

2. Übung

Abgabe bis 06.11.2017, 10:00 Uhr

Bitte beachten:

Generelle Hinweise zu den Aufgaben und deren Bearbeitung finden Sie auf dem „organisatorischen Übungsblatt“. Sie können die Aufgaben erst im EST abgeben, nachdem Sie in die Tafelübungen eingeteilt wurden (voraussichtlich Sonntag).

Einzelaufgabe 2.1: Min, Max, Mittelwert und Median

6 EP

Laden Sie zunächst die Datei `Statistik.java` von der Übungshomepage und ergänzen Sie den Quellcode an den mit „TODO“ markierten Stellen gemäß den im Code enthaltenen Kommentaren.

WICHTIG: Sie dürfen in Ihrer Lösung **nichts** aus der Java-API verwenden – also insbesondere auch keine Methoden der Klasse `Math` oder `Arrays`!

Geben Sie Ihre geänderte Klasse `Statistik.java` über EST ab.

Einzelaufgabe 2.2: Faule Auswertung

7 EP

Die folgende Tabelle enthält in der ersten Spalte jeweils einen zusammengesetzten logischen Ausdruck in der Sprache Java, wobei a, b, c, d und e vom Datentyp `boolean` sind. Simulieren Sie die Ausführung jedes Ausdrucks jeweils mit den in der zweiten Spalte angegebenen Variablenbelegungen (f=false, t=true). Zu jedem Ausdruck ist in dieser Aufgabe der Wahrheitswert der in den folgenden beiden Spalten genannten Teilausdrücke anzugeben – falls der Teilausdruck *nicht* ausgewertet werden würde, kennzeichnen Sie dies mit „X“. Geben Sie schließlich noch den Wahrheitswert des Gesamtausdrucks an. Das erste Beispiel ist exemplarisch gelöst. Geben Sie Ihre Tabelle als `FauleAuswertung.pdf` ab.

| Zeile | Ausdruck | Variablenbelegung | atomare Teilausdrücke | nicht-atomare Teilausdrücke | Gesamtausdruck |
|-------|-----------------------|-------------------------|---|-----------------------------|----------------|
| 0 | (a && b) c | a=f, b=t, c=t | a: <u>f</u> ; b: <u>X</u> ; c: <u>t</u> | (a && b): <u>f</u> | <u>t</u> |
| 1 | | a=t, b=f, c=f | a; b; c | (a && b) | |
| 2 | (a b) && c | a=f, b=t, c=t | a; b; c | (a b) | |
| 3 | | a=t, b=t, c=f | a; b; c | (a b) | |
| 4 | | a=f, b=f, c=t | a; b; c | (a b) | |
| 5 | (a b) & (c d) | a=t, b=f, c=t, d=f | a; b; c; d | (a b); (c d) | |
| 6 | | a=f, b=f, c=f, d=t | a; b; c; d | (a b); (c d) | |
| 7 | | a=f, b=f, c=f, d=f | a; b; c; d | (a b); (c d) | |
| 8 | (a ^ b) (!c & d) | a=f, b=f, c=t, d=f | a; b; c; d | (a ^ b); (!c & d) | |
| 9 | | a=t, b=f, c=f, d=t | a; b; c; d | (a ^ b); (!c & d) | |
| 10 | | a=t, b=t, c=f, d=t | a; b; c; d | (a ^ b); (!c & d) | |
| 11 | !(a b) && (c d) | a=f, b=f, c=t, d=f | a; b; c; d | !(a b); (c d) | |
| 12 | | a=t, b=f, c=f, d=t | a; b; c; d | !(a b); (c d) | |
| 13 | | a=f, b=t, c=t, d=t | a; b; c; d | !(a b); (c d) | |
| 14 | a b && !c d & !e | a=f, b=f, c=t, d=f, e=f | c; e | a b && !c; d & !e | |

Einzelaufgabe 2.3: Spannende Widerstände

8 EP

In einem geschlossenen Stromkreis gilt das **Ohmsche Gesetz**:
(Spannung U , Widerstand R , Stromstärke I)

$$U = R \cdot I \quad (1)$$

Der Gesamtwiderstand einer Reihenschaltung berechnet sich dabei als:

$$R_g = R_1 + R_2 + \dots + R_n \quad (2)$$

Für eine Parallelschaltung von 2 Widerständen gilt

$$R_g = \frac{R_1 \cdot R_2}{R_1 + R_2} \quad (3)$$

Laden Sie zunächst die Datei `Ohm.java` von der Übungshomepage und ergänzen Sie den Quellcode an den mit „TODO“ markierten Stellen mit den Formeln (1), (2) und (3) gemäß den im Code enthaltenen Kommentaren. Geben Sie Ihre geänderte Klasse `Ohm.java` über EST ab.

Gruppenaufgabe 2.4: Variablen, Datentypen und Felder

17 GP

Laden Sie den vorgegeben Rumpf des Programms `Declarations.java` von der AuD-Homepage und ergänzen Sie die mit `// TODO:` markierten Stellen gemäß der dort angegebenen Kommentare. Testen Sie Ihre Änderungen mit dem bereitgestellten „Smoke Test“ und fügen Sie bei Bedarf weitere Tests hinzu.

Geben Sie Ihre fertig ergänzte Datei `Declarations.java` über EST ab.

Gruppenaufgabe 2.5: Syntaxdiagramme

22 GP

In den Vorlesungsfolien wird die Schreibweise einiger Java-Anweisungen in Form von Syntaxdiagrammen graphisch dargestellt. Leider fehlt in den Folien (aus gutem Grund) die Darstellung zusammengesetzter arithmetischer Ausdrücke (*ArithExp*), Vergleichsoperationen (*CompExp*) und boolescher Bedingungen (*BoolExp*). Daher sollen Sie diese nun selbst erstellen. Berücksichtigen Sie bitte, dass die Operatoren `&` bzw. `|` „überladen“ sind und sowohl für ganzzahlige (!) als auch für boolesche Ausdrücke funktionieren!

Nehmen Sie im folgenden vereinfachend an, dass zusätzlich zu den Diagrammen aus der Vorlesung noch Syntaxdiagramme zur Darstellung von Wertvariablen „*ValVar*“ (z.B. „merker“) sowie für Wertkonstanten „*Value*“ (z.B. „-42.0815“) vorgegeben sind.

- Stellen Sie das Syntaxdiagramm für beliebige zusammengesetzte arithmetische Ausdrücke in Java dar. Beschränken Sie sich dabei auf die Operatoren `+`, `-`, `*`, `/`, die beliebige Wertdatentypen verarbeiten können. Mit Ihrem Syntaxdiagramm soll es mitunter möglich sein, z.B. „`-(-merker - - -42.0815) / (((4711)))`“ (gültiger Java-Code) auszudrücken. Tipp: Auch eine einfache Zahl stellt bereits einen arithmetischen Ausdruck dar.
- Stellen Sie nun das Syntaxdiagramm für beliebige zusammengesetzte Vergleichsoperationen mit `<`, `<=`, `==`, `!=`, `>=`, `>` in Java unter Verwendung des vorangehend konstruierten Diagramms dar. Bedenken Sie, dass auch Wahrheitswerte mit `==` bzw. `!=` verglichen werden können. Berücksichtigen Sie außerdem die Java-Operatorpräzedenzregeln, damit nur gültige Java-Ausdrücke entstehen können. Nehmen Sie hier zunächst an, dass es schon ein Diagramm für boolesche Ausdrücke „*BoolExp*“ gibt, das gleich noch zu erstellen ist.

- c) Ergänzen Sie nun das Syntaxdiagramm für zusammengesetzte boolesche Bedingungen in Java. Darin können Vergleichsoperationen ebenso vorkommen, wie die Konstanten `true` bzw. `false`. Zusätzlich zu den binären booleschen Operatoren `&&`, `&`, `|`, `||` muss Ihr Diagramm auch die unäre Negation `!` beherrschen.

Geben Sie Ihre Diagramme als `Syntaxdiagramme.pdf` über EST ab.