

Vorbereitung zum Praktikum Compilerbau WS 2019/2020

Einführung:

Die JVM verwendet bei der Ausführung von Bytecode einen virtuellen Stack als auch mehrere virtuelle Register. Der Stack wird verwendet um Operation wie mathematische Berechnungen oder Funktionsaufrufe durchzuführen. Die Register werden wie lokale Variablen verwendet, um Werte zu speichern oder Parameter für Funktionsaufrufe zu übergeben.

Jedes Byte des Bytecodes hat einen aufsteigenden Indexwert beginnend bei null. Diese Indizierung ist besonders bei Sprüngen im Code wichtig. Soll von Byte 2 ein Sprung nach Byte 8 durchgeführt werden, muss ein Offset von 6 angegeben werden (z.B. goto-Anweisung mit zwei Bytes Offset: a7 00 06). Die Bytes, die den Offset angeben, gehören selbst mit dazu. Das heißt, dass im obigen Beispiel effektiv 3 Bytes Code übersprungen werden.

Java arbeitet grundsätzlich mit *signed* Zahlenwerte. Ist also das erste Bit – das sogenannte sign-Bit – einer Zahl gesetzt, erkennt die JVM diese als negative Zahl. Die negative Wertefolge startet dabei bei null und wird von da aus heruntergezählt. Die Zahl -1 entspricht demnach als 8-Bit Zahl dargestellt dem Wert 1111 1111 (FF).

Hinweis: Der Bytecode wird zur besseren Lesbarkeit in byteweiser hexadezimaler Darstellung angegeben.

Die Vorgehensweise einer JVM beim Durchlaufen eines Programms soll an folgendem Beispiel veranschaulicht werden. Nehmen Sie an dieser Stelle das Hilfsblatt dazu:

```
x = 3;
if x < 6 {
    x = x + 5;
}
```

Bytecode Beschreibung

10 03	Der Wert 3 wird auf den (leeren) Stack abgelegt. Nun liegt ein Wert auf dem Stack.
36 00	Der oberste Wert auf dem Stack wird entfernt und in die lokale Variable mit dem Index 0 (bzw. Register 0) geschrieben.
15 00	Der Wert der Variable mit dem Index 0 wird auf den Stack gelegt.
10 06	Der Wert 6 wird auf den Stack abgelegt.
a2 00 0a	Vergleich von $x < 6$: Logik auf größer/gleich (if_icmpge) ändern, um beim Wahrheitswert <i>true</i> das Statement innerhalb der Condition zu überspringen
15 00	Der Wert der Variable mit dem Index 0 wird auf den Stack gelegt.
10 05	Der Wert 5 wird auf den Stack abgelegt.
60	Die obersten beiden Werte vom Stack werden entfernt. Die beiden Werte (3 und 5) werden addiert und das Ergebnis (8) wird auf den Stack gelegt. Der Stack enthält nun einen Wert weniger als vorher.
36 00	Der oberste Wert vom Stack wird entfernt und in die lokale Variable mit dem Index 0 (bzw. Register 0) geschrieben.