

Aufgabenblatt A04: Datum und Uhrzeit

1 Java Denksportaufgaben

1. Sie wollen folgende Klasse aus einer Bibliothek verwenden:

```
public class Thing {  
    public Thing(int i) { ... }  
    ...  
}
```

Die Klasse können Sie also nicht verändern und Sie wissen auch nicht, wie Sie implementiert ist, Sie kennen nur die Schnittstelle. Für diese Aufgabe haben Sie also nur die Signatur des Konstruktors zur Verfügung.

Ihre Aufgabe: Schreiben Sie bitte eine Unterklasse *MyThing* von *Thing* mit folgenden Eigenschaften:

- 1.1. Sie hat einen Konstruktor, in dem der Wert des Parameters „i“ des Konstruktors der Oberklasse *Thing* ermittelt wird. Dafür verwenden Sie eine Klassenmethode irgend einer Klasse, etwas *Something.func*.
- 1.2. Der Wert, mit dem der Oberklassenkonstruktor aufgerufen wird, kann also jedes Mal ein anderer sein.
- 1.3. Den Wert wollen Sie in einem *final* Instanz-Attribut speichern.

Hinweise: Jedes Problem kann durch eine zusätzliche Indirektionsebene gelöst werden. Ich hoffe, dass ist hinreichend kryptisch.

2. Compiler-Warnungen sollten Sie nie ignorieren. Wenn Sie die Ursache verstanden haben und überzeugend begründen können, dass diese Warnung keine unliebsamen Auswirkungen haben kann, annotieren Sie den Code-Teil mittels einer *@SuppressWarnings* Annotation und einem Javadoc-Kommentar mit der Begründung.

Nun zur Aufgaben: Zunächst ein Code-Schnipsel:

```

public class Null {
    public static void greet() {
        System.out.println("Hello world!");
    }
    public static void main(String[] args) {
        ((Null) null).greet();
    }
}

```

- 2.1. Was passiert, wenn Sie dieses Programm ausführen?
- 2.2. Wie kommt die Ausgabe zustande?
- 2.3. Warum „tut man das nicht“? Nur weil man dickfällig genug ist, Compiler-Warnungen zu ignorieren?

2 Datum und Uhrzeit

In dieser Aufgabe sollen Sie üben mit Datums-, Zeitpunktsangaben und Zeiträumen zu arbeiten.

1. Bei der Aufgabe zwei von Aufgabenblatt a01 hatten Sie es bereits mit Zeitpunkten und Zeiträumen zu tun. Ohne Berücksichtigung irgendwelcher Vorschriften für Zinsberechnung, die in Vertragsbeziehungen gelten, sollen Sie nun verschiedenen Größen in geeigneten Zeiteinheiten ermitteln und damit arbeiten. Dazu sollen Sie in diesem Teile einige Dinge überlegen und testen, die dann in folgenden Aufgabenteilen benötigt werden.

In welchen Einheiten können Sie die „Länge“ eines Zeitraums ermitteln, wie machen Sie das? Es geht also um Objekte der Zeitraumklassen und Zeiträume zwischen Objekten der Datumklassen in Java. Es geht also um die möglichen Darstellungen, z. B. einer *Period* in Jahren, Monaten, Tagen. Die Anzahl Jahre, Monate, Wochen, Tage, Minuten usw. zwischen Zeitpunkten. Was geht da?

2. Mit diesen Info können Sie dann verschiedene Dinge leicht verallgemeinern, die Sie in Aufgabe zwei von Aufgabenblatt a01 bereits für Spezialfälle programmiert haben:

- 2.1. Ermitteln des Barwerts einer Zahlung, die in der Zukunft erfolgt. Dafür benötigen Sie:
 - Das Datum (z. B. heute) zu dem Sie den Barwert berechnen wollen,
 - Den Zahlungsbetrag mit Betrag und Zahlungsdatum,
 - Den Zinssatz, mit dem abgezinst werden soll. Dazu wird der jährliche Zinssatz entsprechend 365 Tage langer Jahre (Schaltjahre werden nicht berücksichtigt) auf den Zeitraum heruntergerechnet.

- 2.2. Ermitteln des Barwerts einer Zahlungsreihe, die z. B. als Liste von Zahlungen angegeben wird, auf einen Zeitpunkt.
 - 2.3. Eine Methode, um die Nullstelle einer Funktion z. B. mit dem Intervallhalbierungsverfahren zu finden, sollte Sie ja schon in Aufgabenblatt a01 geschrieben haben. Wenn Sie z. B. die Zahlungsreihe aus der vorstehenden Aufgabe mit einer Auszahlung (Darlehensbetrag) beginnen lassen, können Sie nun auch für flexible Zahlungen den Effektivzinssatz berechnen, in dem Sie die Nullstelle dieser Funktion berechnen..
 - 2.4. Wer schon mal vorgucken will, kann dafür auch gerne Methodenreferenzen verwenden (im Skript die Kapitel Lambdas und Streams).
3. Sie möchten ein Datum oder eine Uhrzeit nicht in in der internen Darstellung sehen. Mit welchen Elementen - vor allem aber nicht nur, im package `java.time` - können Sie die Ausgabe bedarfsgerecht formatieren? Ich sehe bzw. lerne gerne Formatierungen aus Gegenden, die ich nicht so gut kenne, Schriften, die mit anderen Zeichen als lateinischen Buchstaben, von rechts nach links etc. Daten, Uhrzeiten, mit/one Zeitzone.

Der Abgabetermin für alle ist:

Mittwoch, 22.04.2020, 08:00 .