

## 7. AuD Tafelübung T-C3

Simon Ruderich

8. Dezember 2010

# Garbage-Collector

## Beispiel

```
class A {  
    Integer x;  
    A(Integer x) {  
        this.x = x;  
    }  
}
```

```
Integer x = 42; // neues Integer Objekt!  
A a = new A(x);  
// ...  
x = null;  
// ...  
a = null;
```

# Rekursion

## Definition

- Funktion ruft sich selbst auf um Problem zu lösen
- nutzt (meist) keinen externen Zustand (z. B. Instanzvariablen)
- besteht (immer) aus zwei Teilen
  - 1 Abbruchbedingung
  - 2 Rekursionsschritt
- jede iterative Aufgabe lässt sich rekursiv lösen (und umgekehrt)

# Beispiel

Fakultät  $n! = \prod_{i=1}^n i$  (iterativ)

```
public long fakultaetIterativ(long n) {  
    long result = 1;  
    while (n > 1) {  
        result *= n--;  
    }  
    return result;  
}
```

# Beispiel

Fakultät  $n! = \prod_{i=1}^n i$

$$f(n) = \begin{cases} 1 & \text{für } n \leq 1, \\ n \cdot f(n-1) & \text{sonst.} \end{cases}$$

# Beispiel

Fakultät  $n! = \prod_{i=1}^n i$

$$f(n) = \begin{cases} 1 & \text{für } n \leq 1, \\ n \cdot f(n-1) & \text{sonst.} \end{cases}$$

Fakultät (rekursiv)

```
public long fakultaetRekursiv(long n) {  
    if (n <= 1) {  
        return 1;  
    }  
    return n * fakultaetRekursiv(n - 1);  
}
```

# Beispiel

## Endlosschleife

```
public void endlosIterativ () {  
    while (true) {  
        // ...  
    }  
}  
public void endlosRekursiv () {  
    // ...  
    endlosRekursiv ();  
}
```

# Rekursionstypen

## Rekursionstypen

- linear** ein Rekursionsaufruf pro Aufruf (z. B. Fakultät)
- repetitiv** lineare Rekursion, rekursiver Aufruf als letztes
- kaskadenartig** mehrere Rekursionsaufrufe pro Aufruf  
(z. B. Fibonacci)
- verschränkt** Funktionen rufen sich zyklisch gegenseitig auf
- verschachtelt** rekursive Aufrufe in Parametern  
(z. B. Ackermann-Funktion)

# Beispiel: repetitive Rekursion

## Fakultät (lineare Rekursion)

```
public long fakultaetRekursiv(long n) {  
    if (n <= 1) {  
        return 1;  
    } else {  
        return n * fakultaetRekursiv(n - 1);  
    }  
}
```

# Beispiel: repetitive Rekursion

## Fakultät (repetitive Rekursion)

```
public long fakultaetRepetitiv(long n) {  
    return fakultaetRepetitiv(n, 1);  
}  
private long fakultaetRepetitiv(long n, long r) {  
    if (n <= 1) {  
        return r;  
    }  
    return fakultaetRepetitiv(n - 1, r * n);  
}
```

# Beispiel: kaskadenartige Rekursion

Definition  $g(n)$

$$g(n) = \begin{cases} -4 & \text{für } n \leq 0, \\ 2 & \text{für } n = 1, \\ 5 * g(n - 1) + 3 * g(n - 2) & \text{sonst.} \end{cases}$$

# Beispiel: kaskadenartige Rekursion

## Definition $g(n)$

$$g(n) = \begin{cases} -4 & \text{für } n \leq 0, \\ 2 & \text{für } n = 1, \\ 5 * g(n - 1) + 3 * g(n - 2) & \text{sonst.} \end{cases}$$

## Beispiel $g(n)$

```
public long g(long n) {  
    if (n <= 0) {  
        return -4;  
    } else if (n == 1) {  
        return 2;  
    }  
    return 5 * g(n - 1) + 3 * g(n - 2);  
}
```

# Aufgaben

Kettenbrüche, Flächenfüllen,  
Glatte Kurven, ...

# Fragen

Fragen?