

7.7.1 Kontrollfragen

• Was ist der Unterschied zwischen syntaktischer und semantischer Information?

Syntaktisch:

- Entropie einer Nachricht
- vergleichsweise einfach, z.B. von Compiler auf Korrektheit überprüfbar

Semantisch:

- Bedeutung einer Nachricht
- komplex und schwerer zu erfassen
- syntaktisch korrektes Programm auf semantischer Ebene möglicherweise Fehler

• Wie ist Informationsgehalt auf syntaktischer Ebene quantifizierbar? Auf semantischer?

Syntaktisch:

Syntaktische Information ist durch die Entropie quantifizierbar, durch Kenntnis des Alphabets und der Auftretswahrscheinlichkeiten. Sie steckt dadurch objektiv in der Nachricht und braucht keine Interpretation durch einen Empfänger.

Semantisch:

Sie ist nicht quantifizierbar, da verschiedene Interpretationen möglich sind und auf der semantischen Ebene keine einfach objektivierbaren Wahrscheinlichkeiten zugeordnet werden können.

• Was ist Entropie im Kontext der Informationstheorie?

Die Entropie entspricht der „Unsicherheit“ oder „Unbestimmtheit“ der Nachrichtenquelle. Sie entspricht dem erwarteten Informationsgehalt einer Nachricht, noch bevor diese empfangen wurde.

• Wodurch wird die Entropie bestimmt?

Da vor dem Eintreffen der Nachricht die Auswahl noch nicht getroffen wurde, müssen die Wahrscheinlichkeiten aller erlaubten Zeichen des Alphabets berücksichtigt werden. Die Entropie ist also gleich dem Mittelwert der Informationsgehalte aller erlaubten Zeichen.

• Wie hängt die Entropie mit der Codelänge zusammen? Unter welchen Annahmen gilt dieser Zusammenhang?

- **Was ist Redundanz?**

Die Differenz

$$R = L - H$$

wird Redundanz genannt und gibt an um wie viel bit ein Codewort im Mittel länger ist, als es im optimalen Fall sein müsste.

R ... Redundanz

L ... mittlere Wortlänge

H ... Entropie

- **Was ist die Präfix-Bedingung?**

Damit ein Code diese Eigenschaft hat, darf kein Codewort Anfangswort eines anderen Codeworts sein.

- **Wie ist eine Fließkommazahl aufgebaut?**

VZ | Exponent | Mantisse

Die Basis der Darstellung ist innerhalb eines Fließkomma-Zahlensystems fix und muss daher nicht zusätzlich gespeichert werden. Der Exponent legt die Position des Kommas fest.

- **Was ist *Plain-Text*?**

Ein Text, als eine Folge abstrakter Zeichen, wird oft auch *Plain Text* ("reiner Text") genannt. Die Repräsentation von *Plain Text* soll gerade so viel Information beinhalten, wie zur lesbaren Darstellung des Textes notwendig ist, und nicht mehr. Formatierung, Schriftart und Schriftgröße gehören nicht dazu.

- **Was ist *Unicode*?**

Ein Zeichensatz, wobei Zeichen hier durch eine Bitfolge der Länge 16 (2 Byte) dargestellt wurden. Damit lassen sich $2^{16} = 65536$ verschiedene Werte darstellen. Die ersten 128 Zeichen (0–127) entsprechen den Zeichen des ASCII-Codes

- **Was bedeutet UTF?**

UnicodeTransformation-Formats(UTF), die je nachdem wie viele Bits im Codewort zu einer Einheit zusammengefasst werden, UTF-8, UTF-16 und UTF-32 genannt werden

- **In welcher Weise werden relative Häufigkeiten von Zeichen in einem Text bei der Codierung der Zeichen berücksichtigt?**

Durch Datenkompression lassen sich Codierungen von Nachrichten der Länge l erzeugen, deren Länge kleiner ist als die durch die Entropie gegebene Minimallänge $H \cdot l$. Das wird dadurch erreicht, dass die Zeichen nicht einzeln binär codiert werden, sondern adaptiv, d.h., es werden abhängig von den der zu codierenden Zeichenfolge bestimmte wiederholt auftretende Zeichengruppen zu Superzeichen zusammengefasst.

- **Für welche Art von Daten verwendet die JVM den *Heap*, wofür den *Stack*?**

Stack:

Der Stack dient im Wesentlichen zum Speichern von lokalen Variablen und Parametern von Methoden. Handelt es sich dabei um Variablen mit Referenztyp, enthält die Variable eine Referenz auf das referenzierte Objekt. Die Objekte selbst befinden sich jedoch nicht im Stack sondern im Heap.

Heap:

Aufgabe des Heap ist es, alle dynamisch erzeugten Daten zu speichern. Das sind alle Objekte, also alles, was mit dem new-Operator erzeugt wird, darunter auch Strings und alle Arten von Arrays. Wird ein Objekt erzeugt, wird eine Referenz auf das im Heap neu angelegte Objekt zurückgeliefert.

- **Was ist die *Method-Area*?**

Die Method-Area ist ein (abhängig von der Implementierung der JVM oftmals innerhalb des Heap angelegter) Speicherbereich für alle statischen Daten, das sind also alle Variablen, die pro Klasse existieren (alle static deklarierten Variablen)

- **Zu welchem Zeitpunkt (Laufzeit oder Compilezeit) werden lokale Variablen aufgelöst?**
Compilezeit

- **Wie funktioniert das dynamische Binden mittels VFT?**

- **Was ist der Konstantenpool?**

Im sogenannten Konstantenpool dienen die Namen als Unicode-Zeichenketten (UTF-8-Code) als Schlüsseln, um damit die Offsets und Adressen der entsprechenden Variablen und Methoden ermitteln zu können. Durch bestimmte Klassen und Methoden im Java-System kann man über Namen, die erst zur Laufzeit ermittelt werden, auf Variablen zugreifen und Methoden aufrufen.

- **Warum haben Untertypen einen höheren Informationsgehalt als deren Obertypen?**

Untertypen beschreiben Daten auf genauere Weise als Obertypen und verfügen dazu im Allgemeinen über zusätzliche Objektvariablen und Methoden bzw. überschreiben Methoden des Obertyps so, dass das Verhalten auf genauere Weise spezifiziert wird. Durch die Präzisierung einer Beschreibung im Untertyp erhöht sich der Informationsgehalt.

- **Welchen Möglichkeiten zur Serialisierung von Objekten gibt es?**

toString, OutputStream, Writer