Informatik Abitur Bayern 2013 / II - Beispiellösung

Autor:
Sigl B.

1a Das Pflichtenheft ist ein Katalog, der die eindeutig und überprüfbar formulierten Ansprüche an das, vom Auftraggeber gewünschte, System enthält. Im Gegensatz zum Lastenheft werden die Anforderungen im Pflichten aus technischer Sicht festgehalten. 3

1b In der Softwareentwicklung wird vorzugsweise auf erprobte, vorhandene Lösungsansätze von ähnlichen Problemstellungen zurückgegriffen. Solche Lösungsmöglichkeiten werden als Softwaremuster bezeichnet. 2

1c Die Grundidee des MVC- Modells besteht darin, dass das Softwaresystem in drei Komponenten aufgeteilt wird. Zum einen gibt es eine Datenschicht, genannt das „Model“, und zum anderen eine Präsentationsschicht, bezeichnet als die „View“, und als drittes die Kontrollschicht, den „Controller“. 7

- Im Model sollen die Daten, die für das System benötigt werden, gespeichert und aktualisiert werden.
- Die View dient dazu die Daten aus dem Model für den Benutzer darzustellen und gegebenenfalls desen Eingaben weiterleiten.
- Im Controller werden die Benutzeraktionen aus einer oder mehreren Views entgegengenommen und entsprechend Folgeaktionen eingeleitet.

Ein Vorteil dieses speziellen Softwaremusters liegt in seiner Komponentenstruktur. Benötigt man z.B. eine Änderung in der Logik, so muss man nur das Model verändern und die beiden anderen Komponenten bleiben unberührt.

2a In einer sogenannten Queue wird nach dem FIFO- Prinzip vorgegangen. Mit Hilfe der Methoden „einfuegen(datenobjekt)“ und „entfernen()“ kann solch eine Warteschlange realisiert werden. Beim Einfügen wird ein Datenobjekt an das Ende der Schlange angefügt und beim Entfernen wird immer am Anfang der Schlange das erste Datenobjekt entfernt. 3

2b Da das Hochregallager drei Fächer besitzt, kann es passieren, dass ein Container eher sein Fach verlässt als ein früher eingelagerter Container. Somit wäre das FIFO-Prinzip verletzt 3

2c Das Hochregallager dürfte nur ein Fach besitzen und nur von derselben Seite be- und entladen werden. 2

3 12

```
public class DURCHLAUFREGAL
{
   private FACH[] faecher;

   public DURCHLAUFREGAL(int faecherAnzahl)
   {
     faecher = new FACH[faecherAnzahl];
     for (int i = 0; i< faecherAnzahl; i++){
       faecher[i] = new FACH(i+1);
       }
   }

   public boolean einlagernMoeglich(double containerlaenge, int fachnummer)
   {
      if( faecher[fachnummer-1].nochFreieLaengeGeben() >= containerlaenge){
       return true;
       }
       else return false;
   }

   public void einlagern(CONTAINER container, int fachnummer)
   {
       faecher[fachnummer-1].einfuegen(container);
   }
}
```

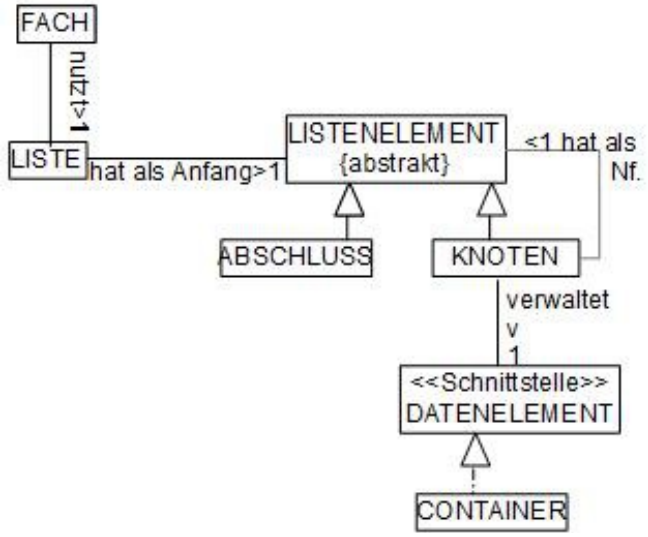
4 ➢ Ein Aspekt: 4

Da man von vornherein nicht sagen kann, wie viel Container in einem Fach untergebracht werden (können), müsste man zu Beginn ein sehr großes Feld instanziieren, welches aber deswegen auch viel unnötigen Speicherplatz benötigt.

➢ Zweiter Aspekt:

Bei einer Warteschlange wird immer das erste Element entnommen und Element am Ende eingefügt. Für eine Realisierung der enthält- Beziehung würde das heißen, dass nach jeder Entnahme eines Containers das Feld umkopiert werden müsste.

5a 7

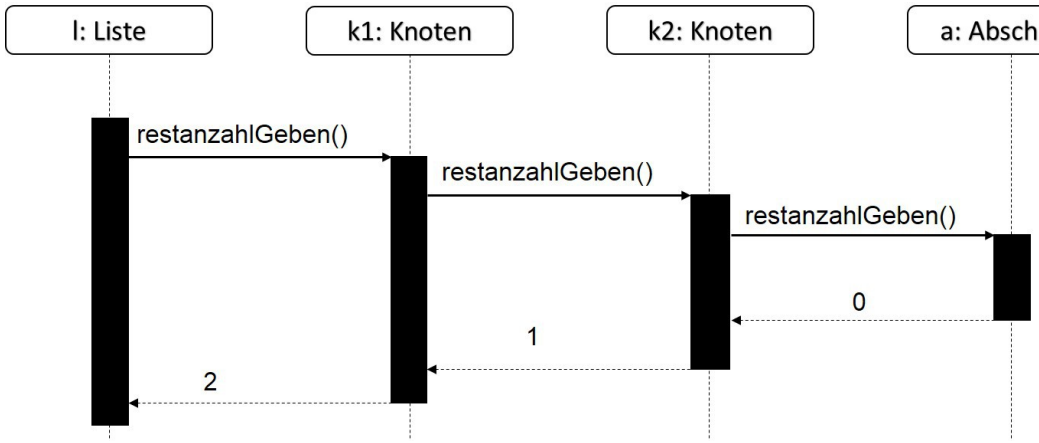


5b 11

```
public class LISTE
{
   private LISTENELEMENT anfang;

  public int anzahlGeben(){
   return anfang.restanzahlGeben();
   }
}

public abstract class LISTENELEMENT
{
   abstract int restanzahlGeben();
}


public class KNOTEN extends LISTENELEMENT
{
  private LISTENELEMENT nachfolger;
  private DATENELEMENT cont;


  public int restanzahlGeben(){
   return nachfolger.restanzahlGeben()+1;
   }

}

public class ABSCHLUSS extends LISTENELEMENT
{
   public int restanzahlGeben(){
```

```
        return 0;
    }
}
```

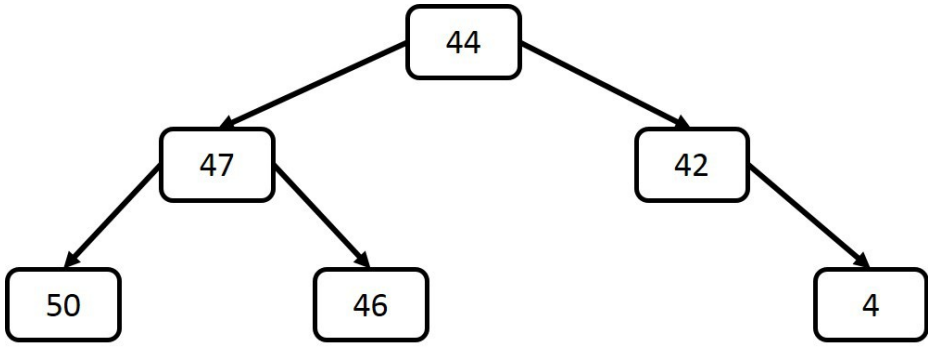
5c



6

6a Ausgehend von der ID des jeweiligen Vaterknoten werden die Knoten mit größerer ID in den linken Teilbaum eingeordnet und die Knoten mit kleinerer ID als der Vaterknoten in den rechten Teilbaum 2

6b 5



Die ID 50 wird in den linken Teilbaum eingefügt, da 50 > 44, und die ID 4 wird aufgrund der Tatsache, 4<44, in den rechten Teilbaum eingefügt. Somit ist es egal, in welcher Reihenfolge die beiden IDs eingefügt werden.

6c 7

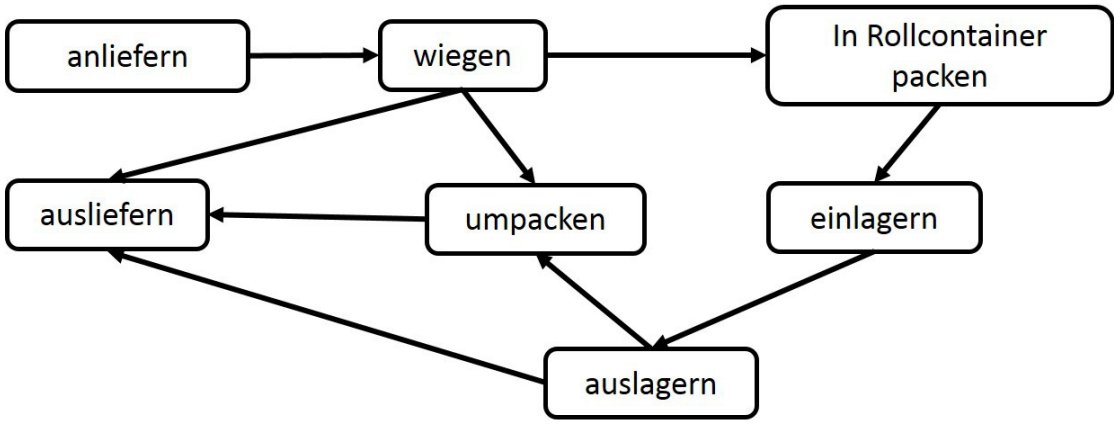
Referenzattribute der Klasse KNOTEN:

- linkerNachfolger mit Referenz auf den linken Nachfolgerknoten
- rechterNachfolger mit Referenz auf den rechten Nachfolgerknoten
- daten mit Referenz auf das Container-Objekt

Algorithmus der Methode „spezialdatenAusgeben()“:

```
rechterNachfolger.spezialdatenAusgeben()
if (daten.containerlaengeGeben()>=1) {
        daten.datenAufKonsoleAusgeben()
}
linkerNachfolger.spezialdatenAusgeben()
endeMethode
```

7



5