



# Grunddatentypen, Ausdrücke und Variablen

---

Gilbert Beyer und Annabelle Klarl

Zentralübung zur Vorlesung Einführung in die Informatik

<http://www.pst.ifi.lmu.de/Lehre/wise-11-12/infoeinf>



## Vorlesung heute:

- Grunddatentypen
- Ausdrücke
- Variablen
- Zustände



# Grunddatentypen

## Ganze Zahlen

Typ	Größe	Wertebereich	
▪ <code>byte</code>	1 Byte	-128	bis 127
		$-2^7$	bis $2^7-1$
▪ <code>short</code>	2 Byte	-32768	bis 32767
▪		$-2^{15}$	bis $2^{15}-1$
▪ <code>int</code>	4 Byte	-2 147 483 648	bis 2 147 483 647
		$-2^{31}$	bis $2^{31}-1$
▪ <code>long</code>	8 Byte	$-2^{63}$	bis $2^{63}-1$
		-9 223 372 036 854 775 808	bis 9 223 372 036 854 775 807



# Grunddatentypen

## Gleitkommazahlen

Typ	Größe	Wertebereich	Genauigkeit
float	4 Byte	ca. $(+/-)10^{-45}$ bis $(+/-)10^{38}$	7 Stellen
double	8 Byte	ca. $(+/-)10^{-324}$ bis $(+/-)10^{308}$	15 Stellen



# Grunddatentypen

**byte** < **short** < **int** < **long** < **float** < **double**



## Aufgabe 1 a)

Was ist die Ausgabe des folgenden Programmcodes?

```
short zahlAlsShort = 12;  
byte zahlAlsByte = zahlAlsShort;  
System.out.println(zahlAlsByte);
```



## Aufgabe 1 a)

Was ist die Ausgabe des folgenden Programmcodes?

```
short zahlAlsShort = 12;  
byte zahlAlsByte = zahlAlsShort;  
System.out.println(zahlAlsByte);
```



Type mismatch: cannot convert from short to byte



## Aufgabe 1 a)

Erzwingen der Typkonversion (Abwärtskonversion):

```
short zahlAlsShort = 12;  
byte zahlAlsByte = (byte) zahlAlsShort; //Type Cast  
System.out.println(zahlAlsByte);
```

Eine sogenannte Explizite Typkonversion, um zu bestätigen, dass man sich eines möglichen Wertverlusts bewusst ist.



## Aufgabe 1 b)

Was ist die Ausgabe des folgenden Programmcodes?

```
short zahlAlsShort = 128; //nun mit 128 statt 12  
byte zahlAlsByte = (byte)zahlAlsShort; //Type Cast  
System.out.println(zahlAlsByte);
```



## Aufgabe 1 b)

Was ist die Ausgabe des folgenden Programmcodes?

```
public class Main {  
    public static void main(String[] args) {  
        short zahlAlsShort = 128;  
        byte zahlAlsByte = (byte) zahlAlsShort;  
        System.out.println(zahlAlsByte);  
    }  
}
```



## Aufgabe 1 b)

Was ist die Ausgabe des folgenden Programmcodes?

```
public class Main {  
    public static void main(String[] args) {  
        short zahlAlsShort = 128;  
        byte zahlAlsByte = (byte) zahlAlsShort;  
        System.out.println(zahlAlsByte);  
    }  
}
```

A screenshot of a Java IDE's console window. The title bar shows tabs for 'Problems', '@ Javadoc', 'Declaration', and 'Console'. The console output shows the program has terminated and printed the value -128. The path to the Java runtime is visible: C:\Program Files (x86)\Java\jre7\bin\javaw.exe (31.10.2011 18:54:14).

```
<terminated> Main2 [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (31.10.2011 18:54:14)  
-128
```



## Aufgabe 1 b)

### Vergleich short und byte: Darstellung von 127

short: 

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 2 Byte = 16 Bit

$2^2$   $2^1$   $2^0$

←  $-2^{15}$   $2^{15}-1$  →

byte: 

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

 1 Byte = 8 Bit

$2^2$   $2^1$   $2^0$

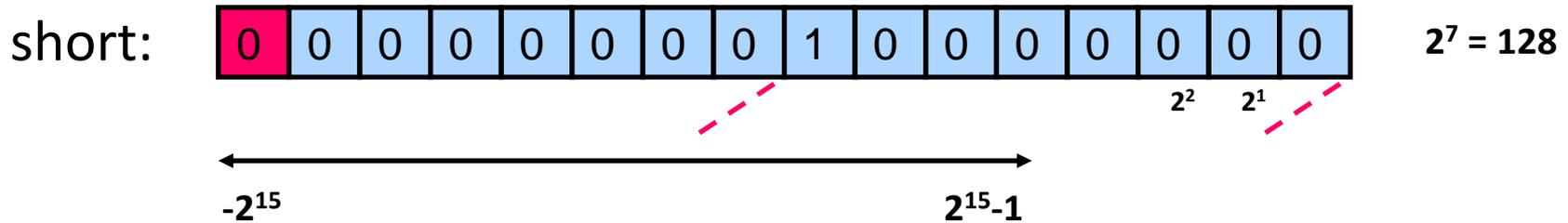
←  $-2^7$   $2^7-1$  →



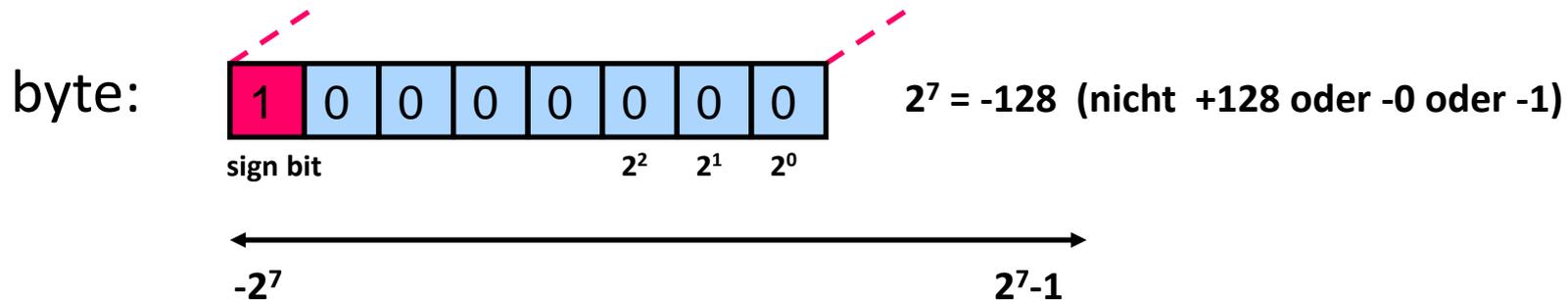


## Aufgabe 1 b)

short Darstellung von 128:



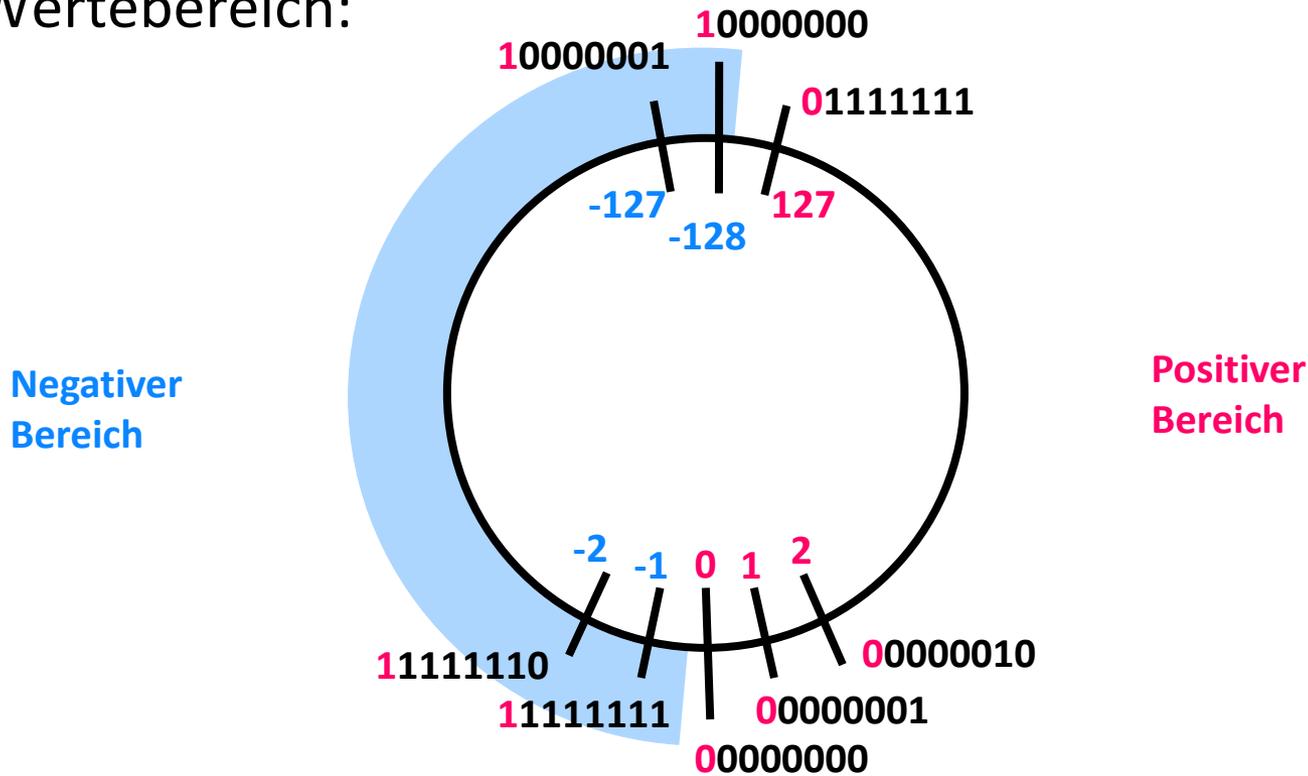
Cast auf byte ergibt folgende Darstellung:





# Aufgabe 1 b)

byte Wertebereich:



Die grösste positive Zahl + 1 ergibt die kleinste negative Zahl.



## Aufgabe 2 a)

Was ist die Ausgabe der folgenden Programmfragmente?

```
byte zahlAlsByte = 12;  
int ZahlalsInt = zahlAlsByte+1;  
System.out.println(ZahlalsInt);
```

```
byte ersteZahl = 12;  
byte zweiteZahl = ersteZahl + 1;  
System.out.println(zweiteZahl);
```



## Aufgabe 2 a)

Was ist die Ausgabe der folgenden Programmfragmente?

```
byte zahlAlsByte = 12;  
int ZahlalsInt = zahlAlsByte+1; //hier kein Type Cast nötig  
System.out.println(ZahlalsInt);
```

```
byte ersteZahl = 12;  
byte zweiteZahl = ersteZahl + 1;  
System.out.println(zweiteZahl);
```



Type mismatch: cannot convert from **int** to byte



## Aufgabe 2 a)

Bei unterschiedlichen Datentypgrößen bei arithmetischen Operatoren werden vor der Anwendung der Operation alle Operanden auf den grössten vorkommenden Typ gebracht (*implizite Aufwärtskonversion* oder *Promotion*).

```
byte zahlAlsByte = 12;  
int ZahlalsInt = zahlAlsByte+1; //Aufwärtskonversion zu int  
System.out.println(ZahlalsInt);  
  
byte ersteZahl = 12;  
byte zweiteZahl = (byte) (ersteZahl + 1); //Korrektur: Type Cast  
System.out.println(zweiteZahl);
```



## Aufgabe 2 b)

Was ist die Ausgabe des folgenden Programmcodes?

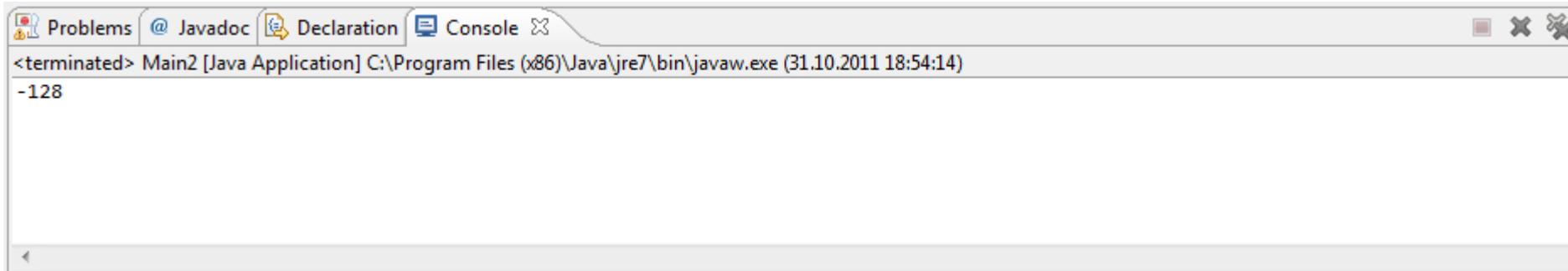
```
byte ersteZahl = 127; //nun mit 127 statt 12  
byte zweiteZahl = (byte)(ersteZahl + 1); //Type Cast  
System.out.println(zweiteZahl);
```



## Aufgabe 2 b)

Was ist die Ausgabe des folgenden Programmcodes?

```
byte ersteZahl = 127; //nun mit 127 statt 12
byte zweiteZahl = (byte) (ersteZahl + 1); //Type Cast
System.out.println(zweiteZahl);
```

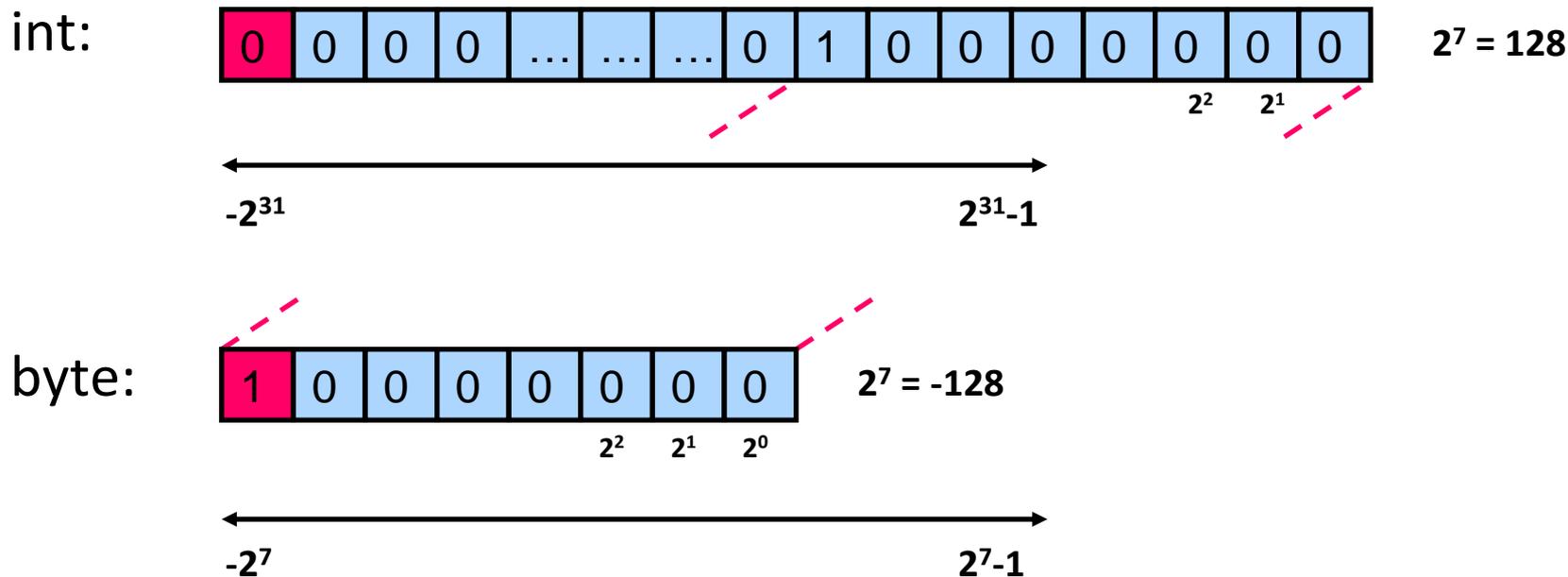




## Aufgabe 2 b)

Vergleich int und byte:

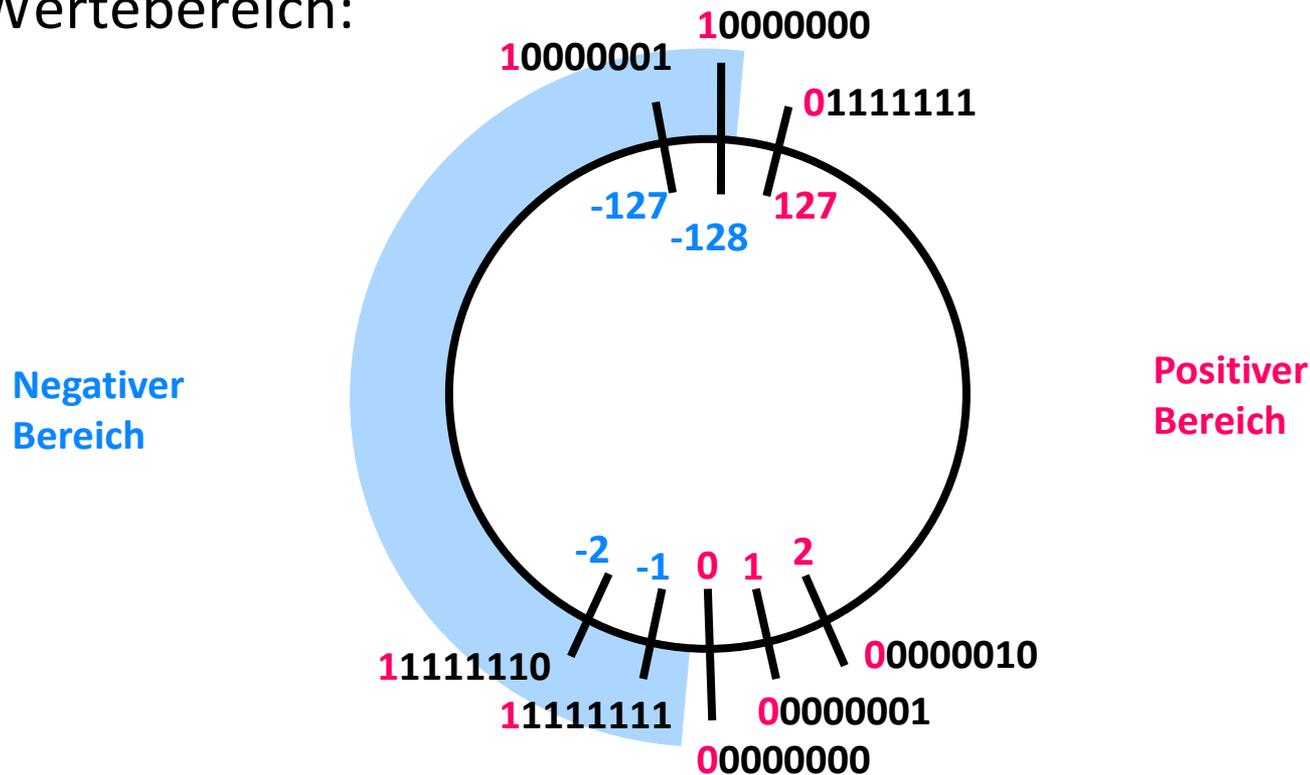
Der Datentyp int hat eine Grösse von 4 Byte = 32 Bit.





# Wiederholung

byte Wertebereich:





## Aufgabe 2 c)

Alle Operanden werden in einem Datentyp gespeichert, dessen Größe für das Ergebnis auf alle Fälle passt.

```
byte zahlAlsByte1 = 1;
```

```
byte zahlAlsByte2 = zahlAlsByte1 + zahlAlsByte1;
```

```
byte zahlAlsByte3 = (byte) (zahlAlsByte1 + zahlAlsByte1);
```

Type mismatch: cannot



## Vorlesung heute:

- Grunddatentypen
- Ausdrücke
- Variablen
- Zustände



## Aufgabe 3

Gegeben seien folgende Variablendeklarationen:

```
double fahrenheit = 40;  
double celsius = 4.44;
```

Welcher Zustand  $\sigma$  wird durch diese Deklarationen beschrieben?



## Aufgabe 3

Gegeben seien folgende Variablendeklarationen:

```
double fahrenheit = 40; //automatische Typkonversion  
double celsius = 4.44;
```

Welcher Zustand  $\sigma$  wird durch diese Deklarationen beschrieben?

```
 $\sigma = [(fahrenheit, 40.0), (celsius, 4.44)]$ 
```



## Aufgabe 3

Werten Sie folgenden Ausdruck bezüglich des Zustands  $\bar{0}$  aus:

```
fahrenheit - 32 * 5/9
```

```
fahrenheit - ((32 * 5) / 9) =ε //von links auswerten
```

```
40.0 - ((32 * 5) / 9) =ε //unter Berücksichtigung
```

```
40.0 - (160/9) =ε //der Präzedenzen
```

```
40.0 - 17 =ε
```

```
23.0
```



## Aufgabe 3

Werten Sie folgenden Ausdruck bezüglich des Zustands  $\bar{0}$  aus:

```
fahrenheit - 32 * 5/9
```

```
fahrenheit - ((32 * 5) / 9) =ε //von links auswerten
```

```
40.0 - ((32 * 5) / 9) =ε //unter Berücksichtigung
```

```
40.0 - (160/9) =ε //der Präzedenzen
```

```
40.0 - 17 =ε
```

```
23.0
```

Ok, Berechnungsidee war falsch, 4.44 sollte in etwa rauskommen!



## Aufgabe 3

Werten Sie folgenden Ausdruck bezüglich des Zustands  $\bar{0}$  aus:

```
(fahrenheit - 32) * 5/9 //explizite Klammersetzung
```

```
((fahrenheit - 32) * 5)/9 =ε //von links auswerten
```

```
(40.0 - 32) * 5)/9 =ε //unter Berücksichtigung
```

```
(8.0 * 5)/9 =ε //der Präzedenzen
```

```
40.0 / 9 =ε //Gleitkommadivision
```

```
4.4444.....
```



## Aufgabe 4

Schreiben Sie ein Java-Programm, das den Ausdruck zur Umrechnung von Fahrenheit in Celsius auswertet und das Ergebnis am Bildschirm zeigt.



## Aufgabe 4

Schreiben Sie ein Java-Programm, das den Ausdruck zur Umrechnung von Fahrenheit in Celsius auswertet und das Ergebnis am Bildschirm zeigt.

```
public class Main{  
    public static void main(String[] args){  
        double fahrenheit = 40;  
        double celsius = (fahrenheit - 32) * 5 / 9;  
        System.out.println(celsius);  
    }  
}
```