

Externe Daten

Aufgabe: Ein Array soll mit Werten aus einer Datei gefüllt und dann angezeigt werden.

Aufgabe externe Datei:

Lade die Datei `punkte.csv` mit einem Texteditor (nicht mit einer Tabellenkalkulation!):

1. Die Datei hat auf jeden Fall zwei Spalten mit den Namen "Punkte" und "Name". In der Überschriftszeile sind die Texte in Anführungszeichen gesetzt. In den übrigen Zeilen nicht. Es fällt auf, dass jede Zeile die gleichen Informationen enthält (eben Name und die Punktzahl). Die Anzahl der Zeilen hängt davon ab, wie oft ihr gespielt habt. Auf jeden Fall darf man die Zeile mit der Überschrift nicht mitzählen, da sie keinen Datensatz enthält.

Aufgabe Implementation:

Lade die Datei `balkendiagramm_csv_roh`:

```
// Liste mit allen Werten
int zahlen[];
String namen[];

...

// Initialisiere Arrays, in die alle Zeilen der Tabelle passen
zahlen = new int[csv.getRowCount()];
namen = new String[csv.getRowCount()];

// Fülle die Arrays mit Werten aus der Tabelle
for (int i = 0; i < zahlen.length; i++) {
    zahlen[i] = csv.getInt(i, "Punkte");
    namen[i] = csv.getString(i, "Name");
}

...

// Beschriftung
fill(#FFFFFF);
text(namen[i], 2, 35+i*15);
text(zahlen[i], 70, 35+i*15);
```



Testen der Implementation:

Mögliche Fehler:

- Datei existiert nicht.
- Datei ist gar keine CSV-Datei.
- Datei enthält keine Überschriften.
- Datei enthält andere Daten.
- Es gibt keine Spalte "Punkte" oder "Name".
- Die Werte in der Spalte Punkte sind keine Zahlen.
- Die Werte in der Spalte Punkte sind negativ oder zu groß, um sie als Balken anzuzeigen.
- In einer Zeile fehlt der Eintrag für Punkte oder für den Namen.

Alle Fehler abzufangen ist nur mit sehr guten Programmierkenntnissen möglich:

- Datei existiert nicht => die Variable csv hat den Wert null. Das kann getestet werden:

```
if (csv != null) { // Laden der Datei fortsetzen }
```
- In einer Zeile fehlt der Eintrag für Punkte => Man muss gar nichts machen, da automatisch der Wert 0 verwendet wird.
- Datei enthält keine Überschriften => das ist schwierig abzufangen. Lässt man den Parameter "header" weg, dann arbeitet die Tabelle ohne Überschriften. Man muss dann auch bei getInt und getString mit Spaltennummern statt Namen arbeiten.
- Datei enthält andere Daten => kaum abzufangen, man könnte höchstens kontrollieren, ob der erste Eintrag wie ein Text und der zweite wie eine Zahl aussieht.
- Werte zu groß oder klein => man könnte z.B. mit

```
if( punkte[i] < 0) { punkte[i] = 0; }
```

die untere Grenze 0 überprüfen und die Werte ggf. korrigieren.

Expertenaufgabe:

```
public void ladeTabelle(String name) {
    // Tabelle aus XML-Datei laden
    XML xml = loadXML(name);
    XML[] rows = xml.getChildren("Row");

    // Initialisiere Arrays, in die alle Zeilen passen
    zahlen = new int[rows.length];
    namen = new String[rows.length];

    // Fülle die Arrays mit Werten aus der Tabelle
    for (int i = 0; i < zahlen.length; i++) {
        // Lies Werte aus der i. Zeile, Unterpunkte "Punkte" bzw. "Name"
        XML eintragPunkte = rows[i].getChild("Punkte");
        XML eintragName = rows[i].getChild("Name");

        zahlen[i] = eintragPunkte.getIntContent();
        namen[i] = eintragName.getContent();
    }
}
```