

Rekursion Lösungen

Aufgabe 1:

Finden Sie heraus, was die folgende Applikation auf die Konsole schreibt:¹

```
Beispiel für eine Rekursion:  
24  
Value ist nun 10:  
3628800
```

"Fakultät"

```
1 package ch.bbw.pr.fakultaet;  
2  
3 /**  
4  * Fakultae Class  
5  * @author Peter Rutschmann  
6  * @version 10.01.2018  
7  */  
8 public class Fakultae {  
9  
10     public int byLoop(int value)  
11     {  
12         int ergebnis = 1;  
13         for(int i=1; i<=value;i++)  
14         {  
15             ergebnis = ergebnis * i;  
16         }  
17         return ergebnis;  
18     }  
19  
20     public int byRekursion(int value)  
21     {  
22         if (value != 0)  
23         {  
24             return (value * byRekursion(value-1));  
25         }  
26         return 1;  
27     }  
28 }
```

```
byRekursion(4)           return 6 * 4 = 24  
  byRekursion(3)         return 2 * 3  
    byRekursion(2)       return 1 * 2  
      byRekursion(1)     return 1
```

¹ 24
10_5_RekursionAusprobierenLSG.docx / 04.03.2024
bbw Stefan Düetsch / Rinaldo Lanza / angepasst P. Rutschmann

" String umdrehen "

```
1 package ch.bbw.pr.stringdrehen;
2
3 import java.util.Scanner;
4
5 /**
6  * String drehen Applikation
7  * @author Peter Rutschmann
8  * @version 11.01.2018
9  */
10 public class Application {
11
12     public static void main(String[] args) {
13         Scanner myScanner = new Scanner(System.in);
14         String eingabe = "";
15         FlippString flipper = new FlippString();
16
17         System.out.println("String drehen:");
18         System.out.print("Bitte geben Sie einen String ein: ");
19         eingabe = myScanner.nextLine();
20         System.out.println();
21         System.out.println("Umgedreht: " + flipper.flipp(eingabe));
22         myScanner.close();
23     }
24
25 }
```

```
8 public class FlippString {
9
10     public String flipp(String text)
11     {
12         /* Aufruf mit halber Laenge des Strings
13          * Korrektur um -1, da Nummerierung 0..(length-1)
14          */
15         return rekursiveFlipp(text, (text.length()/2-1));
16     }
17
18     public String rekursiveFlipp(String text, int value){
19         System.out.println("FlippString.rekursiveFlipp: " + text + " " + value);
20         if (value == -1) return text;
21         text = exchangeChar(text, value, indexB: text.length()-value-1);
22         return rekursiveFlipp(text, value: value-1);
23     }
24
25     @private String exchangeChar(String text, int indexA, int indexB)
26     {
27         //Zugriff ueber charArray
28         char[] textAsArray = text.toCharArray();
29         //tauschen
30         char tmp = textAsArray[indexA];
31         textAsArray[indexA] = textAsArray[indexB];
32         textAsArray[indexB] = tmp;
33         //zurueck in den String
34         return (new String(textAsArray));
35     }
36 }
```

String drehen:

Bitte geben Sie einen String ein: **abcde**

FlippString.rekursiveFlipp: abcde 1

FlippString.rekursiveFlipp: adcbe 0

FlippString.rekursiveFlipp: edcba -1

Umgedreht: edcba

Treppe auf Treppe ab

String abcde, länge 5, halbeLänge-1 = $5/2 - 1 = 2 - 1 = 1$

```
rekursiveFlipp (abcde, 1)           return(return text)
  exchange 1 mit 3
  rekursiveFlipp (adcbe, 0)         return(return text)
    exchange 0 mit 4
    rekursiveFlipp (edcba, -1)      return text
```

" Palindrom"

```
Application.java
1 package ch bbw.pr.palindrom;
2
3 import java.util.Scanner;
4
5 /**
6  * Palindrom
7  * @author Peter Rutschmann
8  * @version 02.11.2021
9  */
10 public class Application {
11
12     public static void main(String[] args) {
13         Scanner keyboard = new Scanner(System.in);
14
15         System.out.println("Bitte geben Sie einen Text ein:");
16         String userInput = keyboard.nextLine();
17
18         if (isPalindrom(userInput.toCharArray(), 0, userInput.length()-1)) {
19             System.out.println("Die Eingabe ist ein Palindrom");
20         } else {
21             System.out.println("Die Eingabe ist kein Palindrom");
22         }
23
24         keyboard.close();
25     }
26
27     private static boolean isPalindrom(char[] data, int front, int rear) {
28         if(front<rear) {
29             if (data[front] != data[rear]) return false;
30             else return isPalindrom(data, front+1, rear-1);
31         }
32         return true;
33     }
34 }
```

```
Terminated: Application (org.openjfx.javafx.swing)
Bitte geben Sie einen Text ein:
AnnA
Die Eingabe ist ein Palindrom
```

```
Bitte geben Sie einen Text ein:
AnAn
Die Eingabe ist kein Palindrom
```

Treppe auf Treppe ab

Wort sugus, 5 Buchstaben, index 0 bis 4

```
isPalindrom( sugus, 0, 4)          return(return(return true))
    isPalindrom( sugus, 1, 3)      return (return true)
        isPalindrom( sugus, 2, 2)  return true, da front<rear
```

Wort sugAs, 5 Buchstaben, index 0 bis 4

```
isPalindrom( sugus, 0, 4)          return(return fals))
    isPalindrom( sugus, 1, 3)      return false,
        da buchstaben front!= buchstabe rear
```

" Fibonacci-Reihe "

© Fibonacci.java ×

```
10  */
11  public class Fibonacci {
12
13      public int calculate(int value)
14      {
15          if(value <= 1){
16              return value;
17          }
18          return calculate( value: value - 1) + calculate( value: value - 2);
19      }
20
21  @ public static List<Integer> fibonacciReiheBisN(int n) {
22      List<Integer> fibReihe = new ArrayList<>();
23      if (n <= 0) {
24          return fibReihe;
25      } else if (n == 1) {
26          fibReihe.add(0);
27          return fibReihe;
28      } else if (n == 2) {
29          fibReihe.add(0);
30          fibReihe.add(1);
31          return fibReihe;
32      } else {
33          fibReihe = fibonacciReiheBisN( n: n - 1);
34          fibReihe.add(fibReihe.get(fibReihe.size() - 1) + fibReihe.get(fibReihe.size() - 2));
35          return fibReihe;
36      }
37  }
38  }
```

```
12  > public static void main(String[] args) {
13      Fibonacci myFibonacci = new Fibonacci();
14      int value = 0;
15      Scanner myScanner = new Scanner(System.in);
16      String eingabe = "";
17
18      System.out.println("Berechnen von Fibonacci-Zahlen:");
19      System.out.print("Bitte geben Sie ein, die wievielte Fibonacci-Zahl Sie haben wollen: ");
20      eingabe = myScanner.nextLine();
21      value = Integer.parseInt(eingabe);
22      System.out.println();
23      System.out.println("Die " + value + ".te Fibonacci Zahl lautet: "
24          + myFibonacci.calculate( value: value-1));
25
26      System.out.println("Berechnen der ersten " + value + " Fibonacci-Zahlen:");
27      System.out.print("0 ");
28      for (int i=1; i<value; i++)
29      {
30          System.out.print(myFibonacci.calculate(i) + " ");
31      }
32      System.out.println();
33      System.out.println("alle per Rekursion");
34      System.out.println(myFibonacci.fibonacciReiheBisN(value));
35
36      myScanner.close();
37  }
38  }
```

Berechnen von Fibonacci-Zahlen:

Bitte geben Sie ein, die wievielte Fibonacci-Zahl Sie haben wollen: 6

Die 6.te Fibonacci Zahl lautet: 5

Berechnen der ersten 6 Fibonacci-Zahlen:

0 1 1 2 3 5

alle per Rekursion

[0, 1, 1, 2, 3, 5]

```
40  /*
41  calculate(4)                                return 3
42      calculate(3)                            + calculate(2)        return 2 + 1
43      calculate(2) + calculate(1)      calculate(1) + calculate(0)    (1 + (return 1)) + (return 1
44      calculate(1) + calculate(0)        (return 1 + return 0) = 1
45  */
```

" GGT "

```
1 package ch.bbw.pr.ggt;
2
3 import java.util.Scanner;
4
5 /**
6  * GGT Applikation
7  * @author Peter Rutschmann
8  * @version 25.01.2018
9  */
10 public class Application {
11
12     public static void main(String[] args) {
13         Scanner myScanner = new Scanner(System.in);
14         String eingabe = "";
15         int zahlA = 0;
16         int zahlB = 0;
17         Ggt ggt = new Ggt();
18
19         System.out.println("GGT berechnen:");
20         System.out.print("Bitte geben Sie eine erste Zahl ein: ");
21         eingabe = myScanner.nextLine();
22         zahlA = Integer.parseInt(eingabe);
23         System.out.print("Bitte geben Sie eine zweite Zahl ein: ");
24         eingabe = myScanner.nextLine();
25         zahlB = Integer.parseInt(eingabe);
26
27         System.out.println();
28         System.out.println("GGT ist: " + ggt.calculate(zahlA, zahlB));
29         myScanner.close();
30     }
31 }
```

```
3 /**
4  * GGT Applikation
5  *
6  * @author Peter Rutschmann
7  * @version 04.03.2024
8  */
9 public class Ggt {
10
11     public int calculate(int a, int b) {
12         if (a > b) {
13             return calculate(a - b, b);
14         } else if (a < b) {
15             return calculate(b - a, a);
16         }
17         return a;
18     }
19 }
```

GGT berechnen:
Bitte geben Sie eine erste Zahl ein: 24
Bitte geben Sie eine zweite Zahl ein: 36

GGT ist: 12

```
34 /*
35 calculate(24, 36)           return 12
36     calculate (36-24, 24)   return 12
37         calculate (24-12, 12) return 12
38             calculate (24-12, 12) return 12
39 */
```