

Aufgabe 1: "Alle geraden Zahlen"**[3 Punkte]**

Gesucht ist ein JAVA-Programm, welcher den Algorithmus aus A01 umsetzt.

Das Programm soll somit **alle geraden ganzen Zahlen** zwischen einer vom Benutzer einzugebenden **Unter-** und **Obergrenze** ausgeben (Unter- bzw. Obergrenze ggf. eingeschlossen).

Beispiele:

```
Untergrenze: 3
Obergrenze: 18
4 6 8 10 12 14 16 18
```

```
Untergrenze: 3
Obergrenze: 4
4
```

```
Untergrenze: 10
Obergrenze: 2
```

Abgabe:

→ **JAVA-Programm.**

Aufgabe 2: "n-te Potenz von x"**[3 Punkte]**

Gesucht ist ein Programm, welcher die **n-te Potenz** einer, durch den Benutzer einzugebenden, **Zahl (=Basiszahl)** berechnet.

Beispiele:

```
Basiszahl: 2
Potenzzahl: 7
7-te Potenz von 2 ist 128.
```

```
Basiszahl: 4
Potenzzahl: 3
3-te Potenz von 4 ist 64.
```

```
Basiszahl: -2
Potenzzahl: -2
-2-te Potenz von -2 ist 1/4
```

```
Basiszahl: -3
Potenzzahl: -3
-2-te Potenz von -3 ist -1/27
```

Hinweis:

- Bitte stellen Sie sicher, dass Sie für dieses Beispiel die **letzte Version der VL/UE als Grundlage** verwenden.

Abgabe:

→ **JAVA-Programm.**

Aufgabe 3: "Durchschnittsgeschwindigkeit"**[6 Punkte]**

Gesucht ist ein **JAVA-Programm** der für eine vom Benutzer einzugebende Strecke (einzugeben in Kilometer) und Zeit (einzugeben als **Stunden** und **Minuten**) die gefahrene **Durchschnittsgeschwindigkeit berechnet** und diese wie unten gezeigt ausgibt.

Beispiel:

```
Strecke [km]: 1215
Benötigte Zeit
  Stunden: 43
  Minuten: 50
1215.0 km in 1 Tag 19 Stunden 50 Minuten
entspricht einer Durchschnittsgeschwindigkeit von 27.718631178707223 km/h.
```

```
Strecke [km]: 10
Benötigte Zeit
  Stunden: 0
  Minuten: 59
10.0 km in 59 Minuten
entspricht einer Durchschnittsgeschwindigkeit von 10.16949152542373 km/h.
```

```
Strecke [km]: 12.5
Benötigte Zeit
  Stunden: 1
  Minuten: 11
12.5 km in 1 Stunde 11 Minuten
entspricht einer Durchschnittsgeschwindigkeit von 10.56338028169014 km/h.
```

```
Strecke [km]: 120
Benötigte Zeit
  Stunden: 64
  Minuten: 1
120.0 km in 2 Tagen 16 Stunden 1 Minute
entspricht einer Durchschnittsgeschwindigkeit von 1.8745118458734704 km/h.
```

Hinweis:

- Sie können davon ausgehen, dass vom Benutzer **zulässige Zahlenwerte** eingegeben werden. Es ist hier noch keine Überprüfung der Eingabewerte notwendig.
- Überlegen Sie **welche Datentypen** Sie für die einzelnen Werte verwenden sollten. Überlegen Sie, wo Sie Kommazahlen vorfinden und wo nur Ganze Zahlen vorkommen.
- Beachten Sie in Ihrer Realisierung die **unterschiedlichen Ausgabevarianten**.

Abgabe:

→ **JAVA-Programm.**

Aufgabe 4: "Kaffeesautomat"**[12 Punkte]**

Gesucht ist ein **Programm**, welches es erlaubt den Verkauf von 3 Kaffee-Produkten zu automatisieren. Dabei kostet ein **"Cappuccino"** **1,20 EUR**, ein **"Mokka"** **0,90 EUR** und ein **"Verlängerter"** **1,10 EUR**.

Ein Benutzer, wir stellen uns hier den typischen Studierenden vor, der dringend Koffein benötigt, um die Vorlesung um 08:30 erfolgreich antreten zu können, kann nun zuerst einmal den gewünschten Kaffee wählen.

Er bekommt danach den (noch) zu zahlenden Betrag angezeigt und kann nun Geld einwerfen (es sollen hier für unser Beispiel nur **10 Cent**, **20 Cent**, und **50 Cent** erlaubt sein) – solange bis der gesamte Preis bezahlt ist.

Sollte mehr Geld eingeworfen worden sein, soll der Restbetrag "ausgegeben" werden. Dabei sollen neben dem Gesamtbetrag auch die einzelnen ausgeworfenen Münzen gelistet werden.

Beispiele:

```
Kaffeeautomat
=====
a ... CAPPUCCINO 1.20 EUR
b ... MOKKA 0.90 EUR
c ... VERLAENGERTER 1.10 EUR
Bitte auswaehlen: x
    Bitte waehlen Sie erneut!
Bitte auswaehlen: a

Gewaeahlt CAPPUCCINO !

Offener Betrag: 1.20
Bitte Geld einwerfen: 20

Offener Betrag: 1.0
Bitte Geld einwerfen: 50

Offener Betrag: 0.50
Bitte Geld einwerfen: 33
    Kein gueltiger Betrag!
Bitte Geld einwerfen: 20

Offener Betrag: 0.30
Bitte Geld einwerfen: 20

Offener Betrag: 0.10
Bitte Geld einwerfen: 50

Bezahlt! CAPPUCCINO wird zubereitet!
Retourgeldbetrag: 0.40
Returgeldauswurf:
    0.20
    0.20
```

Hinweis:

- Überlegen Sie welche **Datentypen** Sie zum Einsatz bringen.
- Überlegen Sie die großen Teile des Programms (e.g. Menü-Auswahl anzeigen, Kaffee wählen, ...)
- Realisieren Sie die einfachen Teile zuerst, kümmern Sie sich um die "Eingabeabsicherung" bzgl. der Münzgrößen später und vervollständigen Sie erst am Schluss Ihr Programm mit der "Stückelung" der Restgeldmünzen.

Abgabe:

➔ **JAVA-Programm**