

# Informatik Abitur Bayern 2013 / I - Beispiellösung

Autor:  
Schönlein

12

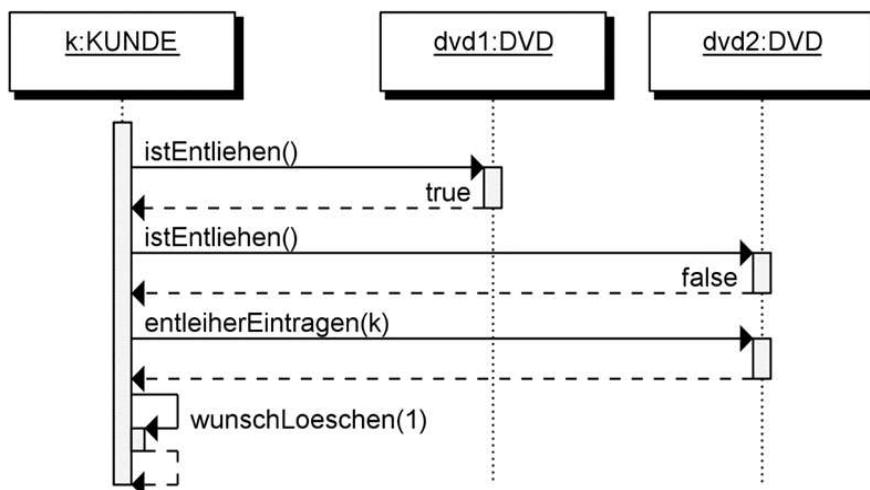
```
1a // Deklaration der Attribute:
private String name;
private String adresse;
private int kundenummer;
private DVD ausgelieheneDvd;
private DVD[] wunschzettel;

// Aufbau der gesuchten Methode:
public void wunschdvdanfordern() {
    int counter = 0;

    if (ausgelieheneDvd != null) {
        System.out.println("Sie haben bereits eine DVD
ausgeliehen");
    } else {
        while (((ausgelieheneDvd != null)
            && (counter < wunschzettel.length)
            && (wunschzettel[counter] != null)) {
            if (!wunschzettel[counter].IstEntliehen()) {
                wunschzettel[counter].entleiherEintragen(this);
                ausgelieheneDvd = wunschzettel[counter];
                wunschLoeschen(counter);
            }
            counter = counter + 1;
        }
        if (ausgelieheneDvd != null) {
            System.out.println("Es wurde keine ausleihbare DVD
gefunden!");
        }
    }
}
```

Zunächst wird geprüft, ob bereits eine DVD entliehen ist. Ist dies nicht der Fall, so wird der Wunschzettel in einer bedingten Wiederholung so weit durchgegangen, bis eine nicht ausgeliehene DVD auf diesem gefunden wurde. Daraufhin wird der aufrufende Kunde als Entleiher gesetzt und der Wunsch vom Wunschzettel gelöscht. Ist keine DVD auf diesem ausleihbar, so wird eine entsprechende Meldung ausgegeben.

1b

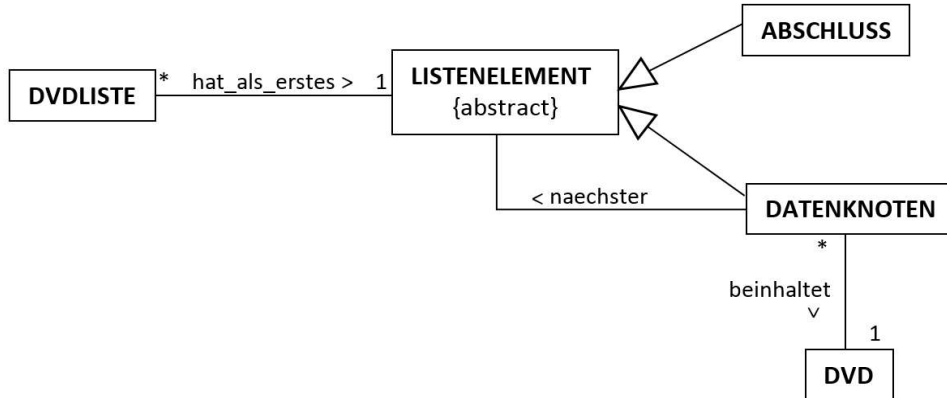


10

Falls nach dem zweiten Aufruf der Methode `istEntliehen()` ein anderer Kunde bei genau dieser DVD die Methode `entleiherEintragen` ausführt kann es zu einer Problemsituation kommen. Würde beispielsweise durch einen Monitor der Zugriff eines Objekts der Klasse KUNDE auf eine DVD so geregelt werden, dass der jeweilige Zugriff nicht unterbrochen werden darf, so könnte das Problem vermieden werden.

2a

6



Hinweis: Zur vollständigen Trennung von Struktur und Inhalt kann zwischen den Klassen DATENKNOTEN und DVD ein Interface DATENELEMENT eingefügt werden.

2b

14

```

class DVDLISTE {
    private LISTENELEMENT hat_als_erstes;

    DVDLISTE() {
        hat_als_erstes = new ABSCHLUSS();
    }

    void DVDHinzufügen(DVD d) {
        hat_als_erstes = new DATENKNOTEN(d, hat_als_erstes);
    }

    void zeigeAlleTitel() {
        hat_als_erstes.zeigeAlleTitel();
    }
}

abstract class LISTENELEMENT {
    abstract void zeigeAlleTitel();
}

class DATENKNOTEN extends LISTENELEMENT {
    private LISTENELEMENT naechstes;
    private DVD beinhaltet;

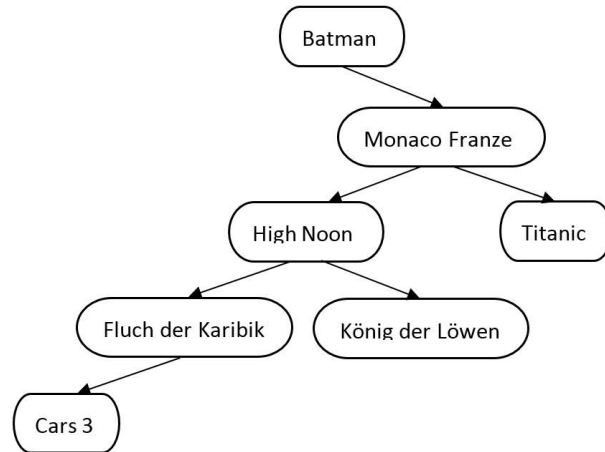
    DATENKNOTEN(DVD dvd1, Listenelement n) {
        beinhaltet = dvd1;
        naechstes = n;
    }

    void zeigeAlleTitel() {
        naechstes.zeigeAlleTitel();
        System.out.println(beinhaltet.titelAusgeben());
    }
}

class ABSCHLUSS extends LISTENELEMENT {
    void zeigeAlleTitel() {
    }
}
    
```

3a

5



Der Baum wäre beispielsweise in der Reihenfolge High Noon, Cars 3, Monaco Franze, Fluch der Karibik, König der Löwen, Batman, Titanic ausbalanciert.

3b Inorder-Durchlauf: Mittels einer rekursiven Methode, welche in jedem Knoten zunächst 4 jeweils den linken Teilbaum, dann dessen Inhalt und daraufhin den rechten Teilbaum abarbeitet, kann der Baum in alphabetischer Reihenfolge ausgegeben werden.

3c Um eine effiziente Suche zu gewährleisten muss der Baum eine möglichst geringe Höhe 7 besitzen.

Anzahl der Baumebenen	Anzahl der speicherbaren Elemente
1 Ebene	$1 = 2^1 - 1$ Elemente
2 Ebenen	$1 + 2 = 3 = 2^2 - 1$ Elemente
3 Ebenen	$1 + 2 + 4 = 7 = 2^3 - 1$ Elemente
...	...
k Ebenen	$2^k - 1$ Elemente

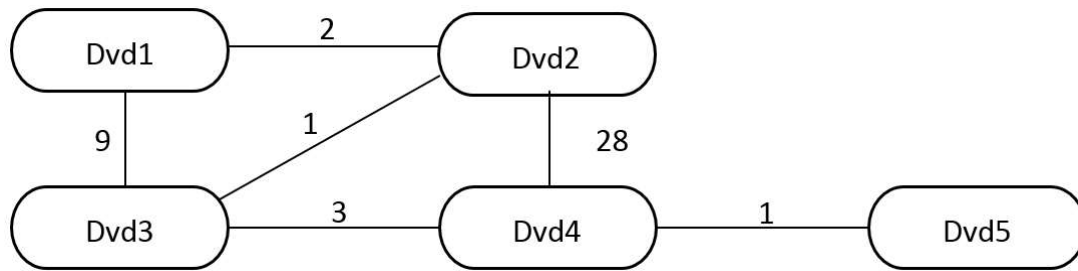
Daher kann ein Baum mit 14 Ebenen 383 Elemente, ein Baum mit 15 Ebenen insgesamt Elemente aufnehmen. Daher werden für 20000 Elemente 15 Ebenen benötigt. Die maximale Dauer einer Suche ist also 15 ns.

4a

4

	Dvd1	Dvd2	Dvd3	Dvd4	Dvd5
Dvd1	-	2	9	-	-
Dvd2	2	-	-	27	-
Dvd3	9	-	-	3	-
Dvd4	-	27	3	-	1
Dvd5	-	-	-	1	-

4b



3

4c Ein Algorithmus zum Durchlauf eines Graphen ist die Tiefensuche.

4

Es werden beispielsweise nicht alle Knoten durchlaufen, falls eine DVD nie ausgeliehen wurde.

4d Die Methode *m3* bestimmt zuerst mithilfe des Aufrufs der Methode *m1* das Maximum *u* der Kantengewichte aller Kanten, welche vom Knoten mit dem Index *k* ausgehen. Falls dieses ungleich 0 ist, werden durch Aufruf der Methode *m2* diejenigen DVD-Titel ausgegeben, welche mit dem Knoten mit dem Index *k* über eine Kante mit dem Gewicht *u* verbunden sind.

Zusammenfassend werden alle DVDs ausgegeben, die mit der *k*-ten DVD am häufigsten in einer Historie aufgeführt waren.