



SCHOOL-SCOUT.DE

Unterrichtsmaterialien in digitaler und in gedruckter Form

Auszug aus:

*Modellierung und Implementierung von Datenstrukturen mit
Java*

Das komplette Material finden Sie hier:

[Download bei School-Scout.de](https://www.school-scout.de)



Vorwort

Dieser Lösungsband bietet Lösungen und Kommentare zu dem gleichnamigen Arbeitsbuch für Schülerinnen und Schüler.

Dieses Arbeitsbuch stellt eine umfassende Unterrichtsreihe zu den Datenstrukturen Queue, Stack, List (d.h. einfach verkettete Liste) und Binärbaum (inklusive binärer Suchbäume) mit Java für Schülerinnen und Schüler der gymnasialen Oberstufe dar. Es baut auf meinem Arbeitsbuch zur Einführung in die objektorientierte Modellierung auf, kann aber — sofern die Lerngruppe mit den Konzepten der OOM vertraut ist — auch unabhängig von diesem verwendet werden, da nur am Rande auf dieses Bezug genommen wird.

Alle Materialien wurden bereits im Unterricht eingesetzt und aufgrund praktischer Erfahrungen ergänzt und ausgebessert.

Jedes Kapitel schließt mit einer Projektarbeit ab, die es den Schülerinnen und Schülern ermöglicht, das Gelernte zu vertiefen und anzuwenden. Außerdem gewinnen sie damit wertvolle Erfahrungen in der Umsetzung größerer Programmiervorhaben.

Bei der Gestaltung wurde sich an den Vorgaben zum Zentralabitur NRW¹ (Bereich I.2) orientiert.

An meiner Schule wird Eclipse als Programmierumgebung eingesetzt. Auf eine Einführung darin wird aber bewusst verzichtet, da viele Schulen andere Werkzeuge bevorzugen. Einen expliziten Bezug darauf gibt es daher nur an wenigen Stellen, so dass dieses Arbeitsbuch ohne Probleme auch mit anderen Programmierumgebungen (oder sogar einem einfachen Texteditor) eingesetzt werden kann. Spezielle Klassenbibliotheken sind für den Einsatz dieses Buches nicht nötig. Ein Lehrerband mit Kommentaren und Lösungen ist bereits erhältlich.

Alle verwendeten Graphiken und Klassen wurden selbst erstellt.

Anregungen und Verbesserungsvorschläge sind herzlich unter appel@ceci.de willkommen.

Dr. D. Appel

Düsseldorf, den 4. Juni 2013

¹<http://www.standardsicherung.schulministerium.nrw.de/abitur-gost/getfile.php?file=3216>

Inhaltsverzeichnis

1	Einführung	3
2	Warteschlangen	4
2.1	Eine Warteschlange in Supermarkt	4
2.2	Entfernen des ersten Wartenden	5
2.3	Artikel auf dem Kassensband	6
2.4	Im Wartezimmer	9
2.5	Queues: Warteschlangen ganz allgemein	12
2.6	PROJEKT KIOSK	15
3	Kellerspeicher	16
3.1	Stapel(n) von Daten	16
3.2	Ordnung im Schuhlager	19
3.3	PROJEKT EISVERKÄUFER	20
4	Listen	22
4.1	Daten flexibel speichern	22
4.2	Spielkarten suchen	27
4.3	Insertionsort	28
4.4	PROJEKT KARTENSPIEL	30
5	Baumstrukturen	31
5.1	Rekursive Methoden	31
5.1.1	Rekursive Zeichnungen mit dem Zeichenroboter	31
5.1.2	Rekursive Zeichnungen mit dem Winkelzeichner	33
5.1.3	Rekursive Methoden in der Mathematik	36
5.2	Binäre Bäume	38
5.2.1	Einführung und Implementierung	38
5.2.2	Anwendung und Traversierung	39
5.2.3	Sortieren mittels binärer Bäume	41
5.3	Binäre Suchbäume	42
5.3.1	Die Klasse Item	42
5.3.2	Implementierung von binären Suchbäumen	44
5.3.3	Entfernen aus binären Suchbäumen	47
5.4	PROJEKT VOKABELTRAINER	50

1 Einführung

Beim Umgang mit großen Datenmengen ist es wichtig, diese in einer *strukturierten* Weise zu speichern. Was man genau mit „strukturiert“ meint, werden wir mit der Zeit besser verstehen. Um aber eine erste Vorstellung davon zu bekommen, sei erwähnt, dass wir bereits *Arrays* als eine Möglichkeit kennengelernt haben, eine große Anzahl von Daten strukturiert zu speichern. Dies bietet einige Vorteile. Zum Beispiel muss man nicht jeder Variablen einen eigenen Namen geben, sondern kann ausnutzen, dass sie alle durchnummeriert sind. Außerdem kann man verschiedene Sortieralgorithmen mithilfe von Arrays sehr leicht umsetzen.

Arrays haben aber auch Nachteile. Zum Beispiel muss man schon beim Anlegen eines Arrays seine Größe festlegen. Diese kann später nicht mehr geändert werden. Außerdem ist es dadurch sehr mühselig, neue Objekte hinzuzufügen oder auch schon vorhandene Objekte zu entfernen.

Wir werden daher verschiedene Alternativen kennenlernen und diese auch selbst implementieren. Dabei werden wir immer versuchen, uns diese anhand von Alltagssituationen vor Augen zu führen.

Neben den Datenstrukturen, die wir hier kennenlernen gibt es noch viele weitere. Da wir aber alle Datenstrukturen selbst implementieren, seid Ihr am Ende dieser Reihe sogar in der Lage, Euren eigenen Bedürfnissen nach neue Datenstrukturen zu entwerfen.

2 Warteschlangen

2.1 Eine Warteschlange in Supermarkt

Aufgabe 1. Erscheint ein neuer Wartender, so hat er noch keinen Nachfolger, sondern ist erst einmal der Letzte in der Schlange. Daher kann man ihm beim Erschaffen keinen Nachfolger zuweisen.

Aufgabe 2. Die Anweisung

```
System.out.println(erster.getNachfolger().getName());
```

liefert die gewünschte Ausgabe.

Aufgabe 3. Man kann die Ausgaben wie folgt prüfen:

```
Wartender erster = new Wartender("Anton");

System.out.println(erster.getNachfolger());

System.out.println(erster.getNachfolger().getName());
```

Als Nachfolger wird `null` zurückgegeben, da der Wartende keinen Nachfolger besitzt. Versucht man, den Namen des Nachfolgers ausgeben zu lassen, erhält man eine Fehlermeldung, da das leere Objekt die Methode `getName()` nicht ausführen kann.

Aufgabe 4. Die einfachste Variante:

```
public class WarteschlangenTest {

    public static void main(String[] args) {

        Wartender erster = new Wartender("Anton");
        erster.setNachfolger(new Wartender("Berta"));
        erster.getNachfolger().setNachfolger(new Wartender("Caesar"));

        System.out.println(erster.getName());
        System.out.println(erster.getNachfolger().getName());
        System.out.println(erster.getNachfolger().getNachfolger().getName()
        );
    }
}
```

Zusatzaufgabe 1. Mithilfe einer Schleife könnte man sich diese Namen wie folgt ausgeben lassen:

```
Wartender aktueller = erster;
while(aktueller!=null){
    System.out.println(aktueller.getName());
    aktueller = aktueller.getNachfolger();
}
```

Aufgabe 5. Hier werden zunächst Anton und Berta hintereinander gestellt. Danach wird Caesar erschaffen. Da ihm Anton als Nachfolger zugewiesen wird, stehen die drei dann in der Reihenfolge Caesar, Anton, Berta hintereinander.

Aufgabe 6. Hier werden wieder zunächst Anton und Berta hintereinander gestellt. Danach wird Caesar erschaffen. Er erhält die Nachfolgerin Dora. Ihm wird schließlich Anton als Nachfolger zugewiesen, so dass die vier in der Reihenfolge Caesar, Dora, Anton, Berta hintereinander stehen.



SCHOOL-SCOUT.DE

Unterrichtsmaterialien in digitaler und in gedruckter Form

Auszug aus:

*Modellierung und Implementierung von Datenstrukturen mit
Java*

Das komplette Material finden Sie hier:

[Download bei School-Scout.de](https://www.school-scout.de)

