

1. Collection API

In der CSV-Datei `bezirke_noe.csv` stehen Daten zu den politischen Bezirken Niederösterreichs zeilenweise wie folgt zur Verfügung:
Kennzeichen;Bezirksname;Einwohnerzahl mit 1.1.2013.
Trennzeichen ist der Strichpunkt.

```
...
303;Waidhofen an der Ybbs (Stadt);11.425
304;Wiener Neustadt (Stadt);41.701
305;Amstetten;112.528
306;Baden;139.496
...
```

- Erstelle eine Klasse `PolBezirk`, in der die zu einem politischen Bezirk verfügbaren Daten gespeichert werden können. Außerdem soll die Klasse eine natürliche Sortierreihenfolge besitzen, bei der alphabetisch aufsteigend nach dem Bezirksnamen sortiert wird.
- Erstelle eine Konsolenanwendung mit folgender Methode:

```
public static Collection<PolBezirk> readCSV(String filename)
                                   throws IOException
```

Diese Methode liest alle Zeilen aus der Datei `bezirke_noe.csv`, verwandelt jede Zeile in ein Objekt vom Typ `PolBezirk` und erzeugt eine `Collection`, in der die Daten nach ihrer natürlichen Sortierreihenfolge gespeichert sind. Die Methode retourniert diese `Collection`.

Gib die Daten anschließend zur Kontrolle auf der Konsole aus. Sortiere nun die Objekte absteigend nach der Einwohnerzahl und gib die Daten wieder auf der Konsole aus.

Hinweis: Die CSV-Datei ist mit Hilfe eines `BufferedReader` zu lesen.

2. Datenströme

In einer Datei `punkte.dat` stehen in binärer Form immer ein Integer-Wert, gefolgt von zwei Double-Werten (int - Quadrant eines Punktes, double - die beiden Koordinaten des Punktes). Dabei sind die Daten korrupt, d.h. der Quadrant passt in der Regel nicht zu den Koordinaten.

- Schreibe eine Klasse `Tools` eine statische Methode, welche die Datei korrigiert, also alle Interwerte auf die richtige Quadrantennummer korrigiert. Die korrigierte Datei soll den Namen `punkte_korr.dat` besitzen.
- Schreibe in der Klasse `Tools` eine weitere Funktion, der ein Dateiname übergeben wird und die aus dieser die Daten der Punkte eines bestimmten Quadranten (dieser wird als int-Wert der Funktion übergeben) ausliest, diese in eine typsichere Liste von Punkten speichert und diese Liste zurückliefert. Implementiere zu diesem Zweck auch eine entsprechende Klasse `Point` (dies kapselt nur die beiden Koordinaten, damit sie keine redundanten Informationen enthält).

- c) Entwickle eine Java-FX App, Hier kann der Benutzer die zu verwendende Datei und den Quadranten (1,2,3,4) auswählen. Nach einem Buttonclick werden alle aus dem eingestellten Quadranten gelesenen Punkte in einem Koordinatensystem visualisiert. Teste die App sowohl mit der gegebenen als auch mit der korrigierten Datei.

Hinweis:

Für die Lösung von Teilaufgabe c) vergleiche: Java FX Scatterplot

3. Java FX

Entwickeln Sie eine grafische Applikation (einfacher Integerrechner) mit folgender (unten dargestellten) Benutzeroberfläche und verwenden Sie zur Implementierung der Logik (möglichst) das Binding-API: Über Slider und Choice können 2 Integerwerte im Bereich von 0 - 10 eingegeben werden, die RadioButton bestimmen die Rechenoperation. Das Ergebnis (die Rechnung) wird auf dem TextField dargestellt. Bei jeder Änderung einer Benutzereingabe erfolgt sogleich die entsprechende Reaktion im Textfeld.

The image shows two screenshots of a Java FX application window titled "Binding".

The top screenshot shows the application in its initial state. At the top, there is a horizontal slider ranging from 0 to 10, with a blue circle indicating the value 1. Below the slider is a choice box (dropdown menu) displaying the number 1. Underneath the choice box are four radio buttons arranged in a 2x2 grid: "Addition" (selected), "Subtraktion", "Multiplikation", and "Division". At the bottom, a text field displays the result of the calculation: "1 + 1 = 2".

The bottom screenshot shows the application after several changes. The slider now has a grey circle indicating the value 4. The choice box now displays the number 0. The "Division" radio button is now selected. The text field at the bottom displays an error message: "Fehler - Division durch 0 nicht möglich!" (Error - Division by 0 not possible!).

4. Finde drei englische aus lauter verschiedenen Buchstaben bestehende Worte der Länge 9, wobei jede Ziffer für einen Buchstaben steht:

1 2 3 4 5 6 7 8 9

2 3 1 4 5 6 7 8 9

8 9 3 1 2 4 5 6 7

Die Lösung ist unter Verwendung der Wortdatei words.txt (über 500000 englische Worte) eindeutig und lautet:

CAUTIONED : AUCTIONED : EDUCATION