

## Lösung Rekursion - Teil II

### Aufgabe 2: Zeichenkette rekursiv invertieren

Schreibe eine rekursive Variante der iterativen Methode. Dabei soll die Methode nichts auf den Bildschirm schreiben, sondern nur die Zeichenkette invertieren. Überlege dabei zuerst, wie man das Problem so zergliedert, dass man einen kleinen Teil selbst erledigt und den Rest delegiert. Vergiss die Abbruchbedingung(en) nicht.

Lade dir die Vorlage zu dieser Aufgabe vom Server und fülle die markierten Stellen.

#### Vorlage:

```
/**
 * Zeichenkette invertieren (iterativ und rekursiv)
 */

public class Invertieren {

    public static String invIterativ(String s){
        String ausgabe = "";

        for (int i=s.length()-1;i>=0;i--){           // Iteration
            ausgabe = ausgabe+s.charAt(i);
        }
        return ausgabe;
    }

    public static String invRekursiv(String s){

        // FÜLLEN

    }

    public static void main(String[] args) {

        System.out.println("Dieses Programm invertiert eine Zeichenkette.\n");
        String s = IOTools.readLine("Geben Sie eine Zeichenkette ein: ");

        if(s.length()>0){
            System.out.println("\nInvertierung Iterativ: "+invIterativ(s));
            System.out.println("\nInvertierung Rekursiv: "+invRekursiv(s));
        }else{
            System.out.println("\nFalsche Eingabe!");
        }
    }
}
```

**Lösung:**

```
/**
 * Zeichenkette invertieren (iterativ und rekursiv)
 */

public class Invertieren {

    public static String invIterativ(String s){
        String ausgabe = "";

        for (int i=s.length()-1;i>=0;i--){
            ausgabe = ausgabe+s.charAt(i);
        }
        return ausgabe;
    }

    public static String invRekursiv(String s){
        switch(s.length()){
            case 1: return s; // Abbruch
            case 2: return (""+s.charAt(1)+s.charAt(0)); // tauschen
            default:
                return (
                    s.charAt(s.length()-1)
                    + invRekursiv(s.substring(1,s.length()-1))
                    + s.charAt(0)
                ); // Rekursion
        }
    }

    public static void main(String[] args) {

        System.out.println("Dieses Programm invertiert eine Zeichenkette.\n");
        String s = IOTools.readLine("Geben Sie eine Zeichenkette ein: ");

        if(s.length()>0){
            System.out.println("\nInvertierung Iterativ: "+invIterativ(s));
            System.out.println("\nInvertierung Rekursiv: "+invRekursiv(s));
        }else{
            System.out.println("\nFalsche Eingabe!");
        }
    }
}
```

**Ausgabe:**

```
Dieses Programm invertiert eine Zeichenkette.
Geben Sie eine Zeichenkette ein: Hallo Welt!
Invertierung Iterativ: !tleW ollaH
Invertierung Rekursiv: !tleW ollaH
```

**Aufgabe 3: Fibonacci-Folge (mehrfache Rekursion)**

Die Fibonacci-Folge ist eine mathematische Zahlenfolge mit besonderen Eigenschaften. Die ersten Werte der Fibonacci-Folge lauten: 1, 1, 2, 3, 5, 8, 13, 21 ... Dabei ergibt sich ein neuer Wert stets aus der Summe der beiden vorhergehenden Werte. Der nächste Wert wäre daher 34.

Entwickle nun ein iteratives und ein rekursives Programm zur Berechnung des  $n$ -ten Folgeglieds der Fibonacci-Folge:

Mathematische Definition:  $a_1 := 1$   
 $a_2 := 1$   
 $a_n := a_{n-1} + a_{n-2}$ , für  $n \geq 3$

Lade dir dazu die Vorlage vom Server und fülle die markierten Stellen.

**Vorlage:**

```
/**
 * Fibonacci (iterativ und rekursiv)
 */

public class Fibonacci {

    public static long fibIterativ(int n){
        long fib_n_minus_2 = 1;
        long fib_n_minus_1 = 1;
        long fib_n = 0;

        switch(n){

            // FÜLLEN

        }
    }

    public static long fibRekursiv(int n){
        switch(n){

            // FÜLLEN

        }
    }

    public static void main(String[] args) {

        System.out.println("Dieses Programm berechnet die n-te Fibonacci-Zahl.\n");
        int n = IOTools.readInt("Geben Sie die Zahl n ein: ");
        System.out.println();

        if(n>0){
            System.out.println("Die "+n+"-te Fibonacci-Zahl lautet (iterativ): "
                               +fibIterativ(n));
            System.out.println("Die "+n+"-te Fibonacci-Zahl lautet (rekursiv): "
                               +fibRekursiv(n));
        }else{
            System.out.println("\nFalsche Eingabe!");
        }
    }
}
```

**Lösung:**

```
/**
 * Fibonacci (iterativ und rekursiv)
 */

public class Fibonacci {

    public static long fibIterativ(int n){
        long fib_n_minus_2 = 1;
        long fib_n_minus_1 = 1;
        long fib_n = 0;

        switch(n){
            case 1: return 1;
            case 2: return 1;
            default:
                for (int i=0;i<n-2;i++) {
                    fib_n = fib_n_minus_1 + fib_n_minus_2;
                    fib_n_minus_2 = fib_n_minus_1;
                    fib_n_minus_1 = fib_n;
                }
                return fib_n;
        }
    }

    public static long fibRekursiv(int n){
        switch(n){
            case 1: return 1; // Abbruch
            case 2: return 1; // Abbruch
            default: return fibRekursiv(n-1) + fibRekursiv(n-2); // Rekursion
        }
    }

    public static void main(String[] args) {

        System.out.println("Dieses Programm berechnet die n-te Fibonacci-Zahl.\n");
        int n = IOTools.readInt("Geben Sie die Zahl n ein: ");
        System.out.println();

        if(n>0){
            System.out.println("Die "+n+"-te Fibonacci-Zahl lautet (iterativ): "
                               +fibIterativ(n));
            System.out.println("Die "+n+"-te Fibonacci-Zahl lautet (rekursiv): "
                               +fibRekursiv(n));
        }else{
            System.out.println("\nFalsche Eingabe!");
        }
    }
}
```

**Ausgabe:**

```
Dieses Programm berechnet die n-te Fibonacci-Zahl.
Geben Sie die Zahl n ein: 20
Die 20-te Fibonacci-Zahl lautet (iterativ): 6765
Die 20-te Fibonacci-Zahl lautet (rekursiv): 6765
```