

**Lernziel:** Felder, die Interfaces/Klassen List, ArrayList, Map/HashMap sowie die Hüllklassen für einfache Typen (wrapper classes) problemgerecht beim Schreiben von Java-Programmen einsetzen können. Verwendung von Singleton-Klassen, SWING, MVC-Konzept und Observer-/Observable-Strategien

## 1. Aufgabe

Modifizieren Sie die 1. Aufgabe der Übung 6. Erstellen Sie eine Klasse Artikel mit Artikelnummer, eine Bezeichnung, ein Preis und eine Menge. Die Artikelnummer wird fortlaufend gezählt und über die Klasse erzeugt.

Die Artikel sollen nun beim Erfassen in ein `DefaultListModel` gespeichert und über eine `JList` dargestellt werden. `JList` und `DefaultListModel` sollen „generisch“ für den Typ Artikel parametrisiert werden. Die GUI soll nun neben den Eingabefeldern für die Artikelerfassung eine `JList` zur Darstellung der Artikel enthalten sowie drei Buttons mit folgenden Funktionen:

- **Speichern:** Liegen keine Fehlereingabefehler wird ein neuer Artikel erstellt und in das `DefaultListModel` eingefügt. Die Eingabefelder werden zurückgesetzt. Ansonsten erfolgen Fehlermeldungen.
- **Update:** der aktuell selektierte Artikel in der `JList` soll geändert werden. Voraussetzung dafür ist, dass die Daten des aktuell selektierten Artikels in die Eingabefelder übertragen werden, damit Sie dort geändert werden können. Artikelnummer und Position in der `JList` müssen erhalten bleiben. Der Button soll nur dann `enabled` sein, wenn ein Artikel selektiert ist.
- **Löschen:** der aktuell selektierte Artikel in der `JList` soll gelöscht werden. Der Button soll nur dann `enabled` sein, wenn ein Artikel selektiert ist.

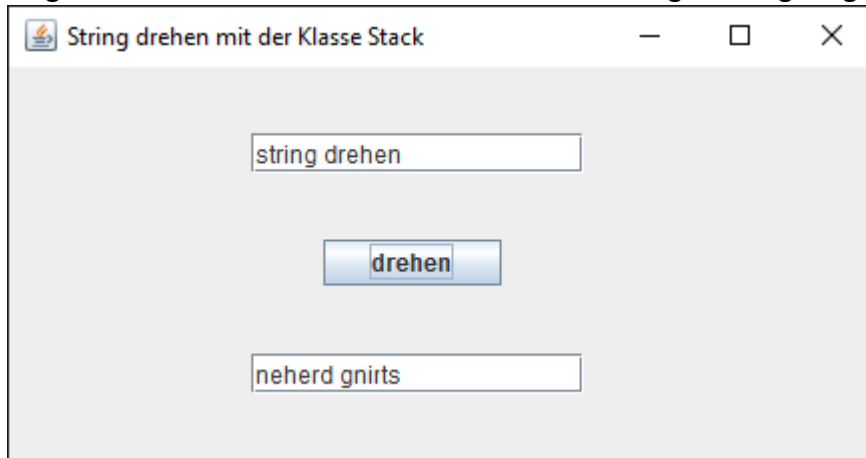
Hinweis: für eine geeignete Darstellung in der `JList` müssen Sie die `toString`-Methode von Artikel überschreiben. Nach „Update“ und „Löschen“ müssen die Eingabefelder nun ebenfalls zurückgesetzt werden.

### Prinzip-Skizze:

Die Skizze zeigt ein GUI-Fenster mit dem Titel 'Artikeldatenerfassung'. Es enthält drei Eingabefelder für 'Bezeichnung:', 'Preis:' und 'Menge:'. Darunter befinden sich drei Buttons: 'speichern', 'update' und 'löschen'. Rechts neben den Eingabefeldern ist eine große Liste mit der Überschrift 'Artikelliste:' dargestellt.

## 2. Aufgabe

Zeigen Sie wie mit Hilfe der Klasse *Stack* ein beliebiger String umgedreht werden kann.



## 3. Aufgabe

Modifizieren Sie die 1. Aufgabe. Ersetzen Sie das `DefaultListModel` durch eine Unterklasse von `AbstractListModel`. Verwenden Sie innerhalb der neuen Datenklasse eine `ArrayList` zu Verwaltung der Artikel. Ergänzen Sie Ihre Unterklasse um Methoden zum Hinzufügen, Ändern und löschen von Artikeln.

## 4. Aufgabe

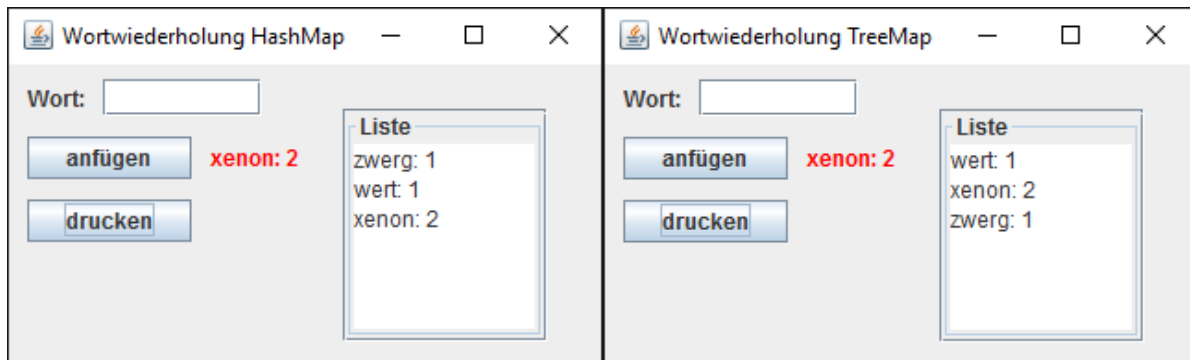
Entwickeln Sie ein GUI-System zur Erfassen, Verwalten und Zählen von Wörtern. Die GUI besteht aus einem Textfeld zur Eingabe eines Wortes. Über ein Label wird angezeigt, wie oft dies Wort schon eingegeben wurde. Zusätzlich gibt es eine TextArea und zwei Buttons. Die Wörter sollen zusammen mit der Anzahl der Eingaben des jeweiligen Wortes in einer `Map` gespeichert werden. Das Wort ist der Key, die Anzahl Eingaben der Value. Die Buttons haben im Detail folgende Funktionen:

**anfügen:** aus dem Textfeld wird ein Wort entnommen. Es wird nachgeschaut ob dieses Wort in der `Map` schon vorhanden ist. Wenn nein, wird es mit der Anzahl 1 in der `Map` gespeichert. Wenn ja, wird die aktuelle Anzahl um 1 erhöht und die neue Anzahl für das Wort gespeichert. Das gerade eingegebene Wort mit der aktuellen Anzahl wird über ein Label angezeigt, das Textfeld zurückgesetzt.

**drucken:** alle Wörter der `Map` werden zusammen mit der Anzahl ihrer Eingaben als Liste in der TextArea dargestellt.

Erstellen Sie zwei Varianten. Initialisieren Sie diese `Map` zuerst als `HashMap`. Ersetzen Sie in der zweiten Variante diese `HashMap` durch eine `TreeMap`. Vergleichen Sie die Listenergebnisse bei der Funktion „drucken“.

**Skizze:**



## 5. Aufgabe

Entwickeln Sie eine Java-Anwendung zur Verwaltung von Studenten. Es sollen Studenten erfasst und in einer Tabelle dargestellt werden. Dazu benötigen Sie die Klassen `Student` und `Studentendaten` (Unterklasse von `AbstractTableModel`). Ein Student hat Matrikelnummer (5-stellig), Name, ECTS (zwischen 0 und 210), Semester (zwischen 1 und 9) - siehe dazu auch das Beispiel im Foliensatz Datenstrukturen. Die Studenten werden in der Klasse `Studentendaten` verwaltet. Die Klasse `Student` soll das Interface `Comparable` implementieren und die Methoden `equals` und `hashCode` überschreiben (entsprechend dem Beispiel im Foliensatz Datenstrukturen – also bzgl. Matrikelnr.).

Erstellen Sie die geeigneten Klassen zusammen mit einer GUI-Klasse zur zentralen Studentenverwaltung. Die Studentenverwaltung soll folgende Funktionen bereitstellen:

- Aufruf eines eigenen Fensters zum Erfassen von Studenten.
- Anzeigen der aktuellen Studentenliste via Tabelle. Selektierte Zeilen/Studenten sollen gelöscht werden können.
- Matrikelnummern können erstmal in beliebiger Reihenfolge erfasst werden. Deswegen soll eine Sortier-Funktion bereitgestellt werden, so dass die Studentenliste bei Bedarf bzgl. Matrikelnummer oder Name neu sortiert werden kann.
- Einfaches editieren der Studentendaten direkt in der Tabelle. Die Matrikelnummer kann nicht geändert werden.
- Man kann sich eine einfache Statistik anzeigen lassen: die durchschnittliche Semesteranzahl aller Studenten; die Anzahl der Studenten die regulär studieren<sup>1</sup>.

Erstellen Sie Ihre Lösungsvariante so, dass es nur ein Studentendaten-Objekt geben kann (*Singleton-Objekt*). Das gleiche soll auch für die Fenster zum Erfassen der Studenten gelten. Hinweise:

- Die Klasse `Studentendaten` muss mehrere abstrakte Methoden implementieren und andere Methoden sinnvoll überschreiben (siehe Foliensatz `SwingKomponenten`, Java-Doku, Google). Sie enthält eine Liste der erfassten Studenten. Sie benötigen zusätzliche Methoden zum Löschen, Anfügen, Sortieren und Statistik der Studenten.
- Die GUI für die Studentenverwaltung verwendet sinnvollerweise ein `Border-Layout` und stellt ihre Funktionen via Menü bereit.
- Bei der Erfassung von neuen Studenten muss überprüft werden, ob die Matrikelnummer schon einmal erfasst wurde. Das Erfassungs-Fenster kann geschlossen und neu geöffnet werden.
- Sortierungen und Statistikwerte können via Streaming bereitgestellt werden.

<sup>1</sup> Semester  $\leq 7$  und ECTS  $\geq$  Semester\*30

Siehe auch die folgenden Prinzip-Skizzen:

