

Table of Contents

Geometry Recipes.....	1
Creating Geometries	1
Procesing Geometries	7
Reading and Writing Geometries	23

Geometry Recipes

Creating Geometries

Create a Point with an XY

```
Point point = new Point(-123,46)
```



Create a LineString from Coordinates

```
LineString lineString = new LineString(  
    [3.1982421875, 43.1640625],  
    [6.7138671875, 49.755859375],  
    [9.7021484375, 42.5927734375],  
    [15.3271484375, 53.798828125]  
)
```



Create a Polygon from a List of Coordinates

```
Polygon polygon = new Polygon([[  
    [-101.35986328125, 47.754097979680026],  
    [-101.5576171875, 46.93526088057719],  
    [-100.12939453125, 46.51351558059737],  
    [-99.77783203125, 47.44294999517949],  
    [-100.45898437499999, 47.88688085106901],  
    [-101.35986328125, 47.754097979680026]  
]])
```



Create a MultiPoint with a List of Points

```
MultiPoint multiPoint = new MultiPoint([  
    new Point(-122.3876953125, 47.5820839916191),  
    new Point(-122.464599609375, 47.25686404408872),  
    new Point(-122.48382568359374, 47.431803338643334)  
])
```



Create a MultiLineString with a List of LineStrings

```
MultiLineString multiLineString = new MultiLineString([
    new LineString (
        [-122.3822021484375, 47.57837853860192],
        [-122.32452392578125, 47.48380086737799]
    ),
    new LineString (
        [-122.32452392578125, 47.48380086737799],
        [-122.29705810546874, 47.303447043862626]
    ),
    new LineString (
        [-122.29705810546874, 47.303447043862626],
        [-122.42889404296875, 47.23262467463881]
    )
])
```



Create a MultiPolygon with a List of Polygons

```
MultiPolygon multiPolygon = new MultiPolygon(  
    new Polygon ([[  
        [-122.2723388671875, 47.818687628247105],  
        [-122.37945556640624, 47.66168780332917],  
        [-121.95373535156249, 47.67093619422418],  
        [-122.2723388671875, 47.818687628247105]  
    ]]),  
    new Polygon ([[  
        [-122.76672363281249, 47.42437092240516],  
        [-122.76672363281249, 47.59505101193038],  
        [-122.52227783203125, 47.59505101193038],  
        [-122.52227783203125, 47.42437092240516],  
        [-122.76672363281249, 47.42437092240516]  
    ]]),  
    new Polygon ([[  
        [-122.20367431640624, 47.543163654317304],  
        [-122.3712158203125, 47.489368981370724],  
        [-122.33276367187499, 47.35371061951363],  
        [-122.11029052734374, 47.3704545156932],  
        [-122.08831787109375, 47.286681888764214],  
        [-122.28332519531249, 47.2270293988673],  
        [-122.2174072265625, 47.154237057576594],  
        [-121.904296875, 47.32579231609051],  
        [-122.06085205078125, 47.47823216312885],  
        [-122.20367431640624, 47.543163654317304]  
    ]])  
    ]])  
)
```



Create a CircularString with a List of Points

```
CircularString circularString = new CircularString([  
    [-122.464599609375, 47.247542522268006],  
    [-122.03613281249999, 47.37789454155521],  
    [-122.37670898437499, 47.58393661978134]  
])
```



Create a *CircularRing* with a List of Points

```
CircularRing circularRing = new CircularRing([
    [-118.47656249999999, 41.508577297439324],
    [-109.6875, 57.51582286553883],
    [-93.8671875, 42.032974332441405],
    [-62.57812500000001, 30.14512718337613],
    [-92.10937499999999, 7.36246686553575],
    [-127.265625, 14.604847155053898],
    [-118.47656249999999, 41.508577297439324]
])
```



Create a CompoundCurve with a List of CircularStrings and LineStrings

```
CompoundCurve compoundCurve = new CompoundCurve([
    new CircularString([
        [27.0703125, 23.885837699862005],
        [5.9765625, 40.17887331434696],
        [22.5, 47.98992166741417],
    ]),
    new LineString([
        [22.5, 47.98992166741417],
        [71.71875, 49.15296965617039],
    ]),
    new CircularString([
        [71.71875, 49.15296965617039],
        [81.5625, 39.36827914916011],
        [69.9609375, 24.5271348225978]
    ])
])
```



```
CompoundRing compoundRing = new CompoundRing([
    new CircularString([
        [27.0703125, 23.885837699862005],
        [5.9765625, 40.17887331434696],
        [22.5, 47.98992166741417],
    ]),
    new LineString([
        [22.5, 47.98992166741417],
        [71.71875, 49.15296965617039],
    ]),
    new CircularString([
        [71.71875, 49.15296965617039],
        [81.5625, 39.36827914916011],
        [69.9609375, 24.5271348225978]
    ]),
    new LineString([
        [69.9609375, 24.5271348225978],
        [27.0703125, 23.885837699862005],
    ])
])
```



Processing Geometries

Get the area of a Geometry

```
Polygon polygon = new Polygon([[
    [-124.80, 48.92],
    [-126.21, 45.33],
    [-114.60, 45.08],
    [-115.31, 51.17],
    [-121.99, 52.05],
    [-124.80, 48.92]
]])
double area = polygon.area
println area
```


62.4026

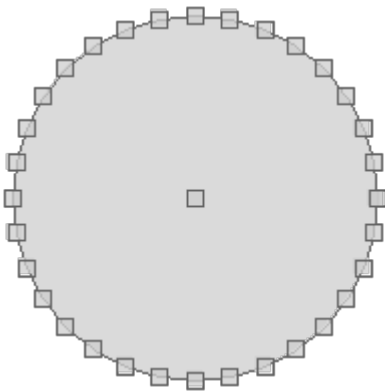
Get the length of a Geometry

```
LineString lineString = new LineString([-122.69, 49.61], [-99.84, 45.33])  
double length = lineString.length  
println length
```

23.24738479915536

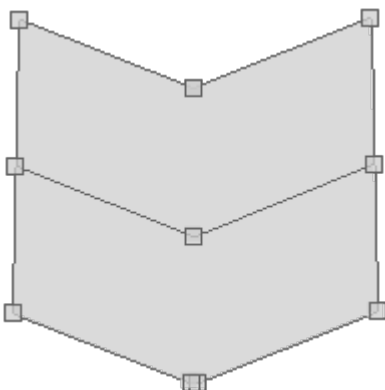
Buffer a Point

```
Point point = new Point(-123,46)  
Geometry bufferedPoint = point.buffer(2)
```



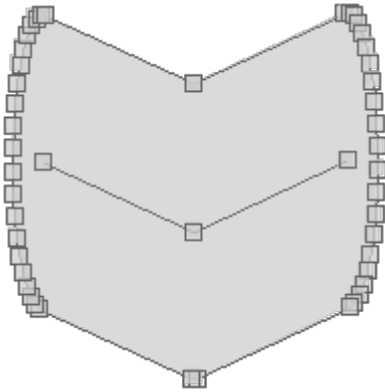
Buffer a LineString with a butt cap

```
LineString line = new LineString([  
    [-122.563, 47.576],  
    [-112.0166, 46.589],  
    [-101.337, 47.606]  
])  
Geometry bufferedLine1 = line.buffer(2.1, 10, Geometry.CAP_BUTT)
```



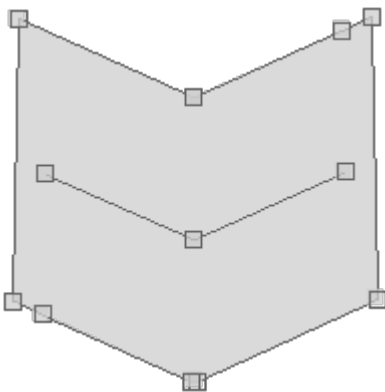
Buffer a LineString with a round cap

```
Geometry bufferedLine2 = line.buffer(2.1, 10, Geometry.CAP_ROUND)
```



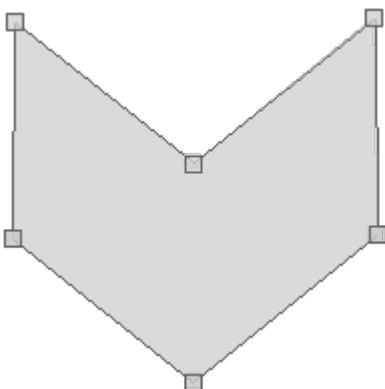
Buffer a LineString with a square cap

```
Geometry bufferedLine3 = line.buffer(2.1, 10, Geometry.CAP_SQUARE)
```



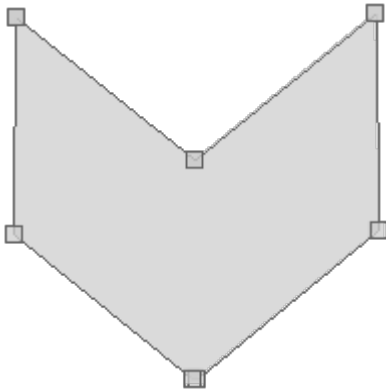
Buffer a LineString on the right side only

```
LineString line = new LineString([  
    [-122.563, 47.576],  
    [-112.0166, 46.589],  
    [-101.337, 47.606]  
])  
Geometry rightBufferedLine = line.singleSidedBuffer(1.5)
```



Buffer a LineString on the left side only

```
Geometry leftBufferedLine = line.singleSidedBuffer(-1.5)
```

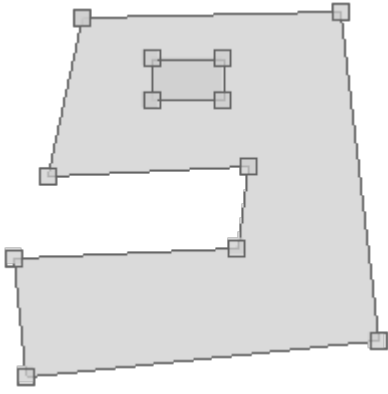


Check whether a Geometry contains another Geometry

```
Polygon polygon1 = new Polygon([[
    [-120.739, 48.151],
    [-121.003, 47.070],
    [-119.465, 47.137],
    [-119.553, 46.581],
    [-121.267, 46.513],
    [-121.168, 45.706],
    [-118.476, 45.951],
    [-118.762, 48.195],
    [-120.739, 48.151]
]])
```

```
Polygon polygon2 = new Polygon([[
    [-120.212, 47.591],
    [-119.663, 47.591],
    [-119.663, 47.872],
    [-120.212, 47.872],
    [-120.212, 47.591]
]])
```

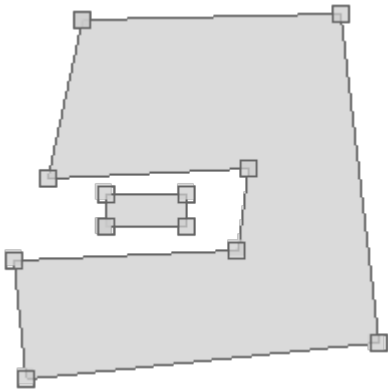
```
boolean contains = polygon1.contains(polygon2)
println contains
```



true

```
Polygon polygon3 = new Polygon([[
    [-120.563, 46.739],
    [-119.948, 46.739],
    [-119.948, 46.965],
    [-120.563, 46.965],
    [-120.563, 46.739]
]])

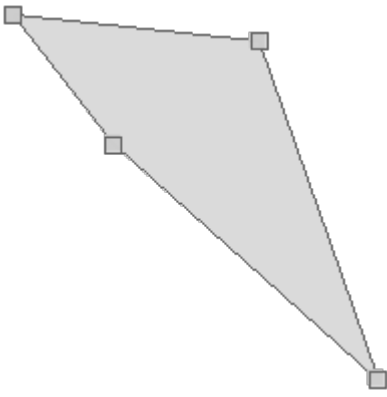
contains = polygon1.contains(polygon3)
println contains
```



false

Create a convexhull Geometry around a Geometry

```
Geometry geometry = new MultiPoint(
    new Point(-119.882, 47.279),
    new Point(-100.195, 46.316),
    new Point(-111.796, 42.553),
    new Point(-90.7031, 34.016)
)
Geometry convexHull = geometry.convexHull
```

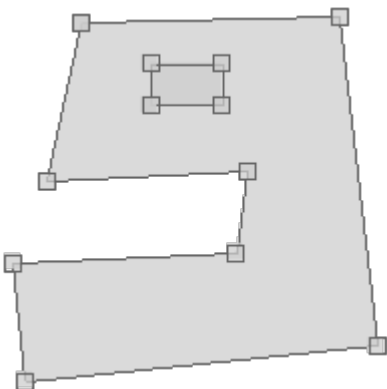


Check whether a Geometry covers another Geometry

```
Polygon polygon1 = new Polygon([[
    [-120.739, 48.151],
    [-121.003, 47.070],
    [-119.465, 47.137],
    [-119.553, 46.581],
    [-121.267, 46.513],
    [-121.168, 45.706],
    [-118.476, 45.951],
    [-118.762, 48.195],
    [-120.739, 48.151]
]])

Polygon polygon2 = new Polygon([[
    [-120.212, 47.591],
    [-119.663, 47.591],
    [-119.663, 47.872],
    [-120.212, 47.872],
    [-120.212, 47.591]
]])

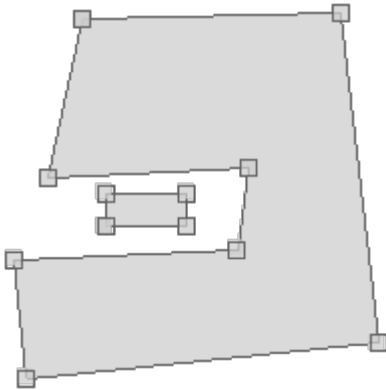
boolean isCovered = polygon1.covers(polygon2)
println isCovered
```



true

```
Polygon polygon3 = new Polygon([[
    [-120.563, 46.739],
    [-119.948, 46.739],
    [-119.948, 46.965],
    [-120.563, 46.965],
    [-120.563, 46.739]
]])

isCovered = polygon1.covers(polygon3)
println isCovered
```



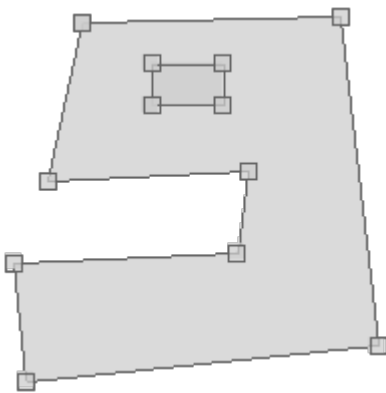
false

Check whether a Geometry is covered by another Geometry

```
Polygon polygon1 = new Polygon([[
    [-120.739, 48.151],
    [-121.003, 47.070],
    [-119.465, 47.137],
    [-119.553, 46.581],
    [-121.267, 46.513],
    [-121.168, 45.706],
    [-118.476, 45.951],
    [-118.762, 48.195],
    [-120.739, 48.151]
]])

Polygon polygon2 = new Polygon([[
    [-120.212, 47.591],
    [-119.663, 47.591],
    [-119.663, 47.872],
    [-120.212, 47.872],
    [-120.212, 47.591]
]])

boolean isCoveredBy = polygon2.coveredBy(polygon1)
println isCoveredBy
```



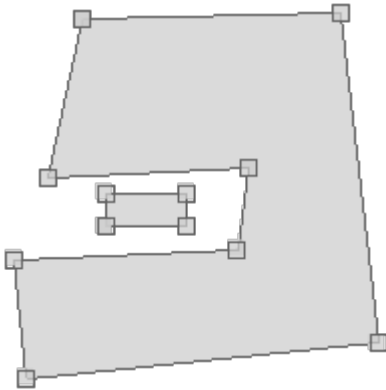
true

```

Polygon polygon3 = new Polygon([[
    [-120.563, 46.739],
    [-119.948, 46.739],
    [-119.948, 46.965],
    [-120.563, 46.965],
    [-120.563, 46.739]
]])

isCoveredBy = polygon3.coveredBy(polygon1)
println isCoveredBy

```



false

Check whether one Geometry crosses another Geometry

```

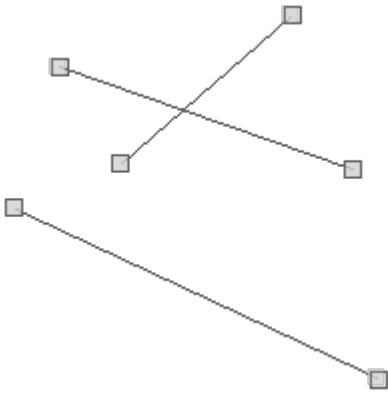
LineString line1 = new LineString([[-122.486, 47.256], [-121.695, 46.822]])
LineString line2 = new LineString([[-122.387, 47.613], [-121.750, 47.353]])
LineString line3 = new LineString([[-122.255, 47.368], [-121.882, 47.746]])

boolean doesCross12 = line1.crosses(line2)
println doesCross12

boolean doesCross13 = line1.crosses(line3)
println doesCross13

boolean doesCross23 = line2.crosses(line3)
println doesCross23

```

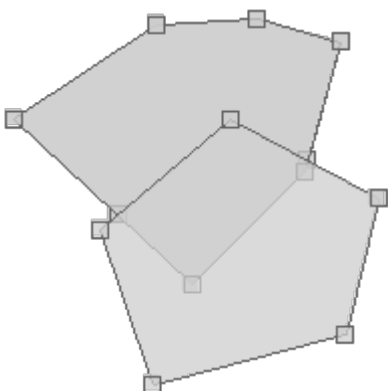
false
false
true

Calculate the difference between two Geometries

```
Polygon polygon1 = new Polygon([[
    [-121.915, 47.390],
    [-122.640, 46.995],
    [-121.739, 46.308],
    [-121.168, 46.777],
    [-120.981, 47.316],
    [-121.409, 47.413],
    [-121.915, 47.390]
]])
```

```
Polygon polygon2 = new Polygon([[
    [-120.794, 46.664],
    [-121.541, 46.995],
    [-122.200, 46.536],
    [-121.937, 45.890],
    [-120.959, 46.096],
    [-120.794, 46.664]
]])
```

```
Geometry difference = polygon1.difference(polygon2)
```



Check whether one Geometry is disjoint from another Geometry

```
Polygon polygon1 = new Polygon([[
    [-121.915, 47.390],
    [-122.640, 46.995],
    [-121.739, 46.308],
    [-121.168, 46.777],
    [-120.981, 47.316],
    [-121.409, 47.413],
    [-121.915, 47.390]
]])

Polygon polygon2 = new Polygon([[
    [-120.794, 46.664],
    [-121.541, 46.995],
    [-122.200, 46.536],
    [-121.937, 45.890],
    [-120.959, 46.096],
    [-120.794, 46.664]
]])

Polygon polygon3 = new Polygon([[
    [-120.541, 47.376],
    [-120.695, 47.047],
    [-119.794, 46.830],
    [-119.586, 47.331],
    [-120.102, 47.509],
    [-120.541, 47.376]
]])

boolean isDisjoint12 = polygon1.disjoint(polygon2)
println isDisjoint12

boolean isDisjoint13 = polygon1.disjoint(polygon3)
println isDisjoint13

boolean isDisjoint23 = polygon2.disjoint(polygon3)
println isDisjoint23
```



```
false  
true  
true
```

Calculate the distance between two Geometries

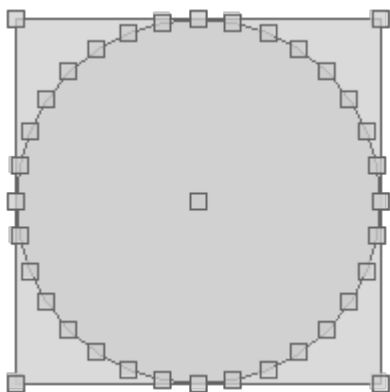
```
Point point1 = new Point(-122.442, 47.256)  
Point point2 = new Point(-122.321, 47.613)  
double distance = point1.distance(point2)  
println distance
```



```
0.37694827231332195
```

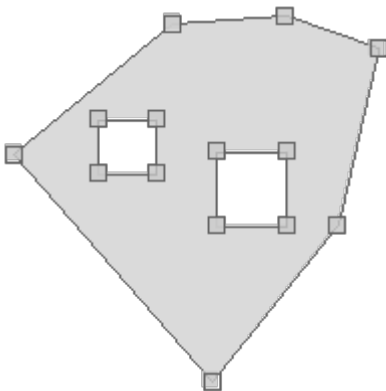
Get Bounds from a Geometry

```
Point point = new Point(-123,46)  
Polygon polygon = point.buffer(2)  
Bounds bounds = polygon.bounds
```



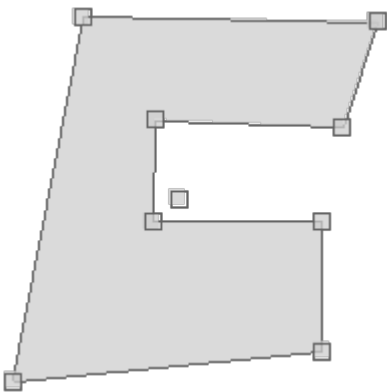
Get the Boundary from a Geometry

```
Polygon polygon = new Polygon([
    [
        [-121.915, 47.390],
        [-122.640, 46.995],
        [-121.739, 46.308],
        [-121.168, 46.777],
        [-120.981, 47.316],
        [-121.409, 47.413],
        [-121.915, 47.390]
    ],
    [
        [-122.255, 46.935],
        [-121.992, 46.935],
        [-121.992, 47.100],
        [-122.255, 47.100],
        [-122.255, 46.935]
    ],
    [
        [-121.717, 46.777],
        [-121.398, 46.777],
        [-121.398, 47.002],
        [-121.717, 47.002],
        [-121.717, 46.777]
    ]
])
Geometry boundary = polygon.boundary
```



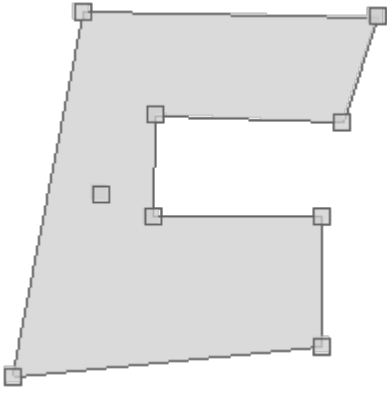
Get the Centroid from a Geometry

```
Polygon polygon = new Polygon([[
    [-118.937, 48.327],
    [-121.157, 48.356],
    [-121.684, 46.331],
    [-119.355, 46.498],
    [-119.355, 47.219],
    [-120.629, 47.219],
    [-120.607, 47.783],
    [-119.201, 47.739],
    [-118.937, 48.327]
]])
Geometry centroid = polygon.centroid
```



Get the Interior Point from a Geometry

```
Polygon polygon = new Polygon([[
    [-118.937, 48.327],
    [-121.157, 48.356],
    [-121.684, 46.331],
    [-119.355, 46.498],
    [-119.355, 47.219],
    [-120.629, 47.219],
    [-120.607, 47.783],
    [-119.201, 47.739],
    [-118.937, 48.327]
]])
Geometry interiorPoint = polygon.interiorPoint
```



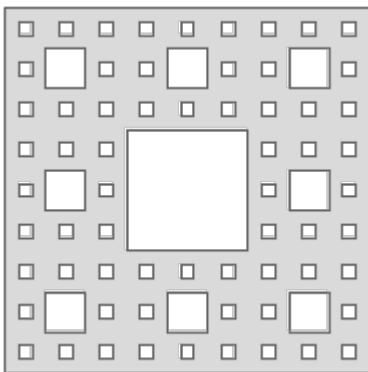
Create a Geometry of a String

```
Geometry geometry = Geometry.createFromText("Geo")
```



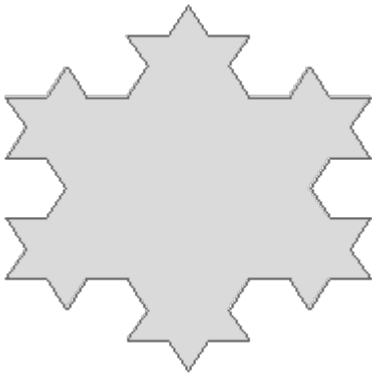
Create a Sierpinski Carpet in a given Bounds and with a number of points

```
Bounds bounds = new Bounds(21.645,36.957,21.676,36.970, "EPSG:4326")
Geometry geometry = Geometry.createSierpinskiCarpet(bounds, 50)
```



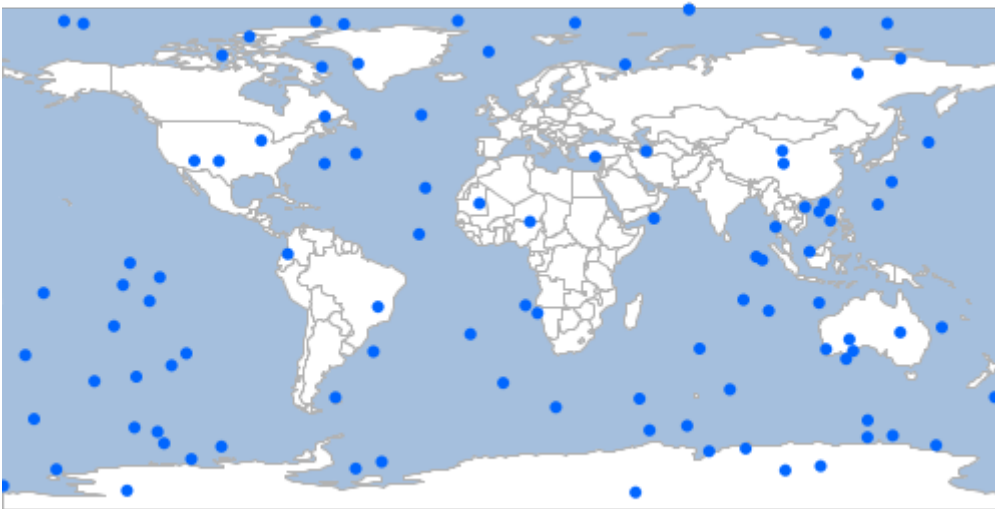
Create a Koch Snowflake in a given Bounds and with a number of points

```
Bounds bounds = new Bounds(21.645,36.957,21.676,36.970, "EPSG:4326")
Geometry geometry = Geometry.createKochSnowflake(bounds, 50)
```



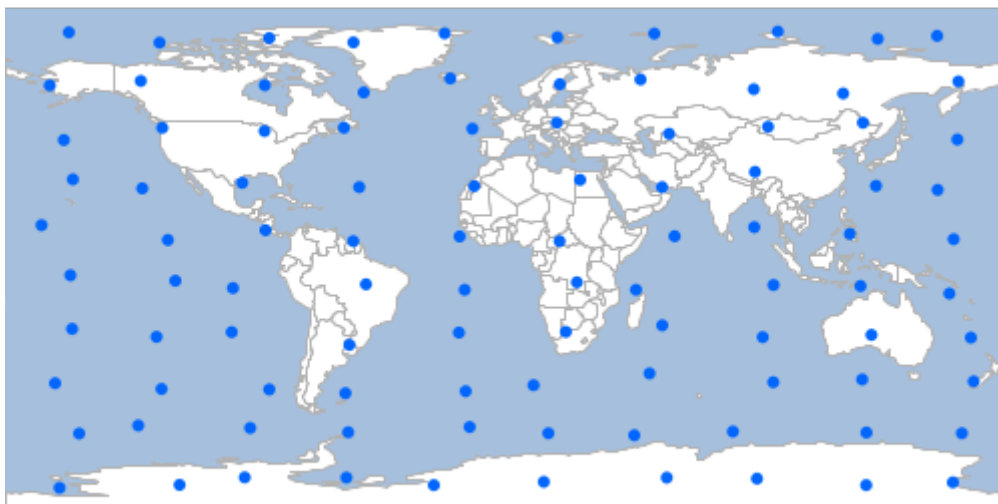
Create a number of random points within a given Geometry

```
Geometry geometry = new Bounds(-180, -90, 180, 90).geometry
MultiPoint randomPoints = Geometry.createRandomPoints(geometry, 100)
```



Create a number of random points within a given Geometry where the points are constrained to the cells of a grid

```
Bounds bounds = new Bounds(-180, -90, 180, 90)
MultiPoint randomPoints = Geometry.createRandomPointsInGrid(bounds, 100, true, 0.5)
```



Reading and Writing Geometries

The `geoscript.geom.io` package has several Readers and Writers for converting `geoscript.geom.Geometry` to and from strings.

Readers and Writers

Find all Geometry Readers

```
List<Reader> readers = Readers.list()
readers.each { Reader reader ->
    println reader.class.simpleName
}
```

```
GeobufReader
GeoJSONReader
GeoRSSReader
Gml2Reader
Gml3Reader
GpxReader
KmlReader
WkbReader
WktReader
GooglePolylineEncoder
```

Find a Geometry Reader

```
String wkt = "POINT (-123.15 46.237)"
Reader reader = Readers.find("wkt")
Geometry geometry = reader.read(wkt)
```




Find all Geometry Writers

```
List<Writer> writers = Writers.list()
writers.each { Writer writer ->
    println writer.class.simpleName
}
```

```
GeobufWriter
GeoJSONWriter
GeoRSSWriter
Gml2Writer
Gml3Writer
GpxWriter
KmlWriter
WkbWriter
WktWriter
GooglePolylineEncoder
```

Find a Geometry Writer

```
Geometry geometry = new Point(-122.45, 43.21)
Writer writer = Writers.find("geojson")
String geojson = writer.write(geometry)
println geojson
```

```
{"type":"Point","coordinates":[-122.45,43.21]}
```

WKB

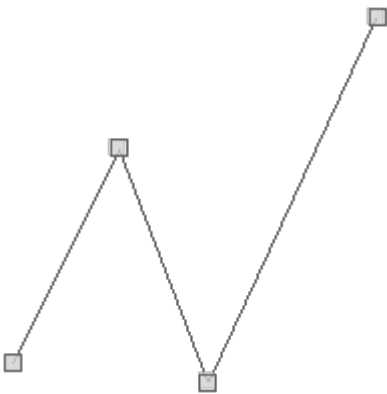
Read a Geometry from WKB using the WkbReader

```
String wkb = "0000000001C05EC9999999999A40471E5604189375"
WkbReader reader = new WkbReader()
Geometry geometry = reader.read(wkb)
```



Read a Geometry from WKB using the Geometry.fromWKB() static method

```
String wkb =  
"0000000002000000004400995810624DD2F404594FDF3B645A2401ADA1CAC0831274048E0A3D70A3D71402  
3676C8B43958140454BC6A7EF9DB2402EA3D70A3D70A4404AE624DD2F1AA0"  
Geometry geometry = Geometry.fromWKB(wkb)
```



Get the WKB of a Geometry

```
Geometry geometry = new Point(-123.15, 46.237)  
String wkb = geometry.wkb  
println wkb
```

```
0000000001C05EC9999999999A40471E5604189375
```

Write a Geometry to WKB using the WkbWriter

```
Geometry geometry = new LineString(  
    [3.198, 43.164],  
    [6.713, 49.755],  
    [9.702, 42.592],  
    [15.32, 53.798]  
)  
WkbWriter writer = new WkbWriter()  
String wkb = writer.write(geometry)  
println wkb
```

```
000000000200000004400995810624DD2F404594FDF3B645A2401ADA1CAC0831274048E0A3D70A3D714023  
676C8B43958140454BC6A7EF9DB2402EA3D70A3D70A4404AE624DD2F1AA0
```

WKT

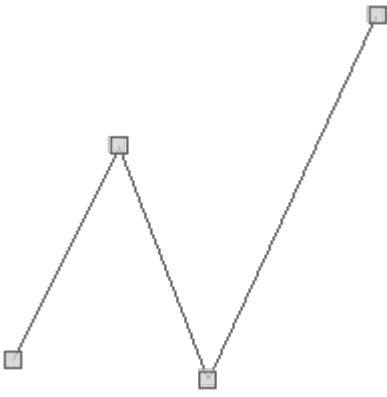
Read a Geometry from WKT using the WktReader

```
String wkt = "POINT (-123.15 46.237)"  
WktReader reader = new WktReader()  
Geometry geometry = reader.read(wkt)
```



Read a Geometry from WKT using the Geometry.fromWKT() static method

```
String wkt = "LINESTRING (3.198 43.164, 6.7138 49.755, 9.702 42.592, 15.327 53.798)"  
Geometry geometry = Geometry.fromWKT(wkt)
```



Get the WKT of a Geometry

```
Geometry geometry = new Point(-123.15, 46.237)
String wkt = geometry.wkt
println wkt
```

```
POINT (-123.15 46.237)
```

Write a Geometry to WKT using the WktWriter

```
Geometry geometry = new LineString(
    [3.198, 43.164],
    [6.713, 49.755],
    [9.702, 42.592],
    [15.32, 53.798]
)
WktWriter writer = new WktWriter()
String wkt = writer.write(geometry)
println wkt
```

```
LINESTRING (3.198 43.164, 6.713 49.755, 9.702 42.592, 15.32 53.798)
```

GeoJSON

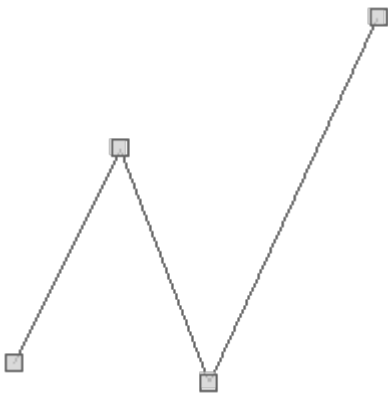
Read a Geometry from GeoJSON using the GeoJSONReader

```
String json = '{"type":"Point","coordinates":[-123.15,46.237]}'
GeoJSONReader reader = new GeoJSONReader()
Geometry geometry = reader.read(json)
```



Read a Geometry from GeoJSON using the Geometry.fromGeoJSON() static method

```
String json =  
'{"type":"LineString","coordinates":[[3.198,43.164],[6.713,49.755],[9.702,42.592],[15.  
32,53.798]]}'  
Geometry geometry = Geometry.fromGeoJSON(json)
```



Get the GeoJSON of a Geometry

```
Geometry geometry = new Point(-123.15, 46.237)  
String json = geometry.geoJSON  
println json
```

```
{"type":"Point","coordinates":[-123.15,46.237]}
```

Write a Geometry to GeoJSON using the GeoJSONWriter

```
Geometry geometry = new LineString(  
    [3.198, 43.164],  
    [6.713, 49.755],  
    [9.702, 42.592],  
    [15.32, 53.798]  
)  
GeoJSONWriter writer = new GeoJSONWriter()  
String json = writer.write(geometry)  
println json
```

```
{"type":"LineString","coordinates":[[3.198,43.164],[6.713,49.755],[9.702,42.592],[15.32,53.798]]}
```

KML

Read a Geometry from KML using the KmlReader

```
String kml = "<Point><coordinates>-123.15,46.237</coordinates></Point>"  
KmlReader reader = new KmlReader()  
Geometry geometry = reader.read(kml)
```



Read a Geometry from KML using the Geometry.fromKml() static method

```
String kml = "<LineString><coordinates>3.198,43.164 6.713,49.755 9.702,42.592  
15.32,53.798</coordinates></LineString>"  
Geometry geometry = Geometry.fromKml(kml)
```



Get the KML of a Geometry

```
Geometry geometry = new Point(-123.15, 46.237)
String kml = geometry.kml
println kml
```

```
<Point><coordinates>-123.15,46.237</coordinates></Point>
```

Write a Geometry to KML using the KmlWriter

```
Geometry geometry = new LineString(
    [3.198, 43.164],
    [6.713, 49.755],
    [9.702, 42.592],
    [15.32, 53.798]
)
KmlWriter writer = new KmlWriter()
String kml = writer.write(geometry)
println kml
```

```
<LineString><coordinates>3.198,43.164 6.713,49.755 9.702,42.592
15.32,53.798</coordinates></LineString>
```

Geobuf

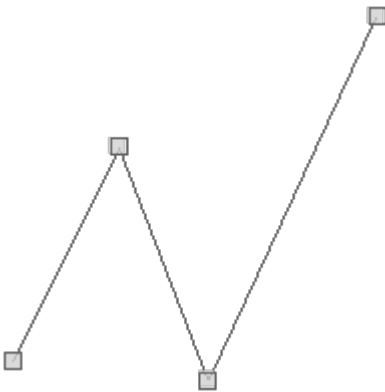
Read a Geometry from Geobuf using the GeobufReader

```
String geobuf = "10021806320c08001a08dffab87590958c2c"
GeobufReader reader = new GeobufReader()
Geometry geometry = reader.read(geobuf)
```



Read a Geometry from Geobuf using the Geometry.fromGeobuf() static method

```
String geobuf =  
"10021806322408021a20e0b08603c0859529f089ad03b0c8a40690efec02efb1ea06a0e5ad05e0f5d70a"  
Geometry geometry = Geometry.fromGeobuf(geobuf)
```



Get the Geobuf of a Geometry

```
Geometry geometry = new Point(-123.15, 46.237)  
String geobuf = geometry.geobuf  
println geobuf
```

```
10021806320c08001a08dffab87590958c2c
```

Write a Geometry to Geobuf using the GeobufWriter

```
Geometry geometry = new LineString(  
    [3.198, 43.164],  
    [6.713, 49.755],  
    [9.702, 42.592],  
    [15.32, 53.798]  
)  
GeobufWriter writer = new GeobufWriter()  
String geobuf = writer.write(geometry)  
println geobuf
```


10021806322408021a20e0b08603c0859529f089ad03b0c8a40690efec02efb1ea06a0e5ad05e0f5d70a