
Programmieren in Java

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

Java-Übung Blatt 1

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.

(Diese Übungsblatt wird *nicht* bewertet.)

Willkommen. Aufgabe 1 (Houston)

Projekt: `ex01_1`. Package: `rocket`. Erstellen Sie in Eclipse ein Java-Projekt namens `ex01_1`. Erstellen Sie in dem Projekt eine Klasse `Houston` im Package `rocket`. Die Klasse `Houston` soll eine main-Methode haben, die folgendes auf die Konsole schreibt:

```
10
9
8
7
6
5
4
3
2
1
Ignition!
```

Sie sollten also am Ende folgende Struktur haben:

```
1 package rocket;
2 /**
3  * Houston control center class.
4  * /
5 class Houston {
6     /**
7      * The countdown is started when calling the main method.
8      * @param args Command line arguments (not used here)
9      */
10    public static void main(String[] args) {
11        /* ... */
12    }
13 }
```

Aufgabe 2 (Primzahlen)

Projekt: `ex01_2`. Package: `prime`.

Erstellen Sie eine Klasse `Primes100` im Package `prime`. Sie soll eine main-Methode

haben, die alle Primzahlen, die kleiner als 100 sind, ausrechnet und auf die Konsole schreibt, und zwar immer 10 Primzahlen pro Zeile (bis auf die letzte Zeile: dort so viele wie übrig sind).

Ihre Struktur sollte also folgende sein:

```
1 package prime;
2 /**
3  * A class that prints all primes smaller 100.
4  */
5 class Primes100 {
6     /**
7      * Print all primes smaller 100.
8      * @param args Command line arguments.
9      */
10    public static void main(String[] args) {
11        /* ... */
12    }
13 }
```

Tipp: Sie brauchen lokale `int`-Variablen, `for`-Schleifen, `if`-Statements.

Aufgabe 3 (Ziffernhäufigkeit)

Projekt: `ex1_3`. Package: `digits`.

Erstellen Sie eine Klasse `DigitCount` im Package `digits`. Die Klasse soll für eine beim Aufruf als Argument übergebene Dezimalzahl berechnen, wie häufig die Ziffern von 0 bis 9 in der übergebenen Zahl vorkommen. Die jeweilige Anzahl soll auf die Konsole ausgegeben werden.

Ihre Struktur sollte also folgende sein:

```
1 package digits;
2 /**
3  * Counting all digits given in an input string.
4  */
5 class DigitCount {
6     /**
7      * Counts all decimals given in the first argument string.
8      * @param args Command line arguments.
9      */
10    public static void main(String[] args) {
11        /* ... */
12    }
13 }
```

Tipp: Sie können die Methode `Integer.parseInt` zum Konvertieren des Argumentstrings in einen `int`-Wert verwenden.