

Programmieren I: Java

Wintersemester 2021/2022

Übungsblatt P06

(Bearbeitung im Rechnerpraktikum in der Zeit vom 07.12.2021 bis zum 13.12.2021)

Dieses P-Übungsblatt thematisiert den Umgang mit Klassen und Methoden. Ebenfalls lässt es Sie das erste Mal die Dokumentation des Application Programming Interface (API) von Java zu Rate ziehen, um geeignete, bereits implementierte Klassen und Methoden zu finden.

Aufgabe P6.1 *

Klassenmethode

- a) Implementieren Sie eine Klasse `MyMath` mit der Klassenmethode `abs`, welche einen Wert vom Typ `int` übergeben bekommt und ebenfalls einen `int`-Wert zurückliefert. Die Methode soll den Absolutbetrag des übergebenen Wertes berechnen und zurückgeben. Testen Sie die Funktionalität, indem Sie in der `main`-Methode die Klassenmethode `abs` mit den folgenden Parametern aufrufen:

`-1, 1, -1337, 2017, 0`

Lassen Sie den Rückgabewert auf der Konsole ausdrucken.

- b) Überladen Sie die Methode `abs`, so dass auch Werte vom Typ `double` übergeben werden können. Führen Sie für diesen Fall einen Typecast nach `int` durch. Testen Sie die erweiterte Funktionalität in der `main`-Methode.
- c) Versehen Sie Ihre fertige Klasse `MyMath` und deren Methoden mit JavaDoc-Kommentaren. Erstellen Sie die HTML-Datei und machen Sie sich mit der Struktur des Dokuments vertraut.
- d) Finden Sie nun die Methode `abs`, die Teil der vordefinierten Klasse `Math` ist, in der Dokumentation des sogenannten Application Programming Interface (API):

<http://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

Verwenden Sie in Ihrer `main`-Methode nicht mehr Ihre selbst implementierte, sondern die vordefinierte Methode.

Aufgabe P6.2 **

Methoden

Das Jahr neigt sich dem Ende zu, der Glühweinkonsum steigt und es kommt, was kommen muss: Weihnachtsstimmung. Da Ihnen die Kleintiere in Ihrer Wohnheimküche nach 8 Wochen jedoch immer noch Angst machen, entscheiden Sie sich, dieses Jahr lieber virtuelle Plätzchen zu backen. Hier kommt Ihnen natürlich Ihr Wissen aus der Programmieren-Vorlesung zu Gute.

a) Gehen Sie folgendermaßen vor:

- Schreiben Sie eine Klasse `Zutat`.
 - Jede `Zutat` hat einen Namen vom Typ `String` und liegt in einer bestimmten Menge vor. Die Menge soll hier über das Gewicht in Gramm ausgedrückt werden. Verwenden Sie dazu eine `int`-Variable.
- Schreiben Sie eine Klasse `Teig`.
 - Jeder `Teig` hat eine boolean-Instanzvariable `geknetet`, die angibt, ob der Teig bereits geknetet wurde.
 - Außerdem hat jeder `Teig` die **Instanzmethoden** `zutatHinzufuegen` und `kneten`:
 - * Die Methode `zutatHinzufuegen` hat als einziges Argument eine `Zutat` und gibt auf der Konsole die Meldung `Zutat hinzugefügt: Zutatname (Menge g)` aus, wobei `Zutatname` und `Menge` durch die jeweiligen Werte ersetzt werden sollten. Die Methode gibt nichts zurück.
 - * Die Methode `kneten` hat kein Argument und gibt nichts zurück. Sie setzt den Wert für `geknetet` auf `true` und gibt auf der Konsole die Meldung `Teig geknetet` aus.
- Schreiben Sie eine Klasse `Ofen`.
 - Diese Klasse hat nur eine einzige **Klassenmethode** `backen`. Sie erwartet als Argument einen `Teig` und einen `int`-Wert für die Backtemperatur. Die Methode hat einen Rückgabewert vom Typ `boolean`, der angibt, ob der Backvorgang erfolgreich war. Sie liefert den Wert `true`, wenn der übergebene Teig vor dem Backen geknetet wurde und die Backtemperatur 200 Grad nicht übersteigt. Andernfalls gibt sie `false` zurück.

b) Schreiben Sie nun die Klasse `Weihnachtsbaeckerei`, in welcher Sie innerhalb der `main`-Methode einen Backvorgang simulieren. Erzeugen Sie erst die drei Zutaten Mehl, Zucker und Eier mit den Mengen 200, 100 und 150 Gramm. Fügen Sie diese Zutaten dem Teig hinzu und kneten Sie ihn gut durch. Anschließend stellen Sie den Teig bei 180 Grad in den Ofen.

Geben Sie dann auf der Konsole das Ergebnis des Backvorgangs aus, indem entweder die Meldung `Hmmmm lecker` oder die Meldung `Leider schief gegangen` angezeigt wird.

Die Konsolenausgabe könnte beispielsweise so aussehen:

```
Zutat hinzugefügt: Mehl (200 g)
Zutat hinzugefügt: Zucker (100 g)
Zutat hinzugefügt: Eier (150 g)
Teig geknetet
Hmmm lecker
```

Experimentieren Sie nun etwas mit dem Programm. Ändern Sie beispielsweise die Temperatur oder lassen Sie das Kneten des Teiges weg.

Aufgabe P6.3 **/**

Rekursion

Gegeben ist folgende iterative Methode:

```
public void druckeDreieck(int a) {
    for (int i = 1; i <= a; i++) {
        for (int j = 1; j <= i; j++){
            System.out.print("*");
        }
        System.out.println("");
    }
}
```

Diese gibt ein Dreieck aus, dessen Größe von dem übergebenen Parameter bestimmt wird.

Beispiel:

```
*
**
***
****
*****
```

Ergänzen Sie den folgenden Lückentext so, dass die rekursive Methode die gleiche Funktionalität, wie die gegebene iterative Methode erfüllt. (6 Punkte)

```
public ----- druckeDreieckRekursiv(int a) {

    if(-----) {
        System.out.println("*");

        -----;
    }
    else {

        ----- s = -----;
        System.out.println(s);

        -----;
    }
}
```

Aufgabe P6.4 ***

Methoden

- a) Zur Speicherung der bei einem physikalischen Experiment gewonnenen Messdaten wird ein Objekt der Klasse

```
class Messung {
    public double[] gewichtungsreihe; // Gewichtungsfaktor jedes
                                   // Messwerts
    public int[] messreihe; // Messreihe mit Messwerten
    public double messwertGewichtet; // gewichteter Messwert
}
```

verwendet. Bei einem befüllten Objekt der Klasse Messung sind beide Felder vollständig mit Zahlen belegt und haben dieselbe Länge. Bitte beachten Sie, dass die Gewichtungsfaktoren in gewichtungsreihe durch eine Besonderheit des Versuchsaufbaus leider in **umgekehrter Reihenfolge** abgespeichert wurden.

Implementieren Sie die Klassenmethode gewichteterMesswert für die Klasse Experimentauswertung. Dieser Methode wird ein befülltes Objekt vom Typ Messung übergeben und deren Aufgabe ist es, in der Variablen messwertGewichtet des übergebenen Objekts die Summe der Produkte der Messwerte mit ihrem jeweiligen Gewichtungsfaktor abzuspeichern.

- b) Schreiben Sie eine Klasse ExperimentauswertungDemo mit einer main-Methode. In dieser lesen Sie vom Nutzer zunächst die Messreihe und dann die Gewichtungsreihe ein und lassen diese danach mit Ihrer in *Aufgabe a)* geschriebenen Methode gewichteterMesswert auswerten und ausgeben. Achten Sie auf die in *Aufgabe a)* beschriebenen Besonderheiten beim Einlesen!

Die Ausgabe soll dabei beispielsweise so aussehen:

```
Gewichteter Messwert für diese Messreihe (berechnet um 17:35:20 Uhr): 50.4
```

Um die Zeit in dieser Form auszugeben, importieren Sie zunächst alle Klassen der Packages text und util, indem Sie die folgenden Befehle verwenden:

```
import java.text.*;
import java.util.*;
```

Um das Format der Zeitangabe zu definieren, bietet sich die Klasse SimpleDateFormat an. Schauen Sie sich die API-Dokumentation zu dieser Klasse an. Verwenden Sie bei der Erzeugung des entsprechenden Objektes den folgenden Befehl:

```
new SimpleDateFormat("HH:mm:ss");
```

Was würde passieren, wenn Sie stattdessen „hh:mm:ss“ verwenden würden?

Die Ausführung der main-Methode könnte folgende Konsolenausgabe erzeugen:

```
Bitte Länge der Messreihe angeben:
3
Bitte Messwert für Komponente 0 eingeben:
7
Bitte Messwert für Komponente 1 eingeben:
2
Bitte Messwert für Komponente 2 eingeben:
10
Bitte Gewichtung für Komponente 2 eingeben:
1,5
Bitte Gewichtung für Komponente 1 eingeben:
10,0
Bitte Gewichtung für Komponente 0 eingeben:
2,2
Gewichteter Messwert für diese Messreihe (berechnet um 11:54:03 Uhr):
50.400000000000006
```

Viel Erfolg!