

# Technische Informatik II

## Rechnerarchitektur

### Hilfe für Übungsblatt 3, Aufgabe 6

Pseudocode in anderen Sprachen

Matthias Dräger

E-Mail: [mdraeger@mi.fu-berlin.de](mailto:mdraeger@mi.fu-berlin.de)

www: [www.matthias-draeger.info/lehre/sose2010ti2/](http://www.matthias-draeger.info/lehre/sose2010ti2/)  
[tinyurl.com/sose2010ti2](http://tinyurl.com/sose2010ti2)

# Übungsblatt 3, Aufgabe 6

## Aufgabe 6: MMIX – Unterprogramme (5+5=10)

Informieren Sie sich über die Erstellung von Unterprogrammen in MMIX.

- a) Implementieren Sie ein Unterprogramm für die Berechnung der Fakultätsfunktion  $n! = 1 \cdot 2 \cdot \dots \cdot n$  und geben Sie ein Hauptprogramm an, vom dem das Unterprogramm aufgerufen wird.
- b) Die rekursive Berechnung des Binomialkoeffizienten  $\binom{n}{k}$  kann mittels folgender Funktion durchgeführt werden:

```
// rekursive Berechnung des Binomialkoeffizienten (Pseudocode)
function binom (n,k : integer) : integer;
begin
    if(k = 0 OR k = n) then binom := 1
    else if k = 1 then binom := n
    else binom := binom(n-1,k) + binom(n-1,k-1)
end;
```

Schreiben Sie ein vollständiges MMIX Assemblerprogramm, welches mittels eines rekursiven Unterprogramms den Binomialkoeffizient  $\binom{n}{k}$  berechnet.

Hinweis: Benutzen Sie PUSHJ und POP.

# Pseudocode in anderen Sprachen (1)

## Pseudocode

```
// rekursive Berechnung des Binomialkoeffizienten
function binom (n,k : integer) : integer;
begin
    if(k = 0 OR k = n) then binom := 1
    else if k = 1 then binom := n
    else binom := binom(n-1,k) + binom(n-1,k-1)
end;
```

## Haskell

```
-- Berechnung des Binomialkoeffizienten (rekursiv)

binom :: Integer -> Integer -> Integer
binom n k
    | (n < 0 || k < 0) = error ("Fehler: Parameter muessen positiv sein!")
    | (k == 0 || k == n) = 1
    | (k == 1) = n
    | otherwise = (binom (n-1) k) + (binom (n-1) (k-1))
```

# Pseudocode in anderen Sprachen (2)

## Pseudocode

```
// rekursive Berechnung des
Binomialkoeffizienten
function binom (n,k : integer) : integer;
begin
    if(k = 0 OR k = n) then binom := 1
    else if k = 1 then binom := n
    else binom := binom(n-1,k) + binom(n-1,k-1)
end;
```

## Python

```
import os

#Binomialkoeffizient
def binom(n,k) :
    if(k == 0 or k == n): return 1
    elif(k == 1): return n
    else: return binom(n-1,k) + binom(n-1,k-1)

#Testfunktion, die binom(n,k) aufruft
def test() :
    n,k = 49,6
    print '(49 ueber 6) = ',
    print binom(n,k)
    os.system('pause')

#Ausfuehrung von Test
test()
```

## Pseudocode in anderen Sprachen (3)

### Pseudocode

```
// rekursive Berechnung des
Binomialkoeffizienten
function binom (n,k : integer) : integer;
begin
    if(k = 0 OR k = n) then binom := 1
    else if k = 1 then binom := n
    else binom := binom(n-1,k) + binom(n-1,k-1)
end;
```

### PHP

```
<?php

// Rekursive Berechnung von Binomialkoeffizienten (n ueber k)
function binom($n,$k){
    // 1.Rekursionsanker
    if($k == 0 || $k == $n) return 1;
    // 2.Rekursionsanker
    elseif($k == 1) return $n;
    // Rekursionsschritt
    else return binom($n-1,$k) + binom($n-1,$k-1);
}

echo "(49 ueber 6) = " . binom(49,6);

?>
```

# Pseudocode in anderen Sprachen (4)

## Pseudocode

```
// rekursive Berechnung des
Binomialkoeffizienten
function binom (n,k : integer) : integer;
begin
    if(k = 0 OR k = n) then binom := 1
    else if k = 1 then binom := n
    else binom := binom(n-1,k) + binom(n-1,k-1)
end;
```

## Java

```
public class Ueb3_Aufg6 {
    public static void main(String[] args) {
        System.out.println("(49 über 6) = " + binom(49, 6));
    }
    /**
     * Rekursive Berechnung des Binomialkoeffizienten (n über k)
     */
    public static int binom(int n, int k){
        // 1.Rekursionsanker
        if(k == 0 || k == n) return 1;
        // 2.Rekursionsanker
        else    if(k == 1) return n;
               // Rekursionsschritt
               else return binom(n-1,k) + binom(n-1,k-1);
    }
}
```