

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
```

- a) 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.
- b) Erstelle eine Konsolenanwendung mit folgender Methode:

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.

- a) 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.
- b) 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).

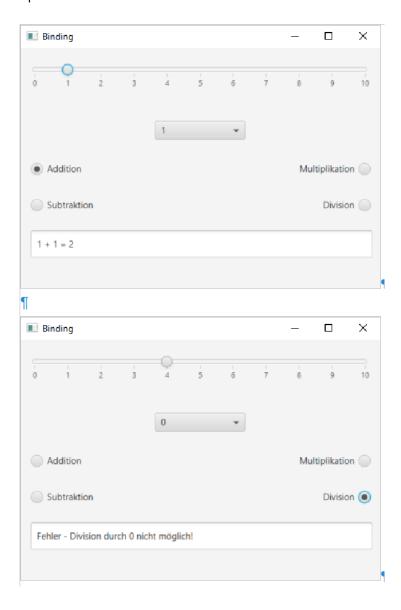
c) Entwickle eine Java-FX App, Hier kann der Benutzer die zu verwendende Datei und den Quadranten (1,2,3,4) auswählen. Nach einem Buttonclick werden alle aus dem eingestellten Quadranten gelesenen Punkte in einem Koordinatensystem visualisiert. Teste die App sowohl mit der gegebenen als auch mit der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



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