

## Übungen zu Einführung in die Informatik I

### Aufgabe 16      **Übersetzung von MiniJava nach MiniJVM**

- a) Übersetzen Sie das folgende MiniJava Programm in ein MiniJVM Programm **nach den Regeln** aus der Vorlesung. Die Variable  $z$  soll dabei in Speicherzelle 0, Variable  $n$  in Speicherzelle 1, Variable  $j$  in Speicherzelle 2, Variable  $y$  in Speicherzelle 3 und Variable  $x$  in Speicherzelle 4 stehen.

```
int z, n, j, y, x;
read(x);
read(n);
z = 1;
while (0<n) {
    j = 1;
    y = x;
    while (2*j<=n) {
        y = y*y;
        j = j*2;
    }
    z = y*z;
    n = n-j;
}
write(z);
```

- b) Was berechnet das Programm?  
Hinweis: Das Ergebnis der Berechnung steht in der Variablen  $z$ .

### Aufgabe 17      **Die Klasse Buch in Java und UML**

Ein Buch ist charakterisiert durch den Namen des Autors, den Buchtitel und den Verlagsdaten. Die Verlagsdaten umfassen den Namen des Verlages, die Nummer der Auflage und das Erscheinungsjahr.

- a) Stellen Sie die Klasse `Buch` in UML dar und definieren Sie sie in Java. Geben Sie dabei auch eine Konstruktordefinition an.
- b) Es soll ein Objekt der Klasse `Buch` erzeugt werden, das ein Buch Ihrer Wahl repräsentiert. Stellen Sie das Objekt in UML dar und instanziiieren Sie es in Java. Das Objekt sollte mit einem geeigneten Namen bezeichnet werden.

## Aufgabe 18      **Modellierung und Implementation eines Bücherregals**

In vorherigen Aufgabe haben wir bereits die Klasse `Buch` modelliert und implementiert. Nun wollen wir dazu ein Bücherregal realisieren.

Modellieren und implementieren Sie eine Datenstruktur `BuecherRegal`, die die folgende Funktionalität bereit stellt:

- Man kann ein leeres Bücherregal anlegen.
- Man kann nach einem Buch mit einem bestimmten Titel suchen.
- Man kann ein weiteres Buch ins Regal stellen. Die dazu geeignete Stelle könnte etwa ganz links, ganz rechts oder an einer anzugebenden Position in der Mitte sein.
- Man kann auch ein Buch, das man gefunden hat, aus dem Regal entnehmen.

Identifizieren Sie aus dieser Beschreibung die Methoden, die eine Klasse `BuecherRegal` zur Verfügung stellen sollte. Überlegen Sie sich, welche Attribute zur Bereitstellung dieser Methoden nötig sind. Modellieren Sie Ihr Bücherregal mit einem Klassendiagramm in UML.

Stützen Sie sich bei Ihrer Implementierung auf die Datenstrukturen `Buch` aus der vorherigen Aufgabe und `List` aus der Vorlesung.

### **Regeln aus der Vorlesung zur Übersetzung von MiniJava nach MiniJVM:**

<code>int x<sub>0</sub>, ..., x<sub>n-1</sub>;</code>	=	ALLOC n
<code>x</code>	=	LOAD i      — sofern x die i-te Variable ist.
<code>17</code>	=	CONST 17
<code>- expr</code>	=	Übersetzung von <code>expr</code> NEG
<code>expr<sub>1</sub> + expr<sub>2</sub></code>	=	Übersetzung von <code>expr<sub>1</sub></code> Übersetzung von <code>expr<sub>2</sub></code> ADD
<code>x = expr;</code>	=	Übersetzung von <code>expr</code> STORE i      — sofern x die i-te Variable ist.
<code>x = read();</code>	=	READ STORE i      — sofern x die i-te Variable ist.
<code>write( expr );</code>	=	Übersetzung von <code>expr</code> WRITE
<code>if ( cond ) stmt<sub>1</sub> else stmt<sub>2</sub></code>	=	Übersetzung von <code>cond</code> FJUMP A Übersetzung von <code>stmt<sub>1</sub></code> JUMP B A: Übersetzung von <code>stmt<sub>2</sub></code> B: ...
<code>while ( cond ) stmt<sub>1</sub></code>	=	A: Übersetzung von <code>cond</code> FJUMP B Übersetzung von <code>stmt<sub>1</sub></code> JUMP A B: ...
<code>stmt<sub>1</sub> ... stmt<sub>k</sub></code>	=	Übersetzung von <code>stmt<sub>1</sub></code> ... Übersetzung von <code>stmt<sub>k</sub></code>
<code>prog</code>	=	ALLOC n Übersetzung von <code>ss</code> HALT