

## Übungen zu Einführung in die Informatik: Programmierung und Software-Entwicklung

### Aufgabe 6-1                      Überprüfen und Auswerten von Ausdrücken                      Präsenz

In dieser Aufgabe sollen Sie Ausdrücke auf Korrektheit untersuchen und gegebenenfalls auswerten. Als Grundlage für Ausdrücke im Kontext von Klassendeklarationen verwenden wir folgende EBNF-Grammatik mit der entsprechenden Erweiterung von `Value` um das Terminalsymbol `"null"`:

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

```
Variable = NamedVariable | FieldAccess  
FieldAccess = Expression "." Identifier
```

```
MethodInvocation = Expression "." Identifier "(" [ActualParameters] ")"
```

```
InstanceCreation = ClassInstanceCreation  
ClassInstanceCreation = "new" ClassType "(" [ActualParameters] ")"
```

Zusätzlich verwenden wir die Klassendeklarationen von `Point` und `Line` aus der Vorlesung. Seien `Point p`; `Line l`; und `int i`; lokale Variablendeklarationen. Gegeben seien folgende Ausdrücke:

1. `l.x`
2. `p.x.y`
3. `p.move(1,2)`
4. `l.length()`
5. `p == null`
6. `p != null`
7. `i == null`
8. `new point(3,4)`
9. `(new Point(1,1)).x`

10. `new Point(1,1).x`

- 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).
- c) Gegeben sei der Zustand  $(\sigma, \eta)$  mit

$$\begin{aligned}\sigma &= [(i, 3), (p, @123), (l, @789)] \\ \eta &= \{ < (@123, Point), [(x, 2), (y, 4)] >, < (@456, Point), [(x, 0), (y, 0)] >, \\ &\quad < (@789, Line), [(start, @123), (end, @456)] > \}\end{aligned}$$

Werten Sie alle syntaktisch korrekten und typkorrekten Ausdrücke bezüglich dieses Zustands aus. Der Scope der Ausdrücke ist dabei jeweils so gewählt, dass die Attribute und Methoden der entsprechenden Klasse sichtbar sind.

### Aufgabe 6-2      Überprüfen und Auswerten von Ausdrücken      Hausaufgabe

In dieser Aufgabe sollen Sie Ausdrücke auf Korrektheit untersuchen und gegebenenfalls auswerten. Als Grundlage für Ausdrücke im Kontext von Klassendeklarationen verwenden wir die EBNF-Grammatik aus Aufgabe 6-1.

Zusätzlich verwenden wir die Klassendeklarationen von `Point` und `Line` aus der Vorlesung. Seien `Point p`; `Line l`; und `int i`; lokale Variablendeklarationen. Gegeben seien folgende Ausdrücke:

1. `i.y`
2. `p.i`
3. `l.start.y`
4. `p.GetX()`
5. `l.move(p)`
6. `i == p`
7. `new Line(3,4)`
8. `Point(3,4)`
9. `(new Line(new Point(i,i), p)).end.getY()`

- 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).
- c) Gegeben sei der Zustand  $(\sigma, \eta)$  mit

$$\begin{aligned}\sigma &= [(i, 3), (p, @123), (l, @789)] \\ \eta &= \{ < (@123, Point), [(x, 2), (y, 4)] >, < (@456, Point), [(x, 0), (y, 0)] >, \\ &\quad < (@789, Line), [(start, @123), (end, @456)] > \}\end{aligned}$$

Werten Sie alle syntaktisch korrekten und typkorrekten Ausdrücke bezüglich dieses Zustands aus. Der Scope der Ausdrücke ist dabei jeweils so gewählt, dass die Attribute und Methoden der entsprechenden Klasse sichtbar sind.

In dieser Aufgabe soll eine Klasse `Figur` deklariert werden, die einfache geometrische Figuren repräsentiert. Beispielsweise können ein Rechteck oder ein Kreis eine geometrische Figur sein. Jede geometrische Figur hat einen Mittelpunkt vom Typ `Point` und eine Farbe vom Typ `String`. Außerdem kann man angeben, ob die Figur mit dieser Farbe ausgefüllt ist.

a) Deklarieren Sie eine Klasse `Figur`, die oben genannte Eigenschaften definiert. Die Klasse soll die folgenden Konstruktoren und zwei Methoden anbieten:

- einen Konstruktor mit formalen Parametern zur Initialisierung der Attribute.
- einen Konstruktor mit formalen Parametern zur Initialisierung der Attribute; für den Mittelpunkt sollen als Parameter die x- und y-Koordinate übergeben werden.
- eine Methode `bewegen` ohne Rückgabetyt, die den Mittelpunkt soweit verschiebt, wie die beiden formalen Parameter `dx` und `dy` es fordern.
- eine Methode `getMittelpunkt()` mit dem Rückgabetyt `Point`, die den Mittelpunkt der Figur zurückgibt.

*Hinweis: Um die Klasse `Point` verwenden zu können, speichern Sie die Datei `Point.java` von der Vorlesungswebseite in den gleichen Ordner wie Ihre Klasse `Figur`. Diese Datei enthält die Klassendeklaration der Klasse `Point`, wie sie in der Vorlesung gezeigt wurde.*

b) Mit folgendem Programmcode können Sie Ihre Klasse `Figur` testen (dieser Code muss in einer Datei `FigurTest.java` im gleichen Ordner wie Ihre Klasse `Figur` gespeichert werden):

```
1 public class FigurTest {
2     public static void main(String[] args) {
3         Point p = new Point(2, 1);
4         Figur figur1 = new Figur(p, "rot", true);
5         Figur figur2 = new Figur(p, "schwarz", false);
6
7         System.out.println("figur1: Mittelpunkt=("
8             + figur1.getMittelpunkt().getX() + ","
9             + figur1.getMittelpunkt().getY() + ")");
10        System.out.println("figur2: Mittelpunkt=("
11            + figur2.getMittelpunkt().getX() + ","
12            + figur2.getMittelpunkt().getY() + ")");
13
14        figur1.bewegen(2, 2);
15
16        System.out.println("figur1: Mittelpunkt=("
17            + figur1.getMittelpunkt().getX() + ","
18            + figur1.getMittelpunkt().getY() + ")");
19        System.out.println("figur2: Mittelpunkt=("
20            + figur2.getMittelpunkt().getX() + ","
21            + figur2.getMittelpunkt().getY() + ")");
22    }
23 }
```

Geben Sie für dieses Programm den Zustand des Speichers (Stack und Heap) nach Zeile 5 und nach Zeile 14 an. Sie können den Zustand entweder in der abstrakten Syntax angeben (siehe Aufgabe 6-1 und 6-2) oder die grafische Repräsentation verwenden.

c) Welches Problem fällt Ihnen bei der Analyse der Zustandsänderung auf und wie können Sie es beheben?

In dieser Aufgabe soll eine Klasse `Fahrzeug` deklariert werden, die beliebige Fortbewegungsmittel repräsentiert. Beispielsweise können ein Fahrrad, ein Auto oder eine Eisenbahn ein Fahrzeug sein. Jedes Fahrzeug hat eine aktuelle Position vom Typ `Point`, eine bestimmte Anzahl an Rädern, ein Leergewicht und eine aktuelle Geschwindigkeit.

- a) Deklarieren Sie eine Klasse `Fahrzeug`, die oben genannte Eigenschaften definiert. Die Klasse soll den folgenden Konstruktor und zwei Methoden anbieten:
- einen Konstruktor mit formalen Parametern zur Initialisierung der Attribute; für den Mittelpunkt sollen als Parameter die x- und y-Koordinate übergeben werden.
  - eine Methode `beschleunigen` ohne Rückgabetypp, die die aktuelle Geschwindigkeit um einen bestimmten Wert erhöht bzw. erniedrigt (bei negativer Beschleunigung). Der Wert soll als Parameter übergeben werden können.
  - eine Methode `getAktuelleGeschwindigkeit` mit dem Rückgabetypp `double`, die die aktuelle Geschwindigkeit des Fahrzeugs zurückgibt.

*Hinweis: Um die Klasse `Point` verwenden zu können, speichern Sie die Datei `Point.java` von der Vorlesungswebseite in den gleichen Ordner wie Ihre Klasse `Fahrzeug`. Diese Datei enthält die Klassendeklaration der Klasse `Point`, wie sie in der Vorlesung gezeigt wurde.*

- b) Mit folgendem Programmcode können Sie Ihre Klasse `Fahrzeug` testen (dieser Code muss in einer Datei `FahrzeugTest.java` im gleichen Ordner wie Ihre Klasse `Fahrzeug` gespeichert werden):

```
1 public class FahrzeugTest {
2     public static void main(String[] args) {
3         Fahrzeug auto = new Fahrzeug(1, 2, 4, 1234.5, 100.0);
4         Fahrzeug fahrrad = new Fahrzeug(3, 4, 2, 33.3, 18.3);
5
6         System.out.println("Auto: aktuelle Geschwindigkeit="
7             + auto.getAktuelleGeschwindigkeit());
8         System.out.println("Fahrrad: aktuelle Geschwindigkeit="
9             + fahrrad.getAktuelleGeschwindigkeit());
10
11        auto.beschleunigen(11.1);
12
13        System.out.println("Auto: aktuelle Geschwindigkeit="
14            + auto.getAktuelleGeschwindigkeit());
15        System.out.println("Fahrrad: aktuelle Geschwindigkeit="
16            + fahrrad.getAktuelleGeschwindigkeit());
17    }
18 }
```

Geben Sie für dieses Programm den Zustand des Speichers (Stack und Heap) nach Zeile 4 und nach Zeile 11 an. Sie können den Zustand entweder in der abstrakten Syntax angeben (siehe Aufgabe 6-1 und 6-2) oder die grafische Repräsentation von Hand malen und abfotografieren oder ein beliebiges Zeichenprogramm (z.B. PowerPoint) verwenden. Bitte geben Sie nur `.txt`-, `.jpg`- oder `.pdf`-Dateien ab!

*Besprechung der Präsenzaufgaben in den Übungen ab 24.11.2011. Abgabe der Hausaufgaben bis Mittwoch, 07.12.2011, 14:00 Uhr über UniworX (siehe Folien der ersten Zentralübung).*

- *Für Textaufgaben geben Sie bitte eine `.txt`-, `.jpg`- oder `.pdf`-Datei ab!*
- *Erstellen Sie zu jeder Aufgabe eine Klasse, die den Namen trägt, der in der Aufgabe gefordert ist. Geben Sie nur die entsprechenden `.java`-Dateien ab. Wir benötigen **nicht** Ihre `.class`-Dateien.*