

## Aufgabenblatt 04

Bearbeitungsende: 06.11.2022

### Empfehlung:

- Bearbeiten Sie die Aufgaben auf der Kommandozeile bzw. in einem einfachen Texteditor.
- Speichern Sie die Dateien zu jeder Aufgabe  $X$  in einem Verzeichnis `eidip/b04aX`.

**Hinweis:** Gehen Sie, wenn Sie sich mit einer Programmieraufgabe befassen, wie in der Vorlesung zu „Entwicklung“ besprochen vor.

### Aufgabe 1 [Programmierung – nicht bewertet]

Schreiben Sie eine Klasse `Eingabe` mit einer Klassenmethode `zaehlen`, die einen `Scanner` und eine Zeichenkette `s` als Argument annimmt. Die Methode soll über den Scanner Zeichenketten aus der Eingabe lesen, bis zu deren Ende. Sie soll zurückgeben, wie oft die Zeichenkette `s` unter den eingegebenen Zeichenketten vorkommt.

**Beispiel:** Wird die Zeichenkette `"und"` gesucht und lautet die Eingabe

```
Grund und Boden und  
Rock'n'Roll
```

so ist das Ergebnis 2.

**Hinweis:** Lesen Sie, falls Sie mit einer IDE arbeiten, in der Anleitung dazu (im Internet oder ggf. im LEA-Kurs unter „Dokumente“) nach, wie man das Eingabeende angibt.

### Aufgabe 2 [Programmierung – nicht bewertet]

Schreiben Sie eine Klasse `Eingabe` mit einer Klassenmethode `durchschnitt`, die einen `Scanner` als Argument annimmt. Die Methode soll über diesen eine Reihe von Gleitkommazahlen einlesen, bis zum Eingabeende. Die Methode soll als Ergebnis den Durchschnitt der gelesenen Werte zurückgeben, also die Summe der Werte geteilt durch ihre Anzahl.

Werden keine Zahlen eingegeben, ist das Ergebnis 0.0.

**Hinweis:** Verwenden Sie zwei Variablen, um sich einerseits die Anzahl, andererseits die Summe der gelesenen Zahlen zu merken.

**Hinweis:** Lesen Sie, falls Sie mit einer IDE arbeiten, in der Anleitung dazu (im Internet oder ggf. im LEA-Kurs unter „Dokumente“) nach, wie man das Eingabeende angibt.

### Aufgabe 3 [Programmierung]

Schreiben Sie eine Klasse `Messdaten` mit einer Klassenmethode `durchschnitt`, die einen `Scanner sc` und eine Zeichenkette `g` als Argumente annimmt. Die Methode soll über `sc` eine Reihe von Messwerten zu verschiedenen Messgrößen einlesen.

Das zweite Argument `g` gibt eine Messgröße an. Alle Messwerte zu dieser Größe sollen identifiziert werden, ihr Durchschnittswert soll ermittelt und (als Gleitkommazahl) zurückgegeben werden.

Genauer: Die Methode liest wiederholt, bis zum Eingabeende, Eingaben im folgenden Format:

1. drei Ganzzahlen, die Jahr, Monat und Tag beschreiben
2. eine Zeichenkette, die die gemessene Größe beschreibt
3. eine Gleitkommazahl, die den Messwert angibt

Sie können hier davon ausgehen, dass die Eingaben korrekt sind.

Die Messwerte aller Eingaben mit der Messgröße `g` werden addiert und gezählt. Am Ende wird ihr Durchschnitt ermittelt und als Ergebnis der Methode zurückgegeben.

Sie können dabei davon ausgehen, dass wenigstens ein Messwert zu der gewählten Messgröße `g` in der Eingabe vorliegt.

**Beispiel:** Die Methode werde aufgerufen mit "Temperatur" als Argument für den Parameter `g`. Die Eingabe sei:

```
2018 10 16 Luftdruck 1014.7
2018 10 17 Niederschlag 1.3
2018 10 15 Temperatur 18.2
2018 10 16 Niederschlag 0.0
2018 10 17 Temperatur 16.8
```

In diesem Fall wäre das Ergebnis  $(18.2 + 16.8)/2 = 17.5$ .

Außerdem soll die Methode auf dem Bildschirm ausgeben, an welchem Tag der maximale Wert für die Messgröße `g` vorlag und wie hoch dieser Wert war. Die Ausgabe soll folgende Form besitzen (am Beispiel der oben gegebenen Werte), abgeschlossen durch einen Zeilenumbruch:

```
max. Temperatur (18.2) am 2018/10/15
```

Tritt der höchste Wert mehrfach auf, soll der zuerst gelesene Tag mit diesem Wert ausgegeben werden.

**Hinweise:**

- Der kleinstmögliche Wert für den Gleitkommatyp `double` ist gegeben durch `-Double.MAX_VALUE`.
- Verwenden Sie zum Testen Ihrer Lösung die Eingabeumlenkung. So müssen Sie Testeingaben nur einmal erstellen und in Dateien abspeichern und können sie dann immer wieder verwenden.

## Aufgabe 4 [Programmierung – nicht bewertet]

Schreiben Sie eine Klasse `GeometrieTest` mit einer Klassenmethode `test`, die keine Argumente annimmt, und die einen Wahrheitswert zurückgibt. Sie testet gegebene Implementierungen der Methoden `flaecheREck` und `umfangREck` der Klasse `Geometrie` aus der folgenden Aufgabe. Die Testmethode soll genau dann `false` zurückgeben, wenn für eine Implementierung ein Fehler gefunden wird.

Beachten Sie dabei, dass Sie für den Vergleich von Gleitkommazahlen die in der Vorlesung vorgestellte Methode `Mathe.equals` verwenden sollten, um Rundungsfehler zu berücksichtigen.

**Hinweis:** Orientieren Sie sich an dem Beispiel für Unit-Tests aus der Vorlesung. Die Klasse sollte aber keine `main`-Methode enthalten. Sie finden in LEA zu dieser Aufgabe eine entsprechende Vorlage, die sich einfach um weitere Testfälle ergänzen lässt. Überlegen Sie, welche unterschiedlichen Fälle von Aufrufen vorkommen können. Sehen Sie jeweils einen repräsentativen Aufruf vor und prüfen Sie, ob das Ergebnis der erwartete Wert ist.

Testen Sie die Klasse mit fehlerhaften Implementierungen von `Geometrie`. Werden diese erkannt?

Laden Sie (nur) die Datei `GeometrieTest.java` in den Praktomat. Dort wird sie verwendet, um verschiedene richtige und falsche Implementierungen von `Geometrie` zu testen. Sie sollte alle Fälle korrekt identifizieren.

**Hinweis:** Wenn der Praktomat einen Fehler in Ihrem Test moniert, bedeutet das:

- Wird von Ihrem Test **false** geliefert, aber **true** erwartet, wurde eine korrekte Lösung als falsch beurteilt. In Ihren Tests ist also wenigstens ein Fall fehlerhaft umgesetzt.
- Wird von Ihrem Test **true** geliefert, aber **false** erwartet, wurde eine falsche Lösung als korrekt beurteilt. Sie haben also wenigstens einen Fall nicht abgedeckt und müssen überlegen, welcher das sein könnte und welche Testfälle entsprechend hinzuzufügen wären.

## Aufgabe 5 [Programmierung – nicht bewertet]

- Erweitern Sie die Klasse **Geometrie** aus der Vorlesung um eine Methode **flaecheREck**, die die gleichen (gleitkommawertigen) Argumente annimmt wie **umfangREck**, die jedoch die Fläche des Rechtecks zurückgibt.
- Schreiben Sie eine Klasse **GeometrieTest** mit einer Klassenmethode **test**. Die Methode nimmt ein **Scanner**-Objekt als Argument an und gibt kein Ergebnis zurück. Sie liest die folgenden Eingaben in dieser Reihenfolge von der Tastatur:
  - zwei Gleitkommazahlen **b** und **h**
  - eine Zeichenkette

Sie können dabei davon ausgehen, dass keine ungültige oder unvollständige Eingabe vorliegt.

Ist die gelesene Zeichenkette

- "Umfang", so wird der Umfang eines Rechtecks mit Breite **b** und Höhe **h** berechnet (mittels der Methode **umfangREck** in Klasse **Geometrie**) und auf dem Bildschirm ausgegeben, gefolgt von einem Zeilenumbruch,
- "Flaeche", so wird analog die Fläche eines solchen Rechtecks berechnet und ausgegeben,
- eine andere Zeichenkette, so wird auf dem Bildschirm

**Falsche Eingabe!**

ausgegeben, gefolgt von einem Zeilenumbruch.

Der gesamte Vorgang von Einlesen und Ausgabe wird so lange wiederholt, bis das Ende der Eingabe erreicht ist. Sie sollen das übergebene **Scanner**-Objekt nicht schließen.

Laden Sie die Dateien beider Klassen in den Praktomat.

**Hinweis:** Sie können die Dateien in ein **zip**-Archiv verpacken und so gemeinsam hochladen.

## Aufgabe 6 [Programmierung]

Schreiben Sie eine Klasse **DatumTest** mit einer Klassenmethode **test**, die keine Argumente annimmt, und die einen Wahrheitswert zurückgibt. Sie testet gegebene Implementierungen der Methode **istSchaltjahr** der Klasse **Datum** aus Aufgabe B03A4. Die Testmethode soll genau dann **false** zurückgeben, wenn für eine Implementierung ein Fehler gefunden wird.

**Hinweis:** Beachten Sie, dass Sie nicht nur positive Fälle überprüfen müssen („Jahr *X* ist ein Schaltjahr“), sondern auch negative („Jahr *X* ist kein Schaltjahr“). Stellen Sie alle Kriterien zusammen, die eine Rolle spielen, und testen Sie für alle Kombinationen Beispiele und Gegenbeispiele. Diskutieren Sie diese mit anderen.

**Hinweis:** Orientieren Sie sich hinsichtlich der Codestruktur an dem Beispiel für Unit-Tests aus der Vorlesung. Die Klasse sollte aber keine `main`-Methode enthalten. Sie finden in LEA zu dieser Aufgabe eine entsprechende Vorlage, die sich einfach um weitere Testfälle zu ergänzen lässt. Überlegen Sie, welche unterschiedlichen Fälle von Aufrufen vorkommen können. Sehen Sie jeweils einen repräsentativen Aufruf vor und prüfen Sie, ob das Ergebnis der erwartete Wert ist.

Testen Sie die Klasse mit fehlerhaften Implementierungen von `Datum`. Werden diese erkannt?

Laden Sie (nur) die Datei `DatumTest.java` in den **Praktomat**. Dort wird sie verwendet, um verschiedene richtige und falsche Implementierungen von `Datum` zu testen. Sie sollte alle Fälle korrekt identifizieren.

**Hinweis:** Wenn der **Praktomat** einen Fehler in Ihrem Test moniert, bedeutet das:

- Wird von Ihrem Test `false` geliefert, aber `true` erwartet, wurde eine korrekte Lösung als falsch beurteilt. In Ihren Tests ist also wenigstens ein Fall fehlerhaft umgesetzt.
- Wird von Ihrem Test `true` geliefert, aber `false` erwartet, wurde eine falsche Lösung als korrekt beurteilt. Sie haben also wenigstens einen Fall nicht abgedeckt und müssen überlegen, welcher das ist und welche Testfälle entsprechend hinzuzufügen sind.

---

Lösungen zu mit [**Programmierung**] markierten Aufgaben sind im **Praktomat** einzureichen.

Lösungen zu mit [**Programmierung – nicht bewertet**] markierten Aufgaben können ebenfalls im **Praktomat** eingereicht werden, werden jedoch nicht bewertet.

Allgemeine **Fragen** zu den Aufgaben können Sie im **LEA-Forum „Übungsaufgaben“** stellen.

**Hilfe** bei der Lösung der Aufgaben erhalten Sie in den **Übungen** und in der **Studierwerkstatt** .