12.12.2024 Was ich über Strings gelernt habe Dominik Labocha

Was ist ein String?

Objekte der finales Klasse String (java.lang.String) → Nicht vererbbar
Nutzt das Interface java.lang.Charsequence → Zeichenfolge

Wie wird ein String erstellt bzw. zusammengestellt?

Diverse Konstruktoren neben String() mit folgenden Parametern in der Klammer:

- String str, also durch Übergabe eines bereits vorhandenen Strings
- byte[] byte, ein Array von byte-Werten bezogen auf den Systemzeichensatz
- char[] value, also ein char-Array, welches durch den String umhüllt wird.

Durch Konkatenation von Strings oder den Literalen primitiver Datentypen als String:

Konkatenation ist die Verkettung von Zeichen oder weiteren Zeichenketten. Das Pluszeichen wird dabei Überladen, solange mindestens ein Element der Verkettung ein nativer String ist, und setzt alles zu einem String zusammen, anstelle einer arithmetischen Addition. Wird eine Addition bei Bildung gebraucht, muss sie in Klammern gesetzt werden, es sei denn sie wird vor Auftauchen des ersten Strings ausgeführt.

Primitive Daten werden im Hintergrund umhüllt und die toString()-Methode der jeweiligen Wrapper-Klasse darauf ausgeführt, um den String zu erhalten. Ist ein anzufügendes Objekt, das das primitive Literal codiert, gleich null, wird der String „null“ eingesetzt. Jedes Objekt in Java hat eine toString-Methode, weil sie in der obersten Klasse Object enthalten ist; diese erzeugt kryptische Zeichenfolgen, demnach sollte man sie, wenn benötigt, überschreiben. Auch möglich ist ein += wie beim Inkrementieren von Zahlen.

Ironisch: Object o = 1 → Integer.toString() (wenn konkateniert), also o + „3“ = „13“, läuft. Object m = „String“ → Compiler: m? Ist das wirklich ein String? → m += „1“ → Error!

Weitere Erkenntnisse

Escape-Sequenzen haben nur die Länge eins und passen somit auch in char-Variablen.

Es gibt einen String Pool im Heap für Literale, die als Referenz den Objekten zugewiesen werden. Existiert ein String mit demselben Inhalt, ist die Referenz gleich, nicht sein Wert! Referenz ist nicht gleich, wenn new String() benutzt wird oder das Literal nach Konkatenation keinem Wert im String Pool entspricht, also selbst eines Hardcode String Literals. s1=“hello“, s2=“hello“, s1 == s2 (true), aber s3=s1+s1 und s4= s1+s1 → s3 != s4 (false)

Strings sind nicht veränderbar. Änderungen werden nur inline als neuer String performt.

Beim Vergleich von Stringvariablen mit Literalen ist folgendes geläufig: „Text“.equals.(s).

compareTo() gibt einen int zurück, der die alphabetische Verschiebung zueinander zeigt. Dies könnte nützlich sein, um zum Beispiel Namen zu sortieren. (0gleich, -vor, +danach)

System.out.printf() → System.out.format() →java.io.PrintStream: var in java.lang.system

Die String Klasse kann das auch selbst: String.format(„Hallo, %s“, name); (kein OutPrint)

Kommentiert [DL1]: Wenn aber s4 = s3.intern() --> gleiche Referenz

Kommentiert [DL2R1]: Intern tut folgendes: Schaut ob die Referenz schon existiert und legt sie an, falls sie fehlt. In beiden Fällen wird sie zurückgegeben. Sprich hat s3 noch keinen Eintrag im String-Pool, wird er erzeugt und dieser dann auch an s4 vergeben.

Kommentiert [DL3]: Weil es den Wert „hellohello“ nicht als Konstante im Stringpool gibt