

JAVA

Tutoriumsblatt 7

Janosch Fuchs
Jürgen Dietel, Abdelrahman Elhabashy, Timon Römer, Daniel Weinholz
MATSE Gruppe
ITC - RWTH Aachen

WS24/25
19.09.2024
Abgabe: —

-
- Die Übungsblätter sollten alleine bearbeitet werden, aber Hilfe von Tutoren oder Studierenden ist gewünscht.
-

Tutoriumsaufgabe 7.1

Folgende Aufgaben sind schriftlich zu bearbeiten:

- Dürfen Konstruktoren andere Methoden des Objekts aufrufen?
- Wann sollten Methoden `private` sein?
- Was ist der Default-Wert für `double`-Attribute?
- Wann müssen Invarianten erfüllt sein?
- Was muss man tun, um einen freien Zugriff von außen auf die Attribute eines Objekts zu verhindern?
- Welches Schlüsselwort verweist in einer Methode eines Objekts auf das Objekt selbst?
- Notieren Sie eine Zeile, in der sie selbst eine Exception werfen.
- In welchen Ausnahmefällen darf man Attribute `public` machen?
- Wann erstellt Java automatisch einen Default-Konstruktor?
- Welches Schlüsselwort benötigt man, um eine Methode in eine Funktion umzuwandeln.
- Worauf haben Methoden Zugriff und Funktionen nicht?

Tutoriumsaufgabe 7.2

Schreiben Sie eine Klasse `Punkt` für einen Vektor im \mathbb{R}^2 . Die Klasse soll folgende Konstruktoren und Methoden enthalten.

- `public Punkt(double x, double y)`: Konstruktor mit der x - und y -Koordinate.
- `public Punkt(Punkt p2)`: Copy-Konstruktor.
- `public void schiebe(double dx, double dy)`: Verschiebt den Punkt um den Wert (d_x, d_y) .
- `public String toString()`: Gibt den Punkt in der Form (x, y) als String zurück.

Tutoriumsaufgabe 7.3

Schreiben Sie eine Klasse `GeradeZahl`. Die Klasse hat ein einziges Attribut, das eine gerade Zahl sein muss. Wird versucht, eine ungerade Zahl zu setzen, wird statt dessen die nächstniedrigere gerade Zahl genommen. Fügen Sie der Klasse `GeradeZahl` folgende Konstruktoren und Methoden hinzu:

- `public GeradeZahl(int z)`: Konstruktor mit der Startzahl z .
- `public GeradeZahl(GeradeZahl z2)`: Copy-Konstruktor.
- `public String toString()`: Gibt die Zahl als String zurück.
- `public GeradeZahl getNext()`: Gibt die nächsthöhere gerade Zahl zurück.
- `public GeradeZahl getSum(GeradeZahl g2)`: Gibt die Summe von `this` und g_2 zurück.

Tutoriumsaufgabe 7.4

Schreiben Sie eine Klasse `Geburtstag`, die den Tag und den Monat eines Geburtstags enthält. Der 29. Februar ist als Geburtstag erlaubt. Die Anzahl der Tage der einzelnen Monate ist: Januar 31, Februar 29, März 31, April 30, Mai 31, Juni 30, Juli 31, August 31, September 30, Oktober 31, November 30, Dezember 31. Fügen Sie der Klasse `Geburtstag` folgende Konstruktoren und Methoden hinzu:

- `public Geburtstag(int monat, int tag)`: Erzeugt einen Geburtstag. Wirft eine `ArithmeticException`, falls der Tag nicht existiert.
- `public Geburtstag(Geburtstag g2)`: Copy-Konstruktor.
- `public String toString()`: Gibt den Tag in der Form "2. Januar" zurück. Der Monat soll dabei als Wort ausgeschrieben sein.
- `public boolean equals(Geburtstag g2)`: Gibt `true` zurück, falls die Geburtstage `this` und `g2` übereinstimmen und ansonsten `false`.

Tutoriumsaufgabe 7.5

Schreiben Sie eine Klasse `Note` für eine Notenverwaltung. Gültige Noten sind an der Hochschule: 1.0; 1.3; 1.7; 2.0; 2.3; 2.7; 3.0; 3.3; 3.7; 4.0; 5.0. Fügen Sie der Klasse `Note` folgende Konstruktoren und Methoden hinzu:

- `public Note(double n1)`: Konstruktor zum Setzen der Note. Wirft eine `NumberFormatException`, falls die Note nicht existiert.
- `public Note(Note n2)`: Copy-Konstruktor
- `public String toString()`: Gibt die Note mit einer Stelle hinter dem Komma zurück.
- `public boolean hatBestanden()`: Gibt `true` zurück, falls die Note maximal 4.0 ist und ansonsten `false`.

Tutoriumsaufgabe 7.6

Schreiben Sie eine Klasse `Dominostein`, die einen Dominostein repräsentiert. Beide Seiten des Dominosteins können die Zahlen 0-6 enthalten. Fügen Sie der Klasse `Dominostein` folgende Methoden hinzu:

- `public Dominostein(int x, int y)`: Erzeugt einen Dominostein mit den gegebenen Werten. Wirft eine `NumberFormatException`, falls ein Wert außerhalb des Bereichs 1...6 liegt.
- `public Dominostein(Dominostein d2)`: Copy-Konstruktor.
- `public String toString()`: Erzeugt einen String wie im folgenden Beispiel: [1,2]
- `public void dreheUm()`: Dreht den Dominostein um, vertauscht also beide Seiten.
- `public int[] getValues()`: Gibt die Werte in einem `int`-Feld der Größe 2 zurück.