

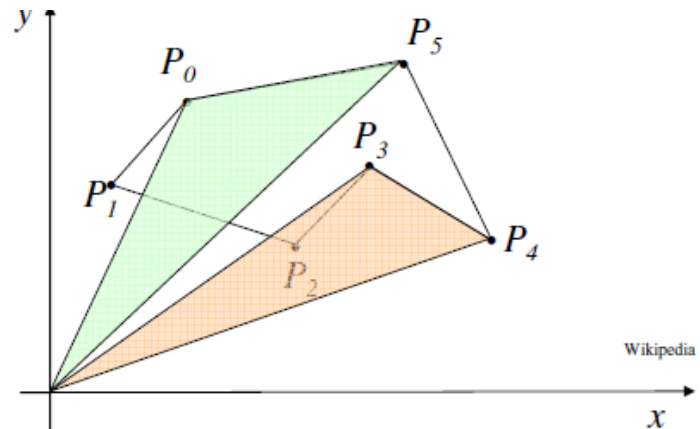
## Aufgabe 2

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

### Gaußsche Dreiecksformel

Im ersten Schritt werden die Flächen der einzelnen Bundesländer berechnet. Hierfür wird das SVG ausgelesen und die einzelnen Polygone erstellt. Anschließend kann mit der Gaußschen Dreiecksformel die Flächenberechnung der einzelnen Polygone vorgenommen werden.

$$\begin{aligned} A &= \sum_{i=1}^n (y_{i+1}x_i - y_ix_{i+1})/2 \\ &= \sum_{i=1}^n (y_ix_{i-1} - y_ix_{i+1})/2 \\ &= \sum_{i=1}^n y_i(x_{i-1} - x_{i+1})/2 \end{aligned}$$



Programmatisch umgesetzt sieht diese Formel wie folgt aus:

```
if (i == vect.size() - 1) {
    area += vect.at(i).getY()
        * ((vect.at(i - 1).getX() - vect.at(0).getX()) / 2);
} else {
    float tmpArea = vect.at(i).getY()
        * ((vect.at(i - 1).getX() - vect.at(i + 1).getX()) / 2);
    area += vect.at(i).getY()
        * ((vect.at(i - 1).getX() - vect.at(i + 1).getX()) / 2);
}
```

Die Ergebnisse werden nun mit jeweils dem gleichen Faktor multipliziert um realitätsnahe Ergebnisse zu erhalten. Damit ergeben sich für die einzelnen Polygone die folgenden Flächen, welche verglichen mit den wirklichen Flächen gut übereinstimmen.

```
resrc/Bayern.txt: 70550.1( Poly: 1) Real-Area: 70550.1 square km Stadt: München
resrc/Saarland.txt: 2710.37( Poly: 1) Real-Area: 2568.75 square km Stadt: Saarbrücken
resrc/badenWuertenberg.txt: 35719.8( Poly: 2) Real-Area: 35751.5 square km Stadt: Stuttgart
resrc/Berlin.txt: 975.092( Poly: 1) Real-Area: 887.7 square km Stadt: Berlin
resrc/Brandenburg.txt: 29488.9( Poly: 2) Real-Area: 29483.1 square km Stadt: Potsdam
resrc/Bremen.txt: -333.361( Poly: 2) Real-Area: 419.24 square km Stadt: Bremen
resrc/Hamburg.txt: 782.568( Poly: 1) Real-Area: 755.16 square km Stadt: Hamburg
resrc/Hessen.txt: 21007.8( Poly: 1) Real-Area: 21114.9 square km Stadt: Wiesbaden
resrc/MeckPommern.txt: 20555( Poly: 7) Real-Area: 23190.8 square km Stadt: Schwerin
resrc/Niedersachsen.txt: 47070.5( Poly: 10) Real-Area: 47612.9 square km Stadt: Hannover
resrc/NordrheinWestfalen.txt: 33654.9( Poly: 1) Real-Area: 34092.3 square km Stadt: Düsseldorf
resrc/ReinlandPfalz.txt: 19574.7( Poly: 1) Real-Area: 19854.1 square km Stadt: Mainz
resrc/Sachsen.txt: 18837.8( Poly: 1) Real-Area: 18419.7 square km Stadt: Dresden
resrc/SachsenAnhalt.txt: 20605.1( Poly: 1) Real-Area: 20449.5 square km Stadt: Magdeburg
resrc/Schleswig-Holstein.txt: 14795( Poly: 11) Real-Area: 15799.3 square km Stadt: Kiel
resrc/Thueringen.txt: 16452( Poly: 1) Real-Area: 16172.5 square km Stadt: Erfurt
```

Real Area Source: <http://www.bernhard-gaul.de/wissen/bundeslaender.php>

Für die Suche nach der richtigen Hauptstadt zu dem jeweiligen Bundesland wurde der Algorithmus Point in Polygon aus der Vorlesung benutzt. Dieser sieht in C++ umgesetzt wie folgt aus:

```
int i = 1;
while(ccw(qA, point, points.at(i)) == 0){
    i += 1;
}
int g = 0;
int s = 0;
int lr = sgn(ccw(qA, point, points.at(i)));
for(unsigned int j = i + 1; j < points.size(); j++){
    int lrnew = sgn(ccw(qA, point, points.at(j)));
    if(abs(lrnew - lr) == 2){
        lr = lrnew;
        if((ccw(points.at(j-1), points.at(j), qA) * ccw(points.at(j-1), points.at(j), point)) <= 0){
            s = s + 1;
        }
    }
}
```