



Technische
Universität
Braunschweig

IAS

INSTITUTE FOR
APPLICATION
SECURITY



Informatik Vorkurs - Vorlesung 2

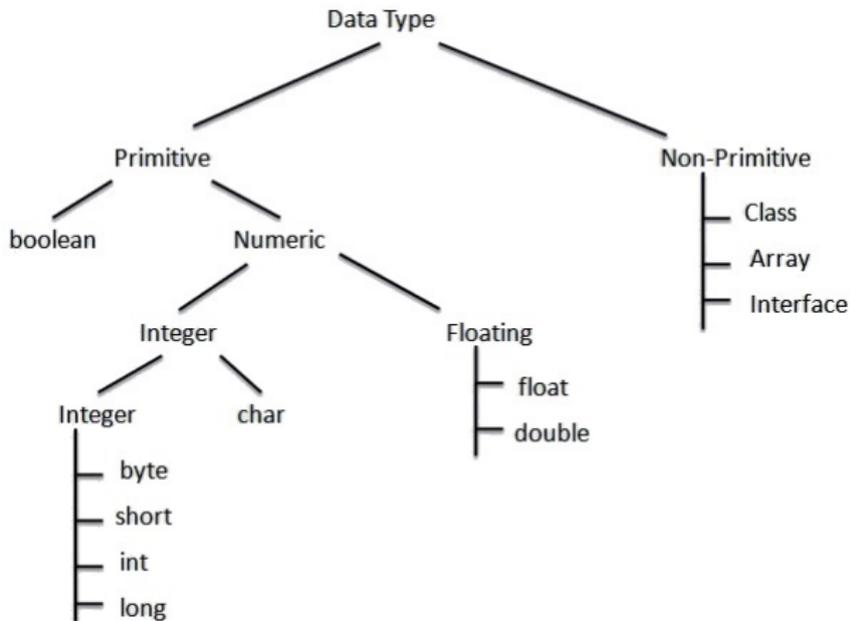
Variablen, Arrays, Zahlensysteme

Torben Achilles, 9. Oktober 2018

Inhalt

- **Variablen & Datentypen**
- **Arrays**
- **Zahlensysteme**

Datentypen



<http://javaseeeedu.blogspot.com/2015/08/data-types-in-java.html>

Ganzzahltypen

- Ganze Zahlen
- Verschiedener Wertebereich

Typen:

- byte \rightarrow 1 Byte = 8 Bits
- short \rightarrow 2 Byte = 16 Bits
- int \rightarrow 4 Byte = 32 Bits
- long \rightarrow 8 Byte = 64 Bits

Wertebereiche

byte:

- $-128 \rightarrow 127$
- $-2^7 \rightarrow 2^7 - 1$

short:

- $-32\,768 \rightarrow 32\,767$
- $-2^{15} \rightarrow 2^{15} - 1$

int:

- $-2\,147\,483\,648 \rightarrow 2\,147\,483\,647$
- $-2^{31} \rightarrow 2^{31} - 1$

long:

- $-9\,223\,372\,036\,854\,775\,808 \rightarrow 9\,223\,372\,036\,854\,775\,807$
- $-2^{63} \rightarrow 2^{63} - 1$

Gleitkommazahlen

- Zahlen mit Nachkommastellen
- Minimum und Maximum sind abhängig von der Genauigkeit!

Typen:

- float → 4 Byte
 - Größe: 4 Byte (32 Bit) – (1 + 8 + 23)
 - Genau bis ca. 8 Dezimalstellen
 - Theoretisch $\pm 3,403 \cdot 10^{38}$
- double → 8 Byte
 - Größe: 8 Byte (64 Bit) – (1 + 11 + 52)
 - Genau bis ca. 16 Dezimalstellen
 - Theoretisch: $\pm 1,798 \cdot 10^{308}$

boolean

- Wahrheitswerte
- Wahr (true) oder Falsch (false)
- Siehe Aussagenlogik (später)

char

- Einzelne Zeichen
- Intern Zahlen → siehe ACSII-Tabelle

Dez	Hex	Okt	Zeichen	Dez	Hex	Okt	Zeichen
65	0x41	101	A	97	0x61	141	a
66	0x42	102	B	98	0x62	142	b
67	0x43	103	C	99	0x63	143	c
68	0x44	104	D	100	0x64	144	d
...

```
1 System.out.println('a'*10); //970
```

Zeichenketten (Strings)

- Speichert Zeichenketten
- Zeichenfolgen in Anführungszeichen z.B. "Hallo Welt!"
- Zusammenfügen mit +

```
1 String s1 = "Hallo ";  
2 String s2 = "Welt!";  
3 System.out.println(s1 + s2);
```

Variablen

- Sind Speicherplätze für Daten
- Haben einen Typ und einen Namen
- Speichern Werte von ihrem Typ
- Deklaration: `Typ bezeichner;`
- Deklaration mit Initialisierung: `Typ bezeichner = ausdruck;`
- Variablen müssen vor ihrer lesenden Verwendung einen Wert zugewiesen bekommen
- Veränderbar mit Zuweisung: `variable = wert`

```
1 int zahl1; //Deklaration
2 int zahl2 = 2; //Deklaration mit Initialisierung
3 zahl1 = 1; //Zuweisung
```

Bezeichner

- Namen von Variablen, Klassen, Methoden
- Zeichenvorrat: Buchstaben (Unicode), Zahlen, `_`, `$`
- Beginnen nicht mit einer Zahl
- Kein reserviertes Schlüsselwort wie z. B. `public`

Arithmetische Operatoren

+ : Addition / Vorzeichen z. B. $1.0 + 1$

- : Subtraktion / Vorzeichen z. B. $1.0 - 1$

* : Multiplikation z. B. $1.0 * 5$

/ : Division z. B. $5 / 2$

% : Modulo z. B. $5 \% 2$

Typ des Ergebnisses:

- Wenn beide Operanden vom selben Typ sind: Typ der Operanden
- Sonst Typ des größten Operanden

```
1 System.out.println(1 + 1);    //2
2 System.out.println(1.0 + 1); //2.0
```

- Variablen & Datentypen
- **Arrays**
- Zahlensysteme

Arrays

- Durchnummerierte Folge von Elementen
- Alle Elemente haben den selben Datentyp
- Angabe mit `Typ[]`
- Feste Länge
- Neues Array wird mit `new` erzeugt: `new Typ[länge]`
- Nummerierung von 0 bis Länge - 1

```
1 int [] a;  
2 String [] namen;  
3 int [] leangefuenf = new int [5];
```

Elementzugriff

- Zugriff auf i-tes Element vom Array a: `a[i]`
- Falls $i \geq$ Länge des Arrays: `ArrayIndexOutOfBoundsException`
- Länge eines Arrays a: `a.length`
- Index i: Ausdruck vom Typ `int`

```
1 int[] zahlen = new int[5];
2 zahlen[0] = 7;
3 zahlen[1] = 11;
4 zahlen[2] = 73;
5 zahlen[3] = 23;
6 zahlen[4] = 56;
7
8 System.out.println(zahlen[4]);
9 //System.out.println(zahlen[5]) <- AIOOB Exception
```

Initialisierung von Arrays

- Bei Array-Initialisierung ist die Angabe der Elemente anstatt eines mit `new` erzeugten Arrays möglich.
- `Typ[] a = Wert1, Wert2, Wert3;`
- Länge wird automatisch ermittelt

```
1 int[] zahlen = {7, 11, 73, 23, 56};
2 System.out.println(zahlen[0]); //7
3 System.out.println(zahlen[4]); //56
4
5 zahlen[4] = 42;
6
7 // System.out.println(zahlen[5]) <- AIOOB Exception
```

- Variablen & Datentypen
- Arrays
- **Zahlensysteme**

Zahlensysteme

- Dezimal
 - $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Dual (Binär)
 - $A = \{0, 1\}$
- Oktal
 - $A = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Hexadezimal
 - $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

Das Dezimale Zahlensystem

- Zahlen wie wir sie kennen
- Wertigkeit ..., 10^2 , 10^1 , 10^0

$235_{10} =$	10^2	10^1	10^0
	2	3	5

- $2 \cdot 100 + 3 \cdot 10 + 5 \cdot 1 = 235_{10}$

Das Binäre Zahlensystem

- Ziffern 0 und 1
- Wertigkeit ... , 2^2 , 2^1 , 2^0
- $235_{10} = 11101011_2$

$$235_{10} =$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	0	1	1

- $11101011_2 =$
 $1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 8 + 1 \cdot 2 + 1 \cdot 1 = 235_{10}$

Das Oktale Zahlensystem

- Ziffern 0, 1, 2, 3, 4, 5, 6, 7
- Wertigkeit ... , 8^2 , 8^1 , 8^0
- $235_{10} = 353_8$

$$235_{10} =$$

8^2	8^1	8^0
3	5	3

- $353_8 = 3 \cdot 64 + 5 \cdot 8 + 3 \cdot 1 = 235_{10}$

Das Hexadezimale Zahlensystem

- Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- A = 10, B = 11, C = 12, D = 13, E = 14, F = 15
- $235_{10} = EB_{16}$

$235_{10} =$	16^1	16^0
	E	B

- $EB_{16} = 14 \cdot 16 + 11 \cdot 1 = 235_{10}$

Umrechnen vom Dezimalsystem

- Teilen durch die Basis des Zahlensystems und Betrachtung des Rests
- $235_{10} =$

$$235 : 2 = 117 \text{ Rest } 1$$

$$117 : 2 = 58 \text{ Rest } 1$$

$$58 : 2 = 29 \text{ Rest } 0$$

$$29 : 2 = 14 \text{ Rest } 1$$

$$14 : 2 = 7 \text{ Rest } 0$$

$$7 : 2 = 3 \text{ Rest } 1$$

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

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

Umrechnen vom Dezimalsystem

- Teilen durch die Basis des Zahlensystems und Betrachtung des Rests
- $235_{10} = 11101011_2$

$$235 : 2 = 117 \text{ Rest } 1$$

$$117 : 2 = 58 \text{ Rest } 1$$

$$58 : 2 = 29 \text{ Rest } 0$$

$$29 : 2 = 14 \text{ Rest } 1$$

$$14 : 2 = 7 \text{ Rest } 0$$

$$7 : 2 = 3 \text{ Rest } 1$$

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

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



Umrechnen vom Dezimalsystem

■ $235_{10} =$

$$235 : 8 = 29 \text{ Rest: } \mathbf{3}$$

$$29 : 8 = 3 \text{ Rest: } \mathbf{5}$$

$$3 : 8 = 0 \text{ Rest: } \mathbf{3}$$

Umrechnen vom Dezimalsystem

- $235_{10} = 353_8$

$$235 : 8 = 29 \text{ Rest: } \mathbf{3}$$

$$29 : 8 = 3 \text{ Rest: } \mathbf{5}$$

$$3 : 8 = 0 \text{ Rest: } \mathbf{3}$$



Umrechnen vom Dezimalsystem

- $235_{10} = 353_8$

$$235 : 8 = 29 \text{ Rest: } \mathbf{3}$$

$$29 : 8 = 3 \text{ Rest: } \mathbf{5}$$

$$3 : 8 = 0 \text{ Rest: } \mathbf{3}$$



- $235_{10} =$

$$235 : 16 = 14 \text{ Rest: } 11 = \mathbf{B}$$

$$14 : 16 = 0 \text{ Rest: } 14 = \mathbf{E}$$

Umrechnen vom Dezimalsystem

- $235_{10} = 353_8$

$$235 : 8 = 29 \text{ Rest: } \mathbf{3}$$

$$29 : 8 = 3 \text{ Rest: } \mathbf{5}$$

$$3 : 8 = 0 \text{ Rest: } \mathbf{3}$$



- $235_{10} = EB_{16}$

$$235 : 16 = 14 \text{ Rest: } 11 = \mathbf{B}$$

$$14 : 16 = 0 \text{ Rest: } 14 = \mathbf{E}$$



Umrechnen vom Dualsystem

- Bilden von Gruppen
- $235_{10} = 11101011_2$
- Oktal: Ziffern 0 – 7 $\implies 8 = 2^3$ Möglichkeiten \rightarrow 3 Bit pro Gruppe

Umrechnen vom Dualsystem

- Bilden von Gruppen
- $235_{10} = 11101011_2$
- Oktal: Ziffern 0 – 7 $\implies 8 = 2^3$ Möglichkeiten \rightarrow 3 Bit pro Gruppe

Binär	011	101	011
Oktal	3	5	3

Umrechnen vom Dualsystem

- Bilden von Gruppen
- $235_{10} = 11101011_2$
- Oktal: Ziffern $0 - 7 \implies 8 = 2^3$ Möglichkeiten \rightarrow 3 Bit pro Gruppe
- Hexadezimal: Ziffern $0 - F \implies 16 = 2^4$ Möglichkeiten \rightarrow 4 Bit pro Gruppe

Binär	1110	1011
Hexadezimal	14 = E	11 = B

Fragen?

- Tipp für Zahlensysteme:
<https://www.arndt-bruenner.de/mathe/scripts/Zahlensysteme.htm>