

---

**Programmieren in Java**<http://proglang.informatik.uni-freiburg.de/teaching/java/2017/>

---

**list-operations***Interaktive Operationen auf einer Liste*

Woche 04 Aufgabe 1/4

Herausgabe: 2017-05-15

Abgabe: 2017-05-26

**Achtung:** beachten Sie unbedingt die allgemeinen Hinweise zur Abgabe auf der Homepage.

Project `list-operations`Package `listoperations`

Klassen

Main
<code>public static void main(String[])</code>

Dieses Programm erlaubt das interaktive Manipulieren von Wörterlisten. Es liest auf `stdin` zeilenweise Kommandos ein, die Listenoperationen darstellen. Die Kommandos sollen dann auf einer Liste von `Strings` ausgeführt werden, soweit dies möglich ist.

Die Kommandos sollen zeilenweise eingelesen werden und entsprechen folgendem Format:

- `append w`: Fügt das Wort *w* zum Ende der Liste hinzu.
- \* `pop`: Entfernt das letzte Wort von der Liste.
- \* `set i w`: Überschreibt das Wort an Index *i* mit *w*.
- `remove-first w`: Entfernt das erste Vorkommen von *w* in der Liste.
- `remove-all w`: Entfernt alle Vorkommen von *w* in der Liste.
- `print`: Gibt „:“ auf `stdout` aus, gefolgt von den Wörter der Liste. Jedem Wort soll ein Leerzeichen („ ”) vorangestellt werden.
- `sort`: Sortiert die Liste.
- `reverse`: Dreht die Reihenfolge der Elemente in der Liste um.

Gültige Wörter bestehen aus den Zeichen `a-z`, `A-Z` und `0-9`.

Bei der Eingabe einer ungültigen Zeile, die also keinem der Kommandos entspricht, soll die Fehlermeldung

`INVALID COMMAND`

ausgegeben werden. (Danach sollen weitere Kommandos eingegeben werden können)

Falls ein Kommando auf der Liste nicht ausführbar ist, soll die Fehlermeldung

DOES NOT COMPUTE

ausgegeben werden. Nur Kommandos die oben mit einem Stern (\*) gekennzeichnet sind, können unter Umständen diese Fehlermeldung verursachen.

### Beispieleingabe 01:

```
append Hallo
append Welt
print
set 0 Hi
pop
print
append World
append Welt
append World
append Welt
print
remove-first World
print
remove-all Welt
print
reverse
print
sort
print
```

### Erwartete Ausgabe 01:

```
:: Hallo Welt
:: Hi
:: Hi World Welt World Welt
:: Hi Welt World Welt
:: Hi World
:: World Hi
:: Hi World
```

### Beispieleingabe 02:

```
append Tschau
print
append !
```

### Erwartete Ausgabe 02:

```
:: Tschau  
INVALID COMMAND
```

### Beispieleingabe 03:

```
append ATTACK  
print  
set 2000 BOOM
```

### Erwartete Ausgabe 03:

```
:: ATTACK  
DOES NOT COMPUTE
```

### Hinweise

- Hier gibt es eine Einführung zum Interface `java.util.List`:

<http://docs.oracle.com/javase/tutorial/collections/interfaces/list.html>.

Ansonsten kann auch die API Dokumentation hilfreich sein.

- **Scanner** erlaubt auch Eingaben nach einfachen Mustern mit sog. Regulären Ausdrücken zu durchsuchen. Wie immer sind die entsprechenden Methoden sind in der **Scanner**-API dokumentiert. Hier ist der Java-Syntax für reguläre Ausdrücke beschrieben:

<https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html>

Bei Oracle gibt es auch ein ausführliches Tutorial zu Regulären Ausdrücken in Java.

<https://docs.oracle.com/javase/tutorial/essential/regex/>

Die für diese Aufgabe interessantesten Kapitel sind

- String Literals,
- Character Classes,
- Predefined Character Classes und
- Quantifiers