

## 11. Übungsblatt

**Aufgabe 1** Jack: Variablen und Referenzen

- a) Gegeben seien die beiden unten gezeigten Programme in der Programmiersprache Jack. Wie unterscheiden sich die beiden Programme in ihren Abläufen und Ausgaben? Was zeigt dies über die Argumentübergabe an Funktionen?

```

class Main {
    function void f (int a) {
        a = 2;
        return;
    }

    function void main {
        var int x;

        x = 1;
        f(x);
        Output.printInt(x);

        return;
    }
}

class Main {
    function void f (int[] a) {
        a[0] = 2;
        return;
    }

    function void main {
        var int[] x;
        x = Array.new(1);
        x[0] = 1;
        f(x);
        Output.printInt(x[0]);
        x.dispose();
        return;
    }
}

```

- b) Was gibt das folgende Jack-Programm aus?

```

class Main {
    function void main {
        var int[] a, b;
        a = Array.new(1);
        a[0] = 2;
        b = a;
        b[0] = a[0] * b[0];
        a[0] = a[0] + b[0];
        Output.printInt(b[0]);
        b.dispose();
        return;
    }
}

```

**Aufgabe 2** Jack: Parse-Bäume

- a) Geben Sie einen Parse-Baum für den folgenden Ausdruck an:

$$\text{sqrt}((x+2)*3-\text{min}(4,i))+\text{pi}$$

b) Geben Sie einen Parse-Baum für den folgenden Jack-Quelltext an:

```
class SimpleMath {
  function int negative (int x) {
    if (x < 0) {          // wenn x negativ ist,
      return -1;        // gib 'wahr' zurück,
    } else {             // sonst
      return 0;         // gib 'falsch' zurück
    }
  }
} // class SimpleMath
```

### Aufgabe 3 Jack: Übersetzung und Symboltabellen

a) Geben Sie an, welche Symboltabellen es während der Übersetzung des unten gezeigten Jack-Programms gibt, wenn die Übersetzung an den durch einen Pfeil markierten Stellen angekommen ist! Welchen Inhalt haben diese Symboltabellen?

```
class Car {
  field String model;
  field int mileage;
  field int speed;

  static int nCars;

  constructor Car new (String m) {
    var int initialMileage;
    nCars = nCars + 1;
    model = m;
    initialMileage = 0;
    ⇒ mileage = initialMileage;
    speed = 0;
    return this;
  }

  function int accelerate (int delta, int speedLimit) {
    var int d;
    if (delta > 10) d = 10;
    ⇒ else d = delta;
    speed = speed + d;
    if (speed > speedLimit) {
      speed = speedLimit;
    }
  }

  // more code
} // class Car
```

b) Was läßt sich anhand dieser Symboltabellen über die Anweisungen sagen, die der zweiten markierten Stelle (in der Funktion `accelerate`) direkt folgen?