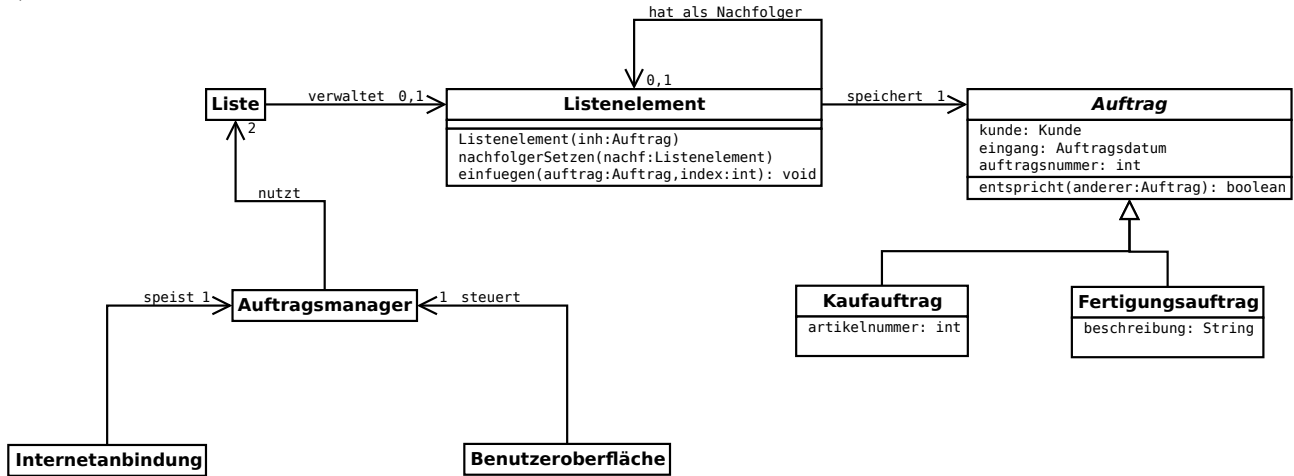


Q11/12 Schulaufgabe Informatik, 09.12.20, Beispiellösung

Aufgabe 1

a)



b)

```

suchen(auftrag: Auftrag): Auftrag
loeschen(auftrag: Auftrag): Listenelement
    
```

c)

Der Klassenbezeichner wurde kursiv gedruckt, um zu signalisieren, dass es sich um eine sogen. abstrakte Klasse handelt, eine Klasse von der keine Objekte erzeugt werden können. Sie erzwingt, dass alle Auftragsklassen – die Unterklassen von *Auftrag* sein müssen – egal für welche Art von Auftrag sie geschaffen werden, die *entspricht*-Methode mit der in der Oberklasse angegebenen Signatur zur Verfügung stellen. Diese Methode wird zum Vergleichen von Auftragsobjekten gebraucht: Ohne sie könnten keine Methoden zum Suchen und Löschen eines bestimmten Objekts (das ja erst einmal über einen Vergleich identifiziert werden muss) implementiert werden.

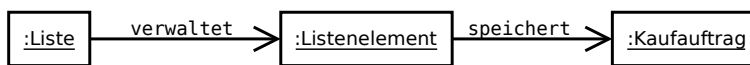
d)

Es handelt sich bei dieser Methode um einen Konstruktor der Klasse. Im gegebenen Fall muss über den Konstruktor gleich bei der Erzeugung eines *Listenelement*-Objekts der zu speichernde Auftrag (Attributwert) gesetzt werden – es darf ja kein *Listenelement* geben, das nichts speichert (vgl. Kardinalität im Klassendiagramm).

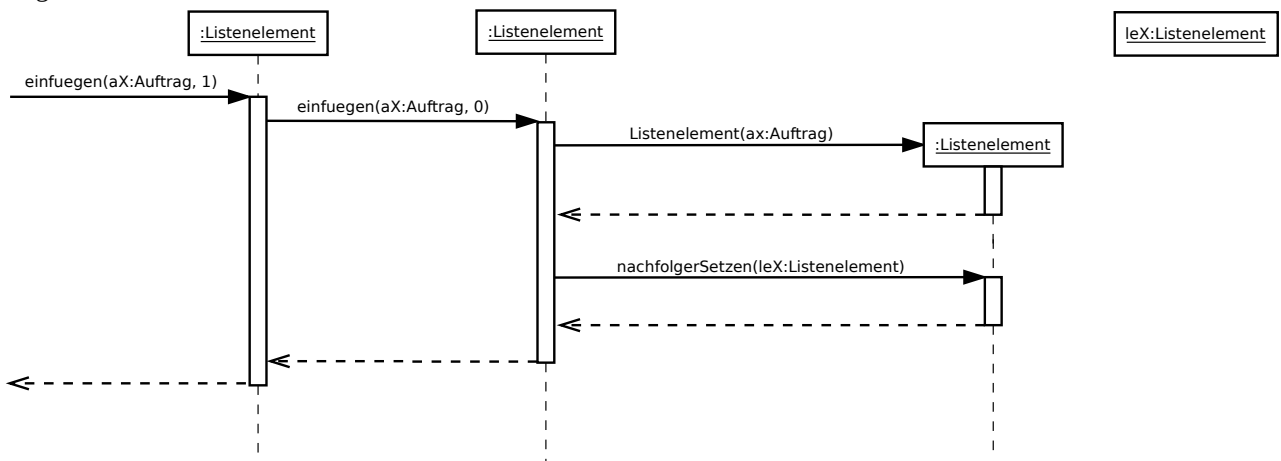
e)

Idealerweise sollte jedes Objekt möglichst weitgehende und möglichst ungeteilte Kontrolle über seine Attribute und deren Werte haben.

Aufgabe 2



Aufgabe 3

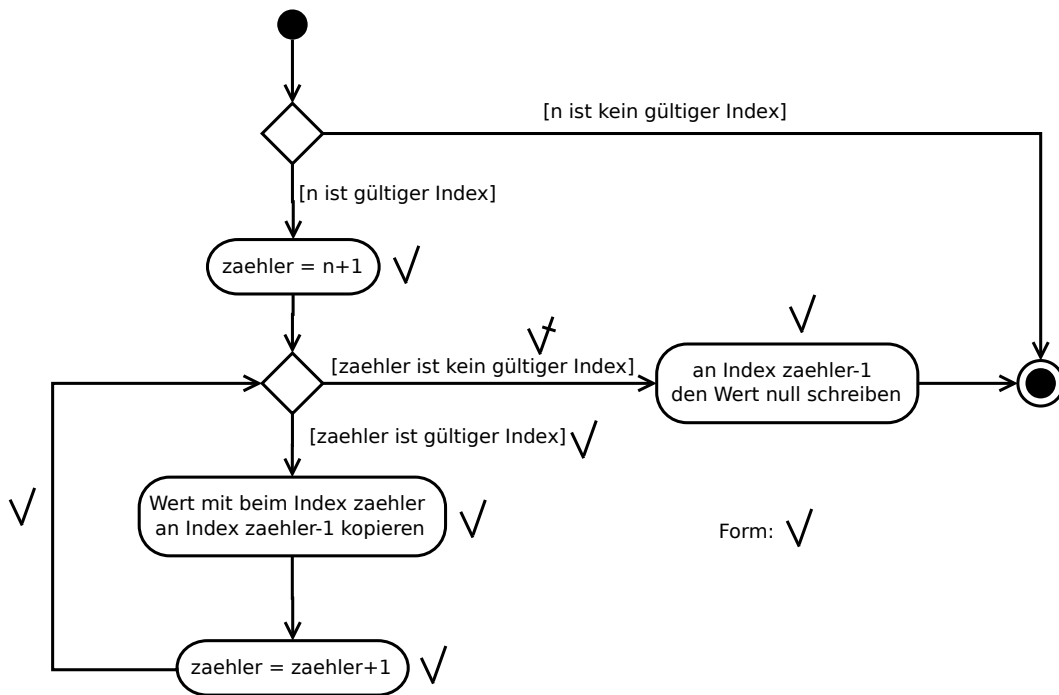


Aufgabe 4

```
private Listenelement nachfolger; // 1 BE
private Auftrag inhalt; // 1 BE

public Listenelement loeschen(int index) { // 3 BE
    if (index == 0) { // 1 BE
        return nachfolger; // 1 BE
    }
    else { // 0,5 BE
        if (nachfolger != null) { // 1 BE
            nachfolger = nachfolger.loeschen(index - 1); // 3 BE
        }
        return this; // 1 BE
    }
} // Form: 0,5 BE
```

Aufgabe 5



Als Nachteil wird hier die Notwendigkeit des „Nachrückens“ aller Werte nach dem gelöschten gesehen. Im gegebenen Fall entspricht dem aber bei der rekursiven Implementierung die Notwendigkeit, alle Elemente vor dem zu löschenden zu durchlaufen.

Aufgabe 6

Rekursion liegt grundsätzlich vor, wenn eine Instanz eines Schemas zur Lösung eines Problems auf eine andere Instanz desselben Schemas zurückgreift. In rekursiven objektorientierten Datenstrukturen äußert sich die Rekursion in Assoziationen zwischen Objekten derselben Klasse (bzw. rekursiven Beziehungen von Klassen „zu sich selbst“). In rekursiv arbeitenden Methoden ruft dann eine Methode eines Objekts einer Klasse dieselbe Methode eines anderen Objekts derselben Klasse auf (denkbar wäre auch, dass die Methode eine andere Instanz ihrer selbst für dasselbe Objekt aufruft).