

Textfiles einlesen & strukturieren

Lernziele:

- Sie können in einem Programm aus einem File Daten lesen
- Strukturierte Textzeilen zerlegen und diese in geeignete Datenobjekte abfüllen
- Ausgaben in Textfiles durch Formatierung strukturieren

3 Strukturierte Textfiles

Oft werden Daten in strukturierten Text Files abgelegt. So können zum Beispiel Excel-Tabellen im CSV-Format gespeichert werden, so dass jede Zeile aus Excel eine Zeile im Text File ergibt. Innerhalb der Zeilen sind die Spalten durch ein spezielles Trennzeichen getrennt (CSV steht für Comma Separated Values, wobei das Komma eher selten als Trennzeichen verwendet wird).

Ein strukturiertes File mit Personendaten könnte also so aussehen:

```
Huber;Peter;1955
Meier;Irene;1960
Muster;Anna;1974
```

Um nun, nachdem eine Zeile gelesen wurde, auf die einzelnen Felder zugreifen zu können, muss der String, welcher eine Zeile enthält, zerlegt werden. Das ist sehr ähnlich wie Sie es schon im Modul 404 gesehen haben:

```
String line = ... // contains one line which must be processed
String[] lineParts = line.split(";"); // Split line at occurrences of semicolon

for ( String p: lineParts ) { // Loop over all parts of the line
    // Do something with this part (Variable p)
}
```

Nummerische Felder

Wenn ein Feld eine Zahl enthält (im Beispiel das Geburtsjahr), dann wird das natürlich zunächst einmal als String eingelesen, wie alle anderen Felder auch. Solange wir damit nicht rechnen wollen, spielt das auch keine Rolle.

Meistens ist aber nötig, diese Felder in richtige Zahlen umzuwandeln. Dazu können die beiden folgenden Aufrufe verwendet werden:

```
String intStr;    // contains the String representation of an int

String doubleStr; // contains the String representation of a double
int intVal = Integer.parseInt(intStr); // Converts a String to an int value
double doubleVal = Double.parseDouble(doubleStr); // Converts a String to a double
```

Übung Teil 1

Erstellen Sie so ein Textfile mit folgender Struktur und verwenden Sie das dann:

```
Huber;Peter;1955
Meier;Irene;1960
Muster;Anna;1974
```

Erstellen Sie ein Programm, welches dieses File liest und Zeile für Zeile wieder auf der Konsole ausgibt, ergänzt um das Alter der Person, bezogen auf das aktuelle Jahr welches fest einprogrammiert sein darf.

Übung Teil 2

Erweitern Sie das Programm so, dass es den Benutzer nach einem Vornamen fragt und dann von allen Personen mit diesem Vornamen den ganzen Namen und das Alter auf der Konsole ausgibt.

Beachten Sie dabei, dass zum Vergleich von zwei String-Variablen nicht der Operator `==` verwendet werden kann. Stattdessen muss die Methode `equals` der Klasse `String` verwendet werden.

Quelle: BBW Unterlagen, St.Dütsch. Aktualisiert August 2020.