

Schreiben Sie Klassen für das folgende Szenario.

Es gibt **Dinge**. Ein Ding hat einen (nicht notwendigerweise eindeutigen) Namen (String). Ein Ding, das man kaufen kann, ist ein **Artikel**. Ein Artikel hat also zusätzlich einen Preis (double), in Euro. Der Preis kann sowohl 0 als auch negativ sein.

Erstellen Sie entsprechende Klassen unter Verwendung von Vererbung. Für Dinge gibt es einen Konstruktor mit dem Namen als Parameter, für Artikel ebenfalls einen Konstruktor mit dem Namen als Parameter, und einen weiteren Konstruktor mit Name und Preis als Parameter. Falls kein Preis angegeben wird, hat der Artikel den Standardpreis 10 €.

Der Name soll nach der Erzeugung eines Objekts nicht mehr änderbar sein. Weiters soll er (außerhalb des default package) nur in Subklassen zugreifbar sein.

Auch der Preis soll entsprechend nur in den Subklassen zugreifbar sein. Er soll jedoch, mittels einer Setter-Methode, veränderbar sein.

Es gibt keine Dinge, die man nicht kaufen kann. Daher soll es keine Objekte geben, die zwar Dinge representieren würden, nicht jedoch Artikel.

Erstellen Sie ein Interface für die **Formatierung** der Artikeldaten, das eine parameterlose Methode `format()` enthält. Sie gibt einen String zurück, mittels dessen ein Artikel ausgegeben werden kann. Der String hat folgende Form:

```
"<name>: EUR <preis>",
```

z.B. "Teekanne: EUR 20.0"

Ergänzen Sie Ihr Programm entsprechend.

Neben Artikeln, deren Preis wie oben beschrieben in Euro festgelegt ist, gibt es Artikel mit Preisen in anderen Währungen, z. B. in US-Dollar. Für sie gilt alles über Artikel gesagte. Darüberhinaus soll die Formatierung zusätzlich den Preis in der Fremdwährung ausweisen, insgesamt also:

```
"<name>: EUR <preis> (<währung> <preis_in_währung>)",
```

z. B. "Kaffeetasse: EUR 3.808 (USD 5.0)".

Für solche **Fremdwährungsartikel** ist die Bezeichnung der Währung (String), z. B. "USD", sowie der Wechselkurs zum Euro (double), z. B. 1.313, bekannt.

Wichtig: der Preis für Fremdwährungsartikel soll in der Fremdwährung gesetzt, aber in Euro gespeichert werden. D.h. die Setter-Methode für den Preis muss diesen auf Euro umrechnen, bevor sie ihn speichert. Auch für Fremdwährungsartikel ist der Standardpreis 10 € (bzw. der entsprechende Betrag in der Fremdwährung).

Sowohl die Umrechnung auf Euro als auch die Formatierung sollen so programmiert sein, dass darin keine währungsspezifischen Werte (Bezeichnung, Wechselkurs) vorkommen.

Erstellen Sie Klassen für Artikel in **Dollar** ("USD", Wechselkurs 1.313) und **Yen** ("JPY", Wechselkurs 136.92).

Ergänzen Sie die main-Methode der Klasse Main um die Erzeugung der folgenden Artikel und um diese Ausgabe auf Standard-Output:

Mozartkugeln	EUR	10.00		
Skischuhe	EUR	150.00		
Erdnussbutter	EUR	10.00	(USD	13.13)
Baseball	EUR	38.08	(USD	50.00)
Bentobox	EUR	10.00	(JPY	1369.20)
Kimono	EUR	292.14	(JPY	40000.00)
Bentobox	EUR	14.61	(JPY	2000.00)

Diese Ausgabe soll in einer einzigen Schleife erzeugt werden; verwenden Sie dazu ein Array (die Anzahl der Leerzeichen sowie Nachkommastellen ist irrelevant).

Bitte beachten Sie weiters:

Das Programm soll nur Klassen enthalten, die Artikel repräsentieren (abgesehen von der Klasse für Dinge, dem Interface für die Formatierung und der Klasse Main).

Lösen Sie die Aufgabenstellung ohne Zuhilfenahme von static Variablen.

Vermeiden Sie die Duplizierung von Code (Aufrufe von Konstruktoren innerhalb anderer Konstruktoren stellen keine Codeduplizierung dar und sind sogar erforderlich).

Vermeiden Sie auch die Duplizierung von Daten (mehrfache Speicherung derselben Information in Objektvariablen).

Vermeiden Sie unnötigen Code (z. B. eine Methode, die nur den Parameterwert zurückliefert) und unnötige Berechnungen (z. B. Umrechnung von Euro in Euro).

Legen Sie für jede Klasse eine eigene Java-Quelldatei an.