

Aufgabe Buchverein

Arbeitszeit 60 min

Einleitung:

Der "Verein für intensiven Buchgenuss" verfügt über einige Bücher, die den Mitgliedern zur Verfügung stehen. Dabei hat jedes Mitglied die Möglichkeit, sich für ein Buch vorzumerken, das er/sie lesen will. Ein Buch wird, nachdem es gelesen wurde, von einem Mitglied an das nächste vorgemerkte Mitglied weitergegeben. Bisher ist die Organisation dieses Systems mit viel Papieraufwand verbunden, der nun durch eine Softwarelösung ersetzt werden soll.

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:

3+2+4 BE

Ein Mitglied soll durch ein Objekt der Klasse **Mitglied** repräsentiert werden. Dabei werden bei Aufruf des Konstruktors Vor- und Nachname (in dieser Reihenfolge) übergeben. Über die Methoden **String vornameGeben()** und **String nachnameGeben()** kann man diese Daten wieder abrufen.

1a) Implementiere diese Klasse

1b) Erstelle die Methode **nachnameAendern(String neuerName)**, die es erlaubt, den Nachnamen eines Mitglieds (z.B. nach Eheschließung) anzupassen.

1c) Erweitere den Konstruktor, dass er sowohl für Vor- als auch für Nachnamen überprüft, dass eine Zeichenkette mit einer Mindestlänge von einem Zeichen übergeben wurde. Falls das nicht der Fall ist, wird der Vor- UND Nachname auf "!!!FEHLER!!!" festgelegt.

Hinweis: Objekte der Klasse String verfügen über die Methode `int length()` ;

Aufgabe 2:**3+7+5 BE**

Die Struktur wird im wesentlichen durch Objekte der Klasse "Knoten" repräsentiert.

2a) Implementiere den Konstruktor der Klasse Knoten derart, dass er ein Objekt der Klasse Mitglied als Parameter empfängt. Schreibe ebenfalls die Methode **mitgliedGeben()**, die besagtes Mitglied wieder ausgibt.

2b) Weitere Knoten können mit der rekursiven Methode "**nachfolgerHinzufuegen(Knoten k)**" 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.

2c) Es soll den Mitgliedern möglich sein, sich wieder aus der Liste auszutragen. Dazu soll die rekursive Methode **void knotenEntfernen(Mitglied m)** genau den Knoten aus der Liste entfernen, der für die Verwaltung dieses Mitglieds verantwortlich ist.

Hinweis: mit der Methode bla.equals(blub) kann überprüft werden, ob die Referenzen bla und blub auf das selbe Objekt zeigen.

Aufgabe 3:**6+6+4 BE**

Die Klasse "BuchListe" verwaltet die Liste im Gesamten.

3a) Implementiere diese Klasse mit folgenden Methoden:

- Der Konstruktor erzeugt eine leere Liste ohne eingetragene Knoten/Mitglieder
- Die Methode **mitgliedAnstellen(Mitglied m)** erstellt einen neuen Knoten mit dem übergebenen Mitglied und stellt diesen am Ende der Liste an.
- Die Methode **naechstesMitgliedEntnehmen()** entfernt den ersten Knoten des Stapels und gibt das darin enthaltene Mitglied zurück. Wenn die Liste leer ist, soll *"null"* zurückgegeben werden.

3b) Die Methode **letztesMitgliedGeben()** gibt das letzte in der Liste enthaltene Mitglied aus, ohne es aus der Liste zu entfernen. Wenn die Liste leer ist, soll *"null"* zurückgegeben werden.

3c) Implementiere die Methode **mitgliedEntfernen(Mitglied m)**, die das Mitglied (entsprechend 2c) aus der Liste entfernt.