



Glossar
zu den drei Bänden
Informatik mit Java
Eine Einführung mit BlueJ
und der Bibliothek
Stifte und Mäuse
von
Bernhard Schriek
zusammengestellt von

Friedrich Hattendorf

17. Februar 2014

Bitte beachten !!!

Dieses Papier ist noch unfertig.

Fehler? fehlt etwas?

Schicken Sie bitte eine Mail an

friedrich@hattendoerfer.de

Inhaltsverzeichnis

1 Schriek Band	5
1.2 Klassen und Objekte	6
1.3 Aufbau eines SuM-Programms	8
1.4 Kontrollstrukturen I	10
1.5 Kontrollstrukturen II	12
1.6 Eigene Klassen 1	13
1.7 Eigene Klassen 2	15
1.8 Abstrakte Klassen 1	16
1.9 Abstrakte Klassen 2	18
1.10 Ereignisorientierung	19
1.11 Ereignisbearbeiter	20
1.12 Phasen eines Softwareentwurfs	21
1.13 Model View Controller	22
2 Schriek Band II	23
2.2 Rekursion	24
2.3 Felder	25
2.4 Lineare Strukturen I - Schlangen	26
2.5 Lineare Strukturen II - Listen	26
2.6 Lineare Strukturen III - Stapel	27
2.7 Binäre Bäume	27
2.8 Sortierverfahren	28
2.9 Tiefensuche - Backtracking	29
2.10 Breitensuche	29
3 Schriek Band III	31
3.2 Netzwerkprogrammierung I	32
3.3 Netzwerkprogrammierung II	32
3.4 Netzwerkprogrammierung III	33
3.4.1 SQL-Datenbanken I	34
3.4.2 SQL-Datenbanken II	34
3.4.3 Kryptographie	35
3.4.4 Akzeptoren	36
3.4.5 Scanner und Parser	36

Kapitel 1

Schriek Band

Jeweils Abschnitt „Neue Begriffe in diesem Kapitel“
(ggf. mit einigen Erweiterungen)

Bernhard Schriek:
Informatik mit Java
Band 1
ISBN 978-3-00-017092-8

1.2 Klassen und Objekte

Klasse

Eine Klasse ist der Bauplan für ein oder mehrere Objekte. In einer Klasse werden Dienste zur Verfügung gestellt. Klassen-Namen beginnen mit einem Großbuchstaben.

Objekt

Ein Objekt ist die Realisierung einer Klasse. Jedes Objekt gehört zu einer Klasse. Die Dienste eines Objekts können aufgerufen werden. Objektnamen beginnen mit Kleinbuchstaben.

Exemplar

Ein Exemplar (Instanz) einer Klasse ist eine Realisierung dieser Klasse, also ein Objekt.

Instanz

→ Exemplar

Dienst

Dienste beschreiben Fähigkeiten von Objekten. Dienste werden in Klassen definiert. Es gibt zwei Arten von Diensten: Anfragen und Aufträge. Dienstbezeichner beginnen mit einem Kleinbuchstaben und enden mit einer evtl. leeren Klammer für Parameter.

Anfrage

Eine Anfrage ist ein Dienst, den eine Klasse für ihre Objekte zur Verfügung stellt. Eine Anfrage liefert ein Ergebnis zurück.

Auftrag

Ein Auftrag ist ein Dienst, den eine Klasse für ihre Objekte zur Verfügung stellt. Wenn ein Objekt einen Auftrag erhält, tut es etwas, es liefert kein Ergebnis zurück.

Konstruktor

Der Konstruktor ist ein spezieller Dienst einer Klasse, der bei der Erzeugung eines Objekts dieser Klasse ausgeführt wird. Er wird mit dem Schlüsselwort new aufgerufen. Viele Klassen haben mehrere Konstruktoren, die sich in der Anzahl oder der Art ihrer Parameter unterscheiden.

Werkbank

Die Werkbank (engl. work bench) ist der untere Bereich des Projektfensters. Dort werden mit new erzeugte Objekte dargestellt. Diese Objekte kann man dann inspizieren und ihnen direkt Anfragen und Aufträge schicken.

Punktnotation

Wenn der Dienst eines Objekts aufgerufen wird, sagt man auch, dass dem Objekt eine Botschaft geschickt wird. Dazu benutzt man die Schreibweise Objekt.Dienst(evtl. Parameter), die als Punktnotation bezeichnet wird.

Parameter

Parameter ermöglichen es, Dienste flexibel zu gestalten. Parameter werden in Klammern hinter den Dienst geschrieben und beeinflussen die Art, wie der Dienst ausgeführt wird.

CamelCase

Schreibweise für Bezeichner in Java, die aus mehreren Worten zusammengesetzt sind.

int

ganze Zahl z.B.: -123

double

Dezimalzahl (Kommazahl) z.B. 3.1415

void

direkt übersetzt: Lücke, nichts. Gibt an, dass ein Dienst keinen Wert zurück liefert, also ein Auftrag ist.

new

Aufruf des Konstruktors einer Klasse. Dient zum Erzeugen eines neuen Objekts.

1.3 Aufbau eines SuM-Programms

Programm-Vorlage

(Template) Viele Anweisungen kommen in jedem Programm vor. Deshalb bietet BlueJ die Möglichkeit diesen Text in eine Vorlage zu schreiben, die dann automatisch erzeugt wird. Dadurch vermeidet man Fehler und muss nicht soviel schreiben.

Template

→ Programm-Vorlage

Uebersetzen

Ein Java-Programm kann erst ausgeführt werden, wenn es in den sogenannten Bytecode (für uns nicht lesbar) übersetzt wurde. Dieser Bytecode wird dann von einem Programm, der sogenannten Virtuellen Maschine (VM) ausgeführt.

Programm uebersetzen

→ Uebersetzen

Programm starten

→ Buch, Bd.I, Kap. 3.3

- zuerst übersetzen, (Editor schließen ist bei mir nicht notwendig, ob das unter Windows auch so ist, müssten Sie ausprobieren)
- im BlueJ- Fenster: Rechtsklick auf „Mein Programm“
- Rechtsklick auf „new Mein Programm()“
- Im Dialogfenster den Instanznamen „MeinProg1“ bestätigen
- öffnet sich einmal das SuM-Fenster, in dem nachher gezeichnet wird und unten links in der Werkbank ein rotes Objekt-Symbol
- dort Rechtsklick und dann „void fuehreAus,,“

Editor

Spezielles Textverarbeitungsprogramm mit Unterstützung der Java-Syntax durch Färbung bestimmter Worte. Im Editor schreibt und ändert man Programme.

Einruecken

Allgemein übliche Methode zur optischen Verdeutlichung der Programmstruktur. Sollte unbedingt eingehalten werden.

Kommentar

Eine Kommentarzeile beginnt mit einem Doppelslash '//' . Kommentare werden bei der Programmübersetzung ignoriert und dienen nur der besseren Lesbarkeit. Sparen Sie nicht an Kommentaren!

Fehlermeldung

Beim Übersetzen werden oft Schreib- und Syntaxfehler erkannt. BlueJ versucht die passenden Fehlermeldungen auszugeben. Es gibt die Möglichkeit, Fehlermeldungen ausführlich anzuzeigen.

Barberpole

Balken unten links im Projektfenster. Wenn er rot schraffiert ist, zeigt er an, dass gerade ein Programm ausgeführt oder übersetzt wird. In Amerika war ein senkrechter Barberpole auf der Straße das Symbol für einen Friseurladen (barber shop).

import

Anweisung, um Klassen zur Verfügung zu stellen, die in einer Bibliothek im Bytecode zur Verfügung stehen. Der Quelltext ist dazu nicht notwendig.

public

öffentlich, Markierung für Klassen oder Dienste, die von „außen“ benutzt werden sollen.

class

Schlüsselwort für die Deklaration einer Klasse.

new

Schlüsselwort zur Erzeugung eines neuen Objekts. Dahinter muss der Konstruktor einer Klasse aufgerufen werden

=

Zuweisung, wird als wird zu ausgesprochen. Der Bezeichner links erhält den Wert des Ausdrucks rechts.

1.4 Kontrollstrukturen I

Kontrollstruktur

Sprachkonstrukt um den Programmfluss (die Reihenfolge der Anweisungen) zu steuern. Es gibt Schleifen und Verzweigungen.

Schleife

Wenn eine Reihe von Anweisungen mehrmals hintereinander ausgeführt werden soll, benutzt man dazu eine Schleife. Die Durchlaufbedingung kontrolliert, ob die Schleife noch mal durchlaufen werden soll. Wenn Sie am Anfang steht, wird die Schleife eventuell gar nicht durchlaufen (abweisende Schleife). Wenn sie am Ende steht, wird die Schleife mindestens einmal durchlaufen.

Durchlaufbedingung

Sie dient zur Kontrolle, ob eine Schleife noch mal durchlaufen werden muss. Die Bedingung muss einen Wahrheitswert (wahr bzw. falsch) liefern.

Verzweigung

Abhängig von einer Bedingung wird entschieden, welche Anweisungen ausgeführt werden. Es gibt einseitige, zweiseitige und mehrseitige Verzweigungen.

Struktogramm

Es dient dazu, eine Kontrollanweisung grafisch darzustellen. Struktogramme können verschachtelt werden, da ihre Form immer ein Rechteck ist. Der Text im Struktogramm kann in Java oder Pseudocode geschrieben sein.

Pseudocode

Wenn ein Programm(ausschnitt) in Umgangssprache und nicht in Java geschrieben ist, nennt man dies Pseudocode.

ist-Beziehung

Zwischen Klassen kann eine ist-Beziehung bestehen. Die eine Klasse nennt man dann Oberklasse, die andere Unterkasse. Oberklassen sind allgemeiner, Unterklassen sind spezieller.

Vererbung

Unterklassen erben die Dienste ihrer Oberklassen. Diese Dienste können aber auch angepasst werden.

Tastaturpuffer

Die in die Tastatur getippten Zeichen kommen in eine Warteschlange, den Tastaturpuffer. Sie bleiben im Puffer bis sie daraus entfernt werden.

Farbe

Die Farbe eines Objektes wird mit `setzeFarbe(Farbe.<Farbname>)` bzw. `setzeFarbe(<Farbnummer>)` geändert.

SCHWARZ	0	MAGENTA	7
BLAU	1	ORANGE	8
CYAN	2	PINK	9
DUNKELGRAU	3	ROT	10
GRAU	4	WEISS	11
GRUEN	5	GELB	12
HELLGRAU	6		

while

(deutsch solange) leitet die Durchlaufbedingung einer Schleife ein. Die Bedingung muss in Klammern stehen.

if

(deutsch wenn) steht vor der Bedingung zur ein- oder zweiseitigen Verzweigung. Die Bedingung muss eingeklammert werden.

else

steht vor dem Alternativteil der zweiseitigen Verzweigung. Eine einseitige Verzweigung hat keinen else-Teil.

switch

leitet eine Mehrfachverzweigung ein.

case

leitet einen Fall der Mehrfachverzweigung ein.

break

beendet einen Teil der Mehrfachverzweigung.

default

behandelt alle nicht mit case behandelten Fälle einer Mehrfachverzweigung

.

==

ist der Vergleichsoperator. Er besteht aus zwei Gleichheitszeichen und wird als gleich ausgesprochen.

char

steht für Zeichen (engl. character). Zeichen werden zwischen Hochkommata gesetzt.

1.5 Kontrollstrukturen II

Animation

Bewegung einer Zeichnung durch wiederholtes Löschen und Neuzeichnen an einer etwas verschobenen Stelle

Zufallszahl

Zahl, deren Wert man bei der Erzeugung nicht vorhersagen kann. Zufallszahlen können ganze Zahlen (evtl. in einem bestimmten Bereich, auch negativ!) sein oder Dezimalzahlen zwischen 0 und 1. Zufallszahlen werden mit der Klasse Rechner erzeugt.

Stoppuhr

Dient zur Zeitmessung. Man kann eine Stoppuhr starten, anhalten und die gestoppte Zeit abfragen. Man kann auch die verflossene Zeit seit Erzeugung der Uhr abfragen.

Zusammengesetzte Bedingung

Eine Bedingung kann aus mehreren Teilbedingungen zusammengesetzt sein. Als Verknüpfungen dient der und- bzw. oder-Operator. Eine mit und verknüpfte Bedingung ist nur dann erfüllt (wahr), wenn sämtliche Teilbedingungen wahr sind. Eine mit oder verknüpfte Bedingung ist schon dann wahr, wenn eine Teilbedingung wahr ist. Die und-Verknüpfung bindet stärker als die oder-Verknüpfung, nicht bindet stärker als und. (Ähnliches gilt in der Mathematik: Punktrechnung geht vor Strichrechnung.)

&&

ist der *und*-Operator und verknüpft zwei Bedingungen bzw. Wahrheitswerte.

&& wird als *und* gesprochen.

||

ist der *oder*-Operator und verknüpft zwei Bedingungen bzw. Wahrheitswerte.

|| wird als *oder* gesprochen.

1.6 Eigene Klassen 1

hat-Beziehung

Wenn ein Objekt ein anderes Objekt erzeugt und verwaltet, so spricht man von einer hat-Beziehung. Solche Bezugsobjekte erkennt man am Wort hat vor dem Objektnamen z.B. hatStift. Bei einer hat-Beziehung ist die besitzende Klasse für die Erzeugung und Freigabe der besessenen Objekte verantwortlich. Eine hat-Beziehung wird im Beziehungsdiagramm durch eine Verbindungsline mit einer gefüllten Raute bei der besitzenden Klasse dargestellt.

Bezugsobjekt

Zu Beginn der Klassendefinition werden die Bezugsobjekte deklariert. Dabei handelt es sich um hat- oder kennt-Beziehungen.

Attribut

Eine Eigenschaft eines Objekts bezeichnet man als Attribut. Verschiedene Objekte einer Klasse haben verschiedene Ausprägungen des Attributs. Ein Attribut ist z.B. die Farbe einer Kugel. Attribute sind keine Objekte, sondern primitive Datentypen wie int oder boolean. Ihre Bezeichner beginnen mit einem z (Zustand) wie zFarbe.

kennt-Beziehung

Wenn ein Objekt ein anderes Objekt im Konstruktor oder einem anderen Dienst kennen lernt, so spricht man von einer kennt-Beziehung. So kann das Objekt auf Dienste des kennt-Objekts zugreifen. Solche Bezugsobjekte erkennt man an dem Wort kennt vor dem Objektnamen, z.B. kenntBildschirm. Eine kennt-Beziehung wird im Beziehungsdiagramm durch eine Verbindungsline mit einer Pfeilspitze beim bekannten Objekt dargestellt.

Zustandsvariable

ist ein anderes Wort für Attribut.

Klassendiagramm

Ein Klassendiagramm ist ein Rechteck mit drei Feldern. Oben wird der Name der Klasse eingetragen, danach folgen die Attribute und zum Schluss kommen die Dienste. So erhält man in knapper Form Informationen über die Klasse.

Beziehungsdiagramm

Im Beziehungsdiagramm werden die ist-, hat- und kennt-Beziehungen zwischen Klassen durch Symbole gekennzeichnet.

Implementierung

Die Klasse wird mit Programmtext gefüllt. Die Implementierung liegt in der Hand des Programmierers, andere Programmierer, die die Klasse verwenden, erhalten nur eine Schnittstellenbeschreibung. Andere Objekte können nur über die Schnittstelle, nicht direkt zugreifen (Geheimnisprinzip).

Schnittstelle

Beschreibung einer Klasse sowie der Dienste, die eine Klasse anderen Objekten zur Verfügung stellt. Die Klassen der SuM-Bibliothek und der Java-

Bibliothek sind vorübersetzt, d.h. nicht im Quelltext verfügbar. Um sie benutzen zu können, muss ihre Schnittstelle bekannt sein. Dazu dient die Dokumentation, die über das Hilfemenü aufgerufen wird.

schreibender Zugriff auf ein Attribut

Dienste, die einen schreibenden (verändernden) Zugriff auf ein privates Attribut einer Klasse ermöglichen, werden üblicherweise mit setzeAttributsbezeichner bezeichnet.

lesender Zugriff auf ein Attribut

Dienste, die einen lesenden (nicht verändernden) Zugriff auf ein privates Attribut einer Klasse ermöglichen, werden üblicherweise mit dem klein geschriebenen attributsbezeichner() bezeichnet. Da Attribute generell als privat gekennzeichnet werden, erfolgt ein Zugriff nur über diese Dienste.

formaler Parameter

Ein formaler Parameter wird in der Kopfzeile eines Dienstes in Klammern deklariert. Sein Bezeichner beginnt mit einem 'p' und davor muss sein Typ angegeben werden.

aktueller Parameter

Der aktuelle Parameter wird beim Aufruf des Dienstes hinter dem Dienstbezeichner in Klammern angegeben. Er enthält einen konkreten Wert oder einen Term, aus dem der entsprechende Wert berechnet wird.

extends

wird bei der Deklaration einer Unterklassie benutzt. Hinter extends steht dann der Name der Oberklasse.

boolean

bedeutet Wahrheitswert (true oder false). Nach George Boole (1815 - 1864)

super(...)

steht als erste Anweisung im Konstruktor einer Unterklassie. Damit wird der Konstruktor der Oberklasse aufgerufen.

super.dienst(...)

ist der Aufruf des Dienstes dienst der Oberklasse. Die Anweisung super.dienst(...) kann im Gegensatz zum Aufruf des Konstruktors der Oberklasse super(...) an beliebiger Stelle stehen. Diese Anweisung wird typischerweise dann benutzt, wenn ein Dienst der Oberklasse in der Unterklassie erweitert wird.

this(...)

ist eine Referenz auf das aktuelle Objekt

1.7 Eigene Klassen 2

Laufzeitfehler

Wenn ein Fehler während der Übersetzung des Programms nicht erkannt wird, also erst während der Programmausführung auftritt, nennt man diesen Fehler Laufzeitfehler (engl. runtime-error). Solche Fehler werden auch als logische Fehler bezeichnet, während man Übersetzungsfehler als Syntaxfehler bezeichnet. Diese logischen Fehler sind das größte Problem bei der Softwareentwicklung. Man geht davon aus, dass auch ein guter Programmierer je 1000 Programmzeilen einen logischen Fehler macht. Um solche Fehler frühzeitig zu finden, sind umfangreiche Tests notwendig. Es gibt dazu spezielle Testumgebungen, die Sie in einem späteren Kapitel kennen lernen werden.

Nullpointer-Exception

Häufig auftretender Laufzeitfehler. Ein Objekt, das noch nicht erzeugt wurde, hat den Wert null. Dies bedeutet soviel wie „nichts“. Wenn man einem solchen nicht existierendem Objekt einen Auftrag oder eine Anfrage schickt, entsteht eine Nullpointer-Exception.

null

ist der Wert für ein nicht existierendes Objekt. Wenn in Java ein Objekt deklariert wird, erhält es automatisch zuerst einmal den Wert null. Man kann einem Objekt diesen Wert aber auch direkt zuweisen, z.B. der Ball = null

1.8 Abstrakte Klassen 1

Substantiv-Verb-Methode

Beim objektorientierten Entwurf bietet die Substantiv-Verb-Methode eine Möglichkeit Klassen und Dienste zu finden. In der Aufgabenstellung werden die Substantive gesucht und überlegt, ob diese sich als Klassenbezeichner eignen. Die Verben der Aufgabenstellung sind Kandidaten für mögliche Dienste.

dynamische Referenz

Wenn zu einem Objektbezeichner kein neues Objekt erzeugt wird, er stattdessen durch eine Zuweisung einem schon existierenden Objekt zugeordnet wird, nennt man dies einen dynamische Referenz (Verweis). Somit verweisen zwei oder mehrere Objektbezeichner auf das gleiche Objekt.

abstrakte Klasse

Generalisierung (Verallgemeinerung) einer Klasse durch Bildung einer Oberklasse, die nicht als Objekt realisiert wird. Eine Klasse muss als abstrakt deklariert werden, wenn sie abstrakte Dienste enthält.

abstrakter Dienst

Ein abstrakter Dienst wird in der Klasse nicht implementiert. Die Deklaration des Dienstes endet mit ;. Der Dienst muss in einer Unterklasse implementiert werden. Man sagt auch, der Dienst ist aufgeschoben.

statischer Typ - dynamischer Typ

Zu einer Oberklasse wird ein Objekt deklariert, aber nicht erzeugt. Der statische Typ ist dann die Oberklasse. Wenn diesem Objekt dann ein Objekt einer Unterklasse zugewiesen wird, so nennt man die Unterklasse den dynamischen Typ des Objekts. Es heißt dynamisch, weil dem Bezeichner ein Objekt einer anderen Unterklasse zugewiesen werden kann und sich somit der dynamische Typ ändert.

Polymorphie

Ein Dienst einer Oberklasse wird als polymorph bezeichnet, wenn er in Unterklassen unterschiedlich realisiert ist. Das bedeutet, bei gleichem Dienstbezeichner werden abhängig von der Unterklasse unterschiedliche Implementierungen ausgeführt.

späte Bindung

Da bei der Programmübersetzung eines polymorphen Dienstes noch nicht entschieden wird, welcher konkrete Dienst später ausgeführt wird, geschieht dies erst während der Laufzeit des Programms. Daher spricht man hier von später Bindung.

Sichtbarkeit

Es gibt drei Arten der Sichtbarkeit von Attributen und Diensten einer Klasse: private bedeutet, dass diese Attribute und Dienste nur in der aktuellen Klasse angesprochen werden können, public bedeutet, dass sie von allen anderen Klassen angesprochen werden können, protected bedeutet, dass sie nur von der aktuellen Klasse und ihren Unterklassen angesprochen werden können.

Darstellung der Sichtbarkeit in Klassendiagrammen

Die Sichtbarkeit von Operationen und Attributten wird wie folgt gekennzeichnet:

- + für public - (engl. öffentlich), unbeschränkter Zugriff
- # für protected - (engl. geschützt), Zugriff nur von der Klasse sowie von Unterklassen (Klassen, die erben)
- für private - (engl. privat), nur die Klasse selbst kann es sehen
- ~ für package - (engl. Paket), innerhalb des Pakets sichtbar (nur in wenigen Programmiersprachen, etwa Java und C#, implementierbar)

abstract

ist eine Eigenschaft einer Klasse oder eines Dienstes. Eine Klasse ist abstrakt, wenn sie abstrakte Dienste besitzt.

protected

ist ein Sichtbarkeitsmerkmal für Bezugsobjekte, Attribute und Dienste. Diese Komponenten sind dann nur in der Klasse selbst und Unterklassen ansprechbar. protected wird im Klassendiagramm durch „#“ dargestellt.

1.9 Abstrakte Klassen 2

lokale Variable

Bezeichner einer Variablen, die in einem Dienst definiert ist und deshalb nur innerhalb dieses Dienstes gültig ist. Die Bezeichner einer lokalen Variablen sollten mit „l“ beginnen.

globale Variable

Bezeichner einer Variablen, die in der Klasse als Attribut (Zustandsvariable) definiert ist und deshalb in allen Diensten gültig ist. Die Bezeichner einer globalen Variablen sollten mit „z“ beginnen.

lokales Objekt

Ein Objekt das nur innerhalb eines Dienstes deklariert und benutzt wird, nennt man lokales Objekt. Dabei kann ein neues Objekt erzeugt sein oder es handelt sich um eine dynamische Referenz. Bezeichner für lokale Objekte sollten mit „l“ beginnen.

Verkettung

Wenn mehrere Objekte eine Reihung (Schlange) bilden und jedes Objekt seinen Nachfolger kennt, spricht man von Verkettung. Um eine beliebiges Objekt ansprechen zu können, muss man nur einen Zugriff auf das erste Element haben.

1.10 Ereignisorientierung

Ereignis

Durch die Betätigung eines Eingabegeräts, der Maus oder der Tastatur, wird ein Ereignis ausgelöst, das im Programm erkannt wird. Auf dieses Ereignis reagiert das Programm mit dem Aufruf des entsprechenden Ereignisdienstes.

Ereignisdienst

Dienst einer Anwendung, in dem auf ein Ereignis reagiert wird. Die notwendigen Zusatzinformationen werden als Parameter übergeben.

Reaktionstabelle

In diese Tabelle wird eingetragen, wie ein Programm auf die verschiedenen Ereignisse reagieren soll. Diese Tabelle sollte vor der Implementierung aufgestellt werden.

ereignisorientierte Programmierung

Programmierung!ereignisorientierte Bei dieser Art der Programmierung wartet das Programm auf das Eintreten von Ereignissen, auf die dann mit den entsprechenden Ereignisdiensten reagiert wird.

Leerlaufereignis

Besonderes Ereignis, das immer dann auftritt, wenn kein anderes Ereignis aufgetreten ist. Es wird dazu benutzt, periodisch auftretende Aufgaben zu erledigen, z.B. Animationen, Uhrzeitanzeige usw.

Algorithmus

Formulierung von Handlungsanweisungen in hinreichend einfacher Form. Ein Algorithmus muss einen definierten Anfangszustand haben, endlich viele Anweisungen haben und eindeutig sein. Computerprogramme sind die Umsetzungen von Algorithmen in Programmiersprachen, aber auch ein Kochrezept ist ein Algorithmus. Algorithmen kann man durch Struktogramme darstellen.

Gummilinie

In Zeichenprogrammen wird das Zeichnen einer Linie dadurch erleichtert, dass die Linie während des Zeichnens vom Anfangspunkt zum aktuellen Mauspunkt mit wandert. Dies kann man wie ein Gummiband vom Ausgangspunkt zum aktuellen Mauspunkt auffassen.

’+’

Das „+“-Zeichen wird kontextabhängig für zwei unterschiedliche Operationen benutzt. Ist mindestens einer der Operanden ein String, so wird er als Verkettungsoperator zum Verbinden zweier Zeichenketten aufgefasst, sind beide Operatoren dagegen Zahlen, wird er als Additionsoperator aufgefasst.

static

bezeichnet Klassendienste, die aufgerufen werden können, ohne dass ein Objekt der entsprechenden Klasse erzeugt wurde. Der Dienst main eines Java-Hauptprogramms ist als static deklariert. Ist ein solcher Dienst vorhanden, kann eine doppelklickbare Anwendung erzeugt werden.

1.11 Ereignisbearbeiter

GUI engl. Graphical User Interface, graphische Benutzerschnittstelle, die verschiedene Komponenten wie Knöpfe, Menüs usw. enthält. GUIs lösten in den 80er Jahren die bis dahin üblichen Textbildschirme ab. GUIs werden mit der Maus bedient. Bahnbrechende Entwicklungsarbeiten dazu wurden in Kalifornien am PARC-Forschungszentrum geleistet.

Ereignisbearbeiter

Klasse, die Ereignisse mit Hilfe von Ereignisdiensten verarbeiten kann.

Ereignisregistrierung

Wenn Ereignisbearbeiter sich bei der EBAwendung registrieren, erhalten Sie automatisch Informationen über aufgetretene Ereignisse und können darauf reagieren.

Hotkey

Zeichen, die bestimmten GUI-Elementen zugeordnet sind. So kann die Aktion, die zu einem GUI-Element gehört, mit der Tastatur aufgerufen werden.

Befehlsnummer

Ganze Zahl, die einem bestimmten GUI-Element zugeordnet ist. Bei einem Aufruf informiert das GUI-Element die Anwendung darüber, dass die Aktion mit der entsprechenden Nummer ausgeführt werden soll.

Bug

Fehler in einem Programm.

Debugger

Programm zur Fehlersuche. Ermöglicht es, im Einzelschrittmodus durch ein Programm zu gehen und dabei die Variablen zu untersuchen.

Haltepunkt

(engl. breakpoint) Markierte Zeile in einem Programm, an der das Programm anhält und den Debugger aufruft. Haltepunkte setzt und entfernt man durch Klicken auf die Zeilennummer.

Aufruffolge

Reihenfolge, in der die Dienste eines Programms zu einem bestimmten Zeitpunkt aufgerufen wurden.

Exemplarvariable

Bezugsobjekt oder Attribut des Objekts, das gerade im Debugger angezeigt wird. Sie werden auch als Instanzvariable bezeichnet.

1.12 Phasen eines Softwareentwurfs

Phasenmodell

der Softwareentwicklung: Ein Projekt durchläuft bis zu seiner Fertigstellung mehrere Phasen: Pflichtenhefterstellung, Entwurfsphase, Implementierung, Testphase.

Pflichtenheft

Basis für einen Vertrag zwischen dem Abnehmer und Hersteller einer Software. Das Pflichtenheft sollte vollständig und eindeutig sein.

Ist-Analyse

Untersuchung, wie eine Aufgabe, die mit einem Computerprogramm gelöst werden soll, zur Zeit gelöst wird.

Soll-Analyse

Entwurf der Oberfläche und Funktionalität eines Softwareprodukts. Die Implementierung bleibt dabei unberücksichtigt.

Regressionstest

Testverfahren, bei dem Tests wiederholt unter gleichen Bedingungen möglichst automatisiert durchgeführt werden können. Dazu wird ein Test aufgezeichnet, der dann mit veränderten Diensten unter gleichen Bedingungen wiederholt wird.

assert

(in deutsch zusichern) Anweisung in einem Testprogramm, welche überprüft, ob zu einem bestimmten Zeitpunkt eine Variable einen bestimmten Wert hat.

1.13 Model View Controller

MVC-Modell

Trennung der Oberfläche (View) von den Daten (Model). Die Steuerungsschicht (Control) stellt die Verbindung zwischen der Oberfläche und der Datenschicht her. Die Anwendung dieses Prinzips sorgt dafür, dass ein Programm robuster und leichter zu warten ist.

Programmgenerator

Hilfsprogramm zur Erzeugung der Bildschirmoberfläche (GUI) eines Programms. Damit kann der Prototyp eines Programms erstellt werden.

Prototyp

Programm, dass nur aus der Bildschirmoberfläche besteht. Die eigentliche Funktionalität muss noch ergänzt werden. Mit einem Prototyp kann die Oberfläche eines Programms getestet werden.

Fokus

Die Komponenten einer grafischen Benutzeroberfläche können den Fokus erhalten oder verlieren. Ein Komponente, die den Fokus besitzt, erhält die Information über Tastatureingaben. Mit der Tabulatortaste kann der Fokus von einer Komponente zur nächsten verschoben werden. Komponenten, die den Fokus besitzen, werden vom Betriebssystem grafisch hervorgehoben.

Textfeld

Komponente zur Eingabe von Text oder Zahlen. Ein Textfeld besteht immer aus einer einzigen Zeile.

XML

Die Extensible Markup Language (XML) ist ein Standard zur maschinenlesbaren und von Menschen lesbaren Darstellung strukturierter Daten. Wird vom Programmgenerator zur Speicherung einer Bildschirrmaske verwendet.

Applet

(kleines Programm von engl. application = Anwendung) Ein Applet ist ein kleines Java-Programm, das mit Hilfe einer HTML-Seite im Internet veröffentlicht werden kann. SuM-Programme, die die Ereignisbibliothek benutzen, können als Applets veröffentlicht werden.

Kapitel 2

Schriek Band II

Jeweils Abschnitt „Neue Begriffe in diesem Kapitel“
(ggf. mit einigen Erweiterungen)

Bernhard Schriek:
Informatik mit Java
Band 2
ISBN 978-3-00-019637-9

2.2 Rekursion

Rekursion

(dt. Selbstbezug) Als Rekursion bezeichnet man den Aufruf oder die Definition durch sich selbst. Ohne geeignete Abbruchbedingung geraten solche rückbezüglichen Aufrufe in einen so genannten infiniten Regress (Umgangssprachlich: Endlosschleife)

Einfache Rekursion kann durch Iteration (Anwendung einer Schleife) ersetzt werden. Rekursion ist oft natürlicher und eleganter.

rekursiver Aufruf

In einem Dienst wird derselbe Dienst ein- oder mehrmals aufgerufen. Da sich der Computer die Rücksprungadresse merken muss, kann ein Stack-Overflow auftreten

Fakultaet

$n!$ (gelesen n Fakultät)

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n; 0! = 1 \quad (2.1)$$

Die Fakultät ist eine wichtige Funktion in der Kombinatorik

Fibonacci-Folge

Die Fibonacci-Folge ist eine Folge von nicht-negativen ganzen Zahlen, wobei das nächste Folgenglied die Summe der beiden vorhergehenden Glieder ist.

Binaerbaum

Ein Baum, bei dem von jeder Verzweigung genau zwei Äste abgehen, heißt Binaerbaum

Selbstähnlichkeit

Selbstähnlichkeit ist die Eigenschaft von Gegenständen, Körpern Mengen oder geometrischen Objekten, in größeren Maßstäben, d.h. bei Vergrößerung dieselben oder ähnliche Strukturen aufzuweisen, wie im Anfangszustand. Diese Eigenschaft wird unter anderem von der Fraktalen Geometrie untersucht, da fraktale Objekte eine hohe bzw. perfekte Selbstähnlichkeit aufweisen.

Türme von Hanoi

Die Türme von Hanoi sind ein Spiel, bei dem drei Scheiben-Stapel von einem Platz zu einem anderen bewegt werden. Dabei darf ein dritter Platz als Zwischenlager benutzt werden. Es darf nie eine größere auf einer kleineren Scheibe liegen.

for

Die for-Schleife ist eine Zählschleife, die benutzt wird, wenn die Anzahl der Schleifendurchläufe bekannt ist und während des Schleifendurchlaufs nicht verändert wird

```
1 for (Initialisierungsklausel; Boolescher Ausdruck; Iteration)
2 {
3     Anweisung
4 }
```

Beispiel:

```
1 for (int i=0; i <10 ; i++)
2 {
3     summe = summe + i;
4 }
```

2.3 Felder

Feld

(engl.array) Ein Feld dient zur Verwaltung einer festen Menge gleichartiger Daten. Dabei werden einzelne Feldelemente mit ihrem Index angesprochen. Felder können mehrdimensional sein und primitive Datentypen und Objekte enthalten.

Taktgesteuerte Simulation

(auch diskrete Simulation genannt) ist eine Simulation, bei der bei jedem Takt gewisse Aktionen ausgeführt werden. Simulationen können auch durch Ereignisse gesteuert werden.

feld

feld[i][j] ist ein Element eines zweidimensionalen Feldes (arrays). i,j sind die Indizes. Die Inizes beginnen in Java immer mit 0. Falls ein Element mit einem ungültigen Index angesprochen wird, löst dies eine IndexOutOfBoundsException aus.

Ein Feld wird z.B. so deklariert:

```
Typ Name[ ][ ] = new Typ[Anzahl][Anzahl];
int[][] array1 = new int[10][20];
```

2.4 Lineare Strukturen I - Schlangen

Schlange

(engl. queue) Eine Schlange ist eine Lineare Struktur, in der Objekte nach dem FIFO (First-In-First-Out) verwaltet werden. Wenn eine Schlange beliebige Objekte verwalten kann, spricht man von einer generischen Schlange

FIFO

steht für First-In-First-Out, das Prinzip, wie die Elemente der Schlange verwaltet werden. Die Elemente verlassen die Schlange in der Reihenfolge Ihrer Ankunft. Beispiel: Kassenschlange.

Klassenbibliothek

jar-Datei (jar = Java Archiv), die übersetzte Klasse (ohne den Quellcode) enthält, die anderen Klassen zur Verfügung gestellt werden können

package

(deutsch Paket) Jede Klasse einer Java-Bibliothek gehört zu einem Paket. Als erste Zeile im Quelltext steht die Paketinformation, z.B. `package bspg.Strukturen;`. Sämtliche Klassen eines Pakets sind in einem gemeinsamen Projekt implementiert. Aus diesem Projekt kann man eine jar-Datei erzeugen, die den übersetzten Quelltext in komprimierter Form enthält.

jar

`jar` = Java Archiv
Endung einer Datei, die ein Paket von übersetzten Java-Klassen enthält. Eine jar-Datei ist entweder ein Paket, dessen Klassen von einem Projekt benutzt werden können oder ein ausführbares Java-Programm, das mit einem Doppelklick gestartet werden kann.

2.5 Lineare Strukturen II - Listen

Liste (lineare Liste)

Liste!lineare Eine Liste ist eine lineare Struktur (Sammlung), in der Objekte verwaltet werden. Wenn eine Liste beliebige Objekte verwalten kann, spricht man von einer generischen Liste.

doppelte Verkettung

Die Listenelemente werden vorwärts und rückwärts verkettet.
Dies ermöglicht eine Vorwärts- und Rückwärtstraversierung der Liste.

Sentinel (dt. Markierung)

Um Einfüge- und Löschoperationen bei Listen einfacher implementieren zu können, werden an den Anfang und am Ende zwei besondere Listenelemente angehängt, die keine Daten enthalten. Diese besonderen Listenelemente bezeichnet man als Sentinels, in unserer Listenimplementierung werden sie mit Bug und Heck bezeichnet.

2.6 Lineare Strukturen III - Stapel

Stapel oder Kellerspeicher (engl. Stack)

Ein Stapel ist eine lineare Struktur (Sammlung), in der Objekte nach dem LIFO-Prinzip verwaltet werden. Wenn ein Stapel beliebige Objekte verwalten kann, spricht man von einem generischen Stapel.

LIFO

LIFO steht für *Last In First Out*, das Prinzip, wie Elemente eines Stapels verwaltet werden. Die Elemente verlassen den Stapel in der umgekehrten Reihenfolge ihrer Ankunft.

Ruecksprungadresse

Wenn ein Unterprogramm (Dienst bei der OOP) aufgerufen wird, wird die Speicheradresse der danach auszuführenden Anweisung auf einem Stapel abgelegt. Wenn das Unterprogramm abgearbeitet ist, wird diese Adresse vom Stapel geholt und das Programm weiß, an welcher Stelle es weiterarbeiten muss. So ist es problemlos möglich, Unterprogramme beliebig tief zu verschachteln.

2.7 Binäre Bäume

BinaerBaum

Hierarchische, rekursive Struktur zur Organisation und Verwaltung von Daten. Ein Baum besteht aus einer Wurzel und zwei eventuell leeren Teilbäumen.

Morsebaum

Im Morsebaum werden die zu codierenden Zeichen an einer bestimmten Position verwahrt, so dass sich der Code aus dem Weg von der Wurzel zum entsprechenden Knoten ergibt. Jede Kante ist mit einem Punkt oder Strich gekennzeichnet.

Fano-Bedingung

Kein Code darf der Anfang eines anderen Codes sein. Dann können Codes problemlos aneinandergehängt werden und eine Decodierung ist möglich. Wenn die Fano-Bedingung erfüllt ist, befinden sich die zu codierenden Zeichen nur in den Blättern des Codebaumes.

Huffman-Algorithmus

Optimales Verfahren zur Komprimierung von Text durch Entropiecodierung. Sie ordnet einer festen Anzahl an Quellsymbolen jeweils Codewörter mit variabler Länge zu. Der bei der Huffman-Kodierung gewonnene Baum liefert garantiert eine optimale und präfixfreie Kodierung, d.h. es existiert kein symbolbezogenes Kodierverfahren, das einen kürzeren Code generieren könnte, wenn die Auftrittswahrscheinlichkeiten der Symbole bekannt sind.

Entropiecodierung

Die Zahl der Codesymbole für ein Zeichen hängt von der Häufigkeit des

Vorkommens des Zeichens im Text ab, Häufigere Zeichen erhalten kürzere Codewörter

preorder

rekursives Verfahren zur Baumtraversierung. Zuerst wird die Wurzel besucht, anschließend der linke Teilbaum und zum Schluss der rechte Teilbaum.

inorder

rekursives Verfahren zur Baumtraversierung. Zuerst wird linke Teilbaum besucht, anschließend die Wurzel und zum Schluss der rechte Teilbaum.

postorder

rekursives Verfahren zur Baumtraversierung. Zuerst wird der linke Teilbaum, anschließend der rechte Teilbaum und zum Schluss die Wurzel.

2.8 Sortierverfahren

O-Notation

Verfahren zur Beschreibung der Komplexität hinsichtlich des Zeitverhaltens von Algorithmen.

Bubblesort

langsamtes rekursives Sortierverfahren, das die Sortierung durch Vertauschung benachbarter Elemente durchführt.

Selectionsort

schnelles Sortierverfahren, bei dem die größten (kleinsten) Elemente der unsortierten Liste gesucht und an die sortierte Teilliste angehängt werden.

Insertionsort

schnelles Sortierverfahren, bei dem jeweils das erste Element der unsortierten Liste in der sortierten Liste an passender Stelle eingefügt wird.

Quicksort

schnelles rekursives Sortierverfahren, das nach dem Prinzip „Teile und herrsche“ arbeitet. Zunächst wird die zu sortierende Liste in zwei Teillisten („linke“ und „rechte“ Teilliste) getrennt. Dazu wählt Quicksort ein sogenanntes Pivotelement aus der Liste aus. Alle Elemente, die kleiner als das Pivotelement sind, kommen in die linke Teilliste, und alle, die größer sind, in die rechte Teilliste. Die Elemente, die gleich dem Pivotelement sind, können sich beliebig auf die Teillisten verteilen. Nach der Aufteilung sind die Elemente der linken Liste kleiner oder gleich den Elementen der rechten Liste. Danach werden (rekursiver Aufruf) die beiden Teillisten für sich sortiert.

2.9 Tiefensuche - Backtracking

Backtracking

(dt. Rückverfolgung) Lösungsstrategie bei Problemen, die aus mehreren Stufen mit jeweils mehreren Möglichkeiten (Kandidaten) bestehen. Als falsch erkannte Lösungsschritte werden dabei zurückgenommen. *Gehe so tief wie möglich, gehe so weit zurück wie notwendig.*

Tiefensuche

Lösungsstrategie bei Problemen, die aus mehreren Stufen mit jeweils mehreren Möglichkeiten (Kandidaten) bestehen. Dabei versucht man im Lösungsbaum möglichst in die Tiefe zu gehen. Eine Alternative dazu ist die Breitensuche, bei der man möglichst in allen Teilbäumen gleichzeitig in die Tiefe geht.

2.10 Breitensuche

Breitensuche

Lösungsstrategie bei Problemen, die aus mehreren Stufen mit jeweils mehreren Möglichkeiten bestehen. Dabei versucht man im Lösungsbaum möglichst in die Breite zu gehen, also jede Stufe vollständig zu verarbeiten. Dies wird durch den Einsatz einer Schlange erreicht.

Branch und Bound

(dt. Verzweigung und Beschränkung) Lösungsstrategie bei Problemen, bei denen der Lösungsbaum viele Verzweigungen besitzt. Durch logische Überprüfungen werden Äste des Lösungsbaums abgeschnitten, so dass der Lösungsbaum kleiner wird.

Kapitel 3

Schriek Band III

Jeweils Abschnitt „Neue Begriffe in diesem Kapitel“
(ggf. mit einigen Erweiterungen)

Bernhard Schriek:
Informatik mit Java
Band 3
ISBN 978-3-00-022995-4F

3.2 Netzwerkprogrammierung I

Schichtenmodell

Die Aufgaben der Datenübertragung werden im OSI-Schichtenmodell auf 7 Schichten aufgeteilt, die voneinander unabhängig sind. So kann in jeder Schicht das Übertragsverfahren verbessert werden, ohne dass die anderen Schichten geändert werden müssen.

IP-Nummer

Jeder Computer im Internet besitzt eine eindeutige IP-Nummer, über die er angesprochen werden kann. Man kann die IP-Nummer mit der Adresse einer Person oder Firma vergleichen, die benötigt wird, um ein Paket zuzustellen.

DNS

Das Domain Name System dient dazu, den IP-Nummern leichter merkbare IP-Namen zuzuordnen.

TCP

Das TCP-Protokoll (Transmission Control Protocol) dient dazu, Daten im Internet sicher zu übertragen, indem der Empfänger Meldungen über den Empfang einzelner Teilstücke an den Absender zurücksendet. Falls ein Paket nicht ankommt, wird es dann nochmal versendet.

UDP

Das UDP-Protokoll (User Datagram Protocol) wird benutzt, um Daten wie Musiksendungen oder Filme zu übertragen, bei denen auch kleinere Fehler auftreten können, ohne dass die Qualität stark beeinflusst wird. UDP ist bedeutend schneller als TCP, da auf Empfangsbestätigungen verzichtet wird.

Private Netze

Bestimmte IP-Nummern sind für die Nutzung in privaten Netzen reserviert. Nach außen erscheint dieses LAN (Local Area Network) wie ein einzelner Computer mit nur einer IP-Nummer.

Port

zusätzlich zur IP-Nummer muss bei einer Datenübertragung noch der Port angegeben werden, der den zugehörigen Dienst kennzeichnet. So können auf einem Computer gleichzeitig mehrere Datenübertragungen stattfinden

3.3 Netzwerkprogrammierung II

Client-Server-System

Ein Server (Anbieter) bietet im Netz Dienste an, die vom Client (Kunden) in Anspruch genommen werden. Dies ist der Standard bei der Internet-Kommunikation. Es gibt allerdings auch Verbindungen, bei denen die beteiligten Computer gleichwertig sind (Peer-to-Peer-Verbindungen). Welchen Dienst des Servers der Client nutzen möchte, teilt er mit der Portnummer mit.

Daytime

Der Daytime-Dienst ist ein Dienst im Internet, bei dem das Datum und die Uhrzeit als String an den Client geliefert werden.

QOTD

Der Quote-of-the-Day-Dienst liefert Sprüche (Zitate) an den Client. Nach jeder Anmeldung eines Clients wird diesem ein zufälliger Spruch geschickt, anschließend trennt der Server die Verbindung.

Echo

Nach der Anmeldung eines Clients, kann dieser dem Server Nachrichten schicken, die wie eine Echo zurückgeschickt werden. Die Verbindung wird gehalten, bis sie der Client beendet,

POP3

Das Post Office Protokoll in der Version 3 dient dazu, Email von einem Mail-Server abzurufen. Das Passwort wird unverschlüsselt übertragen.

SMTP

Das Simple Mail Transfer Protokoll dient dazu, Emails an einen Mailserver zu versenden. Da keine Benutzer-Authentifizierung vorgesehen ist, wurden verschiedene Methoden entwickelt, um diese Schwäche zu verhindern.

POP3-vor-SMTP

Verfahren zu Benutzer-Authentifizierung beim Email-Versand. Bevor man Emails versenden kann, muss man sich mit POP3 anmelden.

3.4 Netzwerkprogrammierung III

Nebenläufigkeit

Mehrere Teile eines Programms arbeiten quasi parallel, indem der Prozessor jedem Programmteil für eine kleine Zeitspanne zur Verfügung steht.

Thread

(dt. Faden) Java-Klasse zur Implementierung von Nebenläufigkeiten. Dabei benutzen Threads den Dienst `run`, in dem steht, was der Thread leisten soll (normalerweise eine Schleife). Dieser Dienst wird mit dem Aufruf `start` gestartet. Mit Hilfe von boolschen Variablen kann man die Schleife im `run`-Dienst des Thread beenden und damit den Thread beenden. Es ist nicht möglich, den `run`-Dienst zweimal zu starten. Stattdessen muss ein neuer Thread erzeugt werden

Dialogorientiertes Protokoll

Bei dieser Form eines Protokolls befindet sich die Kommunikation mit einem Client jeweils in einem bestimmten Zustand, der bestimmte Protokoll—Anweisungen erfordert. Mit Hilfe einer Unterklasse der Klasse Server-Verbindung lassen sich im Dienst `bearbeiteProtokoll` diese Zustände so implementieren, als ob nur die Verbindung zu einem Client bestünde. Mit Hilfe von lokalen Variablen kann man Eigenschaften des entsprechenden Clients verwalten.

RFC

Abkürzung für Request For Comments (dt. Forderung nach Kommentaren). Dokument zur Standardisierung von Protokollen im Internet. RFC 2821 beschreibt z.B. das SMTP-Protokoll. Ein nicht ganz ernst gemeintes RFC hat die Nummer 2549¹ und beschreibt die drahtlose Übertragung von Informationen mit Hilfe von Brieftauben. Besonders problematisch hat sich dabei allerdings das Antwortverhalten erwiesen. Außerdem hat man inzwischen festgestellt, dass auf diese Art auch leicht Viren übertragen werden können.

3.4.1 SQL-Datenbanken I

relationale Datenbank

Eine Datenbank besteht aus mehreren Tabellen, die auch Relationen genannt werden. Jede Tabelle enthält zusammengehörige (relationale) Daten in Form von Datensätzen, die die Zeilen der Tabelle bilden.

MySQL

MySQL Relationales Datenbanksystem (RDBS), das frei benutzt werden kann. Es wurde ab 1994 in Schweden entwickelt. Da es sich um ein Open Source Projekt handelt, stehen die Quelltexte jedem Interessierten offen.

SQL

SQL (Structured Query Language) Programmiersprache zur Arbeit mit relationalen Datenbanken. Insbesondere die SELECT-Anweisung bietet umfangreiche Möglichkeiten zur Auswertung von Daten aus mehreren Tabellen gleichzeitig.

phpMyAdmin

In der Programmiersprache PHP geschriebene komfortable Schnittstelle zur Arbeit mit der Datenbank MySQL.

Relationenalgebra

Mathematisches Modell zur Arbeit mit Relationen und SQL, in dem es 8 Operationen gibt, die eine vollständige algebraische Struktur bilden.

3.4.2 SQL-Datenbanken II

Inkonsistenz

Eine Datenbank ist konsistent, wenn ihre Daten widerspruchsfrei sind. Inkonsistenzen entstehen oft, wenn Daten redundant (also mehrfach) gespeichert sind und Tippfehler aufgetreten sind. Auch Verletzungen der referentiellen Integrität werden als Inkonsistenzen bezeichnet. Mit der Normalisierung von Datenbanken wird versucht Inkonsistenzen zu verhindern.

¹ <http://tools.ietf.org/pdf/rfc2549.pdf>

Anomalie

Anomalien führen zu Inkonsistenzen Datenbanken. Man unterscheidet zwischen Einügeanomalien, Löschanomalien und Änderungsanomalien.

Normalisierung

Bei der Normalisierung einer Datenbank wird versucht, Redundanzen aufzulösen, so dass Daten an nur einer Stelle gespeichert werden. Daten dürfen nur noch von Primärschlüsseln abhängen. Andere Abhängigkeiten zwischen Attributen einer Relation dürfen nicht vorhanden sein. Bei der Normalisierung werden drei Stufen durchlaufen (1., 2. und 3. Normalform (kurz 1NF, 2NF und 3NF)).

ER-Diagramm

In einem Entity-Relationship-Diagramm werden die Entitäten und Beziehungen dazwischen mit ihren Kardinalitäten grafisch dargestellt. Mit Hilfe von Hilfsrelationen wird versucht nur noch 1:n-Beziehungen zu erhalten.

Primaerschluessel

Ein Primärschlüssel ist eine Attribut oder er ist aus mehreren Attributen einer Relation zusammengesetzt. Der Primärschlüssel ist einmalig für jeden Datensatz (Tupel) der Relation.

Fremdschluessel

Ein Fremdschlüssel ist ein Attribut einer Relation, das den gleichen Wert hat, wie der Primärschlüssel einer anderen Relation. So werden Beziehungen zwischen Relationen hergestellt.

Referentielle Integrität

Die referentielle Integrität (RI) einer Datenbank ist verletzt, wenn in einer Relation ein Fremdschlüsselwert keine Entsprechung in einem Primärschlüsselwert der zugeordneten anderen Relation enthält. Dies kann z.B. durch eine Löschanomalie entstehen.

3.4.3 Kryptographie

Symmetrische Verschlüsselung

Wenn zur Verschlüsselung und Entschlüsselung der gleiche Schlüssel benutzt wird, spricht man von symmetrischer Verschlüsselung. Dabei entsteht das Problem der sicheren Schlüsselübermittlung.

Asymmetrische Verschlüsselung

Wenn jeder Partner einen öffentlichen und einen privaten Schlüssel benutzt, spricht man von asymmetrischer Verschlüsselung. Der öffentliche Schlüssel wird veröffentlicht. Eine mit dem öffentlichen Schlüssel verschlüsselte Nachricht kann nur mit dem privaten Schlüssel wieder entschlüsselt werden.

Digitale Signatur

Wenn der Sender an eine Nachricht einen Text, der mit seinem privaten Schlüssel verschlüsselt wurde, anhängt, kann der Empfänger mit dem zugehörigen privaten Schlüssel prüfen, ob die Nachricht wirklich von dem Absender

stammt.

Zertifizierungsstelle

Dienstleister, die überprüfen, ob ein öffentlicher Schlüssel wirklich zu seinem Besitzer gehört. Solche Stellen garantieren, dass der öffentliche Schlüssel wirklich echt ist.

3.4.4 Akzeptoren

Mealy-Automat

Ein Mealy-Automat ist ein endlicher deterministischer Automat mit einer Menge von Zuständen, einem Eingabealphabet, einem Ausgabealphabet, einer Übergangsfunktion, die angibt bei welcher Eingabe von welchem Zustand in welchen Zustand gewechselt wird, einer Ausgabefunktion, die angibt, was bei jedem Übergang ausgegeben wird und einem Startzustand.

Erkennender Automat

Ein EA wird auch Akzeptor genannt. Er besitzt keine Ausgaben, dafür aber Endzustände. So kann man entscheiden, ob ein Wort von diesem Automaten akzeptiert wird.

regulaerer Ausdruck

Kurzschreibweise für die Wörter, die ein erkennender Automat akzeptiert.

regulaere Sprache

Typ 3-Sprache in der Chomsky-Hierarchie, deren Wörter eine reguläre Grammatik erfüllen.

regulaere Grammatik

Eine reguläre Grammatik besteht aus einem Startsymbol, einer Menge von Terminalen, einer Menge von Non-Terminalen und einer Menge von Produktionen, die angeben, wie ein Wort der zugehörigen Sprache abgeleitet werden kann. Die Produktionen dürfen nur die Form $A \rightarrow xB$ bzw. $A \rightarrow z$ haben.

3.4.5 Scanner und Parser

Scanner

Scanner auch Lexer genannt, dienen zur lexikalischen Analyse eines Quelltextes. Der Scanner fasst zusammengehörige Zeichen zu Tokens zusammen und liefert eine Tokenliste, die vom Parser weiter untersucht wird.

Token

Zusammengehöriges Teil eines Quelltextes z.B. ZAHL oder BEZEICHNER. Token bestehen aus einem (MULT = '*') oder mehreren Zeichen (ZAHL = '1234'). Token werden vom Scanner erzeugt und vom Parser weiterverarbeitet.

Syntaxausdruck

Kurzschreibweise für die Regeln (Produktionen) einer Grammatik. Dabei

werden Symbole wie '?' = einmal oder keinmal oder '+' für mindestens einmal oder '*' für beliebig oft an Terminalen oder Non-Terminalen angehängt.

Syntaxdiagramm

Grafische Darstellung von Syntax-Regeln, ähnlich wie Flussdiagramme, die veranschaulichen, was eine Regel beinhaltet.

Parser

Parser dienen zur syntaktischen Analyse einer Tokenliste. Dazu muss die zugehörige Grammatik vorgegeben sein. Der Parser untersucht, ob die Reihenfolge der Tokens die Syntaxregeln erfüllt, und gibt anderenfalls passende Fehlermeldungen aus.

Syntaxbaum

Grafische Darstellung der Reihenfolge der benutzten Regeln bei der lexikalischen Analyse eines konkreten Quelltextes durch den Parser.

rekursiver Abstieg

Bei der syntaktischen Analyse werden die Dienste des Parsers rekursiv aufgerufen. Der Syntaxbaum verdeutlicht den rekursiven Abstieg.

Parsergenerator

Programm zur automatischen Erzeugung eines Scanners (Lexers) und Parsers zu einer vorgegebenen Grammatik. ANTLR ist ein bekannter Parsergenerator.

Index

, 12
'+' *, 19
//, 8
== *, 11
= *, 9
&& *, 12
, 19, 26
, 26
Anomalie, 35
Ableitung, 36
Abstieg
rekursiver, 37
abstract, 17
abstract *, 17
abstrakte Klasse, 16
abstrakter Dienst, 16
aktueller Parameter, 14
Akzeptor, 36
Algorithmus , 19
Analyse
Ist-, 21
Soll-, 21
Anfrage, 6
Animation, 12
Applet, 22
Array, 25
array, 25
assert, 21
Ast, 24
Asymmetrische Verschluesselung, 36
Attribut, 13
Aufruf
rekursiver, 24
Aufruffolge, 20
Auftrag, 6
Ausdruck
regulärer, 36
Automat
erkennender, 36
Mealy, 36
Backtracking, 29
Barberpole, 9
Baum
Morse, 27
Bedingung, 10, 11
Fano, 27
zusammengesetzte, 12
Befehlsnummer , 20
Beziehung
-sdiagramm, 13
hat, 13
kennt, 13
Beziehungsdiagramm, 13
Bezugsobjekt, 13
Bibliothek, 9
Java, 26
BinaerBaum, 27
Binaerbaum, 24
Bindung
spaaete, 16
boolean *, 14
Botschaft, 6
Branch und Bound, 29
break *, 11
breakpoint, 20
Breitensuche, 29
Bubblesort, 28
Bug , 20
Bytecode, 8
CamelCase, 7
case *, 11
char *, 11
Chomsky-Hierarchie, 36
class *, 9
Client-Server-System, 33
Control, 22
Controller
Model View, 22

- MVC, 22
- Darstellung der Sichtbarkeit in Klassendiagrammen, 17
- Datenbank
relationale, 34
- Datentyp
primitiver, 25
- Daytime, 33
- Debugger, 20
- default *, 11
- Dialogorientiertes Protokoll, 34
- Dienst, 6, 9, 13, 14
abstrakter, 16
- Digitale Signatur, 36
- DNS, 32
- Domain Name System, 32
- doppelte Verkettung, 26
- double *, 7
- Durchlaufbedingung, 10
- dynamische Referenz, 16
- Echo, 33
- Editor, 8
- Einruecken, 8
- else *, 11
- Email, 33
- Entity-Relationship-Diagramm, 35
- Entropiecodierung, 27
- ER-Diagramm, 35
- Ereignis, 19
Leerlauf, 19
- Ereignisbearbeiter, 20
- Ereignisdienst, 19
- ereignisorientierte Programmierung, 19
- Ereignisregistrierung, 20
- Erkennender Automat, 36
- Exemplar, 6
- Exemplarvariable, 20
- extends *, 14
- Fakultaet, 24
- Fano-Bedingung, 27
- Farbe, 10
- Fehler
Laufzeit-, 15
logischer, 15
Syntax-, 15
- Fehlermeldung, 9
- Feld, 25
- feld *, 25
- Fibonaccifolge, 24
- FIFO, 26
- Fokus, 22
- for *, 25
- formaler Parameter, 14
- fraktal, 24
- Fremdschlüssel, 35
- Geheimnisprinzip, 13
- globale Variable, 18
- Grammatik
reguläre, 36
- Gummilinie, 19
- Haltepunkt, 20
- hat-Beziehung, 13
- Hotkey, 20
- Huffman-Algorithmus, 27
- if *, 11
- Implementierung, 13
- import *, 9
- Inkonsistenz, 35
- inorder, 28
- Insertionsort, 28
- Instanz, 6
- int *, 7
- Integrität
Referentielle, 35
- IP-Nummer, 32
- Ist-Analyse, 21
- ist-Beziehung, 10
- jar-Datei, 26
- jar *, 26
- Java-Bibliothek, 26
- Kellerspeicher, 27
- kennt-Beziehung, 13
- Klasse, 6, 9, 13
abstrakte, 16
Ober-, 14
Oberklasse, 10
Rechner, 12
Unter-, 10, 14
Unterklasse, 10
- Klassenbibliothek, 26
- Klassendiagramm, 13
- Kommentar, 8
- Konstruktor, 6, 7, 9

- Kontrollstruktur, 10
LAN, 32
Laufzeitfehler, 15
Leerlaufereignis, 19
lesender Zugriff auf ein Attribut, 14
Lexer, 36
LIFO, 27
Liste
 generische, 26
Liste (lineare Liste), 26
lokale Variable, 18
lokales Objekt, 18
Markierung, 26
Mealy-Automat, 36
Model, 22
Morsebaum, 27
MVC, 22
MVC-Modell, 22
MySQL, 34
n, 24
Nebenlaeufigkeit, 33
new *, 7, 9
Normalform, 35
Normalisierung, 35
null *, 15
Nullpointer-Exeption, 15
O-Notation, 28
Oberklasse, 10, 16
Objekt, 6, 9, 13
 Bezugs-, 13
 lokales, 18
oder, 12
Operator
 oder, 12
 und, 12
package, 26
package *, 26
Paket, 26
Parameter, 7, 14
 aktueller, 14
 formaler, 14
Parser, 36, 37
Parsegenerator , 37
Peer-to-Peer—Verbindungen, 33
Pflichtenheft, 21
Phasenmodell, 21
phpMyAdmin, 34
Polymorphie, 16
POP3, 33
POP3-vor-SMTP, 33
Port, 32
Post Office Protokoll, 33
postorder, 28
preorder, 28
Primaerschlüssel, 35
private, 16
Private Netze, 32
Produktion, 36
Programm starten, 8
Programm uebersetzen, 8
Programm-Vorlage, 8
Programmgenerator, 22
Programmierung
 ereignisorientierte, 19
Projektfenster, 9
protected, 16, 17
protected *, 17
Protokoll
 dialogorientiertes, 34
Prototyp, 22
Pseudocode, 10
public, 16
public *, 9
Punktnotation, 6
QOTD, 33
queue, 26
Quicksort, 28
Reaktionstabelle, 19
Redundanz, 35
Referentielle Integrität, 35
Referenz
 dynamische, 16, 18
Regressionstest, 21
regulaere Grammatik , 36
regulaere Sprache, 36
regulaerer Ausdruck, 36
Reihung, 18
Rekursion, 24, 32
rekursiver Abstieg, 37
rekursiver Aufruf, 24
Relation, 34
relationale Datenbank, 34
Relationenalgebra, 34
RFC, 34

- Ruecksprungadresse, 27
runtime-error, 15
- Scanner, 36
Schichtenmodell, 32
Schlange, 18, 26
 generische, 26
Schleife, 10, 11
 abweisende, 10
Schnittstelle, 13
schreibender Zugriff auf ein Attribut, 14
Selbstahnlichkeit, 24
Selectionsort, 28
Sentinel, 26
Sentinel (dt. Markierung), 26
Sichtbarkeit, 16
Signatur
 digitale, 36
Simple Mail Transfer Protokoll, 33
Simulation
 ereignisgesteuerte, 25
 taktgesteuerte, 25
SMTP, 33
Softwareentwurf
 Phasen, 21
Soll-Analyse, 21
sortieren, 28
spaeete Bindung, 16
Sprache
 reguläre, 36
SQL, 34
Stack, 27
Stack-Overflow, 24
Stapel, 27
Stapel oder Kellerspeicher (engl. Stack), 27
static, 19
static *, 19
statischer Typ - dynamischer Typ, 16
Stoppuhr, 12
Structured Query Language, 34
Struktogramm, 10
Substantiv-Verb-Methode, 16
super(...) *, 14
super.dienst(...) *, 14
switch *, 11
Symmetrische Verschluesselung, 35
Syntaxausdruck, 37
Syntaxbaum, 37
- Syntaxdiagramm, 37
Syntaxfehler, 9
- Taktgesteuerte Simulation, 25
Tastaturpuffer, 10
TCP, 32
Teilbaum, 27
Template, 8
Terminalsymbol, 36
Test
 Regressions-, 21
Textfeld, 22
this(...)*, 14
Thread, 33
Tiefensuche, 29
Token, 36, 37
Transmission Control Protocol, 32
Tuerme von Hanoi, 24
Typ
 dynamischer, 16
 statischer, 16
- UDP, 32
Uebersetzen, 8
Umgangssprache, 10
und, 12
Unterkasse, 10, 16
User Datagram Protocol, 32
- Variable
 globale, 18
 Instanz-, 20
 lokale, 18
Vererbung, 10
Vergleichsoperator, 11
Verkettung, 18
 doppelte, 26
Verschlüsselung
 asymmetrische, 36
 symmetrische, 35
Verweis, 16
Verzweigung, 10, 11
 Mehrfachverzweigung, 11
Verzweigungen, 10
View, 22
Virtuelle Maschine, 8
VM, 8
void *, 7
- Werkbank, 6

while *, 11
XML, 22
Zeichen, 11
Zertifizierungsstelle, 36
Zufallszahl, 12
Zugriff, 14
 lesender, 14
 schreibender, 14
Zusammengesetzte Bedingung, 12
Zustand, 13
Zustandsvariable, 13
Zuweisung, 9