

Übungen zu Einführung in die Informatik I

Hinweis: Die Aufgabe dieses Übungsblatts beschreibt ein kleines Programmierprojekt, das in den nächsten beiden Übungen bearbeitet werden soll. Hierbei soll ein kleiner einfacher Editor implementiert werden.

In der ersten Übung werden Sie die Grundoperationen des Editors modellieren und mit deren Implementierung beginnen. Bis zur nächsten Übung sollten sie diese Grundoperationen vollständig implementiert haben, so dass Sie anspruchsvollere Operationen, wie etwa den Befehl, der Textoperationen rückgängig macht, einbauen können.

Aufgabe 31 **Zeileneditor**

Entwickeln Sie einen kleinen zeilenbasierten Editor `Ed` in Java.

Funktionen des Editors:

Befehl		Bedeutung
a	show all	gesamtes Dokument anzeigen
2	<i>numpad up</i>	in die folgende Zeile springen
8	<i>numpad down</i>	eine Zeile zurück
4	<i>numpad left</i>	den Cursor ein Zeichen nach links bewegen
6	<i>numpad right</i>	den Cursor ein Zeichen nach rechts bewegen
l n	line	zur Zeile mit Nummer <i>n</i> springen
c n	column	zur Spalte mit Nummer <i>n</i> springen
n l	new line	unter der aktuellen Zeile die Zeile mit Inhalt <i>l</i> einfügen
r	remove line	aktuelle Zeile löschen
i t	insert	den Text <i>t</i> an der aktuellen Cursor-Position einfügen
d	delete	das Zeichen unter dem Cursor löschen
u	undo	macht den letzten Befehl rückgängig – dieser Befehl kann auch wiederholt werden, bis die erste Operation rückgängig gemacht wurde; es können nur solche Befehle rückgängig gemacht werden, die das Dokument ändern (<i>n, r, i, d</i>)
s	store	speichert das Dokument
q	quit	beendet <code>Ed</code>

Beim Aufruf von `Ed` wird die erste Zeile des Dokuments (dieses wird `Ed` als Kommandozeilen-Parameter übergeben) in folgender Weise angezeigt:

```
1: Das ist die erste Zeile meines Dokuments.  
  ^  
>
```

Die Ausgabe einer Zeile beginnt immer mit der Zeilennummer. Das Zeichen `^` gibt die aktuelle Cursor-Position an, die mit den Befehlen 4 und 6 geändert werden kann. Nach `>` wird die Befehlseingabe erwartet. Jeder Befehl wird mit `RETURN` abgeschlossen und dann ausgeführt. Danach wird die aktuelle (evtl. geänderte) Zeile mit der aktuellen Cursor-Position angezeigt.

a) Überlegen Sie sich geeignete Datenstrukturen/Klassen für:

- Zeilen
- Dokumente, zusammengesetzt aus Zeilen
- Befehle (mit Argument)
- ...

Geben Sie das Klassenmodell des Editors an.

- b) **Navigation:** Zunächst soll der Editor soweit implementiert werden, dass ein bestehendes Dokument eingelesen, dargestellt und innerhalb des Dokuments navigiert werden kann. Implementieren Sie die erforderlichen Klassen und Methoden. Der Undo-Befehl bleibt vorläufig unberücksichtigt.
- c) **Einfügen/Löschen:** Nun sollen auch die Methoden implementiert werden, mit deren Hilfe Text bzw. Zeilen eingefügt und gelöscht werden können.
- d) **Undo-Stack:** Erweitern Sie das Modell derart, dass Befehle rückgängig gemacht werden können. D.h. führen Sie einen Befehls-Stack ein, um den Vorgänger-Zustand zu rekonstruieren.
- e) Implementieren Sie das erweiterte Modell.
- f) **Javadoc:** Kommentieren Sie Ihr Programmpaket so, dass man mit javadoc eine Dokumentation erstellen kann, die Struktur und Arbeitsweise Ihres Programms erläutert.