

Einführung in die Programmierung für Kunst und Multimedia

Übungsblatt 3

Ende der Abgabefrist: 13.11.2013 12:00 Uhr

Hinweise zur Abgabe:

Übungsblätter dürfen NICHT in Teams abgegeben werden, da Sie sich durch eine erfolgreiche Bearbeitung 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 in einem zip-Archiv loesung03.zip. Dieses zip-Archiv können Sie schließlich in UniWorX abgeben.

Wichtig: Achten Sie bitte darauf, dass Ihre Lösungsdateien den korrekten Namen und das korrekte Format haben! Beides wird in der Angabe explizit angegeben. Dateien im falschen Format oder mit falschem Namen werden im Allgemeinen nicht korrigiert.

Aufgabe 3-1 Einfache Anweisungen in Java

5 Punkte

Ein KFZ-Besitzer möchte ein Java-Programm erstellen, das ihm vor jeder Fahrt den entstehenden Benzin- und Ölverbrauch berechnet. Der Benzinverbrauch des Fahrzeugs beträgt 6.7 Liter pro 100 km, der Ölverbrauch beträgt 0.6 Liter pro 1000 km.

Schreiben Sie in einer Klasse `Verbrauch` ein Java-Programm, das für eine gegebene Fahrtstrecke den entstehenden Benzin und Ölverbrauch berechnet und ausgibt. Testen Sie Ihr Programm mit den Fahrtstrecken 0.1km, 3km und 100.13 km.

Überlegen Sie zunächst, welche lokalen Variablen Sie benötigen (mit Typ). Achten Sie außerdem auf ausführliche Kommentare in Ihrem Code (sowohl Javadoc als auch beschreibende Kommentare).

Bitte geben Sie Ihre Lösung als kompilierbare Java-Datei `Verbrauch.java` ab. Bitte achten Sie unbedingt darauf, dass Ihre Lösung fehlerfrei und kompilierbar ist. **Es werden ausschließlich lauffähige Lösungen im korrekten Format bewertet.**

Aufgabe 3-2 Gültigkeitsbereiche**6 Punkte**

1. Gegeben sei folgender Java-Codeblock:

```
{
    int m = 13, n = 2, z = 0;
    m = m + n;
    {
        double x;
        x = m / n;
        int i = (int) x;
        z = i - n;
    }
    n++;
    int h = n + 1;
}
```

Bestimmen Sie den Gültigkeitsbereich von m, n, z, x, i und h.

2. Gegeben sei folgender Java-Codeblock:

```
{
    double a = 2.0, b = 3.4;
    {
        a = a + b;
        int c = (int) a;
        a = c;
    }
    a--;
    double d = a + b;
    boolean test = a != d;
}
```

Bestimmen Sie den Gültigkeitsbereich von a, b, c, d und test.

Bitte geben Sie Ihre Lösung als *aufgabe2.pdf* ab.

Aufgabe 3-3 EBNF-Grammatik

9 Punkte

Im Folgenden betrachten wir eine einfache Sprache, mit der man den Inhalt von Spielzeugkisten in einem Kinderzimmer beschreiben kann. Generell wird dabei eine Spielzeugkiste durch das Wort "Kiste" angegeben und anschließend wird in runden Klammern entweder "leer" geschrieben oder es wird der Inhalt der Spielzeugkiste aufgezählt. Der Inhalt einer Spielzeugkiste besteht aus einer Reihe von Gegenständen, wobei zwischen zwei Gegenständen immer angegeben werden soll, ob der erste "neben" oder "auf" dem zweiten liegt. Ein Gegenstand ist entweder ein Spielzeug oder wiederum eine Spielzeugkiste, die ihrerseits (wie oben beschrieben) angegeben wird.

Spielzeuge sind "Puppe", "Auto", "Ball" oder "Teddy".

Beispiele für gültige Ausdrücke dieser Sprache sind also:

- Kiste (leer)
- Kiste (Puppe neben Auto auf Kiste (Teddy neben Auto) neben Ball)

Ungültig sind Konstruktionen, in denen vor oder nach einer Präposition kein Gegenstand steht, also z.B.:

- Kiste (neben Auto)
- Kiste (Auto neben)

1. Definieren Sie eine EBNF-Grammatik für die oben beschriebene Sprache. Das Startsymbol soll den Namen Spielzeugkiste haben.
2. Leiten Sie mit der oben definierten EBNF-Grammatik folgenden Ausdruck ab:
Kiste (Auto auf Kiste (leer))
Hinweis: Sie können in jedem Schritt bei Anwendung einer Regel gegebenenfalls auf der rechten Seite der Regel vorkommende EBNF-Operatoren gleich mit auswerten.

Bitte geben Sie Ihre Lösung als *aufgabe3.pdf* ab.

Abgabe

Zulässige Dateiformate für die Lösung dieses Übungsblattes sind PDF und JAVA. Bitte geben Sie Ihre Lösung als ZIP-Datei bis zum 13.11.13 12:00 Uhr in UniWorX (<https://uniworx.ifi.lmu.de>) ab.

Hinweis: Verspätete Abgaben, Abgaben im falschen Dateiformat und nicht lauffähige Java-Dateien werden nicht bewertet.