CSV-File Viewer

Verantwortlichkeiten, Klassenstruktur, Schnittstellen, Testing

# Verantwortlichkeiten

* Einlesen eines CSV-Files
* Anzeigen der Daten
* Navigieren innerhalb der Daten
* CLI-Kommandos zu Verfügung stellen und verarbeiten

# Klassen

Für jede Verantwortlichkeit eine Klasse (single responsibility)

|  |  |
| --- | --- |
|  | Einlesen eines CSV-File  Methoden für Rückgabe des Headers und der Records. |
|  | Die Übergebenen Daten werden mittels der Methode show() auf den entsprechenden OutputStream geschrieben. |
|  | Innerhalb der übergebenen Liste kann mittels entsprechender Methoden navigiert werden.  *Alternative:* Nur eine Methode pageForCommand(Command) die eine entsprechende Liste zurück liefert. |
|  | Gibt die Kommandos (FirstPage, LastPage etc.) auf den OutputStream aus und empfängt die Eingaben mittels InputStream. |

## Hilfsklassen

|  |  |
| --- | --- |
|  | Main-Klasse mittels der das Programm gestartet werden kann. Orchestriert die Hauptklassen (siehe oben) und stellt den gewünschten Ablauf sicher:   * Delegiert das Einlesen des CSV-Files * Delegiert das Anzeigen der Daten * Reagiert auf die Kommandos |
|  | Auswahl aller Kommandos |

# Testing

Da die Hauptklassen voneinander unabhängig sind, können sie mittels JUnit-Test's unabhängig voneinander getestet werden.

## Testklassen

* ReaderTest
* ViewerTest
* PagingTest
* CommandLineInterfaceTest