Programmierkurs

Java – Programmierkurs
Philipp Herzig, B.Sc.
2009

Rekursion

Thinking iterativ is human,
Thinking recursive devine

Iterativ zu denken ist menschlich,
Rekursiv zu denken göttlich

:-)

Rekursion

- Bisher haben wir iterativ gedacht!
- Das fällt leicht, weil alle Dinge auf dieser Welt iterativ gemacht werden (Anleitungen (denkt mal an euer LEGO von früher) usw.)
- Daher fällt Rekursion erfahrungsgemäß schwer (am Anfang). Man kann aber oft tolle Sachen damit machen. Die iterativ schwer bis kaum zu lösen sind und wenn dann tierisch umständlich.

Rekursion

- Prinzipiell ist Rekursion einfach erklärt.
- Im Körper einer Funktion X wird die Funktion X wieder aufgerufen.
- z.B.

```
void X() {
    return X();
}
```

- Das ist schon Rekursion!
- **Was ist daran schlimm???**

Rekursion

- **Genau, es ist quasi ein Endlosaufruf!**
- Deswegen benötigen wir ein zweites Element was wir **Abbruchbedingung** nennen!
- z.B.:

```
void X(int zahl) {
    System.out.println(zahl);
    if(zahl==100)     //annahme: eingegebene zahl<100
        return;
    else
        return X(zahl+1);
}
```

- **Was macht das? Wie würde das iterativ aussehen?**

Rekursion – Summe bilden

```
int X(int startzahl) {
    if(startzahl==100)
        return 100;
    else
        return startzahl + X(startzahl+1);
}
```

- **Wie sieht das ganze iterativ aus?!**

Summe rückwärts

- Wie können wir eine Summe von einer startzahl aus rückwarts erstellen, natülrich rekursiv ;-) ?

Rekursion

- Ohne es zu merken habt ihr schon alles an der Hand, um eine hübsche Funktion zu implementiern!
- Die Fakultät
- **Wie lautet die Bildungsvorschrift für die Fakultät?**
- **Wie können wir genau das rekursiv implementieren?**

Rekursion

- Eine interessante Zahlenfolge ist die Fibonacci Zahlenfolge.
- 1, 1, 2, 3, 5, 8, 13, 21, …
- **Wer erkennt das Muster?**

Rekursion

- offensichtlich gilt folgendes:

  Element(n) = Element(n-1) + Element(n-2);

- Wer möchte das mal rekursiv in einer Funktion beschreiben? An der Tafel... ;-)

Rekursion

- Rekursiver Abstieg
- Rekursiver Aufstieg
- Meiner Meinung verkompliziert das folgende Bild das Verständnis. Ich möchte es aber aus Gründen der Vollständigkeit angeben!

Zusammenfassung

- Rekursion besteht aus 3 Teilen
  - Funktion X wird in X aufgerufen (Rekursiver Abstieg)
  - Abbruchbedingung (Ende des rekursiven Abstiegs)
  - Rückgabe (Rekursiver Aufstieg)
- Rekursion als Denkensart (Paradigma) kann helfen Probleme zu lösen, welche mit Iteration schwer oder umständlich lösbar sind.

Until next week....

Vielen Dank für die Aufmerksamkeit
Viel Spaß!!!

Nächste Woche schinken wir unseren Turtle rekursiv durch die Kante :-)