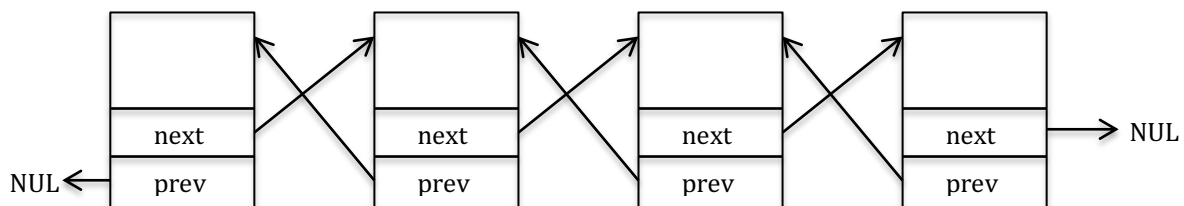


Einführung in die Objektorientierte Programmierung - SS 2021 Praktikum 2 (Stand 29.03.2021)

- Bereiten Sie diese Aufgaben vor der Praktikumsstunde vor, so dass Sie im Praktikum noch Zeit haben eventuelle Fehler zu beheben und Erweiterungen zu machen.
- Achten Sie darauf, aussagekräftige Namen für Variablen, Programmnamen und Methoden zu geben
- Dokumentieren Sie den Programmkopf und die Methoden im Javadoc-Format.
- Strukturieren Sie Ihre Programme sauber, z.B. durch entsprechende Einrückung
- Ziele des Praktikums: Verkettete Listen, Datei Ein- und Ausgabe in Java

Sie erinnern sich sicher noch an Praktikum 5 aus Einführung in die Informatik, in dem Sie eine verkettete Liste in C schreiben mussten? In diesem Praktikum müssen Sie das gleiche Programm in der Programmiersprache Java implementieren.



Aufgabe 1: Doppelt verkettete Liste

Schreiben Sie ein Java-Programm, das Daten in einer doppelt verketteten Liste verarbeitet. Anstelle von C-Strukturen sollen Sie hier Java-Klassen als Knoten der Liste verwenden. Es sollen vom Benutzer Daten hinzugefügt, gelöscht und gesucht werden können. Kapseln Sie die Funktionalität in den Methoden der Klasse `VokabelTrainer` und testen Sie das Programm sorgfältig.

Aufgabe 2: Vokabelprogramm

Vervollständigen Sie jetzt das Java-Kommandozeilen-Programm VokabelTrainer, sodass Sie damit Vokabeln üben können. Das Programm soll beliebig viele Vokabeln in der **doppelt verketteten Liste** aus Aufgabe 1 speichern und mindestens die folgenden Funktionen in einem Menü unterstützen:

- **Vokabeln einlesen.** Lies eine Datei mit beliebig vielen Vokabeln im folgenden Format ein
to study;studieren
university;Universität
...
d.h. in jeder Zeile soll ein Vokabelpaar stehen und die Fremdvokabel ist von der deutschen Vokabel durch ein Semikolon getrennt.
Tipp: schauen Sie sich die Bibliotheksklasse FileReader an.
- **Vokabeln speichern.** Speichere die aktuell in der verketteten Liste gespeicherten Vokabeln in einer Datei.
Tipp: schauen Sie sich die Bibliotheksklasse FileWriter an.
- **Neue Vokabel hinzufügen.** Der Benutzer soll manuell eine neue Vokabel über die Kommandozeile hinzufügen können.
- **Vokabel löschen.** Der Benutzer soll eine fehlerhafte Vokabel über einen Menüeintrag löschen können. Dazu gibt er den Namen der Vokabel ein. Das Programm gibt Feedback, z.B. dass die Vokabel nicht gefunden wurde oder eine Nachfrage, ob die Vokabel wirklich gelöscht werden soll.
- **Vokabeln abfragen.** Das Programm zeigt eine zufällige Vokabel und fragt den Benutzer nach der Übersetzung. Ist die eingegebene Übersetzung falsch, so gibt das Programm die korrekte Übersetzung aus.

Das Programm soll stabil laufen und Fehler entsprechend behandeln, z.B. nicht existierende Datei, Datei nicht im erwarteten Format (leere Zeile, Zeile mit falschen Daten, kein Semikolon, usw.), Vokabeln konnten nicht gespeichert werden, leere Vokabel, usw..

Wichtig: Zeigen Sie Ihrem Tutor, wie Sie Ihr Programm getestet haben!

Aufgabe 3: Klassen als UML-Diagramm

Zeichnen Sie die von Ihnen erstellten Klassen als UML-Klassendiagramm, wie wir es in der Vorlesung gezeigt hatten.

Tipp: Sie können das freie Programm „dia“¹ verwenden, um die Klassen auf dem Rechner zu zeichnen. Sie können aber auch per Hand zeichnen.

¹ [https://en.wikipedia.org/wiki/Dia_\(software\)](https://en.wikipedia.org/wiki/Dia_(software))