

## Aufgabenblatt 1

Stand: 11.10.2024

Bearbeiten Sie alle Aufgaben aller Übungsblätter – soweit nicht anders angegeben – in festen **Zweiergruppen**. Alle Gruppenmitglieder müssen die Aufgabe gemeinsam bearbeiten und die Lösung allein erläutern können.

### Aufgabe 1.1 - Abstrakte Datentypen

In dieser Aufgabe machen Sie sich mit dem Modellierungswerkzeug [USE](#) (A UML-based Specification Environment) vertraut und modellieren einige gängige ADTs. Eine Einführung in USE findet sich bei [YouTube](#) (englisch) und im Moodle-Raum der Veranstaltung bei dieser Aufgabe.

#### Aufgabe 1.1.1 – Einarbeiten in USE

Machen Sie sich mit USE vertraut, indem sie sich die Beispiele aus dem Ordner „examples“ ansehen und damit „spielen“.

Im Anschluss öffnen Sie die Datei `List.use` die in Moodle als Datei zum Übungsblatt 1 abgelegt ist.

Lassen Sie sich das Klassendiagramm anzeigen:



Ordnen Sie die Klassen so an, dass Sie einen guten Überblick über das geladene Modell bekommen. Blenden Sie auch die Operationen ein (Kontextmenü im Klassendiagramm: *Show operations*).

Führen Sie die Datei `LinkedList.soil` aus dem Teams-Ordner aus, indem Sie den folgenden Befehl in der Shell eingeben:

```
use> open LinkedList.soil
```

(ggf. müssen Sie den Pfad zur Datei angeben)

Öffnen Sie das Objektdiagramm zum aktuellen Systemzustand (ebenfalls über die Symbolleiste) und ordnen Sie die Objektknoten so an, dass die verkettete Liste erkennbar wird.

Nun öffnen Sie den *Class Invariant View* (über den Blitz in der Symbolleiste). Die definierte Invariante sollte erfüllt sein.

Erzeugen Sie ein neues `ListItem`, indem Sie im Menü State den Eintrag *Create Object...* wählen und im geöffneten Dialog ein Objekt mit dem Namen `ListItem3` vom Typ `ListItem` erzeugen.

Prüfen Sie, ob die Invariante weiterhin erfüllt ist. Erläutern Sie, warum sie nicht erfüllt ist. Überlegen Sie sich einen geeigneteren Namen als „mustBeTrue“ und ändern Sie diesen im USE-Modell.

Passen Sie den Systemzustand so an, dass die Invariante wieder erfüllt ist. Tipp: Sie können Verbindungen (Links) zwischen Objekten erzeugen, indem Sie die zu verbindenden Objekte markieren (Shift gedrückt halten). Alternativ gibt es den Befehl `!insert ...` auf der Shell.

Führen Sie das Skript `ArrayList.soil` aus und schauen Sie sich das Sequenzdiagramm und die Ausgabe auf der Shell an. Was ist hier schiefgelaufen?

### **Aufgabe 1.1.1 – Spezifikation eines ADT**

Das vorher analysierte Listenmodell und vor allem der abstrakte Datentyp `List` haben noch Schwächen. Erste Hinweise auf diese Schwächen liefert die ebenfalls in Moodle zur Verfügung gestellte Testsuite `List.testsuite` mit dieser Testsuite werden einige Annahmen geprüft. Schauen Sie sich die Definition der Testsuite an und führen Sie diese aus, indem Sie den Befehl `open List.testsuite` auf der Shell ausführen.

Erweitern Sie den abstrakten Datentypen `List` um geeignete Constraints, so dass die Testsuite erfolgreich ausgeführt werden kann.

Sichern Sie die Operationen `insertAt` und `remove` mit geeigneten Constraints ab. Testen Sie diese, indem Sie die Testsuite erweitern.

**Abgabe:** Erstellen Sie für sich eine Dokumentation über Ihre Tätigkeiten und Erkenntnisse. Im Praktikum gehen wir dann gemeinsam Ihre Lösung durch. Ihre Dokumentation geben Sie dann in Moodle ab.