Externe Daten

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

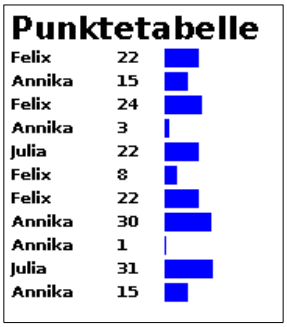

Bild: Screenshot von Ausführung des Programms „balkendiagramm_csv“ (Schaller)

Externe Dateien: CSV oder XML?

1. *Schritt: Geeigneten Dateityp auswählen*
   CSV - Daten in Form einer Tabelle, in jeder Zeile hat man gleichartige Daten
   - Werte durch Komma(,) oder Strichpunkt (;) getrennt (comma separated values)
   - Texte z.T. in Anführungszeichen (")
   - können eine Zeile mit Überschrift enthalten
   XML - Daten in Form eines Baums (extended markup language)
   - geeignet, wenn nicht für jeden Eintrag die gleichen Daten vorliegen.
2. *Schritt: Datei laden/speichern*
   CSV - Verwende die Klassen Table.
   XML - Verwende die Klasse XML.
3. *Schritt: Übertrage die Werte in ein Array.*
   In der Regel werden in einer `for`-Schleife alle Werte des Arrays einzeln belegt.
   z.B.

```
for(int i=0; i < kommazahlen.length; i++) {
    // i => i. Zeile, 0 => 1. Spalte
  kommazahlen[i] = csv.getFloat(i,0);
}
```

Aufgabe externe Datei:

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

1. *Untersuche, wie viele Zeilen und wie viele Spalten sie hat. Ist es eine csv-Datei mit Überschriften? Welches Trennzeichen wurde zwischen den Werten verwendet? Wurden alle Texte in Anführungszeichen eingeschlossen?*
   Ergänze die Datei um zwei eigene Einträge (so kannst du cheaten...).

Datentypen

Nachdem man die Struktur der Daten kennt, kann man sich überlegen, welche Datentypen dazu passen:

Art der Daten	Datentyp	Befehl zum Auslesen aus CSV-Datei
Ganze Zahlen	int	csv.getInt(i,"Punkte");
Dezimalzahlen	double oder float	csv.getFloat(i, "Messwert");
Texte	String	csv.getString(i, "Name");

Punktschreibweise

Verwendet man vorgegebene Bibliotheken, muss man oft zunächst ein sogenanntes "Objekt" einer "Klasse" erschaffen (Befehl new):

```
z.B. Table csv =  new Table(name,"header",',','"');
```

Hier wird das Objekt `csv` vom Typ `Table` erschaffen. Dieses Objekt hat dann bestimmte Fähigkeiten. Man kann sie mit `objektname.faehigkeit()` (Punktschreibweise) nutzen:

```
z.B. csv.getInt(i, 2);
```

Welche Fähigkeiten das Objekt hat, musst du der Dokumentation entnehmen.

Aufgabe Implementation:

Lade die Datei alg12_balkendiagramm_csv:

2. *Deklariere ein Array `zahlen` für die Punkte (welcher Typ ist geeignet?). Deklariere ein Array `namen` für die Spielernamen (welcher Typ ist geeignet?).*
3. *Implementiere die Methode `ladeTabelle(String name)`. Initialisiere dazu dort die Arrays in der richtigen Größe (siehe Arbeitsblatt Zufallszahlen) und fülle sie dann mit den Werten aus der Tabelle (siehe Schritt 3).*
4. *Implementiere die Methode `zeichenBalken()`, so dass das Bild wie oben dargestellt aussieht. Die Namen der Spieler musst du dazu aus dem Array `namen` herauslesen.*

Testen der Implementation:

Da hier Daten aus einer externen Datei kommen, können hier viele Fehler passieren. Zum Beispiel könnte die Datei gar nicht existieren oder keine Überschriften enthalten.

5. *Überlege dir noch zwei Fehler, die die Daten haben könnten.*
6. *Kopiere die Datei punkte.csv, passe das Programm so an, dass deine Kopie geladen wird, und experimentiere, was bei den Fehlern passiert.*
7. *Lies die Dokumentation zum Befehl, der zum Laden der Datei verwendet wird, und versuche das Programm so anzupassen, dass es mit einer Datei ohne Überschriften zurecht kommt.*
8. *Versuche möglichst viele Fehler abzufangen (z.B. das Fehlen der Datei).*

Expertenaufgabe:

9. *Kopiere dein Programm in ein Programm alg12_balkendiagramm_xml. Lass dir die Datei punkte.xml geben. Untersuche die XML-Datei und versuche dein Programm mit Hilfe der Dokumentation der Klasse XML so anzupassen, dass es in der Lage ist, die XML-Datei zu lesen.*