

FOP Reference Sheet

Jonas Milkovits

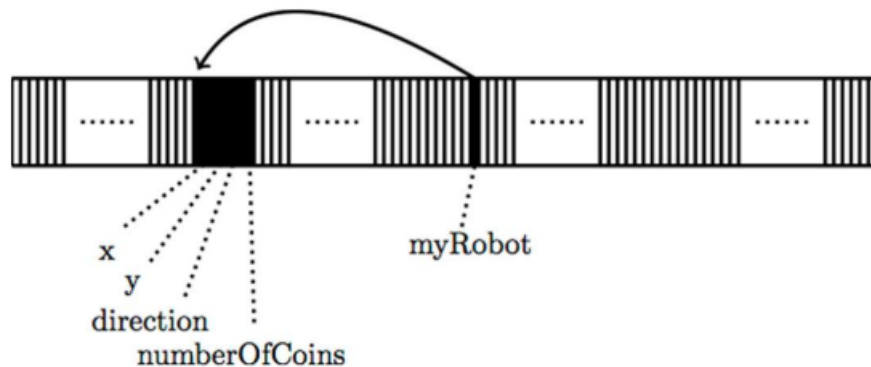
Last Edited: 23. März 2020

Inhaltsverzeichnis

1	Computerspeicher	1
2	Datentypen	1
3	Klassen und Methoden	2
4	Konversionen	2
5	Programme und Prozesse	2
5.1	Schleifen und if	2
6	Syntax	2

1 Computerspeicher

Unsere Vorstellung	▷ großes Feld aus Maschinenwörtern mit eindeutiger Adresse
Erzeugung eines neuen Objekts	▷ Reservierung von ungenutztem Speicher in ausreichender Größe
Referenz	▷ Name der Variable, die die Anfangsadresse des Objekts speichert ▷ Kann auch an komplett anderer Stelle als das Objekt gespeichert sein
Speicherort primitiver Datentypen	▷ Name verweist tatsächlich auf Speicherstelle, an der Wert abgespeichert wird
Prozessablauf	▷ Program Counter enthält Adresse der nächsten Anweisung ⇒ Zählt nach jeder Anwendung hoch und verweist auf nächsten Speicher ▷ CPU verarbeitet parallel die momentane Anweisung aus Program Counter



2 Datentypen

Konstanten	▷ Variable/Referenz wird dadurch unveränderbar ▷ z.B.: <code>final myClass ABC = new myClass();</code> ⇒ Referenz zwar nicht veränderbar, Objekt aber schon ▷ <code>Integer.MAX_VALUE</code> / <code>Integer.MIN_VALUE</code> ▷ Unendlich: <code>Double.POSITIVE_INFINITY</code> / <code>Double.NEGATIVE_INFINITY</code>
Primitive Datentypen	▷ Ganze Zahlen: <code>byte</code> → <code>short</code> → <code>int</code> → <code>long</code> ▷ Gebrochene Zahlen: <code>float</code> → <code>double</code> ▷ Logik: <code>boolean</code> ▷ Zeichen: <code>char</code>
Literale	▷ wörtlich hingeschriebene Werte eines Datentyps ▷ Zahlen standardmäßig <code>int</code> , falls <code>long</code> gewünscht: <code>123L</code> oder <code>123l</code> ▷ Bei gebrochenen <code>double</code> , falls <code>float</code> gewünscht: <code>12.3F</code> oder <code>12.3f</code>
Boolean	▷ nur <code>true</code> und <code>false</code> ▷ Negation <code>!a</code> ▷ Logisches Und: <code>a && b</code> ▷ Logisches Oder: <code>a b</code> (inklusive) ▷ Gleichheit: <code>a == b</code>
Zeichentyp char	▷ z.B.: <code>char c = 'a';</code> ▷ Interne Kodierung als Unicode ▷ <code>'\t'</code> Horizontaler Tab ▷ <code>'\b'</code> Backspace ▷ <code>'\n'</code> Neue Zeile ▷ Auch Darstellung im Hexacode (<code>'\u0039A'</code>)

3 Klassen und Methoden

4 Konversionen

Implizit	▷ Immer möglich, wenn kein Informationsverlust entstehen kann ▷ z.B.: kleinerer Datentyp in größeren
Explizit	▷ Meist Informationsverlust ▷ Durchführung durch Angabe des Datentyps in Klammern davor ▷ z.B.: <code>int i = (int)testDouble;</code>

5 Programme und Prozesse

Quelltest	▷ z.B. selbst geschriebener Java-Code
Java-Bytecode	▷ Wird durch Übersetzung des Java-Quelltextes erzeugt
Programm	▷ Sequenz von Informationen
Aufruf eines Programms	▷ Starten eines Prozesses, der die Anweisungen des Programmes abarbeitet
Prozesse	▷ CPU besteht aus mehreren Prozessorkernen ▷ Mehrere Prozesse laufen dementsprechend parallel ▷ Allerdings bearbeitet jeder Kern nur einen Prozess gleichzeitig (sehr kurz) ⇒ Illusion von Multitasking
	▷

6 Schleifen und if

while-Schleife	▷ <code>while (Bedingung) {Anweisung;}</code> ▷ Schleife wird ausgeführt, solange die Bedingung wahr ist ▷ <code>{}</code> kann bei einzelner Anweisung auch weggelassen werden
do-while-Schleife	▷ <code>do {Anweisung;} while (Bedingung);</code> ▷ Anweisungsblock wird immer mindestens einmal ausgeführt
for-Schleife	▷ <code>for (Anweisung davor; Bedingung; Anweisung danach) {Anweisung}</code> ▷ z.B.: <code>for (int i = 0; i < 10; i++) {...}</code> ⇒ Zehnmalige Ausführung der Anweisung
if-Anweisung	▷ <code>if (Bedingung) {...}</code> ▷ Führt den Code in der Anweisung nur aus, falls die Bedingung erfüllt ist ▷ <code>if (Bedingung) {} else {}</code> ▷ Code, der ausgeführt wird, falls Bedingung nicht erfüllt ist

7 Syntax

Keywords	▷ Können nur an bestimmten Stellen im Code stehen ▷ z.B. <code>class</code> , <code>import</code> , <code>public</code> , <code>while</code> ,...
Identifizier	▷ Namen für Klassen, Variablen, Methoden,... ▷ Erstes Zeichen darf keine Ziffer sein ▷ Keine Keywords als Identifizier ▷ Identifizier sind case-sensitive
Konventionen	▷ Variablen / Methoden beginnen mit Kleinbuchstaben (<code>testInt</code>) ▷ Klassen beginnen mit Großbuchstaben (<code>testClass</code>) ▷ Wortanfänge im Inneren mit Großbuchstaben ▷ Konstanten bestehen aus <code>_</code> und Großbuchstaben (<code>CENTS_PER_EURO</code>)
Kommentare	▷ <code>//</code> Einzelne Zeile ▷ <code>/*...*/</code> Mehrere Zeilen ▷ <code>/**...*/</code> Erzeugung von Javadoc