

Einführung in die Programmierung für Kunst und Multimedia

Übungsblatt 5

Ende der Abgabefrist: 27.11.2013 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 loesung05.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 5-1 Arrays

12 Punkte

In dieser Aufgabe soll eine Zahl in ein bereits sortiertes Array an die richtige Stelle eingefügt werden. Da ein Array in Java nicht erweitert werden kann, muss das Array dazu in ein größeres (eine Stelle mehr) umkopiert werden.

Schreiben Sie dazu ein Programm `Array.java` mit einer Funktion

```
public static int[] sortInArray(int number, int[] array)
```

die eine übergebene ganze Zahl `number` in ein bereits sortiertes Integer-Array `array` einsortiert. Die Funktion gibt dann das neue erweiterte Array zurück. Rufen Sie Ihr Programm und Ihre Funktion mit folgender Main-Methode auf:

```
public static void main(String[] args) {  
    int[] numbers = new int[]{2, 5, 11, 17, 32, 47, 48, 59, 61, 73, 88};  
    int new_number = 35;  
    int[] new_numbers = sortInArray(new_number, numbers);  
}
```

Erweitern Sie Ihre main-Methode so, dass sie das neue Array `new_numbers` auf die Konsole ausgibt.

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

Aufgabe 5-2 Überprüfen von Ausdrücken

8 Punkte

In dieser Aufgabe sollen Sie Ausdrücke auf ihre Korrektheit untersuchen. Als Grundlage für Ausdrücke in Java verwenden wir die folgende EBNF-Grammatik. `Variable` und `Value` stehen dabei selbsterklärend für Variablen und Werte in Java, `PrimitiveType` steht für einen primitiven Datentyp, das sind alle Datentypen, die in Java direkt erstellt werden können, also z.B. `int`, `double` oder `boolean`.

```
Expression = Variable |
            Value |
            Expression BinOp Expression |
            UnOp Expression |
            "(" Expression ")"
```

```
BinOp      = "&" | "&&" | "|" | "||" | "+" | "-" | "*" | "/" | "%" | "=="
            | "!=" | "<" | "<=" | ">" | ">="
```

```
UnOp       = "!" | "-" | "+" | "(" PrimitiveType ")"
```

Gegeben seien folgende Ausdrücke:

1. `!(3 > 5)`
2. `!3 < 5`
3. `5 ! <= 5`
4. `14 < 3 || 1/5 == 4567`
5. `false && (false || true)`
6. `false & -1`
7. `5 < 7 == !`
8. `3 == (7 && 0) < 1`

- a) Geben Sie an, welche dieser Ausdrücke syntaktisch nicht korrekt sind (mit Begründung)
- b) Geben Sie an, welche dieser Ausdrücke nicht typkorrekt sind (mit Begründung)

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

Abgabe

Zulässige Dateiformate für die Lösungen dieses Übungsblattes sind JAVA und PDF. Bitte geben Sie Ihre Lösung als ZIP-Datei bis zum 27.11.13 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.