

Informatik I Übung, Woche 46

Giuseppe Accaputo

12. November, 2015

Plan für heute

1. Lernziele für die heutige Übungsstunde
2. Vorbesprechung Übung 9
3. Nachbesprechung Übung 8

Lernziele

- ▶ Arrays von Datensätzen
- ▶ Rekursion

Rekursion: Definition

Eine Prozedur, die sich selber aufruft, heisst *rekursiv*.

“In order to understand recursion, one must first understand recursion”

Rekursion: Aufbau einer rekursiven Funktion

Jede rekursive Prozedur besteht aus den folgenden 3 Bestandteilen:

1. Abbruchbedingung (bestimmt wann die Prozedur fertig ist)
2. Schritt in Richtung Erfüllung der Abbruchbedingung (wir vereinfachen das Problem)
3. Rekursiver Aufruf (die Prozedur sollte sich selbst aufrufen)

Beispiel einer rekursiven Funktion: Fakultät (1/3)

Die Fakultät ist wie folgt definiert:

$$\text{Fakultaet}(n) = n * (n-1) * (n-2) * \dots * 2$$

Für den Fall $n = 0$ erhalten wir:

$$\text{Fakultaet}(0) = 1$$

Die Funktion `Fakultaet` kann auch rekursiv definiert werden:

$$\text{Fakultaet}(n) = n * \text{Fakultaet}(n-1)$$

Frage: Wie sieht die Abbruchbedingung aus für die rekursive Implementierung von `Fakultaet`?

Beispiel einer rekursiven Funktion: Fakultät (2/3)

Die Fakultät ist wie folgt definiert:

$$\text{Fakultaet}(n) = n * (n-1) * (n-2) * \dots * 2$$

Für den Fall $n = 0$ erhalten wir:

$$\text{Fakultaet}(0) = 1$$

Die Funktion `Fakultaet` kann auch rekursiv definiert werden:

$$\text{Fakultaet}(n) = n * \text{Fakultaet}(n-1)$$

Frage: Wie sieht die Abbruchbedingung aus für die rekursive Implementierung von `Fakultaet`?

Antwort: `IF n = 0 THEN Fakultaet := 1;`

Beispiel einer rekursiven Funktion: Fakultät (3/3)

```
FUNCTION  Fakultaet (n : LONGINT): LONGINT;  
BEGIN  
  IF n = 0 THEN Fakultaet := 1  
  ELSE  
    BEGIN  
      Fakultaet := n * Fakultaet(n-1);  
    END;  
END;
```

```
Writeln('3! = ', Fakultaet(3));
```

```
Konsole: 3! = 6
```


Visualisierung eines Aufrufs einer rekursiven Funktion (1/9)

Wie sieht nun beispielsweise die Verarbeitung des Aufrufs

```
Fakultaet (3)
```

genau aus?

⇒ Ihr könnt euch einen Stapel vorstellen, auf welchem alle einzelnen rekursiven Funktionsaufrufe der Reihe nach reingeworfen werden. Bei Erreichung der Abbruchbedingung wird der Stapel mit den Funktionsaufrufen abgearbeitet.

Visualisierung eines Aufrufs einer rekursiven Funktion (2/9)

Funktionsaufruf:

```
f := Fakultaeet(3)
```

Stapel:

Fakultaeet(3)

Visualisierung eines Aufrufs einer rekursiven Funktion (3/9)

Stapel:

Fakultaet(2)
Fakultaet(3)

Visualisierung eines Aufrufs einer rekursiven Funktion (4/9)

Stapel:

Fakultaet(1)
Fakultaet(2)
Fakultaet(3)

Visualisierung eines Aufrufs einer rekursiven Funktion (5/9)

Stapel:

Fakultaet(0)
Fakultaet(1)
Fakultaet(2)
Fakultaet(3)

Die **Abbruchbedingung** wurde nun mit dem Aufruf von `Fakultaet(0)` erreicht, d.h. im nächsten Schritt beginnt die Abarbeitung des Stapels (von oben nach unten, begonnen bei `Fakultaet(0)`).

Visualisierung eines Aufrufs einer rekursiven Funktion (6/9)

Stapel:

Fakultaet(1)
Fakultaet(2)
Fakultaet(3)

Abarbeitung:

Fakultaet(0)	Fakultaet := 1

Visualisierung eines Aufrufs einer rekursiven Funktion (7/9)

Stapel:

Fakultaet(2)
Fakultaet(3)

Abarbeitung:

Fakultaet(1)	Fakultaet := 1 * F..(0) := 1*1
Fakultaet(0)	Fakultaet := 1

Visualisierung eines Aufrufs einer rekursiven Funktion (8/9)

Stapel:

Fakultaet(3)

Abarbeitung:

Fakultaet(2)	Fakultaet := 2 * F..(1) := 2*1*1
Fakultaet(1)	Fakultaet := 1 * F..(0) := 1*1
Fakultaet(0)	Fakultaet := 1

Visualisierung eines Aufrufs einer rekursiven Funktion (9/9)

Stapel:

Abarbeitung:

Fakultaet(3)	Fakultaet := 3 * F..(2) := 3*2*1*1
Fakultaet(2)	Fakultaet := 2 * F..(1) := 2*1*1
Fakultaet(1)	Fakultaet := 1 * F..(0) := 1*1
Fakultaet(0)	Fakultaet := 1

Resultat: n := 6

Aufgabe 9.3

- ▶ Die Implementation der `Split` Funktion findet ihr auf dem Slide 7-29: Ändert die Implementation so ab, dass sie für **INTEGER** funktioniert.
- ▶ Die `Split(arr, pivot)` Funktion gibt ein Index `s`, sodass gilt: `arr[Low..s-1] <= pivot < arr[s..High]`
 - ▶ `s` ist also der Startindex des Teilarrays, welches die Elemente enthält, welche grösser als das Pivot sind
- ▶ Unbedingt Tipps beachten!