

## Was ist Rekursion?

Rekursion lässt sich auch mit Parametern und Rückgabewerten durchführen. Bei den Parametern ist zu beachten, dass bei jedem rekursiven Aufruf einer Methode mit einem Parameter eine neue, aber gleichnamige lokale Variable angelegt wird.

### Eine rekursive Methode mit Parameter ohne Rückgabewert

```
class WasIstDas
{
    public void p(int n)
    {   if (n>0) p(n-1); // Rekursion!
        Out.println(n);
    }

    public void action()
    {   Out.println("Die Ausgabe von p(4): ");
        p(4);
    }
} // class
```

Bei jedem Aufruf der Methode wird im Prinzip eine weitere Kopie der Methode mit eigenen lokalen Variablen angelegt. Stelle dies für das Beispiel graphisch dar. *Rekursion* bedeutet so viel wie „Zurückgehen“. Erläutere dies mit Hilfe der Graphik.

### Eine rekursive Methode mit Parametern und Rückgabewert

```
double vielfaches (double x, int n)
{   if (n==1)
        { return x; }
    else { return x + vielfaches(x, n-1); }
}
```

Der Parameter `x` ist vom Typ `double`, also eine rationale Zahl bzw. Dezimalzahl. Dieser Parameter wird bei jedem rekursiven Aufruf weiter gereicht.

## Aufgaben

1. Schreibe zur Methode `vielfaches` ein Hauptprogramm bzw. eine vollständige Klasse.
2. Schreibe eine Methode `potenz`, die die  $n$ -te Potenz einer Zahl  $x$  rekursiv berechnet.
3. Schreibe eine Methode `fakultaet`, die die Fakultät einer natürlichen Zahl berechnet. Mathematisch:  $n! = (n - 1)! \cdot n$ ;  $0! = 1$ .
4. Schreibe eine Methode `fibonacci`, die die  $n$ -te FIBONACCI-Zahl  $f(n)$  wie folgt berechnet:

$$f(1) = f(2) = 1; \quad f(n) = f(n - 2) + f(n - 1) \text{ für } n > 2; n \in \mathbb{N}.$$

5. Die Binomialkoeffizienten (PASCAL-Dreieck) sind rekursiv wie folgt definiert:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}; \quad \binom{n}{0} = 1; \quad \binom{n}{k} = 0 \text{ für } k > n$$

Schreibe damit ein Programm, das das PASCAL-Dreieck darstellt.

6. Was ist der Funktionswert `makkaroni(5)`?

```
int makkaroni(int nudel)
{ if (nudel==1)
  { return 1; }
  else { if (nudel==2)
        { return 2; }
        else { return 2*makkaroni(nudel-2)+makkaroni(nudel-1);}
        }
}
```

7. Was ist der Funktionswert `brat(5)`?

```
int brat(int kartoffel)
{ if (kartoffel==1)
  { return 1; }
  else { if (kartoffel==2)
        { return 3; }
        else { return 3*brat(kartoffel-2)+2*brat(kartoffel-1);}
        }
}
```

8. Was passiert hier?

```
class WasIstDiesRekursiv
{
  public int p(int x)
  { if (x==0) return 1;
    else { int y = p(x-1);
          Out.println(x+y);
          return y+1;
        }
  }

  public void action()
  { Out.println("Die Ausgabe von p(3): ");
    int wert=p(3);
    Out.println("Funktionswert ist " + wert);
  } // main
} // class
```

9. Löse das Problem der *Türme von Hanoi* rekursiv.