

Schulaufgabe aus dem Fach Informatik
am 09.12.2020

Polymorphie, Listen (und deren Spezialfall Warteschlange)

Szenario

Der Juwelierladen „Die Goldschmiede“ von Goldschmiedemeisterin Dorothea Goldhammer hat, nachdem derzeit nur wenige Kunden direkt den Laden besuchen und dort etwas kaufen, kleine Kataloge drucken lassen und an die Stammkundschaft verschickt. Und nun gehen immer mehr Aufträge per Telefon und Internet ein. Deshalb soll jetzt eine Software zur Auftragsverwaltung erstellt werden.

Prinzipiell unterscheidet Frau Goldhammer zwei Arten von Aufträgen: „Kaufaufträge“, bei denen jemand ein bereits fertiges Produkt kauft, und „Fertigungsaufträge“, bei denen ein Produkt nach besonderen Kundenwünschen angefertigt werden muss.

Es ist bereits klar, dass die eingehenden Aufträge, sobald sie elektronisch erfasst sind, in einer Liste noch nicht geprüfter Aufträge gespeichert werden sollen. Wenn Frau Goldhammer einen Auftrag geprüft hat, verwirft sie ihn entweder oder sie verbucht ihn als angenommenen Auftrag – dann soll er aus der Liste der noch nicht geprüften Aufträge entfernt werden und in einer weiteren Liste, der Liste geprüfter Aufträge, gespeichert werden. Die beiden Listen sollen von einem `Auftragsmanager`-Objekt verwaltet werden. Der Zugriff darauf erfolgt entweder über ein Objekt einer Klasse `Benutzeroberfläche` (das eine grafische Benutzeroberfläche bereitstellt, über die Frau Goldhammer das Programm steuern kann) oder über ein Objekt einer Klasse `Internetanbindung` (das Aufträge, die über das Internet eingehen, automatisiert in das System einspeist).

Ein noch unfertiger Entwurf für die Software ist im Klassendiagramm auf dem Diagrammblatt dargestellt.

Aufgabe 1: Klassendiagramm (und die Hintergründe dazu)

a) Ergänzen Sie im Klassendiagramm auf dem Diagrammblatt die laut Szenario fehlenden Klassen (ohne Attribute und Methoden). Tragen Sie auch die nötigen Beziehungen (mit allem, was dazugehört) ein. Vervollständigen Sie in diesem Zusammenhang zudem die Listenstruktur, sodass sie (soweit das unter den gegebenen Umständen noch möglich ist) von den übrigen Teilen der Software getrennt wiederverwendet werden könnte. Ergänzen Sie schließlich in den drei Klassen, in denen dafür Platz gelassen wurde, (und nur dort) jeweils ein sinnvolles Attribut.

12 BE

b) Nennen Sie zwei weitere Methoden, über die die Klasse `Listenelement` verfügen müsste, damit die gängigen Anforderungen an eine Liste erfüllt werden können. (Geben Sie deren vollständige Signatur mit Datentypen an.)

5 BE

c) Erklären Sie, warum der Bezeichner der Klasse `Auftrag` kursiv gedruckt wurde. Klären Sie dabei, was es mit der Klasse `Auftrag` demnach auf sich hat, was das mit der `entspricht`-Methode (die wir im Unterricht anders genannt haben) zu tun hat und wozu diese Methode benötigt wird.

2 BE

d) Die Methode mit dem Bezeichner `Listenelement` ist eine besondere Methode, die wir im Unterricht in der Regel nicht in Klassendiagramme eingetragen haben. Geben Sie knapp an, wie man eine solche Methode nennt und was sie im gegebenen Fall leisten muss.

2 BE

e) Geben Sie knapp an, warum nach gängigen Prinzipien der Software-Entwicklung eigentlich möglichst auf die `nachfolgerSetzen`-Methode der Klasse `Listenelement` verzichtet werden sollte. (Wir nutzen sie im Folgenden trotzdem ...)

1 BE

Aufgabe 2: Objektdiagramm

Wir nehmen an, dass in einer der beiden Listen genau ein Kaufauftrag gespeichert ist. Zeichnen Sie ein Objektdiagramm dieser Liste (ohne Attribute und Methoden, aber mit Beziehungen).

4 BE

Aufgabe 3: Sequenzdiagramm

Das bruchstückhafte Sequenzdiagramm auf dem Diagrammblatt soll zeigen, was der ganz links eingezeichnete Aufruf der `einfüegen`-Methode bewirkt. Vervollständigen Sie das Diagramm (ohne noch weitere Objekte zu ergänzen). Entscheiden Sie dabei selbst, ob das ganz rechts eingezeichnete Objekt in die Interaktion einbezogen wird oder ob es außen vor bleibt und ihm also gar keine Nachricht geschickt wird.

Zur Erklärung: Die bereits eingezeichnete Nachricht, die (von links gezählt) vom zweiten zum dritten Objekt verläuft, bedeutet, dass das zweite Objekt das dritte erzeugt. Danach können nun auch Methoden des neu erzeugten Objekts aufgerufen werden.

6 BE

Aufgabe 4: Implementierung einer Methode der Klasse `Listenelement`

Notieren Sie in der Programmiersprache Java die Implementierung der zwei laut Klassendiagramm erforderlichen Referenzattribute und einer Methode der Klasse `Listenelement`, die das `Listenelement`-Objekt „bei Index n “ aus der Listenstruktur entfernt: Wird die Methode also zum Beispiel mit dem Parameter 5 aufgerufen, soll sie das sechste `Listenelement`-Objekt löschen. Gibt es kein `Listenelement`-Objekt „beim angegebenen Index“, weil die Liste gar nicht so lang ist, so soll der entsprechende Methodenaufruf wirkungslos bleiben.

13 BE

Aufgabe 5: Aktivitätsdiagramm

Wir nehmen an, eine Liste sei – anstatt wie oben angegeben – mithilfe eines (nicht zyklisch genutzten) Arrays implementiert worden. Zeichnen Sie ein Aktivitätsdiagramm, das zeigt, wie eine Methode zum Entfernen des Elements mit dem Index n dann funktionieren müsste. Benennen Sie zudem, was am dargestellten Algorithmus in der Regel als Nachteil der Implementierung mithilfe eines Arrays gegenüber der rekursiven Implementierung gesehen wird; und prüfen Sie, ob der genannte Nachteil im gegebenen Fall zum Tragen kommt.

9 BE

Aufgabe 6: Zum Begriff Rekursion

Geben Sie knapp an, was der Begriff Rekursion grundsätzlich bedeutet und wie sich Rekursion allgemein in rekursiven objektorientierten Datenstrukturen sowie im Besonderen in rekursiv arbeitenden Methoden äußert.

4 BE

Viel Erfolg!