

Aufgabe Bestellungen

Arbeitszeit 60 min

Einleitung:

In einer Bar soll das Bestellsystem für Getränke digitalisiert werden. Die Bedienungen geben dabei aufgegebene Bestellungen in das System ein, der Barkeeper entnimmt die Bestellungen in der selben Reihenfolge aus dem System und bereitet die Getränke zu.

Anmerkungen:

Für alle Aufgaben sind Tests vorhanden, die nur korrekt funktionieren, wenn die Klassen und deren Methoden korrekt benannt werden. Die Tests für die Aufgaben können einzeln (Rechtsklick > TestAll) oder im Gesamten ausgeführt werden. Tests, die momentan nicht kompiliert sind, werden nicht ausgeführt.

Die Aufgaben können nicht dadurch erfüllt werden, die Tests zu manipulieren! Daher sollten die Tests einfach unverändert bleiben. Stimmige Lösungsansätze werden auch dann positiv gewertet, wenn sie die Tests nicht erfüllen!

Aufgabe 1:

2+2+2 BE

Eine Getränkebestellung soll durch ein Objekt der Klasse **Bestellung** abgebildet werden. Dabei wird bei Aufruf des Konstruktors die Sorte des Getränks als String übergeben (z.B. "Cola", "Spezi", "Kaffee").

Über die Methode **String sorteGeben()** kann man die Sorte wieder abrufen.

1a) Implementieren Sie diese Klasse wie beschrieben.

1b) Getränke können entweder groß oder klein sein. Neu erstellte Objekte repräsentieren generell kleine Getränke und können dann durch Aufruf der Methode **void grossSetzen()** entsprechend verändert werden. Die Methode **groesseGeben()** soll je nach Zustand den String "gross" oder "klein" zurückgeben. Zudem soll die Methode **boolean istGross()** genau dann **true** zurückgeben, wenn es sich um ein großes Getränk handelt.

Implementieren Sie das beschriebene Verhalten. *(Anmerkung: Achten Sie bei dieser Teilaufgabe besonders auf die Schreibweise)*

1c*) Erweitern Sie die Klasse, damit die Bedienungen alternativ zum Namen des Getränks eine Nummer eingeben können. Es soll jedoch mit der Methode **sorteGeben()** weiterhin die Bezeichnung des Getränks ausgegeben werden. Beschränken Sie sich beispielhaft auf folgende Nummern:

5 = Kaffee, 23 = Spezi, 42 = Cola.

Aufgabe 2:**2+7+4 BE**

Die Struktur wird im Wesentlichen durch Objekte der Klasse **Knoten** repräsentiert.

2a) Implementieren Sie den Konstruktor der Klasse **Knoten** derart, dass er ein Objekt der Klasse **Bestellung** als Parameter empfängt. Schreiben Sie ebenfalls die Methode **bestellungGeben()**, die besagte Bestellung wieder ausgibt.

2b) Weitere Bestellungen bzw. deren Knoten können mit der rekursiven Methode **bestellungHinzufuegen(Bestellung b)** an das Ende der bisherigen Knotenliste angehängt werden.

Die Methode **hatNachfolger()** (Datentyp boolean) soll genau dann true zurückgeben, wenn ein Knoten dahinter angehängt ist.

Die Methode **nachfolgerGeben()** soll den nachfolgenden **Knoten** ausgeben. Falls dieser nicht vorhanden ist, ist *null* als Rückgabewert akzeptabel.

Erweitern Sie die Klasse **Knoten** um diese Methoden.

2c*) Da hin und wieder Fehler bei der Buchung passieren, soll eine Funktion zum Löschen der zuletzt hinzugefügten Bestellung aus der Knotenliste implementiert werden. Implementieren Sie die rekursive Methode **undo()** mit besagtem Verhalten. Wird die Methode auf einem Knoten ohne Nachfolger aufgerufen, dann ist sie funktionslos.

Aufgabe 3:**4+4 BE**

Die Klasse **BestellSystem** verwaltet die Liste im Gesamten.

3a) Implementiere diese Klasse mit folgenden Methoden:

- Der Konstruktor erzeugt eine leere Liste ohne eingetragene Knoten/Bestellungen
- Die Methode **bestellungHinzufuegen(Bestellung b)** erstellt einen neuen Knoten mit der übergebenen Bestellung am Ende der Liste.
- Die Methode **bestellungEntnehmen()** nimmt die älteste Bestellung aus der Liste heraus und gibt diese zurück. Ist keine Vorhanden, soll **null** zurückgegeben werden.

3b*) Erweitere das Projekt um die Methode **int kleineGetraenkeZaehlen()** in der Klasse **BestellSystem**, die die aktuelle Anzahl der Bestellungen kleiner Getränke zurückgeben soll.

Übersicht der Methodensignaturen:

Bestellung
Bestellung(String) groesseGeben() : String grossSetzen() : void istGross() : boolean sorteGeben() : String

Knoten
Knoten(Bestellung) bestellungGeben() : Bestellung bestellungHinzufuegen(b : Bestellung) : void hatNachfolger() : boolean kleineGetraenkeZaehlen() : int nachfolgerGeben() : Knoten undo() : void

BestellSystem
BestellSystem() bestellungEntnehmen() : Bestellung bestellungHinzufuegen(b : Bestellung) : void kleineGetraenkeZaehlen() : int