

# Kurs programowania w QGIS za pomocą Pythona

Dzień 3

Hel wrzesień 2019



```
import processing
processing.runalg('sag
None,"ifelse(gt(a,150),
"D:\\PROJ14\\FYQGIS\\pr
rlayer= QgsRasterLayer
"NMT5")
QgsMapLayerRegistry.i
```



## OGR Simple Feature Library

The OGR Simple Features Library is a C++ [open source](#) library (and commandline tools) providing read (and sometimes write) access to a variety of vector file formats including ESRI Shapefiles, S-57, SDTS, PostGIS, Oracle Spatial, and Mapinfo mid/mif and TAB formats.

OGR is a part of the [GDAL](#) library.

### Resources

- [OGR Supported Formats](#) : ESRI Shapefile, ESRI ArcSDE, MapInfo (tab and mid/mif), GML, KML, PostGIS, Oracle Spatial, ...
- [OGR Utility Programs](#) : ogrinfo, ogr2ogr, ogrtindex
- [OGR Class Documentation](#)
- [OGR C++ API Read/Write Tutorial](#)

Praca na danych wektorowych

documentation »

# Welcome to the Python GDAL/OGR Cookbook!

<https://pcjericks.github.io/py-gdalogr-cookbook/>

## OGR - schemat logiczny

**Drajwer**

`driver=ogr.GetDriverByName('ESRI shapefile')`

**Źródło**

`dataSource=driver.CreateDataSource(ff), dataSource=driver.Open(ff,0)` 0-czytanie,  
1-czytanie i pisanie

`ff='D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\Poligony6.shp'`

**Warstwa**

`vlayer=dataSource.CreateLayer(...), vlayer=dataSource.GetLayer()`

**Obiekt**

**Geometria**

```
from osgeo import ogr
```

```
driver=ogr.GetDriverByName('ESRI shapefile')
```

```
ff='D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\punkty.shp'
```

```
dataSource=driver.Open(ff,0)
```

```
vlayer=dataSource.GetLayer()
```

## OGR - schemat logiczny

*Podstawowe metody stosowane na warstwach i definiowania obiektu:*

**feature**=vlayer.GetFeature(**indeks\_obiekt**) : udostępnia\_dany\_obiekt

vlayer.CreateFeature(**obiekt**) : umieszczenie obiektu w warstwie (jako ostatniego)

vlayer.SetFeature(**obiekt**) : nowe zdefiniowanie istniejącego obiektu (zmiana obiektu)

n=vlayer.GetFeatureCount() : liczba obiektów w warstwie

Extent=vlayer.GetExtent() : tupla (xul,xlr,ylr,yul) Extent[0],Extent[1]....

vlayer.DeleteFeature(**indeks\_obiektu**)

feature\_definitione=vlayer.GetLayerDefn() :ogólna definicja obiektu

**feature**=ogr.Feature(feature\_definitione)

zdefiniowanie obiektu

Zad.

Wydrukuj liczbę obiektów  
w warstwie.

Wydrukuj współrzędne  
zakresu przestrzennego  
danych.

## OGR - SHP tablica atrybutowa (.shp), tablica atrybutowa (.dbf)

### Czytanie danych

obiekt.GetFieldAsInteger()

obiekt.GetFieldAsDouble()

obiekt.GetFieldAsString()

Każde pole ma indeks:

feature.GetFieldIndex('pp')

-1 nie ma takiego pola

```
for i in range (0, numFeature):  
    feature=vlayer.GetFeature(i)
```

```
x = feature.GetFieldAsInteger('ID')  
y=feature.GetFieldAsDouble(1)  
print i,x,y
```

### Wprowadzanie nowego pola

fdef=ogr.FieldDefn(....., ogr.OFTInteger)

warstwa.CreateField(fdef)

Enumerator	
OFTInteger	Simple 32bit integer.
OFTIntegerList	List of 32bit integers.
OFTReal	Double Precision floating point.
OFTRealList	List of doubles.
OFTString	String of ASCII chars.
OFTStringList	Array of strings.
OFTWideString	deprecated
OFTWideStringList	deprecated
OFTBinary	Raw Binary data.
OFTDate	Date.
OFTTime	Time.
OFTDateTime	Date and Time.
OFTInteger64	Single 64bit integer.
OFTInteger64List	List of 64bit integers.



### Zmiana/wpisywanie danych

obiekt.SetField()

warstwa.SetFeature(obiekt)

```
fdef=ogr.FieldDefn('nowe',ogr.OFTInteger)  
vlayer.CreateField(fdef)
```

**Pole widoczne po powtórny załadowaniu !!!!!**

```
for i in range (0, numFeature):  
    feature=vlayer.GetFeature(i)  
    y=feature.GetFieldAsDouble(1)  
    feature.SetField(2,y/2.0)  
    vlayer.SetFeature(feature)
```

**Zmiany w polu widoczne po powtórny załadowaniu !!!!!**

Enumerator	
OFTInteger	Simple 32bit integer.
OFTIntegerList	List of 32bit integers.
OFTReal	Double Precision floating point.
OFTRealList	List of doubles.
OFTString	String of ASCII chars.
OFTStringList	Array of strings.
OFTWideString	deprecated
OFTWideStringList	deprecated
OFTBinary	Raw Binary data.
OFTDate	Date.
OFTTime	Time.
OFTDateTime	Date and Time.
OFTInteger64	Single 64bit integer.
OFTInteger64List	List of 64bit integers.

0

2

4

```
vlayer.GetGeomType()
```

```
: 1 # Listowanie pól
2
3 from osgeo import ogr,gdal
4
5 schema = []
6 ldefn = vlayer.GetLayerDefn()
7 for n in range(ldefn.GetFieldCount()):
8     fdefn = ldefn.GetFieldDefn(n)
9     schema.append(fdefn.name,)
10    print(fdefn.type)
11 print (schema)
12
```

0

2

4

['K1', 'K2', 'K3']

```
0 Geometry
1 Point
2 Line
3 Polygon
4 MultiPoint
5 MultiLineString
6 MultiPolygon
100 No Geometry
```



### Zadanie 3\_1

Napisać skrypt obliczający w nowym polu *P2* procent w danym rekordzie sumarycznych wartości danego pola *K1*.

$$p2 = \frac{K1}{\sum K1} * 100$$

PRZY ZAPISYWANIU DO PLIKÓW

```
driver.Open(...., 1)
```

Na końcu

```
dataSource=None
```

	K1	K2	K3	XXX	P2
1	12	5.54371000000	A	24	4.940000000000000
2	6	16.86800000000	BC	NULL	15.029999999999999
3	3	3.43520000000	FD	6	3.060000000000000
4	15	55.67400000000	G	30	49.619999999999997
5	1	18.47900000000	TT	2	16.469999999999999
6	3	12.20000000000	BB	6	10.869999999999999

## OGR - schemat logiczny

### Geometria

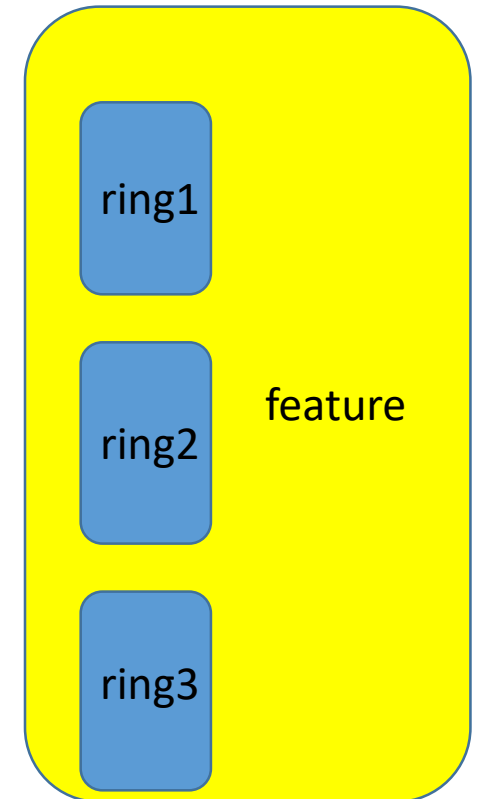
**geometria** = obiekt.GetGeometryRef()

**geometria** = ogr.GeometryRef(wkbPoint)

**geometria** = ogr.GeometryRef(wkbLineString)

**geometria** = ogr.GeometryRef(wkbLineRing)

**geometria** = ogr.GeometryRef(wkbPolygon)





```
>>> dir(geometry)
['AddGeometry', 'AddGeometryDirectly', 'AddPoint', 'AddPointM', 'AddPointZM', 'AddPoint_2D', 'Area', 'AssignSpatialReference', 'Boundary', 'Buffer', 'Centroid', 'Clone', 'CloseRings', 'Contains', 'ConvexHull', 'CoordinateDimension', 'Crosses', 'DelaunayTriangulation', 'Destroy', 'Difference', 'Disjoint', 'Distance', 'Empty', 'Equal', 'Equals', 'ExportToGML', 'ExportToIsoWkb', 'ExportToIsoWkt', 'ExportToJson', 'ExportToKML', 'ExportToWkb', 'ExportToWkt', 'FlattenTo2D', 'GetArea', 'GetBoundary', 'GetCoordinateDimension', 'GetCurveGeometry', 'GetDimension', 'GetEnvelope', 'GetEnvelope3D', 'GetGeometryCount', 'GetGeometryName', 'GetGeometryRef', 'GetGeometryType', 'GetLinearGeometry', 'GetM', 'GetPoint', 'GetPointCount', 'GetPointZM', 'GetPoint_2D', 'GetPoints', 'GetSpatialReference', 'GetX', 'GetY', 'GetZ', 'HasCurveGeometry', 'Intersect', 'Intersection', 'Intersects', 'Is3D', 'IsEmpty', 'IsMeasured', 'IsRing', 'IsSimple', 'IsValid', 'Length', 'Overlaps', 'PointOnSurface', 'Segmentize', 'Set3D', 'SetCoordinateDimension', 'SetMeasured', 'SetPoint', 'SetPointM', 'SetPointZM', 'SetPoint_2D', 'Simplify', 'SimplifyPreserveTopology', 'SymDifference', 'SymmetricDifference', 'Touches', 'Transform', 'TransformTo', 'Union', 'UnionCascaded', 'Value', 'Within', 'WkbSize', '__class__', '__del__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattribute__', '__hash__', '__init__', '__iter__', '__module__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__setstate__', '__sizeof__', '__str__', '__subclasshook__', '__swig_destroy__', '__weakref__', 'next', 'this', 'thisown']
```

## OGR - schemat logiczny - geometria

### PUNKTY

#### Tylko czytanie

```
vlayer=dataSource.GetLayer()  
  
nn=vlayer.GetFeatureCount()  
  
- for i in range(0,nn):  
    ppoint=vlayer.GetFeature(i)  
    geometry=ppoint.GetGeometryRef()  
    x=geometry.GetX()  
    y=geometry.GetY()  
    print i,x,y  
  
dataSource=None
```

X=geometry.GetX()

Y=geometry.GetY()

geometry.AddPoint(x,y)

obiekt.SetGeometry(geometry)

geometry=objekt.GetGeometryRef()

geometry=ogr.Geometry(ogr.wkbPoint) (pusta)

#### Zmiana geometrii

```
point=ogr.Geometry(ogr.wkbPoint)  
  
- for i in range(0,n):  
    feature=vlayer.GetFeature(i)  
    geometry=feature.GetGeometryRef()  
    x=geometry.GetX()  
    y=geometry.GetY()  
    # zmniejszenie x o 100  
    x=x-100.0  
    point.AddPoint(x,y)  
    feature.SetGeometry(point)  
    vlayer.SetFeature(feature)
```

## OGR - schemat logiczny - geometria

### LINIE

#### Tylko czytanie

```
vlayer=dataSource.GetLayer()  
  
nn=vlayer.GetFeatureCount()  
  
for i in range(0,nn):  
    fline=vlayer.GetFeature(i)  
    geometry=fline.GetGeometryRef()  
    nvertex=geometry.GetPointCount()  
    for j in range(0,nvertex):  
        x=geometry.GetX(j)  
        y=geometry.GetY(j)  
        print i,j,x,y  
  
dataSource=None
```

```
geometria=objekt.GetGeometryReff()  
nvertex=geometria.GetPointCount()
```

```
geometria=ogr.Geometry(ogr.wkbLineString) (pusta)
```

#### Zmiana geometrii (nowe)

```
nvertex=geometria.GetPointCount()  
X=geometria.GetX(n)  
Y=geometria.GetY(n)  
  
geometria.AddPoint(x,y)  
  
obiekt.SetGeometry(geometria)
```

```
vlayer=dataSource.GetLayer()  
  
featureDefn=vlayer.GetLayerDefn()  
feature=ogr.Feature(featureDefn)  
  
line=ogr.Geometry(ogr.wkbLineString)  
line.AddPoint(439483,705921)  
line.AddPoint(439943,705944)  
feature.SetGeometry(line)  
feature.SetField('Id',1)  
vlayer.CreateFeature(feature)  
.....
```

## OGR - schemat logiczny - geometria

### POLIGONY

#### Tylko czytanie

```
vlayer=dataSource.GetLayer()
nn=vlayer.GetFeatureCount()

print vlayer.GetGeomType()

for i in range(0,nn):
    flpol=vlayer.GetFeature(i)
    polgeometry=flpol.GetGeometryRef()
    nrings=polgeometry.GetGeometryCount()
    for j in range(0,nrings):
        ring=polgeometry.GetGeometryRef(j)
        nrvertex=ring.GetPointCount()
        for k in range(0,nrvertex):
            x=ring.GetX(k)
            y=ring.GetY(k)
            print i,j,k,x,y

dataSource=None
```

Wydrukuj poligony, ringi i werteksy.

```
nrings=geometria.GetGeometryCount()
geom_rings=geometria.GetGeometryRef(k)

nvertex=geom_rings.GetPointCount()
X=geom_rings.GetX(n)
Y=geom_rings.GetY(n)

geom_rings.AddPoint(x,y)
geometria.AddGeometry(geom_rings)
geom_rings.Empty()

obiekt.SetGeometry(geometria)
```

#### Zmiana geometrii (nowe)

```
vlayer=dataSource.GetLayer()
poly_def=vlayer.GetLayerDefn()
poly=ogr.Feature(poly_def)

poly_geom=ogr.Geometry(ogr.wkbPolygon)
ring=ogr.Geometry(ogr.wkbLinearRing)
ring.AddPoint(439750,705890)
ring.AddPoint(440150,705890)
ring.AddPoint(440150,705530)
ring.AddPoint(439750,705530)
ring.AddPoint(439750,705890)
poly_geom.AddGeometry(ring)
ring.Empty()
ring.AddPoint(439800,705790)
ring.AddPoint(440100,705790)
ring.AddPoint(440100,705630)
ring.AddPoint(439800,705630)
ring.AddPoint(439800,705790)
poly_geom.AddGeometry(ring)
poly.SetGeometry(poly_geom)
vlayer.CreateFeature(poly)
```

Tworzenie geometrii obiektów:

Punkty (nowy shp)

Polilinie

Poligony

```
from osgeo import ogr, osr
import numpy as np

# tablica czterech punktow
ptab=np.zeros((4,2),float)
ptab[0,0]=438700;ptab[0,1]=706400
ptab[1,0]=440260;ptab[1,1]=706400
ptab[2,0]=440260;ptab[2,1]=705500
ptab[3,0]=438700;ptab[3,1]=705500

driver=ogr.GetDriverByName('ESRI shapefile')
ff=r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\Punkty_n.shp'

# uklad wspolrzecznych
spatRef=osr.SpatialReference()
spatRef.ImportFromEPSG(2180)

# stworzymy plik prj
spatRef.MorphToESRI()
fole=open(r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\Punkty_n.prj','w')
fole.write(spatRef.ExportToWkt())
fole.close()
```

Tworzenie geometrii obiektów:

Punkty (nowy shp)

Polilinie

Poligony

Zad

Wykonaj nową warstwę 4  
punktów jako wierzchołków  
prostokąta:

XY lewy górny 438700, 706400

XY prawy 706400, 705500

```
# tworzymy plik prj
spatRef.MorphToESRI()
fole=open(r'C:\JACEK2\QGISHL18\Hel18\dzien3\dane\Punkty_n.prj','w')
fole.write(spatRef.ExportToWkt())
fole.close()

dataSource=driver.CreateDataSource(ff)
vlayer=dataSource.CreateLayer('Punkty_n',geom_type=ogr.wkbPoint)
fieldDef=ogr.FieldDefn('IDD',ogr.OFTInteger)
vlayer.CreateField(fieldDef)

featureDefn=vlayer.GetLayerDefn()
feature=ogr.Feature(featureDefn)

point=ogr.Geometry(ogr.wkbPoint)
for i in range(0,4):
    x=ptab[i,0];y=ptab[i,1]
    point.AddPoint(x,y)
    feature.SetGeometry(point)
    feature.SetField('IDD',i)
    vlayer.CreateFeature(feature)

feature.Destroy()
dataSource=None
```



```
from osgeo import ogr
```

```
driver=ogr.GetDriverByName('ESRI shapefile')  
ff='\\punkty.shp'  
dataSource=driver.Open(ff,0) # 0-czytanie, 1-zmiany
```

Dostęp

```
vlayer=dataSource.GetLayer()
```

```
nn=vlayer.GetFeatureCount()
```

```
for i in range(0,nn):  
    ppoint=vlayer.GetFeature(i)  
    geometry=ppoint.GetGeometryRef()  
    x=geometry.GetX()  
    y=geometry.GetY()  
    print i,x,y
```

Zmiana

```
for i in range(0,nn):  
    if i==5:  
        ppoint=vlayer.GetFeature(i)  
        geometry=ppoint.GetGeometryRef()  
        x=geometry.GetX()  
        y=geometry.GetY()  
        x=x+600  
  
        point=ogr.Geometry(ogr.wkbPoint)  
        point.AddPoint(x,y)  
        ppoint.SetGeometry(point)  
        vlayer.SetFeature(ppoint)
```

ogr

```
dataSource=None
```

PUNKT

Nowy

```
from osgeo import ogr,osr
```

```
driver=ogr.GetDriverByName('ESRI shapefile')  
ff='D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\punkty4.shp'  
# step1  
spatialRef=osr.SpatialReference()  
spatialRef.ImportFromEPSG(2180)  
spatialRef.MorphToESRI()  
file=open('D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\punkty4.prj','w')  
file.write(spatialRef.ExportToWkt())  
file.close()  
# step2...  
dataSource=driver.CreateDataSource(ff)  
vlayer=dataSource.CreateLayer('punkty4',geom_type=ogr.wkbPoint)  
  
dataSource=None
```

Usuwanie

```
for i in range(0,nn):  
    print i  
    if i==6:  
        vlayer.DeleteFeature(i)
```

Dodawanie

```
# Step 1  
featureDefn=vlayer.GetLayerDefn()  
feature=ogr.Feature(featureDefn)  
# Step 2  
point=ogr.Geometry(ogr.wkbPoint)  
x=439870;y=705982  
point.AddPoint(x,y)  
# Step 3  
feature.SetGeometry(point)  
feature.SetField('K1',1)  
feature.SetField('K2',3.333)  
  
vlayer.CreateFeature(feature)
```



Tworzenie geometrii obiektów:

Punkty

Polilinie

Poligony

```
line=ogr.Geometry(ogr.wkbLineString)
x=438634;y=706001
line.AddPoint(x,y)
x=439014;y=705648
line.AddPoint(x,y)
x=440140;y=705666
line.AddPoint(x,y)
feature.SetGeometry(line)
print line.GetPointCount()
feature.SetField('KK',2)

vlayer.CreateFeature(feature)
```

```
from osgeo import ogr
driver=ogr.GetDriverByName('ESRI shapefile')
ff='D:\\Linie2.shp'
dataSource=driver.Open(ff,1) # 0-czytanie,1-zmiany
```

Nowy

```
from osgeo import ogr, osr

driver=ogr.GetDriverByName('ESRI shapefile')
ff='D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\linie3.shp'
# step1
spatialRef=osr.SpatialReference()
spatialRef.ImportFromEPSG(2180)
spatialRef.MorphToESRI()
file=open('D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\linie3.prj','w')
file.write(spatialRef.ExportToWkt())
file.close()
# step2...
dataSource=driver.CreateDataSource(ff)
vlayer=dataSource.CreateLayer('linie3',geom_type=ogr.wkbLineString)
# step3: dodawanie
featureDefn=vlayer.GetLayerDefn()
feature=ogr.Feature(featureDefn)

line=ogr.Geometry(ogr.wkbLineString)
line.AddPoint(439483,705921)
line.AddPoint(439943,705944)
feature.SetGeometry(line)
feature.SetField('Id',1)
vlayer.CreateFeature(feature)
dataSource=None
```

Dostęp

```
vlayer=dataSource.GetLayer()
nn=vlayer.GetFeatureCount()

for i in range(0,nn):
    fline=vlayer.GetFeature(i)
    geometry=fline.GetGeometryRef()
    nvertex=geometry.GetPointCount()
    for j in range(0,nvertex):
        x=geometry.GetX(j)
        y=geometry.GetY(j)
        print i,j,x,y

dataSource=None
```

LINIA

Zmiana

```
ogr
for i in range(0,nn):
    if i==3:
        fline=vlayer.GetFeature(i)
        geometry=fline.GetGeometryRef()
        nvertex=geometry.GetPointCount()
        xx=[0]*nvertex;yy=[0]*nvertex
        # współrzędne wartościów linii do tablic xx i yy
        for j in range(0,nvertex):
            xx[j]=geometry.GetX(j)
            yy[j]=geometry.GetY(j)
        # modyfikacja tablic
        yy[nvertex-1]=yy[nvertex-2]

        line=ogr.Geometry(ogr.wkbLineString)
        for j in range(0,nvertex):
            line.AddPoint(xx[j],yy[j])
        fline.SetGeometry(line)
        vlayer.SetFeature(fline)
```

Dodawanie

Usuwanie

```
vlayer.DeleteFeature(3)
```

```
vlayer=dataSource.GetLayer()

featureDefn=vlayer.GetLayerDefn()
feature=ogr.Feature(featureDefn)

line=ogr.Geometry(ogr.wkbLineString)
line.AddPoint(439483,705921)
line.AddPoint(439943,705944)
feature.SetGeometry(line)
feature.SetField('Id',1)
vlayer.CreateFeature(feature)
```

```
from osgeo import ogr
```

```
driver=ogr.GetDriverByName('ESRI shapefile')  
ff='D:\\Poligony3.shp'
```

```
dataSource=driver.Open(ff,1) #0-czytanie,1-zmiany
```

Nowy

```
from osgeo import ogr, osr
```

```
driver=ogr.GetDriverByName('ESRI shapefile')  
ff='D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\Poligony6.shp'
```

```
# step1  
spatialRef=osr.SpatialReference()  
spatialRef.ImportFromEPSG(2180)  
spatialRef.MorphToESRI()  
file=open('D:\\PRACA8\\HELQGIS\\programy\\day3\\dane1\\Poligony6.prj','w')  
file.write(spatialRef.ExportToWkt())  
file.close()
```

```
# step2...  
dataSource=driver.CreateDataSource(ff)  
vlayer=dataSource.CreateLayer('Poligony6',geom_type=ogr.wkbPolygon)
```

```
poly_def=vlayer.GetLayerDefn()  
poly=ogr.Feature(poly_def)
```

```
poly_geom=ogr.Geometry(ogr.wkbPolygon)  
ring=ogr.Geometry(ogr.wkbLinearRing)
```

```
# ring0  
ring.AddPoint(439750,705890)  
ring.AddPoint(440150,705890)  
ring.AddPoint(440150,705530)  
ring.AddPoint(439750,705530)  
ring.AddPoint(439750,705890)  
poly_geom.AddGeometry(ring)  
ring.Empty()  
ring.AddPoint(439800,705790)  
ring.AddPoint(440100,705790)  
ring.AddPoint(440100,705630)  
ring.AddPoint(439800,705630)  
ring.AddPoint(439800,705790)  
poly_geom.AddGeometry(ring)
```

```
poly.SetGeometry(poly_geom)  
vlayer.CreateFeature(poly)
```

Dostęp

```
vlayer=dataSource.GetLayer()
```

```
nn=vlayer.GetFeatureCount()
```

```
print vlayer.GetGeomType()
```

```
for i in range(0,nn):
```

```
    fpol=vlayer.GetFeature(i)
```

```
    polgeometry=fpol.GetGeometryRef()
```

```
    nrings=polgeometry.GetGeometryCount()
```

```
    for j in range(0,nrings):
```

```
        ring=polgeometry.GetGeometryRef(j)
```

```
        nrvertex=ring.GetPointCount()
```

```
        for k in range(0,nrvertex):
```

```
            x=ring.GetX(k)
```

```
            y=ring.GetY(k)
```

```
            print i,j,k,x,y
```

```
dataSource=None
```

Zmiana

```
vlayer=dataSource.GetLayer()
```

```
nn=vlayer.GetFeatureCount()
```

```
for i in range(0,nn): # zmiana ob. i==1 (ring==1)
```

```
    fpoly=vlayer.GetFeature(i)
```

```
    polgeometry=fpoly.GetGeometryRef()
```

```
    nrings=polgeometry.GetGeometryCount()
```

```
    for j in range(0,nrings):
```

```
        if i==1 and j==0:
```

```
            ring0=polgeometry.GetGeometryRef(j)
```

```
        if i==1 and j==1:
```

```
            ring=polgeometry.GetGeometryRef(j)
```

```
            nrvertex=ring.GetPointCount()
```

```
            xx=[0]*nrvertex;yy=[0]*nrvertex
```

```
            for k in range(0,nrvertex):
```

```
                xx[k]=ring.GetX(k)
```

```
                yy[k]=ring.GetY(k)
```

```
            ring=ogr.Geometry(ogr.wkbLinearRing)
```

```
            for k in range(0,nrvertex-2): #bez ost. i powt. 1
```

```
                ring.AddPoint(xx[k],yy[k])
```

```
            ring.AddPoint(xx[0],yy[0])
```

```
            nrvertex=ring.GetPointCount()
```

```
            poly=ogr.Geometry(ogr.wkbPolygon)
```

```
            poly.AddGeometry(ring0)
```

```
            poly.AddGeometry(ring)
```

```
            fpoly.SetGeometry(poly)
```

```
if i==1:
```

```
    vlayer.SetFeature(fpoly)
```

Dodawanie

```
vlayer=dataSource.GetLayer()
```

```
poly_def=vlayer.GetLayerDefn()
```

```
poly=ogr.Feature(poly_def)
```

```
poly_geom=ogr.Geometry(ogr.wkbPolygon)
```

```
ring=ogr.Geometry(ogr.wkbLinearRing)
```

```
ring.AddPoint(439750,705890)
```

```
ring.AddPoint(440150,705890)
```

```
ring.AddPoint(440150,705530)
```

```
ring.AddPoint(439750,705530)
```

```
ring.AddPoint(439750,705890)
```

```
poly_geom.AddGeometry(ring)
```

```
ring.Empty()
```

```
ring.AddPoint(439800,705790)
```

```
ring.AddPoint(440100,705790)
```

```
ring.AddPoint(440100,705630)
```

```
ring.AddPoint(439800,705630)
```

```
ring.AddPoint(439800,705790)
```

```
poly_geom.AddGeometry(ring)
```

```
poly.SetGeometry(poly_geom)
```

```
vlayer.CreateFeature(poly)
```

POLIGON

Usuwanie

```
vlayer=dataSource.GetLayer()
```

```
vlayer.DeleteFeature(4)
```

```
dataSource=None
```

ogr

## FILTRACJA - Attribut filters

`layer.SetAttributeFilter("pole > 15")`

`'pole1>15 and pole2<=10'`

`'pole1>15 or pole2<=10'`

`not,xor`

`layer.SetAttributeFilter(None)`

DO REALIZACJI FILTRA TRZEBA UŻYWAĆ

`GetNextFeature()`

```
from osgeo import ogr
driver=ogr.GetDriverByName('ESRI shapefile')
ff=r'C:\JACEK2\QGISHL18\Hel18\ dzien3\dane\Punkty2.shp'

dataSource=driver.Open(ff,0)

vlayer=dataSource.GetLayer()

nn=vlayer.GetFeatureCount()
print(nn)

# filtracja atrybutowa
vlayer.SetAttributeFilter("P2>10 and XXX>4") # or, not, xor

nn=vlayer.GetFeatureCount()
print(nn)

- for i in range(0,nn):
    rek=vlayer.GetNextFeature()
    k1=rek.GetFieldAsInteger('K1')
    print(i,k1)

# kasowanie filtracji
vlayer.SetAttributeFilter(None)

nn=vlayer.GetFeatureCount()
print(nn)
dataSource=None
```



## FILTRACJA - Spatial filters

`layer.SetSpatialFilterRect(xmin,ymin,xmax,ymax)`

`layer.SetSpatialFilter(geometry)`

`layer.SetSpatialFilter(None)`

DO REALIZACJI FILTRA TRZEBA UŻYWAĆ

`GetNextFeature()`


```
# filtracja przestrzenna
xmin=439000
ymin=705500
xmax=439700
ymax=706100
vlayer.SetSpatialFilterRect(xmin,ymin,xmax,ymax)

nn=vlayer.GetFeatureCount()
print(nn)

for i in range(0,nn):
    rek=vlayer.GetNextFeature()
    k1=rek.GetFieldAsInteger('K1')
    print(i,k1)

# kasowanie filtracji
vlayer.SetAttributeFilter(None)

nn=vlayer.GetFeatureCount()
print(nn)
dataSource=None
```

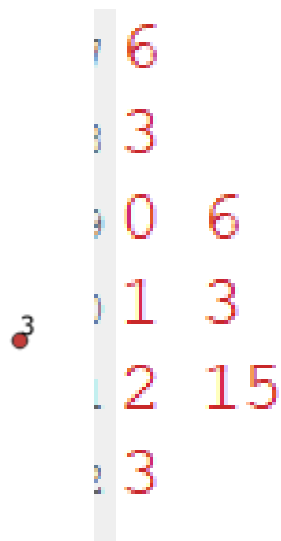
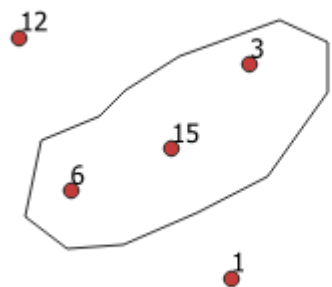


## FILTRACJA - Spatial filters

`layer.SetSpatialFilterRect(xmin,ymin,xmax,ymax)`

`layer.SetSpatialFilter(geometry)`

`layer.SetSpatialFilter(None)`



```
from osgeo import ogr
driver=ogr.GetDriverByName('ESRI:shapefile')
ff1=r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\Punkty2.shp'
ff2=r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\geometria1.shp'

dataSource1=driver.Open(ff1,0)
dataSource2=driver.Open(ff2,0)

vlayer1=dataSource1.GetLayer()
vlayer2=dataSource2.GetLayer()
fpol=vlayer2.GetFeature(0)
poligon=fpol.GetGeometryRef()

nn=vlayer1.GetFeatureCount()
print(nn)

# filtracja przestrzenna

vlayer1.SetSpatialFilter(poligon)

nn=vlayer1.GetFeatureCount()
print(nn)

for i in range(0,nn):
    rek=vlayer1.GetNextFeature()
    k1=rek.GetFieldAsInteger('K1')
    print(i,k1)

# kasowanie filtracji
vlayer1.SetSpatialFilter(None)

nn=vlayer1.GetFeatureCount()
print(nn)
dataSource=None
```

# WZAJEMNE RELACJE GEOMETRYCZNE OBIEKTÓW

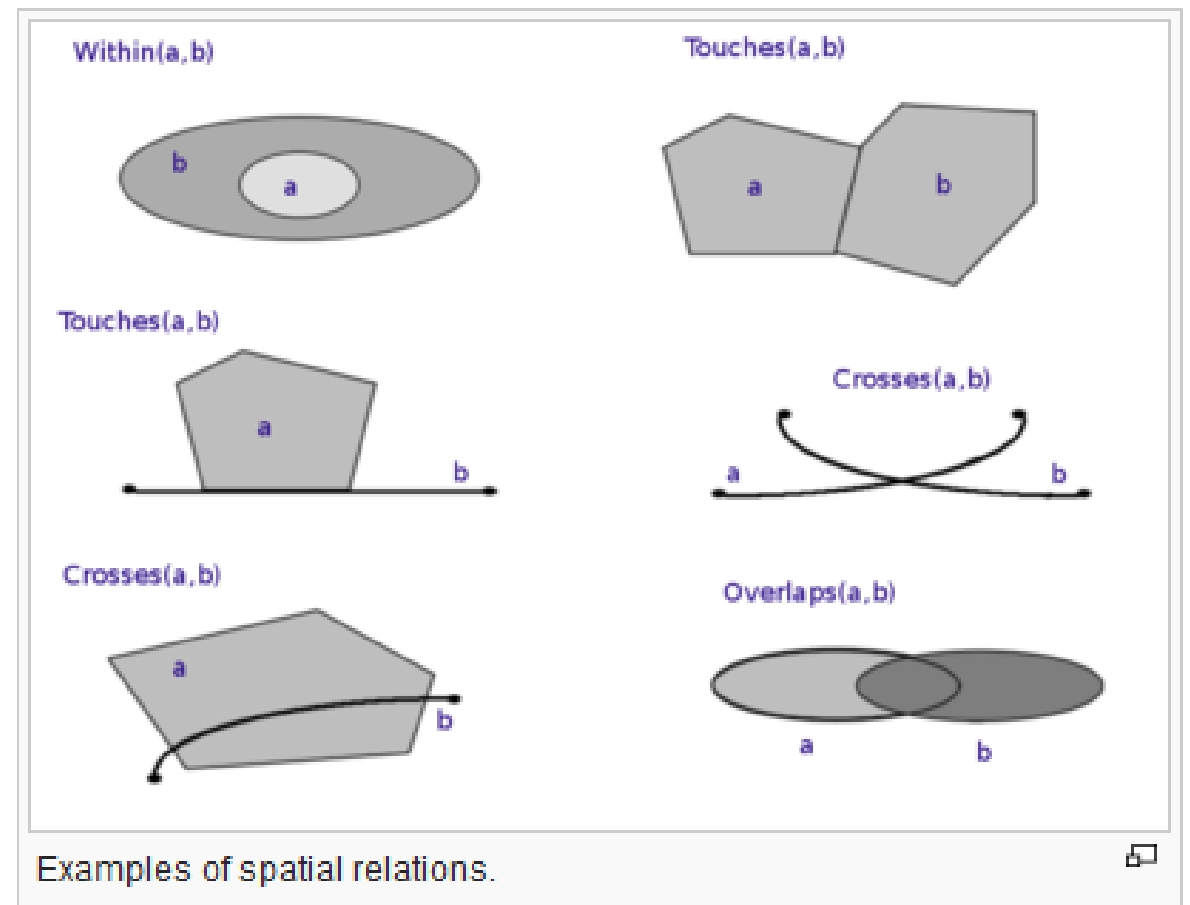
<geom1>.   (<geom2.>

**Intersect**  
**Disjoint**  
**Touches**  
**Crosses**  
**Within**  
**Contains**  
**Overlaps**

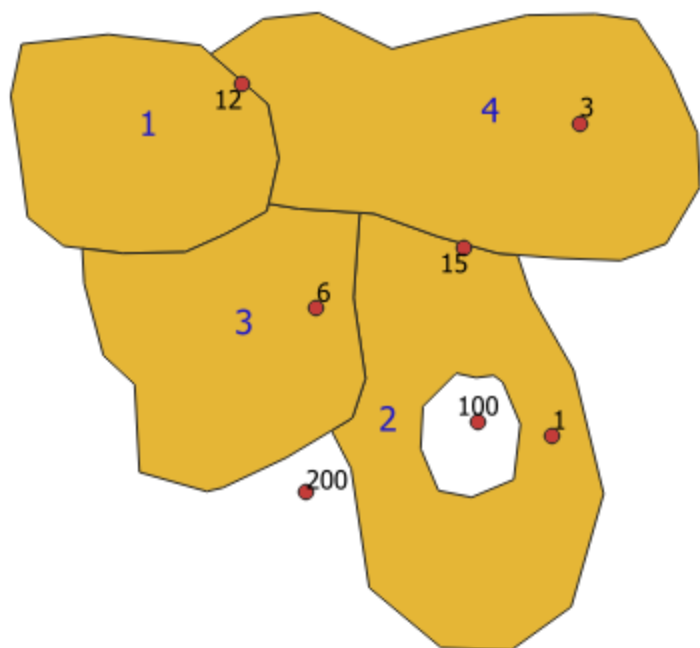
True / False

**Distance**

= xxxx jed. mapy







```

31 punkt 1
32 poligon 2
33 True
34 0.0

```

```

dataSource1=driver.Open(ff1,0)
dataSource2=driver.Open(ff2,0)

vlayer1=dataSource1.GetLayer()
vlayer2=dataSource2.GetLayer()
# punkt
fpunkt=vlayer1.GetFeature(4)
print('punkt',fpunkt.GetFieldAsInteger('K1'))
gpunkt=fpunkt.GetGeometryRef()
# poligon
fpoly=vlayer2.GetFeature(1)
print('poligon',fpoly.GetFieldAsInteger('Id'))
gpoly=fpoly.GetGeometryRef()

print(gpunkt.Intersect(gpoly))
print(gpunkt.Distance(gpoly))

dataSource=None

```

### Zadanie 3\_2

- a) Dokonać wyboru punktów (podając ich identyfikator z pola K1), które znajdują się w poligonie o ID=2.
- b) Dokonać wyboru punktów (podając ich identyfikator z pola K1), które znajdują w poligonie o ID=2 lub nie dalej niż 250 m od niego.

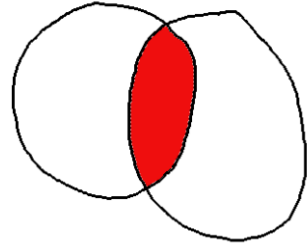
Wykorzystać:

Punkty

Poligony

## OPERACJE NA GEOMETRIACH

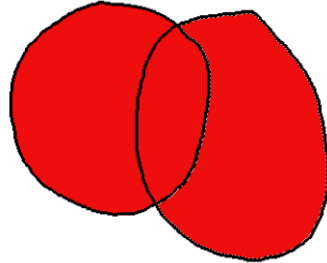
**poly3.Intersection(poly2)**



`<geom>=<geom>.Buffer(<distance>)`

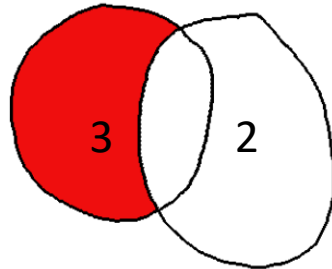
`Boolean=<geom1>.Equal(<geom2>)`

**poly3.Union(poly2)**



`lista = <geom>.GetEnvelope()`

**poly3.Difference(poly2)**

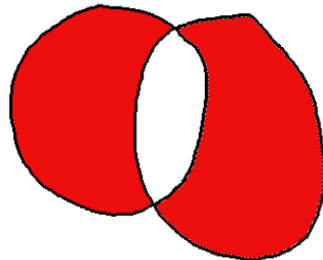


`<geom>=<geom>.Boundary()`

`<geom>=<geom>.ConvexHull()`

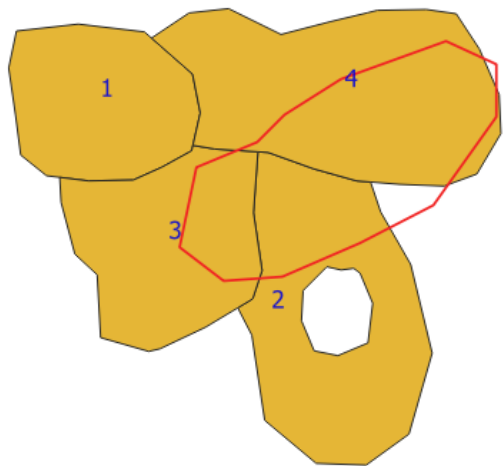
`<geom>=<geom>.UnionCascaded()`

**Poly3.SymetricDifference(poly2)**

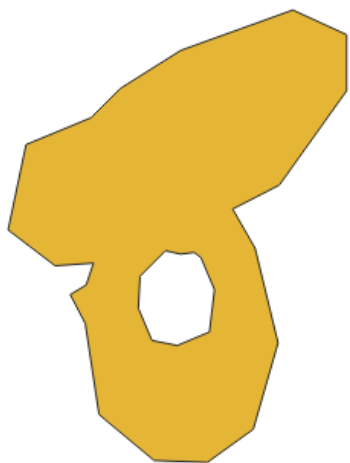


## Przykład 1

### Union 2 geometrii



poligon 2



```
from osgeo import ogr, osr
driver=ogr.GetDriverByName('ESRI shapefile')
ff2=r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\geometrial.shp'
ff1=r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\Poligony.shp'

dataSource1=driver.Open(ff1,0)
dataSource2=driver.Open(ff2,0)

vlayer1=dataSource1.GetLayer()
vlayer2=dataSource2.GetLayer()

# poligon1
fpoly1=vlayer1.GetFeature(1)
print('poligon',fpoly1.GetFieldAsInteger('Id'))
gpoly1=fpoly1.GetGeometryRef()

# poligon2
fpoly2=vlayer2.GetFeature(0)
gpoly2=fpoly2.GetGeometryRef()

gpoly12=gpoly2.Union(gpoly1)

# nowa warstwa
ff=r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\union1.shp'

# uklad wspolrzednych
spatRef=osr.SpatialReference()
spatRef.ImportFromEPSG(2180)

# tworzymy plik prj
spatRef.MorphToESRI()
```

```
# poligon2
fpoly2=vlayer2.GetFeature(0)
gpoly2=fpoly2.GetGeometryRef()

gpoly12=gpoly2.Union(gpoly1)

# nowa warstwa
ff=r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\union1.shp'

# uklad wspolrzecznych
spatRef=osr.SpatialReference()
spatRef.ImportFromEPSG(2180)

fole=open(r'C:\JACEK2\QGISHEL18\Hel18\dzien3\dane\union1.prj','w')
fole.write(spatRef.ExportToWkt())
fole.close()

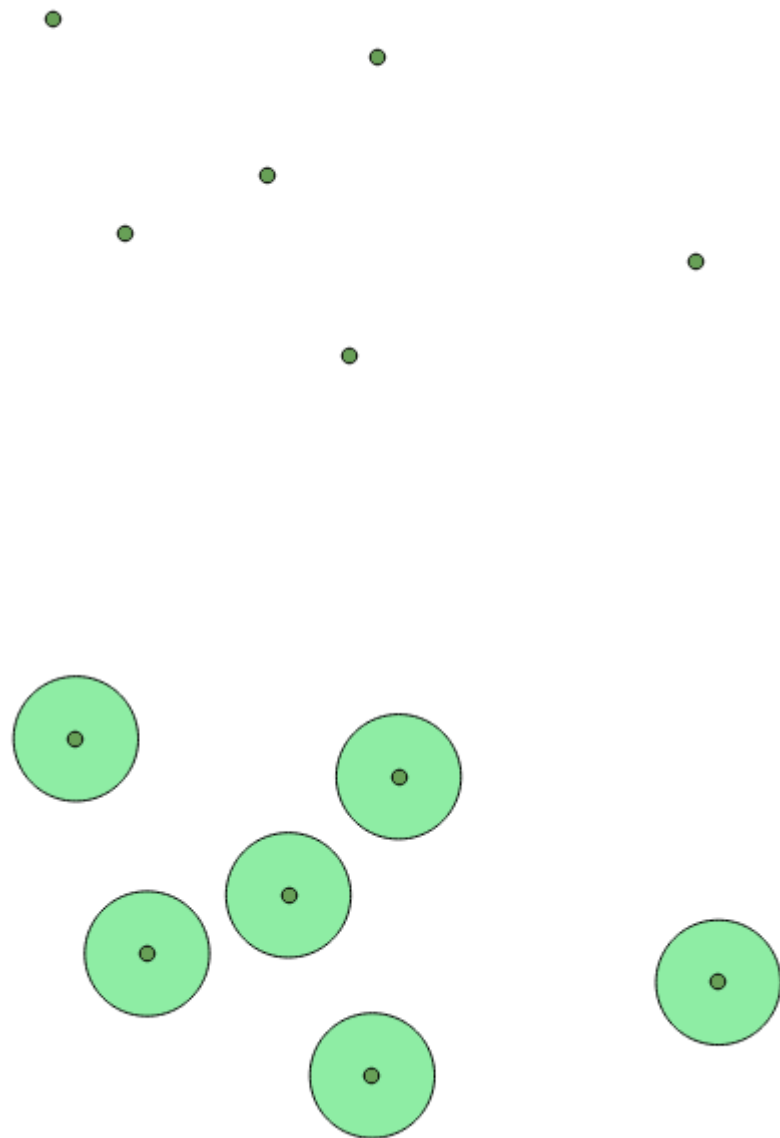
dataSource=driver.CreateDataSource(ff)
vlayer=dataSource.CreateLayer('union1',geom_type=ogr.wkbPolygon)
fieldDef=ogr.FieldDefn('IDD',ogr.OFTInteger)
vlayer.CreateField(fieldDef)

featureDefn=vlayer.GetLayerDefn()
feature=ogr.Feature(featureDefn)

feature.SetGeometry(gpoly12)
feature.SetField('IDD',7)
vlayer.CreateFeature(feature)
feature.Destroy()
dataSource=None
```

## Przykład 2

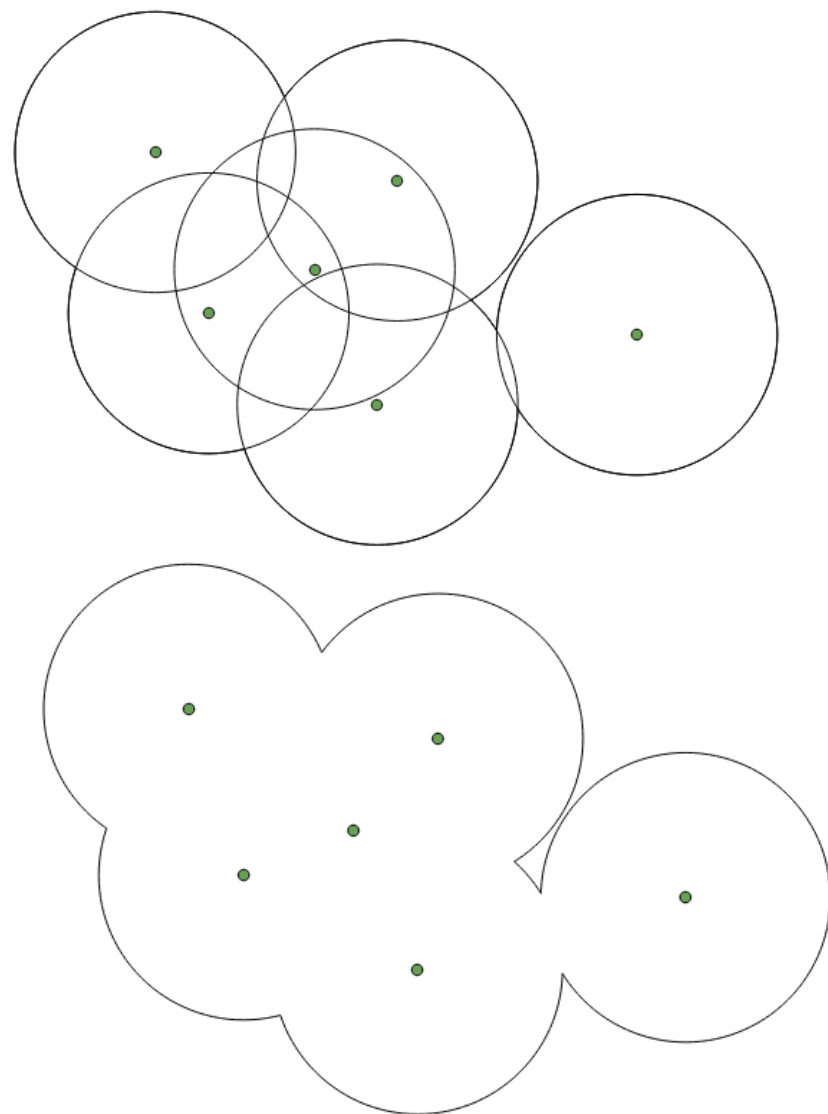
### Buforowanie – tworzenie Nowej warstwy



```
1  from osgeo import ogr,osr
2
3  driver=ogr.GetDriverByName('ESRI shapefile')
4
5  ff1='D:\\PRACA10\\HelQGIS17\\day3\\dane3\\Punkty.shp'
6  dataSource=driver.Open(ff1,0)
7  vlayer=dataSource.GetLayer()
8  nn=vlayer.GetFeatureCount()
9
10 # 1 tworzenie nowej warswy poligonow SHP
11 ff2='D:\\PRACA10\\HelQGIS17\\day3\\dane3\\Poligonybuf4.shp' # nowy
12 # 2 zdefiniowanie ukkladu wsp. i utw. pliku
13 spatialRef=osr.SpatialReference()
14 spatialRef.ImportFromEPSG(2180)
15 #   utworzenie warstwy .prj
16 file=open('D:\\PRACA10\\HelQGIS17\\day3\\dane3\\Poligonybuf4.prj','w')
17 file.write(spatialRef.ExportToWkt())
18 file.close()
19 # 3 utworzenie nowej warstwy (bez prj)
20 dataSource2=driver.CreateDataSource(ff2)
21 vlayerpp=dataSource2.CreateLayer('Poligonybuf4',geom_type=ogr.wkbPolygon)
22 featureDefn=vlayerpp.GetLayerDefn()
23
24 # 4 tworzenie buforow i zapisywanie do nowej warstwy
25 for i in range(0,nn):
26     punkt=vlayer.GetFeature(i)
27     gpunkt=punkt.GetGeometryRef()
28     gbuf=gpunkt.Buffer(220)
29     feature=ogr.Feature(featureDefn)
30     feature.SetGeometry(gbuf)
31     vlayerpp.CreateFeature(feature)
32
33 dataSource.Destroy()
34 dataSource2.Destroy()
```

## Przykład 3

### Buforowanie – tworzenie Nowej warstwy z Dissolve



```
1 from osgeo import ogr,osr
2
3 driver=ogr.GetDriverByName('ESRI shapefile')
4
5 ff1='D:\\PRACA10\\HelQGIS17\\day3\\dane3\\Punkty.shp'
6 dataSource=driver.Open(ff1,0)
7 vlayer=dataSource.GetLayer()
8 nn=vlayer.GetFeatureCount()
9
10 # 1 tworzenie nowej warstwy poligonow SHP
11 ff2='D:\\PRACA10\\HelQGIS17\\day3\\dane3\\Poligonybuf15.shp' # nowy
12 # 2 zdefiniowanie ukladu wsp. i utw. pliku
13 spatialRef=osr.SpatialReference()
14 spatialRef.ImportFromEPSG(2180)
15 # utworzenie warstwy .prj
16 file=open('D:\\PRACA10\\HelQGIS17\\day3\\dane3\\Poligonybuf15.prj','w')
17 file.write(spatialRef.ExportToWkt())
18 file.close()
19 # 3 utworzenie nowej warstwy (bez prj)
20 dataSource2=driver.CreateDataSource(ff2)
21 vlayerpp=dataSource2.CreateLayer('Poligonybuf15',geom_type=ogr.wkbMultiPolygon)
22 featureDefn=vlayerpp.GetLayerDefn()
23 geom=ogr.Geometry(ogr.wkbMultiPolygon)
24 # 4 tworzenie buforow i zapisywanie do nowej warstwy
25 for i in range(0,nn):
26     punkt=vlayer.GetFeature(i)
27     gpunkt=punkt.GetGeometryRef()
28
29     gbuf=gpunkt.Buffer(300)
30     geom.AddGeometry(gbuf)
31
32 result=geom.UnionCascaded()
33 feature=ogr.Feature(featureDefn)
34 feature.SetGeometry(result)
35 vlayerpp.CreateFeature(feature)
36
37 dataSource.Destroy()
38 dataSource2.Destroy()
```