
Programmieren in Java<http://proglang.informatik.uni-freiburg.de/teaching/java/2013/>

Java-Übung Blatt 0 (Daphne Test)

2013-04-16

Hinweise

- Schreiben Sie Identifier *genau so*, wie sie auf dem Blatt stehen (inklusive Groß- und Kleinschreibung), nicht nur ungefähr.
- Laden Sie Ihre Lösungen mit subversion (svn) ins Übungssystem hoch. Den entsprechenden Pfad finden Sie online.
- Das Übungssystem kann überprüfen, ob Sie Ihr Quelltext den Anforderungen genügt und ob Sie alle Klassen erstellt haben, etc.

Nutzen Sie dies!

- Sollte das Übungssystem Ihre Lösungen ablehnen, dann werden sie *nicht* korrigiert! Akzeptanzkriterien:
 - Compiliert erfolgreich
 - Checkstyle bringt keine Fehler
 - Alle Packages, Klassen, Interfaces, Methoden, Typen, Argumente sind exakt wie auf dem Übungsblatt gefordert.

Abgabe: Freitag, 26. April 2013, um 23.59 Uhr.

Dieses Übungsblatt dient zum Testen und Ausprobieren des Daphne Übungsportals. Eine genaue Beschreibung zur Benutzung finde Sie auf der Kurs-Homepage:

<http://proglang.informatik.uni-freiburg.de/teaching/java/2013/tools.html>

In dem Code, den Sie zusammen mit diesem Übungsblatt heruntergeladen haben, werden die folgenden zwei trivialen Aufgaben so gelöst, dass sie *nicht* den Abgabekriterien entsprechen. Ihre Aufgabe ist es nun, mit Hilfe der Anleitung auf der Homepage, die Abgabe zu verbessern, so dass sie akzeptiert wird. Das sollte Ihnen ein Gefühl für die Interpretation der Fehlermeldungen des Validierungsberichts geben.

Aufgabe 1 (HelloWorld)Projekt: `ex00_1`. Package: `helloworld`.

Implementieren Sie das folgende obligatorische Programm:

```
1 package helloworld;
2 /**
3  * Obligatory HelloWorld main class
4  * /
5 class Main {
6     /**
7      * The main method, doing the obligatory thing.
8      * @param args Command line arguments
9      */
10    public static void main(String[] args) {
11        System.out.println("Hello World!");
12    }
13 }
```

Aufgabe 2 (Akkumulator)Projekt: `ex00_2`. Package: `accumulator`.

Vervollständigen Sie das folgende Klassengerüst, so dass die mit der Methode `add` übergebenen Zahlen aufsummiert werden, und die aktuelle Summe über `getResult` ausgelesen werden kann.

```
1 package accumulator;
2 /**
3  * A class that accumulates a sum of integers.
4  */
5 class Acc {
6     /**
7      * Add an integer to the accumulator.
8      * @param i The integer to add.
9      */
10    public void add(int i) {
11        /* ... */
12    }
13    /**
14     * Get the accumulated result so far.
15     * @return The result.
16     */
17    public int getResult() {
18        /* ... */
19    }
20 }
```
