



Codierung und Verschlüsselung Stellenwertsysteme & Speichergrößen



Binärzahlen

Öffne die Zaubershow und erledige folgende Aufgaben:

- Versuche durch Ausprobieren, die Zahl 53 in der Mitte erscheinen zu lassen.
- Versuche durch Ausprobieren, die Zahl 222 in der Mitte erscheinen zu lassen.
- **Wie viel Wert addiert jeder Scheinwerfer zu der Zahl in der Mitte? Notiere die Werte!**
- **Welches Muster steckt hinter der Größe der Werte für jeden Scheinwerfer?**
- Lass die Zahl 14 anzeigen und gehe dabei systematisch vor.
- Lass die Zahl 113 anzeigen und gehe dabei systematisch vor.
- Lass die Zahl 255 anzeigen und gehe dabei systematisch vor.
- Lass die Zahl 157 anzeigen und gehe dabei systematisch vor.
- **Welches System hast du angewandt, um möglichst schnell die Zahlen anzeigen zu lassen?**

Umrechnung Dezimalzahl $\rightarrow$ Binärzahl

- Beispiel anhand 202_{10}

Potenz	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	128	64	32	16	8	4	2	1
202	$202-128=74$	$74-64=10$			$10-8=2$		$2-2=0$	
Binär	1	1	0	0	1	0	1	0

- Formuliere den obigen Algorithmus in eigenen Worten!
- $300_{10} =$ (Tipp: Man braucht mehr als 8 Binärstellen)
- $1024_{10} =$

Binärzahlen

- Es leuchtet die 1., 3., 5. und 7. Lampe von links. Welcher Wert wird wohl in der Mitte angezeigt?
- Es leuchtet die 2., 3., 5., 6., 7. und 8. Lampe von links. Welcher Wert wird wohl in der Mitte angezeigt?
- **Wie hast du die letzten beiden Zahlen berechnet?**

Umrechnung Binärzahl → Dezimalzahl

- Binärzahl 11001010_2

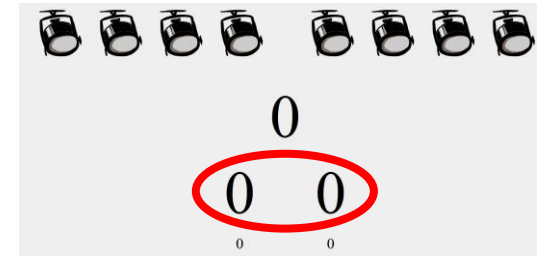
Potenz	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	128	64	32	16	8	4	2	1
Zahl	1	1	0	0	1	0	1	0

$$128 + 64 + \quad \quad \quad 8 + \quad \quad 2 \quad \quad = \mathbf{202}$$

- Übung: Zähle mit einer Hand binär bis 31 hoch
- $10\ 1100\ 1100_2 =$
- $10\ 0000\ 1111_2 =$

Hexadezimalzahlen

- Öffne die Zaubershow und stelle für die Dezimalzahl in der Mitte die Zahl 8 ein:
 - Zähle Schritt für Schritt von 8_{10} bis 18_{10} hoch und beobachte, welche Werte die beiden unteren Zahlen annehmen.
 - Welche Werte haben sie angenommen und für welchen Wert diesen diese?
 - Wie viele verschiedene Werte kann eine Ziffer im Hexadezimalen logischerweise annehmen?



Umrechnung Hexadezimalzahl → Dezimalzahl

- Umrechnung mithilfe eines Polynoms
- Hexadezimalzahl $2c8a_{16}$

Potenz	16^3	16^2	16^1	16^0
Zahl	2	c	8	a

$$2 \cdot 16^3 + 12 \cdot 16^2 + 8 \cdot 16^1 + 10 \cdot 16^0 = \mathbf{11402}$$

- Übung: welchen Dezimalwert haben die folgenden Zahlen?
- $2fa_{16}$
- abc_{16}

Umrechnung Dezimalzahl → Hexadezimalzahl

- Umrechnung durch sukzessives Teilen
- Dezimalzahl 11402_{10} :

– 11402 /R 16 = 712 Rest 10 $\triangleq$ **a**
– 712 /R 16 = 44 Rest **8**
– 44 /R 16 = 2 Rest 12 $\triangleq$ **c**
– 2 /R 16 = 0 Rest **2**

→ $2c8a_{16}$

Potenz	16^3	16^2	16^1	16^0
Zahl	2	c	8	a

- Übung: welchen Hexadezimalwert haben die folgenden Zahlen?
- 1337_{10}
- 221_{10}

Binärzahl $\Leftrightarrow$ Hexadezimalzahl

4 Binärziffern $\triangleq$ 1 Hexadezimalziffer $\rightarrow$ da $2^4 = 16$

Umrechnung durch Zusammenfassen von 4-Bit-Blöcken von rechts:

10 1100 1000 1010
↑↑ ↑↑ ↑↑ ↑↑
2 c 8 a

- Funktioniert in beide Richtungen!
 - Wandle um: 1110010_2 , de_{16} , 1011111001_2 , $2fa_{16}$
- $\rightarrow$ Schultaschenrechner können Umrechnungen zwischen Binär-, Hexadezimal- und Dezimalsystem!

Dez	Bin	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Hexadezimalzahlen

```
FAHRRAD rad = new FAHRRAD();
```

```
System.out.println(rad);
```

- Speicheradressen

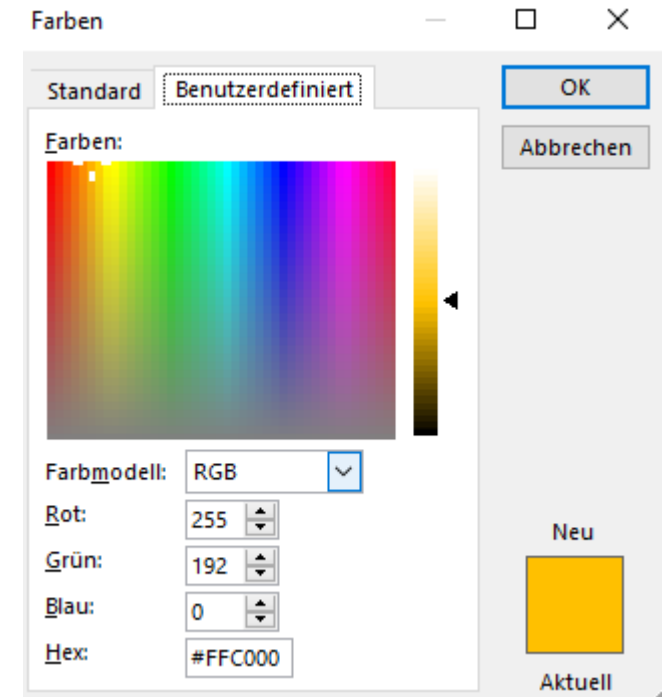
FAHRRAD@1db4d56c

- MAC-Adresse
- IPv6-Adresse

```
Ethernet-Adapter LAN-Verbindung:  
Verbindungsspezifisches DNS-Suffix:  
Beschreibung . . . . . : Realtek RTL8168B/8111B-Familie-PCI-E-Giga  
bit-Ethernet-NIC <NDIS 6.20>  
Physikalische Adresse . . . . . : 00-6F-D0-A6-D0-FA  
DHCP aktiviert . . . . . : Ja
```

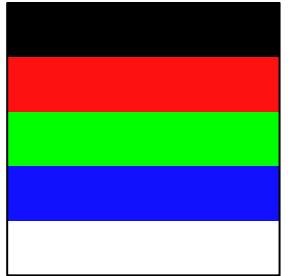
Präfix				Interface Identifier			
IPv6: 2001:0DB8:AC10:FE01:1319:8A2E:0370:7347							
16 Bit	16 Bit	16 Bit	16 Bit	16 Bit	16 Bit	16 Bit	16 Bit

- Farbcodierung



Hexadezimalzahlen

- Hexadezimalzahlen werden häufig dazu verwendet um binär codierte Zeichen als Mensch effizient lesen zu können.
- Nutze diesen Farbmischer, um die Farbencodierung zu verstehen:
http://www.spectrumcolors.de/color_rgb_demo.php
- Öffne Powerpoint, Word, oder Excel und lass ein selbstgeschriebenes Wort in einer selbst zusammengemischten Farbe anzeigen.



Codierung von Zeichen → BlueJ-Projekt

- Sieh dir den Code der Klasse Codierung an. Versuche vorherzusagen, was die Methode umwandeln(..) macht und ausgibt.
- Probieren Sie die Methode umwandeln(...) für ein paar Zahlen aus.
 - Versuche das Zeichen A ausgeben zu lassen. Zahl:___
 - Versuche das Zeichen Z ausgeben zu lassen. Zahl:___
 - Mit welchen Zahlen sind die kleinen Buchstaben codiert?
 - Welche Zeichen hast noch gefunden?
- Diskutiere mit deinem Nachbarn, was diese Codezeile bewirkt:
zeichen = (char) zahl;

Codierung von Zeichen → BlueJ-Projekt

- Kopiere die Methode `umwandeln()` und kehre sie um. Das heißt, dass die Methode das Zeichen erwartet und die dazugehörige Zahl zurückgibt.
 - Welche Zahlen werden den folgenden Zeichen zugeordnet: `+` `{` `&`
- Recherchiere eine ASCII-Tabelle, die zur Codierung von Zeichen verwendet wird und gib an welche Zahlen den folgenden Zeichen zugeordnet werden:
 - Ziffern 0 bis 9:
 - Großbuchstaben:
 - Kleinbuchstaben

Codierung von Zeichen → BlueJ-Projekt

- Das Umwandeln von Datentypen funktioniert mit allen primitiven Datentypen!
- Implementiere eine Methode, die eine ganze Zahl zu einer Kommazahl umwandelt.
- Implementiere eine Methode, die eine Kommazahl zu einer ganzen Zahl umwandelt. → Was passiert hier?

Speichergrößen

- **Bit** $\triangleq$ Bezeichnung für eine Stelle einer Binärzahl und ist die kleinstmögliche Unterscheidung, zwischen zwei Möglichkeiten meist mit 0 und 1 gekennzeichnet, die ein digitaltechnisches System treffen kann.
- 8 Bit = **1 Byte** = 2 Hexziffern = 256 unterschiedliche Zustände → Ziffern von 0 bis 255 darstellbar
- Einheitenzeichen: **b** für Bit und **B** für Byte
 - 1 kB = 1000 B (Kilobyte)
 - 1 MB = 1000 kB (Megabyte)
 - 1 GB = 1000 MB (Gigabyte)
 - 1 TB = 1000 GB (Terabyte)
 - 1 PB = 1000 TB (Petabyte)

Ein Zeichen nach
ASCII/UTF-8 codiert
benötigt 1 Byte
Speicherplatz! → siehe
ASCII-Tabelle

Gefühl für Speichergrößen

Gib für jeden Punkte die ungefähre Datenmenge an und ordne der Größe nach aufsteigend:

- Ein Lied als Mp3-Datei
- gestreamter Full-HD-Film
- Ungefähre Größe des Betriebssystems (z. B. Android, Windows, iOS)
- Foto mit Handykamera
- Gesamter Bibeltext mit 4.410.133 Buchstaben
- E-Mail ohne Anhang oder Textnachricht
- Durchschnittlicher täglicher Upload von Videos bei Youtube
- Installierter Arbeitsspeicher (RAM) an deinem Schulrechner