



Vorbereitung zum Praktikum Compilerbau WS 2019/2020

Ziel: Einführung in Mini-Java und Java-Bytecode.

Lesen Sie sich außerdem das Dokument „CB19Prak0 Einführung.docx“ gut durch.

Aufgabe 1: (Gültigen und ungültigen Mini-Java Code erkennen)

Geben Sie an, ob der jeweilige Mini-Java Code gültig ist:

- a) `final int WOCHENTAGE = 7;`
`int heute = 89, woche;`
`{`
`woche = (heute / WOCHENTAGE) + 1;`
`}`
- b) `final int WOCHENTAGE = 7;`
`int heute = 89;`
`int naechsteWoche;`
`{`
`naechsteWoche = heute + WOCHENTAGE;`
`}`
- c) `print(42);`
- d) `final int MO=0, DI=1, MI=2, DO=3, FR=4, SA=5, SO=6;`
`int chillen, wochenTag = 0;`
`{`
`if (wochenTag >= 5)`
`{`
`chillen = 1;`
`print (1 + 11);`
`}`
`else`
`chillen = 0;`
`}`
- e) `final int WOCHENTAGE = 7;`
`int chillen, wochenTag = 0;`
`{`
`while wochenTag < WOCHENTAGE`
`{`
`if wochenTag < 5`
`{`
`chillen = 0;`
`}`
`if wochenTag >= 5`
`{`
`chillen = 1;`
`print (chillen);`
`wochenTag = wochenTag + 1;`
`}`
`}`
`}`

Vorbereitung zum Praktikum Compilerbau WS 2019/2020

Aufgabe 2: (Java-Bytecode verstehen)

Schreiben Sie in Mini-Java auf was der folgende Bytecode macht:

```
10 00
36 01
15 01
10 0a
a2 00 0d
15 01
10 01
60
36 01
a7 ff f2
b1
```

Aufgabe 3: (JDKs javap kennenlernen)

Schreiben und kompilieren Sie folgendes Programm in einer Entwicklungsumgebung Ihrer Wahl:

```
public static void main(String[] args) {
    final int WOCHENTAGE = 7;
    int heute = 245, woche;
    woche = (heute / WOCHENTAGE) + 1;
}
```

Öffnen Sie eine Kommandozeile im Ordner, in dem sich die kompilierte *.class* Datei Ihrer Funktion befindet und geben Sie in der Konsole folgendes ein:

javap -v „Dateiname“ (ignorieren Sie dabei die Dateiergung *.class*).

- Finden Sie nun in der Ausgabe Ihren Code in Bytecodedarstellung.
- Seien Sie in der Lage diesen erklären zu können (diese Ausgabe wird Ihnen in Praktikum 5 und 6 bei der Fehlerdiagnose weiterhelfen).
- Wie lang ist *Ihr Code* in Bytes?

Aufgabe 4: (Java-Bytecode schreiben)

Übersetzen Sie den vorgegebenen Code in den entsprechenden Bytecode in hexadezimaler Form auf:

Hinweis: *var0, var1, ..., varn* stellt die lokale Variable mit Index 0, 1, ..., n dar.

Hinweis für c): Die Reihenfolge der Operationen entspricht der Postfixnotation (s. Kellerautomat aus TILO)

- Rechenoperation
5 - 3;
- Wertzuweisung
var0 = var1;
- Komplexe Rechenoperation
*var0 = 9 + (4 - 2) * 5;*
- Vergleichsoperation
if var0 > 5
{



Vorbereitung zum Praktikum Compilerbau WS 2019/2020

```
    var0--;  
}  
var1 = 100;
```

Aufgabe 5: (Programmieren in Java-Bytecode)

Die Faktoriätsfunktion $fak: Nat \rightarrow Nat$ ist wie folgt rekursiv programmiert:

$$fak(0) = 1$$
$$fak(n) = n * fak(n-1), \text{ falls } n > 0$$

Schreiben Sie ein Programm in Java-Bytecode, das die Faktoriätsfunktion berechnet, wobei der Eingabewert als Konstante im Programm definiert wird und das Ergebnis der Rückgabewert der Funktion ist.