Für den Auftrag der “Homecontrol OHG” werden ein paar Funktionalitäten immer wieder benötigt. Damit diese nicht immer wieder neu programmiert werden müssen, sollen Sie Lilly dabei unterstützen, ein diese Funktionalitäten als Methoden zur Verfügung zu stellen.

Aus dem Lastenheft der Homecontrol OHG:

* Ein Menü soll vom Benutzer abfragen, was er machen möchte. Dieses Menü soll solange angeboten werden, bis der Benutzer eingibt, dass er das Programm beenden möchte.
* Temperaturen sollen zwischen Celsius, Fahrenheit und Kelvin konvertierbar sein.
* Temperaturdurchschnitte sollen berechnet werden.
* Temperaturwerte, die mehr als eine Nachkommastelle haben sollen auf eine Nachkommastelle gerundet werden.
* Die Temperaturen eines Jahres sollen schön formatiert ausgegeben werden.
* Es sollen die Höchstwerte, die Tiefstwerte und der häufigste Wert ermittelt werden.

Methoden

**Aufgabe 1**

Informieren Sie sich mit Hilfe der Unterlagen des BWV über Methoden.

Nutzen Sie dafür die Vokabelliste und die Kontrollfragen.

**Aufgabe 2**

Erstelle zwei Eingabemethoden in der Klasse MainTemperaturCSV.java im Projekt J05\_Temperatur.

1. Eine soll mit Hilfe einer zu übergebenden Eingabeaufforderung ein Zeichen vom User einlesen.
2. Eine soll mit Hilfe einer zu übergebenden Eingabeaufforderung eine Dezimalzahl vom User einlesen.

**Aufgabe 3**

Erstelle in der Klasse MainTemperaturCSV.java, eine oder mehrere Methoden, welche die geforderten Umrechnungen ermöglichen:  
Formeln:  
Gradzahl in °C = (Gradzahl in °F -32)/ 1,8  
Gradzahl in °F = Gradzahl in °C \*1,8 + 32  
In Kelvin: K = Gradzahl in °C + 273,15

**Aufgabe 4**

Erstelle eine Methode, welche das Menü der zur Verfügung stehenden Funktionen anzeigt und eine der in Aufgabe 2 erstellten Methoden nutzt, um die Auswahl des Benutzers einzulesen. Mit Hilfe der Menüauswahl soll in main dann die gewünschte Operation durchgeführt werden. Wiederhole das solange der Benutzer das möchte.

**Aufgabe 5**

Erstelle eine Methode, welche eine Zahl auf eine angegebene Anzahl Nachkommastellen rundet.

Die Methode **Math.round()** rundet auf volle Ganzzahlen.

Beispiel zum Runden auf 2 Nachkommastellen: 123,456   
🡪   
🡪   
🡪

Arrays

Das Programm TemperatureDataRetriever benutzt an einigen Stellen "String[ ]".

Informiert Euch über diese klassische Datenstruktur.   
Wie wird sie erstellt?   
Wozu braucht man einen Index?  
Mit welcher Schleifenart wird diese Datenstruktur i.d.R. erstellt?

Beantworte die Kontrollfragen zu Arrays, um sicherzugehen, dass alle wichtigen Informationen bekannt sind.

**Aufgabe 1**

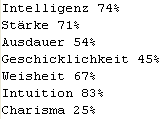
Lese die Temperaturen eines Jahres im Temperatur-Programm in ein geeignetes Array ein.

**Aufgabe 2**

* 1. Erstelle eine Klasse ArrayUtilities mit den folgenden Funktionalitäten als Methoden:

1. Eine Methode, welche ein int-Array mit Zahlen in einem bestimmten Wertebereich (von – bis) füllt.
2. Eine Methode, welche ein double-Array mit Zahlen in einem bestimmten Wertebereich (von-bis) füllt.
3. Eine Methode zum schönen Ausgeben eines Arrays.
4. Eine Methode, welche die Summe aller Zahlen berechnet.
5. Eine Methode, welche den Durchschnitt berechnet. Achte darauf, dass Du keinen Quellcode duplizierst.
6. Eine Methode, welche das Maximum im Array ermittelt.
7. Eine Methode, welche das Minimum im Array ermittelt.
8. Eine Methode, welche das den häufigsten Wert (Modus oder Modalwert genannt) im Array ermittelt.
   1. Nutze die Klasse ArrayUtilities, im Temperatur-Programm.
9. Dafür muss die Klasse im selben Verzeichnis, wie dein Temperatur-Programm, liegen.
10. Jede Methode kann im Temperatur-Programm genutzt werden, wenn beim Aufruf der Klassenname dem Methodennamen vorangestellt wird.   
    Bsp.: ArrayUtilities.ausgabe( array )
11. Erstelle im Temperatur-Programm ein Menü, mit welchem dem Benutzer angeboten wird, die höchste, niedrigste, durchschnittliche Temperatur und den Median zu ermitteln.

**Aufgabe 3**

 Schreiben Sie ein Programm **J06\_Rollenspiel**, welches 7 Prozentwerte zwischen 0% und 100% für die Charaktereigenschaften (Intelligenz, Stärke, Ausdauer, Geschicklichkeit, Weisheit, Intuition und Charisma) einer Spielfigur zufällig erzeugt und in einem Array gespeichert werden. Achtung: Prozentzahlen sind Dezimalzahlen. 0,01 entspricht 1%.

Benutzen Sie auch für die Eigenschaftsbezeichnungen ein Array.

**Aufgabe 4**

**4a)** Schreiben Sie das Programm **J06\_Noten** in dem der Benutzer gefragt wird, wie viele Noten generiert werden sollen. Ein Array wird in entsprechender Größe angelegt. Die Noten werden automatisch generiert. Anschließend wird der Notendurchschnitt berechnet. Alle Noten und der Schnitt werden ausgegeben.   
**4b)**Der Benutzer erhält jetzt die Möglichkeit eine Note zu ändern. Er gibt an welche (z.B. 1. Note). Die zu ändernde Note wird ausgegeben. Er gibt die neue Note ein. Der Durchschnitt soll daraufhin neu berechnet und ausgegeben werden.   
**4c)** Wiederholen Sie 3b) so oft wie der Benutzer möchte.

**4d)** Es dürfen nur Noten zwischen 1 und 6 zugelassen werden.  
 Alternative 1: Der Benutzer muss so oft die zu ändernde Note   
 eingeben, bis die Note im erlaubten Bereich liegt.

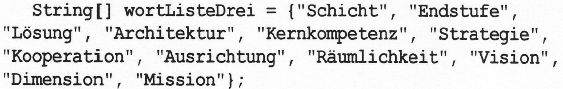
Alternative 2: Die Änderung im Array erfolgt nur bei zulässiger Eingabe.   
 Ansonsten bleibt das Array unverändert.

**Aufgabe 5**



Lassen Sie zufällig Phrasen bilden, solange der Benutzer weiter machen möchte.

J06\_Phrasomat

Quelle: Sierra, Kathy, Bates, Bert: Java von Kopf bis Fuß, O'Reilly Verlag,   
1. Mai 2006

**Zusatzaufgabe Game of Life**

Erstellen Sie ein **zweidimensionales Array** 40x40 boolean-Array. Ein true bedeutet, dass die Zelle lebt. False bedeutet, dass Sie nicht lebt. Zu Anfang sind 8 Zellen mit true belegt.

Erstellen Sie eine Simulation bekannt unter dem Namen "game of life": Informieren Sie sich in Wikipedia, wie die Simulation des Lebens nach Conway funktioniert und starten Sie mit unterschiedlichen Startkonstellationen.