

**Datum:** 10.12.2018

**Abgabe bis:** 7.01.2019 11:30

Die nachfolgenden Aufgaben bilden ein Sammelsurium aus allen bisher behandelten Bereichen.  
Wählen Sie vornehmlich die Aufgabenbereiche aus, bei denen Sie sich noch nicht sicher fühlen.  
*Es sei ihnen selbst überlassen, die anderen Aufgaben dennoch zu bearbeiten.*

Hinweis: Erfinden Sie das Rad nicht neu. Nutzen Sie, wenn Sie es möchten, ihre Lösungen der Vorstunden zur Realisierung.

*Vorher*

**Bevor Sie eine Aufgabe beginnen...**

Erstellen Sie zunächst eine Klasse mit dem Namen `IhrName`.  
Lösen Sie innerhalb dieser Klasse die Aufgaben ihrer Wahl.

-----

**Am Ende...**

Ergänzen Sie ihr Hauptprogramm (also die `public static void main(String[] args)`-Methode) um eine entsprechende Benutzereingabe, die abhängig vom eingegebenen Wert eine ihrer Methoden/Prozeduren ausführt.

*Hinweis:* Sollte die aufgerufene Methode Parameter erwarten, lassen Sie diese durch den Benutzer eingeben.

---

**Grundlagen**

**Einlesen und ausgeben**

Schreiben Sie eine Prozedur `zahlenEinlesen`, die fünf Fließkommazahlen einliest und diese in umgekehrter Reihenfolge wieder ausgibt.

**Einlesen und Berechnung**

Entwickeln Sie eine Prozedur `zahlenBerechnen`, die Sie nach drei Zahlen fragt (auch negative Werte sollen erlaubt sein). Anschließend wird die Summe der drei Zahlen berechnet werden. Nachdem die Summe ausgegeben wurde, soll nach einer neuen Zahl gefragt werden, mit der die Summe dann multipliziert wird. Das Produkt soll ebenfalls ausgegeben werden.

**Modulo**

Erstellen Sie eine Methode `zeit`, die folgende Aufgabe erfüllt:

Es wird eine Anzahl von Sekunden eingegeben. Das Programm muss berechnen, wie viele Stunden, Minuten und restlichen Sekunden in dieser Sekundenzahl enthalten sind.

---

## **Datenstrukturen**

### **kleinste von vier**

Erstellen Sie eine Methode `kleinsteVonVier`, die von vier übergebenen Zahlen die Kleinste ermittelt und zurück gibt.

### **Division zweier Zahlen**

Erstellen Sie eine Methode `division`, die folgende Aufgabe erfüllt:

Es sind zwei Zahlen einzugeben. Der Rückgabewert der Methode ist die ganzzahlige Division der größeren Zahl durch die kleinere Zahl.

*Hinweis:* Die Division durch 0 ist nicht erlaubt.

### **Mittelwert**

Schreiben Sie eine Methode `mittelwert`, die den Mittelwert beliebig vieler einzugebender Zahlen berechnet.

### **Fallunterscheidung**

Schreiben Sie eine Methode `rechnen`, die vom Benutzer zunächst zwei Zahlen fordert und im Anschluss nach Benutzerwahl eine Berechnung folgender Art durchführt:

- 1 : Addition      3 : Multiplikation  
2 : Subtraktion   4 : Division

Das Ergebnis soll als `double`-Zahl ausgegeben werden.

---

## **Strings**

*Hinweis:* Für die Bearbeitung dieser Aufgaben müssen Sie sich selbstständig noch einmal mit dem Datentyp `String` auseinandersetzen.

### **String umwandeln**

Schreiben Sie eine Prozedur `umwandeln`, die einen Text vom Benutzer einliest und diesen in

Großbuchstaben umwandelt und ausgibt.

### Anzahl Zeichen im String

Erzeugen Sie eine Prozedur `zeichenZaehlen`, die in einem vom Benutzer eingegebenen Text die Anzahl des durch den Benutzer vorgegebenen Zeichens bestimmt.

*Beispiel:* In der Zeichenkette "Dies ist das Kapitel ueber die Arbeit mit Zeichenketten" soll die Anzahl der vorkommenden 'e' bestimmt werden.

### Zeichentausch

Schreiben Sie eine Prozedur `zeichentausch`, die in einer vom Benutzer einzugebenden Zeichenkette eine im Quellcode vorgegebene Zeichenkette vertauscht und die Zeichenkette anschließend auf dem Bildschirm ausgibt.

### Palindrom

Schreiben Sie eine Methode `palindrom`, die für eine übergebene Zeichenkette überprüft ob diese ein Palindrom ist.

Ist dies der Fall, gibt die Methode `true` zurück, ansonsten `false`.

---

## Arrays

*Hinweis:* Für die Bearbeitung dieser Aufgaben müssen Sie sich selbstständig noch einmal mit dem Datenkonstrukt **Array** auseinandersetzen.

**Summe im Array** Schreiben Sie eine Methode `summeArray`, die zunächst vom Benutzer eine durch diesen definierte Anzahl an Zahlen erwartet. Im Anschluss berechnet die Methode die Summe über alle Zahlen des Arrays.

**Beispiel:**

`int a[] = {1, 2, 3, 4};` hat die Ausgabe: 10.

Schreiben Sie ein Prozedur `indexAddition`, die zwei Arrays indexweise addiert.

**Beispiel:** `int a[] = {1, 2, 3}; int b[] = {2, 3, 4};` hat die Ausgabe `int c[] = {3, 5, 7};`.

### Tausch von Elementen

Erstellen Sie eine Methode `wertetausch`, die in einem übergebenen Array zwei aufeinander folgende Elemente vertauscht.

**Beispiel:** `int a[] = {1, 2, 3, 4, 5, 6}` wird zu `int a[] = {2, 1, 4, 3, 6, 5}`.

Ist die Anzahl der Elemente ungerade, bleibt das letzte Element unverändert.

`int a[] = {1, 2, 3}` wird also zu `int a[] = {2, 1, 3}`.

## Übung zur Anwendung und Vertiefung

*Für jede dieser Aufgaben erzeugen Sie eine eigene Klasse mit der Bezeichnung `ThrName_Bezeichnung`.*

### Quadrat malen

Schreiben Sie ein Programm, das die Seitenlänge eines Quadrates erfragt und dann im Textmodus ein Quadrat dieser Größe ausgibt.

Gibt der Benutzer beispielsweise 5 ein, so soll die Ausgabe wie folgt aussehen:

```
* * * * *
*       *
*       *
*       *
*       *
* * * * *
```

Zur Lösung dieser Aufgabe verwenden Sie bitte entsprechende Prozeduren (z.B. `ersteLetzteZeile` oder `zwischenZeile`).

### Geschachtelte Schleife

Schreiben Sie ein Programm, das für eine vom Benutzer definierte Zahl folgende Ausgabe erzeugt.

Benutzerdefinierte Zahl: 8

**Ausgabe:**

```
1 abcdefg
12 abcdef
123 abcde
1234 abcd
12345 abc
123456 ab
1234567 a
```

### Ausgabe einer Ganzzahl im Wortlaut

Schreiben Sie ein Programm, das eine eingegebene Ganzzahl im Wortlaut ausgibt. Es reicht, wenn das Programm beispielsweise die Zahl 345 als 'Drei Vier Fünf' ausgibt.

Sorgen Sie dafür, dass alle auftretenden Sonderfälle korrekt behandelt werden.

Eine interessante Erweiterung dieser Aufgabe besteht darin, die Zahl so auszugeben, wie die tatsächlich ausgesprochen wird. In obigem Beispiel also "Dreihundertfünfundvierzig".

### Sieb des Eratosthenes

Eine seit über 2000 Jahren bekannte Form der Primzahlbestimmung wird als "Sieb des Eratosthenes" bezeichnet.

Angenommen, Sie sollen alle Primzahlen im Bereich von 1 bis 1000 bestimmen. Gehen Sie dazu wie folgt vor:

Erstellen Sie ein boolesches Array mit 1000 Elementen und initialisieren Sie alle Elemente auf `true`.

Ändern Sie das erste Element auf `false`.

Nun führen Sie in einer Schleife folgende Schritte aus:

- Suchen Sie das nächste Element, dessen Wert `true` ist. Falls kein weiteres Element diese Bedingung erfüllt, beenden Sie die Schleife.
- Ändern Sie den Wert aller Elemente, deren Position ein ganzzahliges Vielfaches der aktuellen Position ist, auf `false`.

Haben Sie beispielsweise das fünfte Element gefunden, so werden auf diese Weise die Elemente 10, 15, 20, ... 995, 1000 auf `false` gesetzt.

Geben Sie nun die Positionen aller Arrayelemente aus, deren Wert `true`. Diese sind die Primzahlen zwischen 1 und 1000.

Schreiben Sie ein Programm, das das "Sieb des Eratosthenes" realisiert und ermitteln Sie auf diese Weise alle Primzahlen zwischen 1 und 1000.

### Primfaktorzerlegung

Schreiben Sie ein Programm, das zu einer eingegebenen positiven Ganzzahl eine Primfaktorzerlegung durchführt.

Wird beispielsweise 120 eingegeben, so soll die Ausgabe des Programms  $120 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 5$  lauten.

Erstellen Sie zum Einlesen der Zahl und zur Primfaktorzerlegung entsprechende Methoden.

### Perfekte Zahl

Schreiben Sie ein Programm, das für eine eingegebene Zahl  $n$  bestimmt, ob  $n$  eine perfekte Zahl ist.

Mathematiker bezeichnen alle positiven Ganzzahlen, deren Wert gleich der Summe ihrer echten Teiler ist, als perfekte Zahlen.

So ist beispielsweise 6 eine perfekte Zahl, denn  $6 = 1 + 2 + 3$ . Eine andere perfekte Zahl ist 28,

denn es gilt  $28 = 1 + 24 + 7 + 14$ .

Bestimmen Sie auch die nächstgrößere perfekte Zahl über 28.

### **Teilbarkeit durch 2**

Schreiben Sie ein Programm, das zu einer einzugebenden Zahl im Bereich von 1 ... 10000 ermittelt, wie oft diese durch 2 teilbar ist.

Verwenden Sie eine `while`-Anweisungen und geben Sie das Resultat in der Konsole aus.