

# Computational Geometry

## 2.Aufgabe

### Patrick Burger, Michael Wimmer

Lesen Sie die SVG-Datei 'DeutschlandMitStaedten.svg' und ermitteln Sie die Flächen der einzelnen Bundesländer (bezüglich der in der Datei verwendeten Skala). Am Ende der Datei befinden sich Koordinaten von Städten, Versuchen Sie herauszufinden, in welchem Bundesland diese jeweils liegen.

#### Auslesen der Daten aus der SVG-Datei:

Zum Auslesen der Information wurden die Standard Java Funktionen für das Handeln von XML-Dateien verwendet.

#### Pfade auftrennen, Eckpunkte und Kanten extrahieren:

Die Pfade zur Beschreibung der Umrisse der Bundesländer befinden sich unter dem <g>-Tag aufgeteilt in einzelnen <path>-Tags. Die Koordinaten der Ankerpunkte der Pfade sind teilweise in absoluten jedoch meistens in relativen Koordinaten beschrieben.

Die Trennung der Koordinaten erfolgt durch zeilenweises separieren des Pfades. Einzelne Regionen eines Bundeslandes sind im Pfad über das Prefix M (= Startpunkt) und den Endpunkt L (= Endpunkt) gefolgt von einer Zeile mit dem Zeichen z gekennzeichnet.

Im Constructor der FederalState-Klasse wird der komplette Pfad anhand der Kennzeichnungen getrennt und an einzelne Regions-Klassen per constructor übergeben.

Die Regions-Klasse trennt wiederum den Teilpfad auf anhand der Prefixe und speichert daraus den Startpunkt (M), die Kanten (l, L, H) sowie errechnete absolute Vertices ab.

#### Volumen berechnen:

Durch diese Vorbereitungen ist es relativ leicht das Volumen einzelner Regionen zu berechnen. Über die Gauss'sche Summenformel aus der Vorlesung wird der Flächeninhalt der einzelnen Regionen berechnet:

$$A = \text{Summe}[(y_i + y_{i+1})/2 * (x_i - x_{i+1})] \text{ for } i=1 \dots n$$

Die Größe des Bundeslandes wird über die Addition aller Regionsvolumen berechnet. Hierbei ist zu beachten, dass Regionen mit negativen Wert existieren, da es Bundesländer gibt, die im Inneren ein weiteres Bundesland umschließen. Kanten sind counterclockwise orientiert wodurch das Polygoninnere links von einer Kante liegt. Negative Regionen sind clockwise orientiert wodurch man feststellen kann eine Region zur Fläche dazugehört oder nicht.

Beim Vergleich der Größen der berechneten Bundesländer über einen ungefähren Vergrößerungsfaktor von 1,6 (berechnet über ‚Größe Bayern in der Realität‘/ ‚Größe Bayern aus dem SVG‘) ist mir allerdings aufgefallen, dass Mecklenburg-Vorpommern nicht an derselben Stelle steht wie im realen Ranking. Dies liegt daran, dass hier für einige Regionen negative Volumen herauskommen. Ändert man die Berechnungsmethode so, dass alle Volumen mit Ihrem absolut Wert aufgerechnet werden, stimmt das Ranking mit der Realität überein. Dies liegt aber daran liegen, das diese Änderung nur auf wenige Länder Einfluss hat.

#### negativ oder positive Volumen aufaddiert

Bundesland	Realität	SVG-Pfad
Bayern	70.549,19	69.630,39
Niedersachsen	47.618,24	47.134,89
Baden Württemberg	35.751,65	35.387,04
Nordrhein-Westfalen	34.083,52	33.601,043
Brandenburg	29.477,16	29.320,172
Hessen	21.114,72	20.853,91
Sachsen-Anhalt	20.445,26	20.242,643
Mecklenburg-Vorpommern	23.174,17	19.905,195
Rheinland-Pfalz	19.847,39	19.619,627
Sachsen	18.414,82	18.174,695
Thüringen	16.172,14	15.920,645
Schleswig-Holstein	15.763,18	14.573,183
Saarland	2.568,65	2.528,5354
Berlin	891,75	888,80035
Hamburg	755,16	734,678
Bremen	404,23	224,11633

#### Absolute Volumen der einzelnen Regionen aufaddiert.

Bundesland	Realität	SVG-Pfad
Bayern	70.549,19	69.630,39
Niedersachsen	47.618,24	47.754,312
Baden Württemberg	35.751,65	35.405,85
Nordrhein-Westfalen	34.083,52	33.601,043
Brandenburg	29.477,16	31.097,936
Mecklenburg-Vorpommern	23.174,17	2.2804,2
Hessen	21.114,72	20.853,91
Sachsen-Anhalt	20.445,26	20.242,643
Rheinland-Pfalz	19.847,39	19.619,627
Sachsen	18.414,82	18.174,695
Thüringen	16.172,14	1.5920,645
Schleswig-Holstein	15.763,18	1.5609,461
Saarland	2.568,65	2.528,5354
Berlin	891,75	888,80035
Hamburg	755,16	734.678
Bremen	404,23	395,47092

Nach einigen Test kamen wir leider zu keinem besseren Ergebnis.

### **In welchem Bundesland befindet sich die Stadt:**

Die Städte mit ihren Koordinaten wurden anhand des Attributs `sodipodi:type=arc` indentifiziert und aus der SVG-Datei extrahiert. Zur Speicherung der Stadt als Objekt wurde die Datenklasse CapitalCity verwendet.

Die Methode pointInRegion vergleicht jede Kante einer Region mit der Strecke zwischen den Zentrumskoordinaten einer Stadt und einem Punkt oberhalb des maximalen x- und y-Wert der Region. Dies erfolgt über den InsectingEdgeChecker aus Aufgabe 1. Dabei wird gezählt wie oft sich die Strecke mit den Kanten schneidet. Ist dieser Wert am Ende ungerade so befindet sich der Punkt innerhalb der Region.

Wird eine Stadt in einem Bundesland doppelt erkannt, so befindet sich die Stadt in anderen umschlossenen Bundesland und wird nicht als Hauptstadt ausgegeben (bspw. Berlin in Brandenburg).

```
München is in region of Bayern
Berlin is in region of Berlin
Stuttgart is in region of Baden__x26__Württemberg
Saarbrücken is in region of Saarland
Wiesbaden is in region of Hessen
Mainz is in region of Rheinland-Pfalz
Düsseldorf is in region of Nordrhein-Westfalen
Bremen is in region of Bremen
Erfurt is in region of Thüringen
Dresden is in region of Sachsen
Magdeburg is in region of Sachsen-Anhalt
Hannover is in region of Niedersachsen
Hamburg is in region of Hamburg
Kiel is in region of Schleswig-Holstein
Schwerin is in region of Mecklenburg-Vorpommern
Potsdam is in region of Brandenburg
```

### **Anmerkungen:**

Das Programm wurde zum einen Teil in Java geschrieben und zum anderen in Kotlin, eine typisiert Programmiersprache die in ByteCode für die JVM übersetzt wird, geschrieben. Das Vorteilhafte dabei ist, dass man auch schon während des Programmierens die Teile von Java in Kotlin und vice versa verwenden kann.