Programmieren in Java

http://proglang.informatik.uni-freiburg.de/teaching/java/2017/

line-numbers

Zeilennummerierung

Woche 03 Aufgabe 1/4

Herausgabe: 2017-05-08
Abgabe: 2017-05-19

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

Project `line-numbers`
Package `linenumbers`
Klassen

`Main`
`public static void main(String[])`

Das Programm soll die Eingabe auf `stdin` zeilenweise einlesen und die Zeilen dann nummeriert auf `stdout` ausgeben. Über Kommandozeilenargumente soll angegeben werden können, welcher Zeilenbereich ausgegeben werden soll.

- Ohne Argumente werden alle Zeilen ausgegeben.
- Mit einem Argument n werden die Zeilen ab Zeile n ausgegeben. Ist n größer als die Anzahl der Zeilen in der Eingabe, soll gar nichts ausgegeben werden.
- Mit zwei Argumenten n und m werden die Zeilen von Zeile n bis Zeile m ausgegeben (soweit vorhanden). Ist $m < n$, oder n größer als die Anzahl der Zeilen in der Eingabe, soll gar nichts ausgegeben werden.

Zeilennummern sind immer positive, ganze Zahlen (also ≥ 1), die in den Datentyp `long` passen. Wird ihr Programm mit unsinnigen Argumenten aufgerufen, geben Sie folgende Fehlermeldung aus (auf `stdout`)):

```
Bad arguments.
Usage: line-numbers [start-number [end-number]]
```

Beispieleingabe 01 (ohne Argumente):

```
diese datei
wird wieder
ausgegeben
```

Erwartete Ausgabe 01:

```
1 diese datei
2 wird wieder
3 ausgegeben
```

Beispieleingabe 02 (Argumente: `"2"`):

```
diese datei
wird wieder
ausgegeben
```

Erwartete Ausgabe 02:

```
2 wird wieder
3 ausgegeben
```

Beispieleingabe 03 (Argumente: `"a" "2"`):

```
eins
zwei
drei
```

Erwartete Ausgabe 03:

```
Bad arguments.
Usage: line-numbers [start-number [end-number]]
```

Beispieleingabe 04 (Argumente: `"2" "2"`):

```
diese datei
wird wieder
ausgegeben
```

Erwartete Ausgabe 04:

```
2 wird wieder
```

Hinweise:

- Die Kommandozeilenargumente sind im Parameter der `main` Methode als `String` Array gespeichert (typischerweise nennen wir den Parameter daher `args`).
- In IntelliJ lassen sich folgendermaßen Kommandozeilenargumente angeben:
  - Den „Run Configurations" öffnen mit `Run -> Edit Configurations ...`
  - Unter „Program arguments" die gewünschten Argumente eingeben. Mehrere Argumente werden durch Leerzeichen getrennt.

  Weitere Informationen: `https://docs.oracle.com/javase/tutorial/essential/environment/cmdLineArgs.html`
- In den Test-Dateien sind die Argumente im Dateinamen angegeben, durch Punkte („.") getrennt. Beispiele:
  - Der Testinput `line-numbers-02.4.stdin` übergibt das Argument `4`.
  - Der Testinput `line-numbers-03.2.a.stdin` übergitb die Argumente `2` und `a`.
- Wenn `arr` ein Array ist, dann gibt der Ausdruck `arr.length` die Länge von `arr` zurück. (Beachten Sie die fehlenden Klammern) Weiter Informationen zu Java-Arrays gibt es hier: `http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html`
- Die nützliche `Scanner` Klasse lässt sich auch mit `String`s instanziieren: `https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html#Scanner(java.lang.String)`