	Elementare Programmstrukturen (Java) Datenverarbeitung mit Dateien	
Klasse:	Name:	Datum:

Schulklassenverwaltung (☺☺): Datenstruktur Felder

Schreiben Sie ein Programm, das aus einer txt-Datei mit Schul- und Schülerdaten die Daten einliest und gemäß den unten angeführten Anforderungen verarbeitet.



Die Daten sind in den txt-Dateien immer folgendermaßen organisiert (Beispiel):

In der ersten Zeile sind durch Semikolon getrennt folgende Informationen enthalten:

- Schulname
- Bezeichnung der Klasse (Klassenname)
- Schuljahr der Einschulung

Jede Zeile, außer der letzten Zeile in der Datei, schließt mit einem Zeichen CRLF (Carriage Return und Line Feed) (ASCII-Code = 10).


Nach der nächsten und in den darauffolgenden Zeilen kommen die Schülerdaten. Pro Schüler sind, jeweils durch Semikolon getrennt, folgende Informationen pro Zeile enthalten:

- Nachname
- Vorname
- Geburtsdatum in der Syntax: tt.mm.jjjj

Für das Einlesen und ggf. für das Schreiben aus bzw. in Dateien sollen Sie Methoden der vorgegebenen Klasse `FileReaderWriter` benutzen. Sie sollen in der Klasse `FileReaderWriter` keine Änderungen vornehmen.

Das Klassendiagramm zur gesamten Aufgabe sieht so aus:



	Elementare Programmstrukturen (Java) Datenverarbeitung mit Dateien	
Klasse:	Name:	Datum:

Anforderungen (Programmieren Sie die Anforderungen in der Reihenfolge)

Entpacken Sie das auf der Lernplattform eingestellte ZIP-File *Schulklassenverwaltung.zip* in ein Verzeichnis (Vorschlag; DO = Datenorganisation):

\\..\DO\Felder\Schulklassenverwaltung\...

Im ZIP-File ist neben den Klassen *Schulklassenverwaltung* und *FileReaderWriter* auch eine txt - Datei mit Schulklassendaten vorhanden. Weitere txt - Datei mit Schulklassendaten finden Sie für Tests auf der Lernplattform.

Das vorgegebene Programm liest bereits die Daten aus der txt-Datei in eine Zeichenkette ein, und gibt diese zur Kontrolle auf vollständige Übernahme der Daten aus der txt-Datei einfach nur auf der Konsole aus. Die Ausgabe der Zeichenkette können Sie danach aus- und bei Bedarf wieder einkommentieren.

Verarbeiten Sie dann die eingelesene Zeichenkette nun über die nachfolgenden Aufgaben:

Aufgabe 1:

Übernehmen Sie aus der ersten Zeile die 3 darin enthaltenen Informationen in 3 getrennten Variablen, und geben Sie den Inhalt der 3 Variablen nach der Übernahme benutzerfreundlich auf der Konsole aus:

- String *schulname*
- String *klassenBezeichnung*
- String *schuljahr*

Tipp:

Benutzen Sie die Kenntnis, dass die 3 Informationen durch 2 Semikolon getrennt sind. Nach der letzten Information folgt ein Steuerzeichen CRLF (ASCII-Code = 10). Gehen Sie mit einer for-Schleife durch den Anfang der eingelesenen Zeichenkette:

- o Fangen Sie damit an, dass Sie zuerst den String *schulname* füllen, und zwar solange, bis Sie auf das Zeichen Semikolon treffen.
- o Danach füllen Sie nach dem ersten Semikolon den String *klassenBezeichnung*, und zwar solange, bis Sie auf das Zeichen Semikolon treffen.
- o Danach füllen Sie nach dem zweiten Semikolon den String *schuljahr*, und zwar solange, bis Sie auf das Zeichen CRLF treffen.

Aufgabe 2: Übernahme in Datenstruktur Feld

Übernehmen Sie die nachfolgenden **Schülerdaten** in ein geeignetes **zweidimensionales Feld** *schuelerdaten*.

Sie dürfen **keine Klassen aus dem API benutzen**, sondern nur den Referenzdatentyp *Feld* (array).

Geben Sie die Daten aus dem Feld dann mit Hilfe von for-Schleifen benutzerfreundlich auf der Konsole aus.

Aufgabe 3: Löschen eines Datensatzes und Speichern

Löschen Sie einen Datensatz (z.B. den Datensatz Nummer 4 (wenn ein solcher vorhanden ist)), und speichern Sie die neuen Daten in eine Datei, die im Datenamen vor „.txt“ den Zusatz „_neu“ hinzufügt.

Aufgabe 4: Durchschnittsalter (= Differenzierung)

Ermitteln Sie das Durchschnittsalter der Schüler in der eingelesenen Schulklasse (z.B. 17,5 Jahre) in Bezug auf das aktuelle Systemdatum ihres IT-Systems, und geben Sie anschließend das Durchschnittsalter der Schüler benutzerfreundlich auf der Konsole aus.