

Globalübung 2



Überblick

- Typanpassungen
- Iteration mit Schleifen
- Bedingte Anweisungen

- Agile Softwareentwicklung
- Einfaches Ratespiel

- Softwarefehler

Überblick

- **Typanpassungen**
- Iteration mit Schleifen
- Bedingte Anweisungen

- Agile Softwareentwicklung
- Einfaches Ratespiel

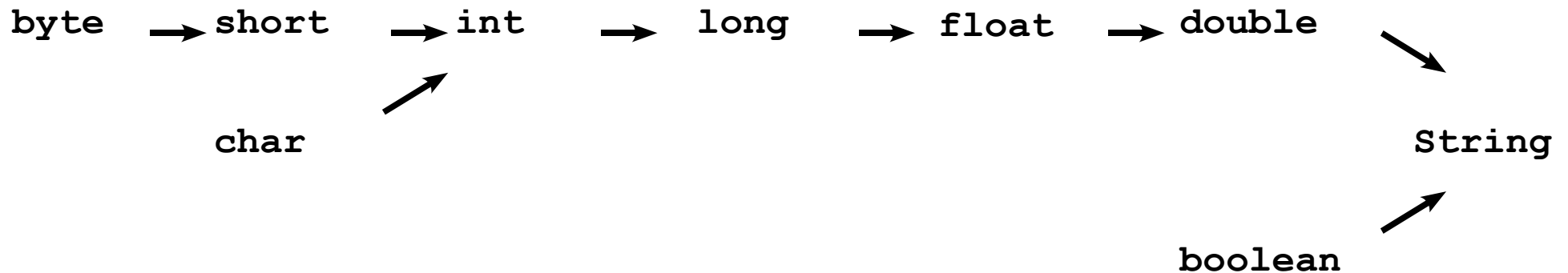
- Softwarefehler

Typanpassungen

- **implizite Typanpassung** falls Wertebereich des Zieltyps ausreichend gross
- Beispiel: `double d = 1.2f;`
- **explizite Typanpassung** falls Wertebereich des Zieltyps zu klein
- Beispiel: `float f = (float) 1.2;`
- Was passiert bei `(float) 1E100` ?

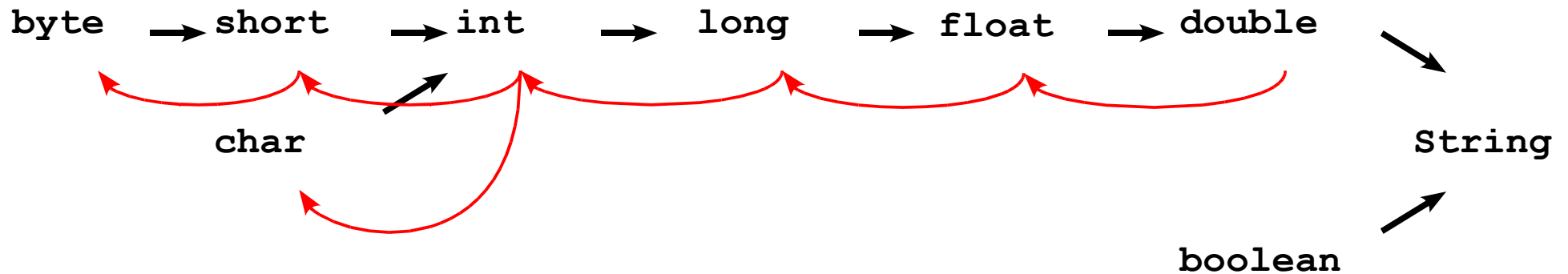
Typanpassungen

- **implizite Typanpassung** falls Wertebereich des Zieltyps ausreichend gross
- **explizite Typanpassung** falls Wertebereich des Zieltyps zu klein



Typanpassungen

- **implizite Typanpassung** falls Wertebereich des Zieltyps ausreichend gross
- **explizite Typanpassung** falls Wertebereich des Zieltyps zu klein



Überblick

- Typanpassungen
- **Iteration mit Schleifen**
- Bedingte Anweisungen

- Agile Softwareentwicklung
- Einfaches Ratespiel

- Softwarefehler

Iteration mit Schleifen

```
do {A} while (B) ;
```

A = Anweisungen

B = Bedingung

Iteration mit Schleifen

```
do {A} while (B) ;
```

```
while (B) {A} ;
```

A = Anweisungen

B = Bedingung

Iteration mit Schleifen

```
do {A} while (B) ;
```

```
while (B) {A} ;
```

```
for (I; B; N) {A}
```

A = Anweisungen

I = Initialisierung

B = Bedingung

N = Zwischenanweisungen

Überblick

- Typanpassungen
- Iteration mit Schleifen
- **Bedingte Anweisungen**

- Agile Softwareentwicklung
- Einfaches Ratespiel

- Softwarefehler

Bedingte Anweisungen

```
if (B) {A1} ;
```

```
if (B) {A1} else {A2} ;
```

B = Bedingung

A1 , **A2** = Anweisungen

Bedingte Anweisungen

```
if (B) {A1};
```

```
if (B) {A1} else {A2};
```

```
switch (V) {  
    case W1: A1; break;  
    case W2: A2; break;  
}
```

B = Bedingung

A1, **A2** = Anweisungen

V = Variable

W1, **W2** = Wert

Überblick

- Typanpassungen
- Iteration mit Schleifen
- Bedingte Anweisungen
- **Agile Softwareentwicklung**
- Einfaches Ratespiel
- Softwarefehler

Agile Softwareentwicklung

Individuen und Interaktionen haben Vorrang vor *Prozessen und Werkzeugen*.

Lauffähige Software hat Vorrang vor *ausgedehnter Dokumentation*.

Zusammenarbeit mit dem Kunden hat Vorrang vor *Vertragsverhandlungen*.

Das **Eingehen auf Änderungen** hat Vorrang vor *striker Planverfolgung*.

Agile Softwareentwicklung

Planung:

- Anwendungsszenarien skizzieren
- Entwicklung in mehrere Schritte einteilen

Entwicklung:

- Programmcode frühzeitig testen
- Paarweise programmieren
- Programmcode überarbeiten
- Nicht zu früh optimieren

Überblick

- Typanpassungen
- Iteration mit Schleifen
- Bedingte Anweisungen

- Agile Softwareentwicklung
- Einfaches Ratespiel

- Softwarefehler

Einfaches Ratespiel

- der Benutzer „denkt“ sich eine Zahl zwischen 1 und 100 aus
- der Rechner rät eine Zahl zwischen 1 und 100
- der Benutzer gibt Feedback
 - Zahl richtig geraten
 - Zahl zu klein
 - Zahl zu groß
- der Rechner hat 7 Versuche

Überblick

- Typanpassungen
- Iteration mit Schleifen
- Bedingte Anweisungen

- Agile Softwareentwicklung
- Einfaches Ratespiel

- Softwarefehler

Softwarefehler

Zwei grundlegende Arten von Softwarefehlern:

- das Programm berechnet etwas Falsches

Softwarefehler

Zwei grundlegende Arten von Softwarefehlern:

- das Programm berechnet etwas Falsches
- das Programm hält nicht an

Softwarefehler

Zwei grundlegende Arten von Softwarefehlern:

- das Programm berechnet etwas Falsches
- das Programm hält nicht an
- je nach Anwendung katastrophale Folgen

Softwarefehler

Zwei grundlegende Arten von Softwarefehlern:

- das Programm berechnet etwas Falsches
- das Programm hält nicht an
- je nach Anwendung katastrophale Folgen

>> Programmverifikation <<<

Carsten Fuhs



Stephan Swiderski

