
II B3 Abstrakte Datentypen

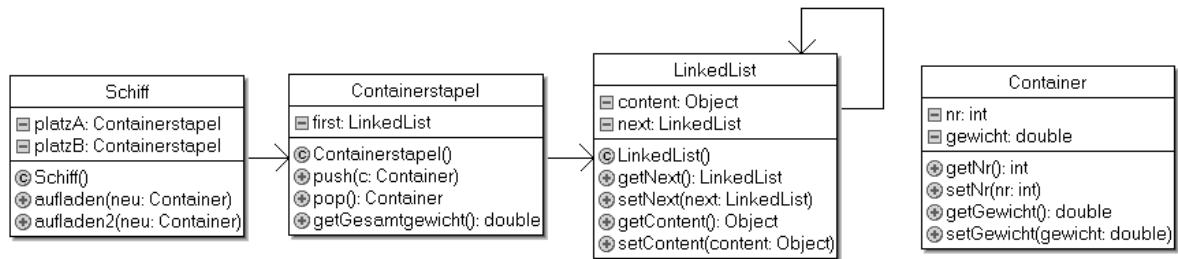
Ein Containerschiff ist ein speziell für den Transport von Containern gebautes Frachtschiff. Die Container sind genormte Behälter, die auf vorgesehenen Plätzen aufeinander gestapelt werden. Ein besonderer Schiffstyp, das sogenannte Feederschiff, hat nur eine geringe Anzahl von Plätzen zur Aufnahme von Containern und kann somit auch kleine Häfen anlaufen.



"Ship Aurora" by Tvabutzku1234 - Own work. Licensed under CC0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Ship_Aurora.jpg#mediaviewer/File:Ship_Aurora.jpg

Eine Reederei besitzt mehrere Feederschiffe mit jeweils genau zwei Stapelplätzen. Die Plätze werden auf jedem Schiff mit platzA und platzB bezeichnet. Jeder Platz kann genau einen Containerstapel aufnehmen. Ein Container auf einem solchen Stapel darf nicht mehr als 28 t wiegen.

- B3.1 Die Reederei möchte die Frachten der Feederschiffe mit einem objektorientierten Computerprogramm verwalten. Das folgende Diagramm zeigt die beteiligten Klassen. Die Klasse Containerstapel stellt dabei eine Erweiterung des ADT Stapel dar:



- Erläutern Sie, welche Auswirkungen die Methoden push und pop auf einen Stapel haben. Begründen Sie, warum für diese Aufgabenstellung die Modellierung mit einem Stapel sinnvoll ist.
- Die Klasse Containerstapel besitzt die Methode `getGesamtgewicht() : double`. Implementieren Sie diese Methode in der im Unterricht verwendeten Programmiersprache. Beachten Sie dabei, dass hierfür keine zusätzlichen Attribute in die Klasse Containerstapel eingefügt werden dürfen..

(6 VP)

B3.2 Das Schiff soll möglichst ausgeglichen beladen werden. Aus diesem Grund werden neue Container immer auf dem leichteren Stapel abgelegt, bei gleich schweren Stapeln auf platzA.

- Implementieren Sie die Methode `aufladen(neu: Container)` in der Klasse Schiff, die den übergebenen Container jeweils auf dem bisher leichteren der beiden Stapel ablegt, bzw. auf platzB, wenn beide Stapel gleich schwer sind.

(4 VP)

B3.3 Eine experimentelle Lademethode `aufladen2(neu: Container)` beruht auf dem folgenden Algorithmus, hier notiert in Pseudocode. Er soll nur dann angewendet werden, wenn auf beiden Plätzen mindestens ein Container liegt. Die Prüfung, ob beide Stapel Container enthalten, ist hier nicht vorgesehen.

Algorithmus aufladen2(neu: Container)

```

a = platzA.getGesamtgewicht();
b = platzB.getGesamtgewicht();
c = neu.getGewicht();
wenn (a < b)
    wenn (a + c > b + 20)
        platzA.push(platzB.pop());
        platzB.push(neu);
    sonst
        platzA.push(neu);
    ende wenn
sonst
    wenn (b + c > a + 20)
        platzB.push(platzA.pop());
        platzA.push(neu);
    sonst
        platzB.push(neu);
    ende wenn
ende wenn
  
```

Bei einem Ladevorgang befinden sich zu Beginn auf platzA und platzB je ein Container mit 10 t. Es sollen nun drei Container mit den Gewichten in der Reihenfolge 21 t, 27 t und 20 t geladen werden.

- Stellen Sie das Ergebnis des Ladevorgangs in einer Skizze dar.
- Analysieren Sie, welche Vorteile dieser Algorithmus gegenüber Ihrem Algorithmus aus Aufgabe B3.2 bietet.
- Beurteilen Sie, ob die experimentelle Lademethode in den Produktivbetrieb übernommen werden sollte.

(6 VP)

B3.4 Zur Vorbereitung des Ladevorgangs wird eine Ladeliste erstellt, in der die Container – bezogen auf die Zielhäfen – in umgekehrter Reihenfolge notiert werden, damit die Container, die zuerst ausgeladen werden müssen, ganz oben stehen. Neu hinzu kommende Container werden an der richtigen Stelle in die Liste eingefügt. Dafür bietet eine verbesserte Containerstapel-Klasse eine Methode `ein fuegenAnKorrekte rPosition(c:Container)` an.

- Beschreiben Sie, wie innerhalb der Methode die Verkettung der Objekte verändert werden muss, um ein neues Objekt innerhalb einer verketteten Liste einzufügen. Veranschaulichen Sie die Beschreibung unter Verwendung einer aussagekräftigen Skizze.

(4 VP)