

Schreiben Sie die Methode

```
public static int[] merge(int[] a, int[] b)
```

die die Werte der beiden aufsteigend sortierten Felder a und b in einem neuen Feld abspeichert, das selbst auch aufsteigend sortiert ist. Das neue Feld wird zurückgegeben.

Erstellen Sie in Eclipse ein neues Projekt mit dem Namen `EidP_VL08`. Fügen Sie dem Projekt eine leere Java-Datei mit dem Namen `Median.java` hinzu.

- a) Schreiben Sie ein Methode `berechneDurchschnitt`, die für ein gegebenes zweidimensionales Feld von `int` das arithmetische Mittel (Durchschnitt) der Werte berechnet und zurückgibt.
- b) Schreiben Sie nun ein Methode `berechneMedian`, die für ein gegebenes eindimensionales Feld von `int` den Median berechnet. Der Median ist der mittlere Wert aller Werte im Feld. Bei gerader Anzahl von Elementen nimmt man den Mittelwert der beiden mittleren Werte.

Beispiele: Das Feld

1	5	4	3	8
---	---	---	---	---

hat den Median 4 (1,3 sind kleiner und 5,8 sind größer als 4), das Feld

1	5	4	3	2	8
---	---	---	---	---	---

hat den Median 3,5 (3 und 4 sind die mittleren Werte).

Das Collatz-Problem ist ein ungelöstes mathematisches Problem. Hierbei geht es um die Frage, ob die Zahlenfolge, definiert durch

$n \rightarrow n/2$, falls n gerade

$n \rightarrow 3*n+1$, sonst

für alle $n > 0$ den Wert 1 erreicht.

Wir betrachten hier eine modifizierte Version, bei der die Folge durch

$n \rightarrow n/2$, falls n gerade

$n \rightarrow n/3$, falls n ungerade und n durch 3 teilbar

$n \rightarrow 3*n+1$, sonst

definiert ist.

Beispiele:

Die Folge zu $n=5$ sieht wie folgt aus:

$5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

Die Folge zu $n=7$ sieht wie folgt aus:

$7 \rightarrow 22 \rightarrow 11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16$
 $\rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

Erstellen Sie in Eclipse ein neues Projekt mit dem Namen `EidP_VL06`. Fügen Sie dem Projekt eine leere Java-Datei mit dem Namen `CollatzModifiziert.java` hinzu.

- a) Erstellen Sie innerhalb der Klasse `CollatzModifiziert` eine `main`-Methode, in der zunächst folgende Auswahlliste auf der Konsole ausgegeben werden soll:

```
-----
1    Collatz-Folge für n
2    Länge der Collatz-Folge für n
3    Fertig
-----
```

Lesen Sie nun die Auswahl (1-3) als ganze Zahl von der Konsole ein. Sollte die eingegebene Zahl nicht zwischen 1 und 3 liegen, so geben Sie die obige Auswahlliste erneut aus und lesen die Auswahl ein. Das Programm soll erst enden, wenn der Nutzer die Auswahl 3 eingibt.

- b) Schreiben Sie nun eine Methode `berechneCollatzFolge`, die für ein gegebenes n die (modifizierte) Collatz-Folge wie in obigen Beispielen auf der Konsole ausgibt. Beachten Sie, dass nach Ausgabe der 1 kein Pfeil mehr ausgegeben wird.
Modifizieren Sie die `main`-Methode, so dass bei Auswahl 1 eine ganze Zahl von der Konsole gelesen und die entsprechende Folge mittels der Methode `berechneCollatzFolge` ausgegeben wird.
- c) Schreiben Sie eine weitere Methode `berechneCollatzLaenge`, die für ein gegebenes n die Länge der (modifizierten) Collatz-Folge auf der Konsole ausgibt.

Beispiel: Bei Eingabe von 5 sieht die Ausgabe (Konsole) wie folgt aus.

5: 6

Bei Eingabe von 7 sieht die Ausgabe (Konsole) wie folgt aus:

7: 17

Beantworten Sie folgende Fragen:

- Was ist ein Objekt?
- Was ist eine Klasse?
- Worin besteht der Unterschied zwischen einer Klasse und einem Objekt?
- Wie greift man auf ein Attribut bzw. eine Methode eines Objektes zu?
- Was ist die „garbage collection“ und wozu dient diese?

In dieser Aufgabe sollen zunächst die Ergebnisse von 3 WM-Spielen eingelesen und die daraus resultierende Tabelle ausgegeben werden. Für einen Sieg werden 3 Punkte für den Sieger vergeben, bei einem Unentschieden bekommt jede der beiden Mannschaften 1 Punkt. Die teilnehmenden Mannschaften sind Italien, Niederlande und Deutschland, wobei jede dieser Mannschaften gegen jede andere genau einmal spielt.

Zunächst soll die Zeile

```
(1) Italien - Niederlande
```

ausgegeben und zwei Zahlen (`int`) nacheinander von der Konsole eingelesen werden, die das Ergebnis der Partie ergeben. Werden z.B. die Zahlen 4 und 1 eingegeben, so hat Italien mit 4 zu 1 gewonnen.

In gleicher Weise sollen dann auch die Ergebnisse der Partien Niederlande – Deutschland und Deutschland – Italien eingelesen werden.

Nach Einlesen der Ergebnisse soll dann die Tabelle in der Form

```
Mannschaft Punkte  
ausgegeben werden, z.B.
```

```
Italien 4  
Niederlande 2  
Deutschland 1
```

Die Tabelle muss hierbei **nicht** nach Punkten sortiert ausgegeben werden!

Haben zwei Mannschaften die gleiche Punktezahl, so soll die gesamte Eingabe der Ergebnisse wiederholt werden, bis sich eine eindeutige Tabelle ergibt.

Ergänzen Sie die `main`-Methode der Klasse `Spielplan` entsprechend.

Geben Sie notwendige `import`-Anweisungen an. Sie dürfen eigenständig Methoden ergänzen, wenn Sie diese benötigen.

```

public class Spielplan
{
    public static void main(String[] args)
    {
        int punkteDeutschland = 0;
        int punkteItalien = 0;
        int punkteNiederlande = 0;
    }
}

```

- a) In dieser Teilaufgabe sollen die in einem eindimensionalen int-Array abgespeicherten Zahlen aufsummiert werden.
 Beispiel: Ist das Feld mit den Zahlen 5, 1, 5, 4 gegeben, so soll die Summe $5+1+5+4 = 15$ berechnet und zurückgegeben werden (keine Ausgabe auf der Konsole).
Hinweis:
 Zur Lösung dürfen Sie keine Methoden aus der Java-Klassenbibliothek verwenden.

```

public static int berechneSumme(int[] feld)
{

```

- b) Nun soll aus einer ganzen Zahl $n \geq 0$ ein int-Array erzeugt werden, dass die Ziffern der Dezimaldarstellung von n enthält.
 Beispiel: Für die Zahl 5145 soll das Feld der Länge 4 mit den Zahlen 5, 1, 4, 5 zurückgegeben werden.

```

public static int[] dezimalDarstellung(int n)
{
    //Annahme:  $n \geq 0$ 

```

- c) Geben Sie nun eine Methode an, die für ganze Zahlen $n \geq 0$ die Quersumme berechnet. Beispiel: Für die Zahl 5145 soll $5+1+4+5 = 15$ zurückgegeben werden.

Hinweis:

Zur Lösung dürfen Sie keine Methoden aus der Java-Klassenbibliothek verwenden. Sie dürfen aber die Methoden aus den Teilaufgaben a) und b) nutzen.

```
public static int quersumme(int n)
{
    //Annahme:  $n \geq 0$ 
```

- b) Programmieren Sie die Methode `gitter`, die für ein beliebiges $n > 0$ ein quadratisches, zweidimensionales Feld mit $2n+1$ Zeilen und $2n+1$ Spalten erzeugt und jede zweite Zeile und Spalte mit dem Zeichen `*` füllt (Gittermuster). Die übrigen Elemente sollen Leerzeichen enthalten.

Beispiele für $n = 1$ und 2 :

	*	
*	*	*
	*	

	*		*	
*	*	*	*	*
	*		*	
*	*	*	*	*
	*		*	

```
public static char[][] gitter(int n)
{
    assert (n > 0);
```