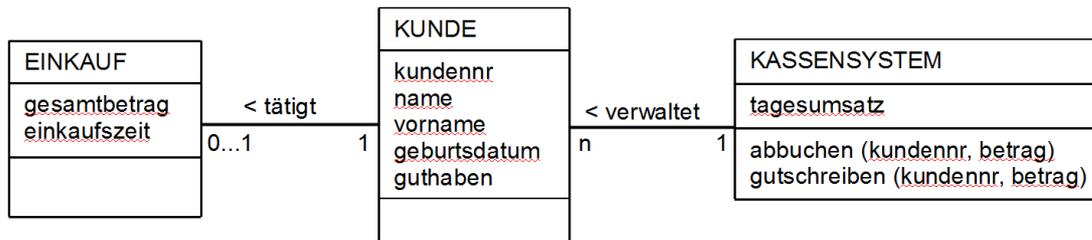


# Informatik Abitur Bayern 2012 / II - Lösung

Autor:  
Wankerl

1



9

2a Die Liste ist dynamisch, d.h. es wird immer der passende Speicherplatz belegt, wohingegen beim Feld oft Plätze leer bleiben oder die Länge des Feldes zu klein werden kann. Die Methoden zur Verwaltung sind einfacher. Im Anwendungsfall sprechen diese Gründe für die Verwendung der Liste. 6

2b In der abstrakten Klasse LISTENELEMENT:  
`abstract int teilsummeGeben();` 11

In der Klasse KNOTEN:

```
int teilsummeGeben() {
    return daten.gesamtbetragGeben()+naechster.teilsummeGeben();
}
```

In Klasse ABSCHLUSS:

```
int teilsummeGeben() {
    return 0;
}
```

In der Klasse EINKAUFLISTE:

```
int gesamtsummeGeben() {
    return erster.teilsummeGeben();
}
```

2c Jeder Einkauf könnte wiederum eine Liste verwalten, in der die Einzelposts verwaltet werden. 3

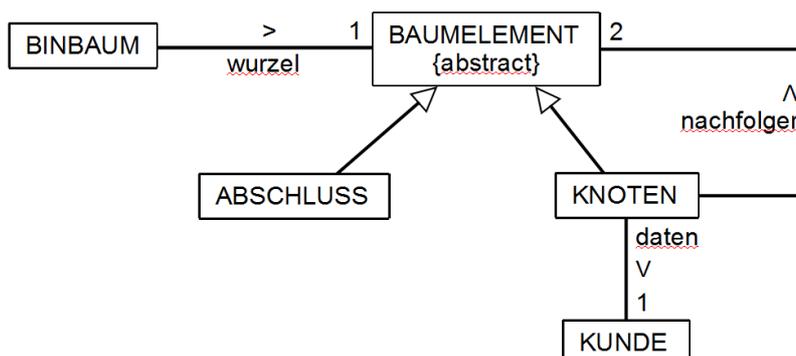
3a Vorteil eines geordneten Binärbaums: 4

Anzahl Ebenen	1	2	3	4	n	10	11
Anzahl Elemente	$1 = 2^1 - 1$	$3 = 2^2 - 1$	$7 = 2^3 - 1$	$15 = 2^4 - 1$	$2^n - 1$	$1023 = 2^{10} - 1$	$2047 = 2^{11} - 1$

Maximal 11 Suchschritte sind nötig.

3b

14



Genauer könnte man die Beziehung zwischen KNOTEN und BAUMELEMENT als zwei Beziehungen mit unterschiedlicher Semantik (hat als linken/rechten Nachfolger) schreiben.

**Klasse BINBAUM:**

```
KUNDE kundeSuchen(int ges_nr){
    return wurzel.kundeSuchen(ges_nr);
}
```

**Klasse BAUMELEMENT:**

```
abstract KUNDE kundeSuchen(int ges_nr);
```

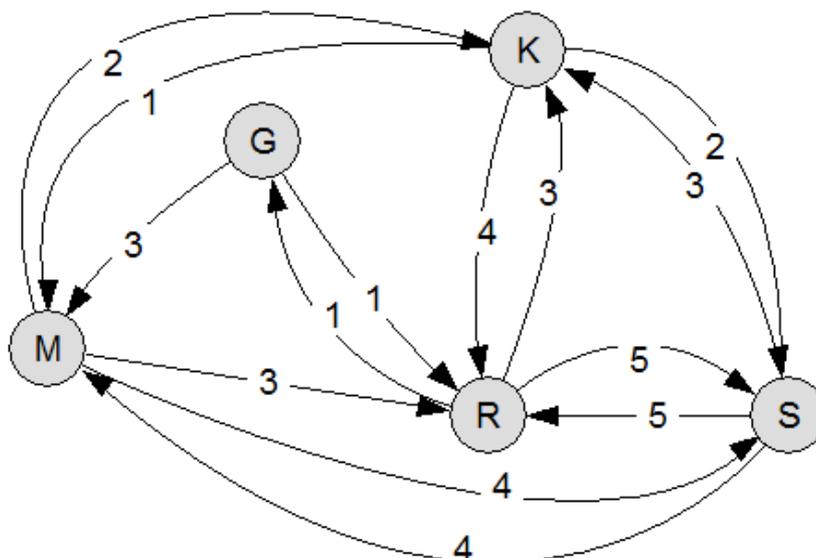
**Klasse KNOTEN:**

```
KUNDE kundeSuchen(int ges_nr){
    if (daten.kundennummerGeben() == ges_nr)
    {
        return daten;
    }
    else
    {
        if (daten.kundennummerGeben() > ges_nr)
        {
            return liNf.kundeSuchen(ges_nr);
        }
        else
        {
            return reNaf.kundeSuchen(ges_nr);
        }
    }
}
```

**Klasse ABSCHLUSS:**

```
KUNDE kundeSuchen(int ges_nr){
    return null;
}
```

- 3c Der neu eingefügte Teilbaum entartet zur Liste, da alle Elemente rechts vom letzten eingefügten Element eingeordnet werden. 3  
 Für den letzten eingefügten Schüler benötigt man nun  $11 + 190 = 201$  Suchschritte. In einem balancierten Baum würde man immer noch mit 11 Schritten auskommen. Die Suchzeit wird unnötig lang.
- 3d Die Inorderausgabe, bei der zunächst rekursiv der linke Teilbaum des Knoten, dann die Daten des Knoten und schließlich der rechte Teilbaum ausgegeben wird, leistet das Gewünschte. Wenn links die kleineren Kundennummern einsortiert wurden, entsteht eine aufsteigend sortierte Ausgabe. 6
- 4a Die Matrix ist nicht symmetrisch → gerichteter Graph 5



- 4b Der Graph kann nicht als Baum dargestellt werden, da er Zyklen enthält. 3
- 4c
- $G \rightarrow R \rightarrow S \rightarrow K$  : Länge 9 3
  - $G \rightarrow M \rightarrow K$  : Länger 5
- 4d z.B.: Tiefensuche 4  
 Dabei geht man ausgehend von einem Knoten so lange in die Tiefe, bis es keine nicht besuchten Nachbarknoten mehr gibt. In zusammenhängenden Graphen lässt sich daraus ein Pfad ermitteln, der jedoch in der Regel nicht optimal ist. Daher ist die Tiefensuche hier nicht geeignet. (Besser  $\rightarrow$  Dijkstra-Algorithmus)
- 5a z.B.: Das Wasserfallmodell 6
- Anforderungsanalyse: Festlegung der Anforderungen an das System. Ergebnis ist eine Anforderungsbeschreibung  $\rightarrow$  Dient als "Vertrag" zwischen Anwender/Entwickler
  - Entwurf: Modellierung des Systems
  - Implementierung: Codierung des Entwurfs in einer Programmiersprache
  - Test und Integration: Test der einzelnen Komponenten, Einbau, Systemtest
  - Einsatz & Wartung: Fehlerbeseitigung nach Inbetriebnahme
- 5b Meilensteine sind eingangs definierte Zeitpunkte, bis zu denen ein bestimmter Teil des Projekts erledigt sein muss. 3  
Zum Beispiel: Bis zum 09.09.2015 muss die Modellierung abgeschlossen sein.

80