

**Einführung in die Programmierung**  
WS 2009/10

**Übungsblatt 1: Kommentare in Java, Zeichenreihen**

Besprechung: 02./04./05.11.2009

Ende der Abgabefrist: Montag, 02.11.2009 10:00 Uhr.

**Hinweise zur Abgabe:**

Übungsblätter dürfen NICHT in Teams abgegeben werden, da Sie sich durch eine erfolgreiche Bearbeitung bereits einen Bonus für die Klausur verdienen können. Es ist zwar sinnvoll in kleinen Teams die Aufgaben zu diskutieren, die Lösungen müssen aber von jedem Studenten EINZELN bearbeitet werden. Bitte beachten Sie, dass abgeschriebene Lösungen mit 0 Punkten bewertet werden!

Sammeln Sie die Lösungen zu diesem Übungsblatt (also `euklid.jar`, `p-adisch.txt` und `boole.txt`) in einem zip-Archiv `loesung01.zip`. Dieses zip-Archiv können Sie schließlich unter `http://www.pst.ifi.lmu.de/uniworx/` abgeben.

**Wichtig:** Achten Sie bitte darauf, dass Ihre Lösungsdateien die korrekten, d. h. die in der Angabe geforderten Namen haben, sonst kann Ihre Lösung nicht der richtigen Aufgabe zugeordnet werden. Java-Dateien, die nicht fehlerfrei kompilierbar sind, werden im Allgemeinen nicht korrigiert.

**Aufgabe 1-1**     *Kommentare in Java*

**10 Punkte**

- (a) Die Klasse `Euklid` (erhältlich über die Datei `Euklid.java`) implementiert den euklidischen Algorithmus, wie in der Vorlesung besprochen. Erweitern Sie diese Klasse sehr ausführlich um geeignete Kommentare. Die Kommentare sollten folgendes beinhalten:
- (i) Ein Dokumentationskommentar, der die gesamte Klasse beschreibt und Sie als Autor nennt.
  - (ii) Ein Dokumentationskommentar, der das Programm mit dem aktuellen Datum versioniert.
  - (iii) Ein Dokumentationskommentar, der die Main-Methode beschreibt. Den Parameter `args` können sie wie im Beispiel aus dem Vorlesungsskript dokumentieren.
  - (iv) Ein- oder mehrzeilige Kommentare an Stellen, an denen Sie es für sinnvoll halten.
- (b) Erzeugen Sie die Dokumentation in einem Unterverzeichnis `\doku`:
- ```
javadoc -d .\doku Euklid.java
```
- (c) Falls alles geklappt hat, erzeugen Sie eine weitere JAR-Datei, in der die dokumentierte Klasse `Euklid` und die erstellte Dokumentation zusammengefasst sind:
- ```
jar cvf euklid.jar Euklid.* .\doku
```

**Bemerkung:**

Um "schöne" javadoc-Kommentare zu schreiben, in denen zum Beispiel Formeln in mathematischer Notation angezeigt werden, ist es nötig, sich mit html auszukennen. Wir empfehlen deshalb zur Einarbeitung: <http://de.selfhtml.org/>.

**Aufgabe 1-2** *p-adische Zahlendarstellungen***10 Punkte**Die Datei `p-adisch.txt` enthält folgende Tabelle:

$p = 2$	$p = 8$	$p = 10$	$p = 16$
1101	15	13	D
		254	
			29A
110100001			
	60		

Ergänzen Sie die Tabelle so, dass in jeder Zeile die verschiedenen  $p$ -adischen Zahlendarstellungen für die selbe Zahl stehen.**Aufgabe 1-3** *Boolesche Algebra***5 Punkte**

In der Vorlesung wurde die Menge der booleschen Werte  $\mathbb{B} = \{TRUE, FALSE\}$  eingeführt. Funktionen auf diesem Wertebereich (boolesche Funktionen) werden häufig mit Hilfe von Wahrheitstabellen (oder Wahrheitstafeln) definiert. Unter einer Wahrheitstabelle versteht man die tabellarische Aufstellung des Wahrheitsverlaufs einer booleschen Funktion. Die Wahrheitstabelle zeigt für alle möglichen Zuordnungen von Wahrheitswerten, welchen Wahrheitswert die Gesamtaussage unter der jeweiligen Zuordnung annimmt.

- Die einstellige Funktion

$$\neg : \mathbb{B} \rightarrow \mathbb{B}$$

(gesprochen “nicht”) ist definiert durch folgende Wahrheitstabelle:

$x$	$\neg x$
<i>TRUE</i>	<i>FALSE</i>
<i>FALSE</i>	<i>TRUE</i>

Das bedeutet: Die Funktion  $\neg$  bildet den Wert *TRUE* auf den Wert *FALSE* und umgekehrt ab.

- Die zweistelligen Funktionen

$$\vee : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

(gesprochen “oder”) und

$$\wedge : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

(gesprochen “und”) sind definiert durch die Wahrheitstabelle:

$x$	$y$	$x \vee y$	$x \wedge y$
<i>FALSE</i>	<i>FALSE</i>	<i>FALSE</i>	<i>FALSE</i>
<i>FALSE</i>	<i>TRUE</i>	<i>TRUE</i>	<i>FALSE</i>
<i>TRUE</i>	<i>FALSE</i>	<i>TRUE</i>	<i>FALSE</i>
<i>TRUE</i>	<i>TRUE</i>	<i>TRUE</i>	<i>TRUE</i>

- Die zweistelligen Funktion

$$\Rightarrow: \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

(gesprochen “impliziert”) ist definiert durch folgende Wahrheitstabelle:

$x$	$y$	$x \Rightarrow y$
<i>FALSE</i>	<i>FALSE</i>	<i>TRUE</i>
<i>FALSE</i>	<i>TRUE</i>	<i>TRUE</i>
<i>TRUE</i>	<i>FALSE</i>	<i>FALSE</i>
<i>TRUE</i>	<i>TRUE</i>	<i>TRUE</i>

Gegeben sei nun folgende dreistellige Funktion:

$$f : \mathbb{B} \times \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

$$f(x, y, z) = ((\neg x) \Rightarrow (y \wedge z))$$

Klammern bezeichnen einen Vorrang, d.h., um den Wert von  $f$  zu bestimmen, ist zunächst der Wert von  $(\neg x)$  bzw.  $(y \wedge z)$  zu bestimmen.

Vervollständigen Sie folgende Tabelle aus Datei `boole.txt`:

$x$	$y$	$z$	$f$
<i>FALSE</i>	<i>FALSE</i>	<i>FALSE</i>	
<i>FALSE</i>	<i>FALSE</i>	<i>TRUE</i>	
<i>FALSE</i>	<i>TRUE</i>	<i>FALSE</i>	
<i>FALSE</i>	<i>TRUE</i>	<i>TRUE</i>	
<i>TRUE</i>	<i>FALSE</i>	<i>FALSE</i>	
<i>TRUE</i>	<i>FALSE</i>	<i>TRUE</i>	
<i>TRUE</i>	<i>TRUE</i>	<i>FALSE</i>	
<i>TRUE</i>	<i>TRUE</i>	<i>TRUE</i>	