

Wintersemester 2024/25
Prof. Dr. J. Rexilius

Abgabetermin: 24.10.2024, 08:00

Aufgabe 1 (9 Punkte)

- Strukturieren Sie ihre Programme mit geeigneten Methoden. Details zur Berechnung dürfen nicht in der main-Methode implementiert werden.
- Ihre Lösungen dürfen nicht zu Programmabstürzen führen. Eingaben über die Konsole müssen nicht überprüft werden.
- *Beachten Sie die aktuelle Version zu den allgemeinen Abgabehinweisen (Praktikumsordner im ILIAS).*

(a) Zahlen zählen (3 Punkte)

Schreiben Sie einen Algorithmus innerhalb einer Methode `zahlenZaehlen()`. Das Programm liest zwei Zahlen (in beliebiger Reihenfolge) von der Konsole ein, und gibt die Summe aller Primzahlen von der kleineren Zahl (einschließlich) bis zur größeren Zahl (einschließlich) auf der Konsole aus. Zusätzlich wird die Anzahl an Primzahlen ausgegeben. Das Ergebnis muss wie in dem unten gezeigten Beispiel formatiert werden.

Die Eingabe der Zahlen über die Konsole erfolgt durch die Funktion `nextInt()` aus der Klasse `Scanner`. Diese muss *zunächst importiert werden mit* `import java.util.Scanner;`

```
Scanner sc = new Scanner(System.in);  
int readInt = sc.nextInt();
```

Beispiel für Programmablauf und Ausgaben (Benutzereingaben in <>):

Zahl1: <3>

Zahl2: <8>

Summe aller Primzahlen zwischen 3 und 8: 3+5+7=15

Anzahl Primzahlen: 3

Wintersemester 2024/25
Prof. Dr. J. Rexilius

Abgabetermin: 24.10.2024, 08:00

(b) Sensordaten (3 Punkte)

Ein Sensor liefert Messwerte als Folge ganzer Zahlen. Dabei ist die Menge an Daten vorab nicht festgelegt. Schreiben Sie eine Methode `verarbeiteMesswerte`, mit der die Sensorwerte verarbeitet werden. Dabei simuliert die Methode den Sensor in dem Zahlen von der Standard-eingabe eingelesen werden. Nutzen Sie dafür, wie in Teil (a), folgenden Code:

```
Scanner sc = new Scanner(System.in);  
int readInt = sc.nextInt();
```

Die eingelesenen Zahlen werden immer in der folgenden Reihenfolge verarbeitet:

1. Alle Gruppen von zwei oder mehr unmittelbar aufeinanderfolgenden Nullen werden zu einer einzigen Null zusammengefasst.
2. Negative Werte werden in den positiven Wert der jeweiligen Zahl umgewandelt.

Die verarbeiteten Messwerte werden direkt auf der Konsole ausgegeben. Die Auswertung der Messwerte stoppt, wenn eine Eingabe von drei aufeinander folgenden Einsen erkannt wurde, also 1, 1, 1. Wenn diese Endsequenz erfüllt ist, wird zusätzlich noch die Summe aller ausgegebenen Werte auf der Konsole ausgegeben, in denen die Ziffer 4 vorkommt. Also z.B. Werte wie 4, 34, 46, etc.

Beispiele:

Input: 1 -2 4 7 0 0 1 1 14 1 1 1
Output: 1 2 4 7 0 1 1 14 1 1 1
Summe = 18

(c) Sanduhr (3 Punkte)

Implementieren Sie eine Methode `printSanduhr(int width)`, mit der eine Sanduhr in der Konsole ausgegeben wird. Die Größe soll über den Parameter `width` festgelegt werden. Dieser legt die Breite der ersten Zeile in Anzahl der Zeichen fest, wobei sich in jeder Zeile die Anzahl der Zeichen um zwei reduziert bzw. erhöht, und diese entsprechend eingerückt werden (siehe Beispiele). Die Methode soll für beliebige Werte von `width > 2` funktionieren.

Testen Sie Ihr Programm für `width=3, 4, 9`.

Beispiele:

<code>width=3:</code>	<code>width=4:</code>	<code>width=5:</code>
***	****	*****
*	**	***
***	****	*

