AP 11: Rekursive Methoden

Bei nicht-linearen rekursiven Datenstrukturen (z.B. Baumstrukturen) vereinfachen rekursive Methoden häufig den Programmcode. Bei rekursiven Methoden handelt es sich um Methoden, die innerhalb des Methodenrumpfes die Methode erneut aufrufen.
Ein Beispiel dafür wäre folgende Methode, welche die Summe von 1 bis einer übergebenen Zahl berechnet:

```
public int summe1BisN(int n){
    if(n<=0){ //Abbruchbedingung
        return 0;
    }
    return n + summe1BisN(n-1); //rekursiver Aufruf
}
```

Wird beispielsweise die Methode mit der Zahl 3 aufgerufen, würde das wie folgt ausgewertet werden:
`summe1BisN(3)` $= 3 +$ `summe1BisN(2)` $= 3 + 2 +$ `summe1BisN(1)` $= 3 + 2 + 1 +$ summe1BisN$(0) = 3 + 2 + 1 + 0 = 6$

Aufgaben

1. Implementieren Sie eine statische Methode `int fakultaet(int n)`, die rekursiv die Fakultät einer Zahl bestimmt.
   Hinweis: `fakultaet(4)` $= 4 \cdot 3 \cdot 2 \cdot 1$

2. Zeichenketten
   (a) Beschreiben Sie die Funktionalität der Methode `funktion1`. Ermitteln Sie beispielsweise die Funktionswerte für die Argumente „HANNAH“, „SARAH“ und „OTTO“.

```
public boolean funktion1(String s){
  if(s.length() == 0 || s.length() == 1){
    return true;
  }
  return (s.charAt(0) == s.charAt(s.length()-1))
    && funktion1(s.substring(1, s.length()-1));
}
```

(b) Implementieren Sie eine weitere rekursive Methode
`public static String umdrehen(String s)`, welche den String `s` umdreht, also beispielsweise beim Argument „REGAL“ das Wort „LAGER“ zurückgibt.

(c) Wie könnte man mithilfe der Methode `umdrehen` die in Teilaufgabe (a) dargestellte Methode vereinfachen?

3. Ein Klassiker der rekursiven Programmierung sind die Fibonacci-Zahlen. Dies ist eine Zahlenreihe der folgenden Form: 1, 1, 2, 3, 5, 8, 13, 21,...
Diese Zahlenreihe beginnt mit den beiden Zahlen 1 (**fib(1)**) und 1 (**fib(2)**). Die weiteren Zahlen setzen sich aus der Summe der beiden vorhergehenden Fibonacci-Zahlen zusammen. Dies wäre z.B. fib(3) = fib(2) + fib(1) = 1 + 1 = 2, fib(4) = fib(3) + fib(2) = 2 + 1 = 3 oder fib(5) = fib(4) + fib(3) = 3 + 2 = 5.

(a) Implementieren Sie eine rekursive Methode
`public static int fib(int n)`, welche die n-te Fibonacci-Zahl rekursiv berechnet und zurückgibt.

(b) Implementieren Sie eine **iterative** Methode
`public static int fibtIt(int n)`, welche die n-te Fibonacci-Zahl iterativ berechnet und zurückgibt. Vergleichen Sie diese Methode mit der rekursiven Methode aus Teilaufgabe (a).

4. Gegeben ist für eine ganze Zahl n folgende Definition einer rekursiven Funktion f:
$$f(\mathrm{n}) = \begin{cases} f(f(\mathrm{n}+11)), & \text{if n} <= 100 \\ \mathrm{n}-10, & \text{sonst} \end{cases}$$
Implementieren sie diese Funktion als statische Methode und testen Sie Ihr Programm, indem Sie die Methode mit unterschiedlichen Zahlen aufrufen. Was fällt Ihnen auf?