen dabei nicht als Vokale betrachtet werden. Beispiel: Das Wort "Exenmeister" wird bei Eingabe von O oder o zu "Oxonmoostor".

Verwenden Sie keine Bibliotheksfunktionen der Klasse `String` außer `char charAt(int i)` und `int length()`.

## Aufgabe 4.4 Rechdschreipschwäche

Schreiben Sie ein MiniJava-Programm, das einen `String` einliest und darin jeden Kleinbuchstaben durch den jeweiligen Großbuchstaben ersetzt und umgekehrt. Alle anderen Zeichen sollen nicht ersetzt werden. Geben Sie den konvertierten `String` via der Methode `write(String s)` aus.

Beispiel: `"Hello Students!"` wird umgewandelt in `"hELLO sTUDENTS!"`.

Verwenden Sie keine Bibliotheksfunktionen der Klasse `String` außer `char charAt(int i)` und `int length()`.