

K(e|ö)nnen Sie Java? (4)

Bernd Müller

Fakultät Informatik
Ostfalia
Hochschule Braunschweig/Wolfenbüttel

Wintersemester 2015/16

Was ist die Ausgabe?

```
public class Multicast {  
  
    public static void main(String[] args) {  
        System.out.println((int) (char) (byte) -1);  
    }  
  
}
```

▶ 65535

- ▶ 65535, warum?

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt
- ▶ Cast nach byte ok, alle 8 Bits gesetzt, also -1

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt
- ▶ Cast nach byte ok, alle 8 Bits gesetzt, also -1
- ▶ char ist unsigned, -1 kann nicht konvertiert werden

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt
- ▶ Cast nach byte ok, alle 8 Bits gesetzt, also -1
- ▶ char ist unsigned, -1 kann nicht konvertiert werden
- ▶ Also *Widening and narrowing primitive conversion*, JLS 5.1.4.
byte → int → char

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt
- ▶ Cast nach byte ok, alle 8 Bits gesetzt, also -1
- ▶ char ist unsigned, -1 kann nicht konvertiert werden
- ▶ Also *Widening and narrowing primitive conversion*, JLS 5.1.4.
byte → int → char
- ▶ Widening mit Sign-Extension, falls Original signed.
Zero-Extension für char.

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt
- ▶ Cast nach byte ok, alle 8 Bits gesetzt, also -1
- ▶ char ist unsigned, -1 kann nicht konvertiert werden
- ▶ Also *Widening and narrowing primitive conversion*, JLS 5.1.4.
byte → int → char
- ▶ Widening mit Sign-Extension, falls Original signed.
Zero-Extension für char.
- ▶ byte signed

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt
- ▶ Cast nach byte ok, alle 8 Bits gesetzt, also -1
- ▶ char ist unsigned, -1 kann nicht konvertiert werden
- ▶ Also *Widening and narrowing primitive conversion*, JLS 5.1.4.
byte → int → char
- ▶ Widening mit Sign-Extension, falls Original signed.
Zero-Extension für char.
- ▶ byte signed \implies char alle Bits gesetzt ($2^{16} - 1 = 65535$)

- ▶ 65535, warum?
- ▶ Javas Zahlendarstellung ist Zweierkomplement
- ▶ Daher bei -1 alle Bits gesetzt
- ▶ Cast nach byte ok, alle 8 Bits gesetzt, also -1
- ▶ char ist unsigned, -1 kann nicht konvertiert werden
- ▶ Also *Widening and narrowing primitive conversion*, JLS 5.1.4.
byte → int → char
- ▶ Widening mit Sign-Extension, falls Original signed.
Zero-Extension für char.
- ▶ byte signed ⇒ char alle Bits gesetzt ($2^{16} - 1 = 65535$)
⇒ Zero-Extension: int mit 65535 da char-int auch
Widening mit Zero-Extension