

2.3 Objektorientierte Programmierung mit Java

1.3

Snap



Benötigte IDEs:

BlueJ

Verfasser:

Niko Diamadis

Erstellungs-/ Änderungsdatum

30. Juni 2020

Inhaltsverzeichnis

1	Timer im Codepad benutzen	1
1.1	Spielregeln & Benutzung des Timers	1
1.2	'Testen'	1
1.3	'Problemlösung'	1
1.4	Privaten Zugriff	2
2	Fachkonzept - Kapselung	3
3	Übungen	4
3.1	Geburtstagsschwindler	4
3.2	private vs. public	4
3.3	Stilfrage	5

1 Timer im Codepad benutzen

1.1 Spielregeln & Benutzung des Timers

➤ **1. Zeile**

eine Timer-Variable `t` wird implementiert

➤ **2. Zeile**

der Variable wird ein Objekt zugewiesen

➤ **3. Zeile**

dem Integer `min` des Objekts `t` wird der Wert 2 zugewiesen

➤ **4. Zeile**

dem Integer `max` des Objekts `t` wird der Wert 10 zugewiesen

➤ **5. Zeile**

die Methode `starten` des Objekts `t` wird ohne Parameter aufgerufen

1.2 'Testen'

Zu dieser Aufgabe sind meines Erachtens nach keine Anleitungen und/oder Erläuterungen nötig.
Wenn doch Fragen aufkommen, schreib' einfach an nikodiamond3@gmail.com.

1.3 'Problemlösung'

Die Methode funktioniert folgendermaßen:

Bei Aufruf der Methode werden als Parameter der minimale und der maximale Wert angegeben.
Diese werden aber nur `min` und `max` zugewiesen, wenn `minimal` größer oder gleich 0 ist und wenn `maximal` kleiner oder gleich 0 ist.

Es ist jedoch immer noch möglich, die Attribute `min` und `max` über die Direkteingabe wieder zu verändern.

1.4 Privaten Zugriff

Zu dieser Aufgabe sind meines Erachtens nach keine Anleitungen und/oder Erläuterungen nötig.

Wenn doch Fragen aufkommen, schreib' einfach an **nikodiamond3@gmail.com**.

2 Fachkonzept - Kapselung

Zu dieser Seite sind meines Erachtens nach keine Anleitungen und/oder Erläuterungen nötig.

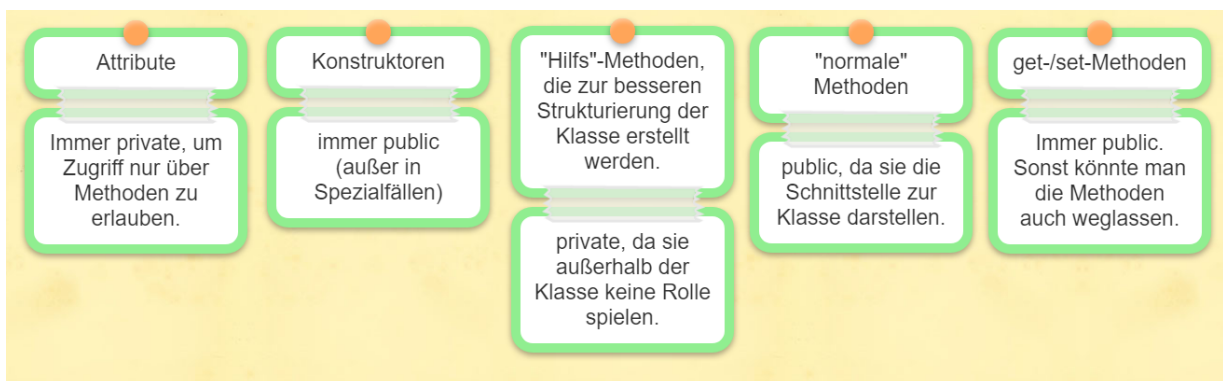
Wenn doch Fragen aufkommen, schreib' einfach an **nikodiamond3@gmail.com**.

3 Übungen

3.1 Geburtstagsschwindler

```
class Mensch
{
    private int alter;
    public Mensch()
    {
        alter = new java.util.Random().nextInt(100);
    }
    public int getAlter()
    {
        return Math.max(18, alter);
    }
    public void setAlter(int alt)
    {
        alter = Math.max(0, alt);
    }
}
```

3.2 private vs. public



3.3 Stilfrage

Dies ist eine gute Idee, da nur so sichergestellt werden kann, dass das Programm so wie geplant funktioniert. Eventuell sind es auch Daten, die die Nutzer nichts angeht oder womit sie womöglich nichts anfangen können.

Die Benutzung einer Klasse ändert sich durch Anpassen der Erreichbarkeit vonseiten des Nutzers dahingehend, dass alle Attribute privat sind und nur die Attribute, welche vom Nutzer geändert werden dürfen, von einer set-Methode beeinträchtigt werden.