

Präsenzaufgaben 10

06./07.12.2023

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

1. Berechnung der Fakultät einer Zahl

Die Fakultät einer ganzen positiven Zahl n kann man mit der folgenden Rekursionsformel berechnen:

$$n! = \begin{cases} 1; & n=0 \\ n \cdot (n-1)!; & n>0 \end{cases}$$

Schreiben Sie eine rekursive Funktion

```
public static long fakultaet(int n)
```

die auf die oben beschriebene Weise die Fakultät von n berechnet.

Beispiele: $5! = 120$; $10! = 3628800$

2. Fibonacci-Zahl

Das n . Glied der Fibonacci-Folge lässt sich mit der folgenden Rekursionsformel berechnen:

$$fib(n) = \begin{cases} 1; & n=1 \\ 1; & n=2 \\ fib(n-1) + fib(n-2); & n>2 \end{cases}$$

Schreiben Sie eine rekursive Funktion

```
public static long fibonacci(int n)
```

die auf die oben beschriebene Weise das n . Glied der Fibonacci-Folge berechnet.

Beispiele: $fib(10) = 55$; $fib(20) = 6765$.

3. Rekursive Summe

Ergänzen Sie den Code der folgenden Funktion

```
public static int summe(int anfang, int ende)
{
    /*Berechnet rekursiv die Summe der Zahlen
    * von anfang bis ende.
    * Ist anfang > ende, so ist das Ergebnis 0
    */
}
```

Beispiele: Summe von 1-6 (inklusive) = 21; Summe von 15-20 (inklusive) = 105

4. Euklidischer Algorithmus

Implementieren Sie den Euklidischen Algorithmus rekursiv. Verwenden Sie zur Berechnung die untenstehende rekursive Funktionsvorschrift.

Der Euklidische Algorithmus zur Berechnung des größten gemeinsamen Teilers zweier positiver ganzer Zahlen a und b ($\text{ggT}(a,b)$) ist wie folgt rekursiv definiert:

- $\text{ggT}(a,b) := a$, falls $a = b$ gilt
- $\text{ggT}(a,b) := \text{ggT}(a-b, b)$, falls $a > b$ gilt
- $\text{ggT}(a,b) := \text{ggT}(a, b-a)$, falls $b > a$ gilt

Beispiele: $\text{ggT}(960,30)=30$; $\text{ggT}(144,27)=9$

5. Das Pascalsche Dreieck

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
...

```

Das Pascalsche Dreieck lässt sich aus folgender Entwicklungsvorschrift konstruieren:

$$a_{i,j} = \begin{cases} 1 & j=0 \text{ oder } j=i \\ a_{i-1,j} + a_{i-1,j-1} & 0 < j < i \end{cases}$$

Schreiben Sie eine Funktion

```
public static int getPascalWert(int i, int j)
```

die den Wert des Pascalschen Dreiecks an der Stelle (i, j) zurückgibt. Sie können davon ausgehen, dass die übergebenen Funktionswerte immer korrekt sind. Benutzen Sie einen rekursiven Funktionsalgorithmus.

Beispiele:

$$a_{0,0}=1; a_{4,1}=4; a_{5,0}=1; a_{5,2}=10$$

6. Rekursive Ausgabe einer Zahl

Schreiben Sie eine **rekursive** Methode zur Ausgabe aller Ziffern einer positiven ganzen Zahl n :

```
public static void rekursiveAusgabe(int n)
```

Die Methode soll außer n keine zusätzlichen Variablen verwenden.

Beispiel: `rekursiveAusgabe(12345)` soll folgendes ausgeben:

```
5
4
3
2
1
```

Ändern Sie nun die Reihenfolge, also

```
1
2
3
4
5
```

Sie müssen dazu nur eine Zeile verschieben, aber welche?

7. Spiegelung einer Zeichenfolge

Schreiben Sie eine Methode

```
public static String umkehren(String s)
```

die einen String übergeben bekommt und diesen in umgekehrter Reihenfolge zurückgibt. Verwenden Sie kein Array oder sonstige Konstrukte zur Zwischenspeicherung der Zeichenfolge, sondern implementieren Sie die Lösung **rekursiv**.

Testen Sie zum Beispiel „*Nie fragt sie: Ist gefegt? Sie ist gar fein.*“ Oder suchen Sie sich andere Palindrome auf <https://de.wikipedia.org/wiki/Palindrom>.

8. Geldautomat

Schreiben Sie eine rekursive Methode `geldautomat(int betrag)`, die einen Geldbetrag (in ganzen Euros) übergeben bekommt und bestimmt, welche Scheine (500EUR, 200EUR, 100EUR, 50EUR, 20EUR, 10EUR, 5EUR) notwendig sind, um den Betrag auszuzahlen. Dabei sollen möglichst große Einheiten verwendet werden. Restbeträge unter 5EUR werden als Trinkgeld behalten.

Beispiel: 463EUR = 200EUR + 200EUR + 50EUR + 10EUR

Implementieren Sie drei verschiedene Varianten:

- Sobald ein passender Schein gefunden wird, wird die Meldung direkt am Bildschirm ausgegeben (Rückgabety `void`),
- Die Methode gibt eine `ArrayList<Integer>` zurück, die die Scheine enthält, aus denen der Betrag zusammengesetzt ist,
- Die Methode gibt ein `int`-Feld zurück, bei dem eine Position einer Sorte von Scheinen entspricht. In dem Feld wird dann die Anzahl gespeichert, wie oft der jeweilige Schein verwendet wird.

9. Kombination von Primzahlen

Man kann die Zahl 10 auf 5 verschiedene Weisen als Kombination von Primzahlen schreiben:

7 + 3
5 + 5
5 + 3 + 2
3 + 3 + 2 + 2
2 + 2 + 2 + 2 + 2

Schreiben Sie eine rekursive Funktion, die zählt, auf wie viele Arten sich eine Zahl n als Kombination von Primzahlen ausdrücken lässt.

Beispiele:

Die Zahl 7 lässt sich auf 3 Arten als Kombination von Primzahlen darstellen (7; 5+2; 3+2+2).
Die Zahl 3 lässt sich nur auf eine Art als Kombination von Primzahlen darstellen (3).