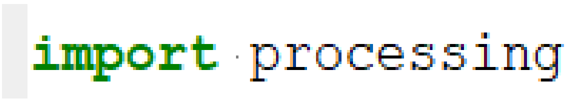
Kurs programowania w QGIS za pomocą Pythona

Wykorzystywanie w programach funkcji lub narzędzi z QGISa (biblioteka *processing*)

Hel wrzesień 2019

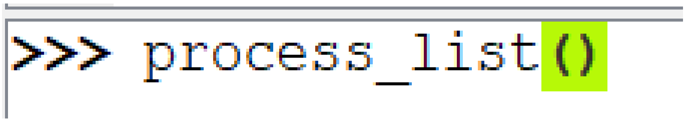
```
import processing

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



```
import processing
```

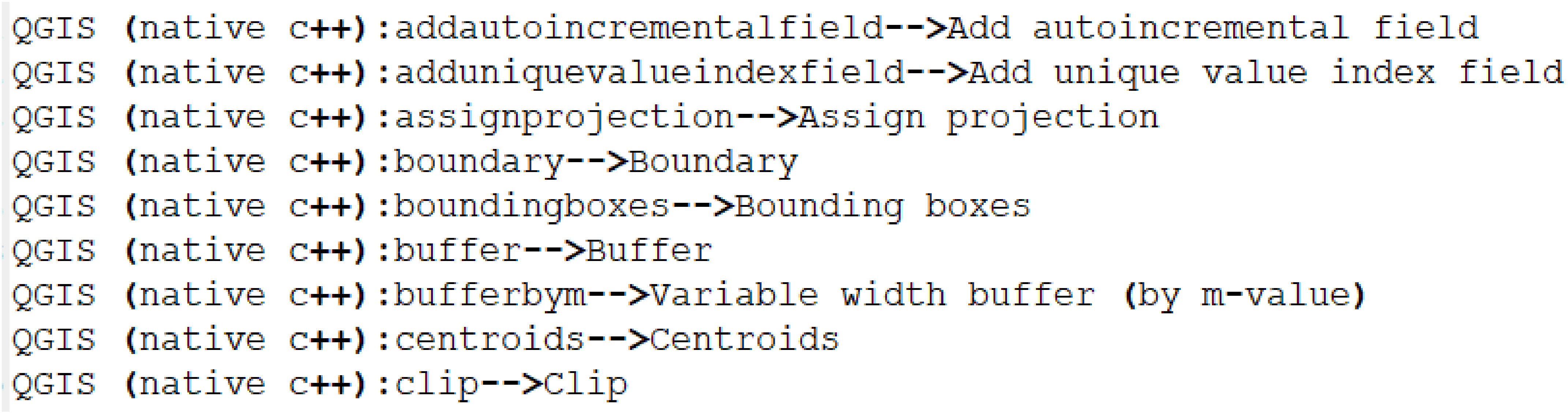
Wydruk listy algorytmów processingu

```
def process_list():
   for alg in QgsApplication.processingRegistry().algorithms():
      print("{}:{}-->{}".format(alg.provider().name(),alg.name(), alg.displayName()))
```



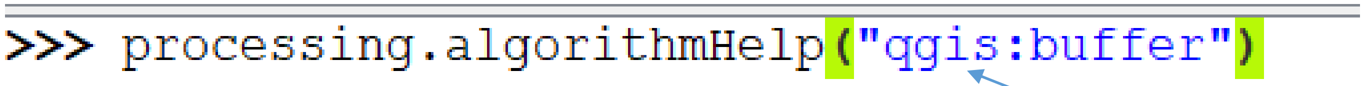
```
>>> process_list()
```

Lista wszystkich algorytmów:



```
QGIS (native c++):addautoincrementalfield-->Add autoincremental field
QGIS (native c++):adduniquevalueindexfield-->Add unique value index field
QGIS (native c++):assignprojection-->Assign projection
QGIS (native c++):boundary-->Boundary
QGIS (native c++):boundingboxes-->Bounding boxes
QGIS (native c++):buffer-->Buffer
QGIS (native c++):bufferbym-->Variable width buffer (by m-value)
QGIS (native c++):centroids-->Centroids
QGIS (native c++):clip-->Clip
```

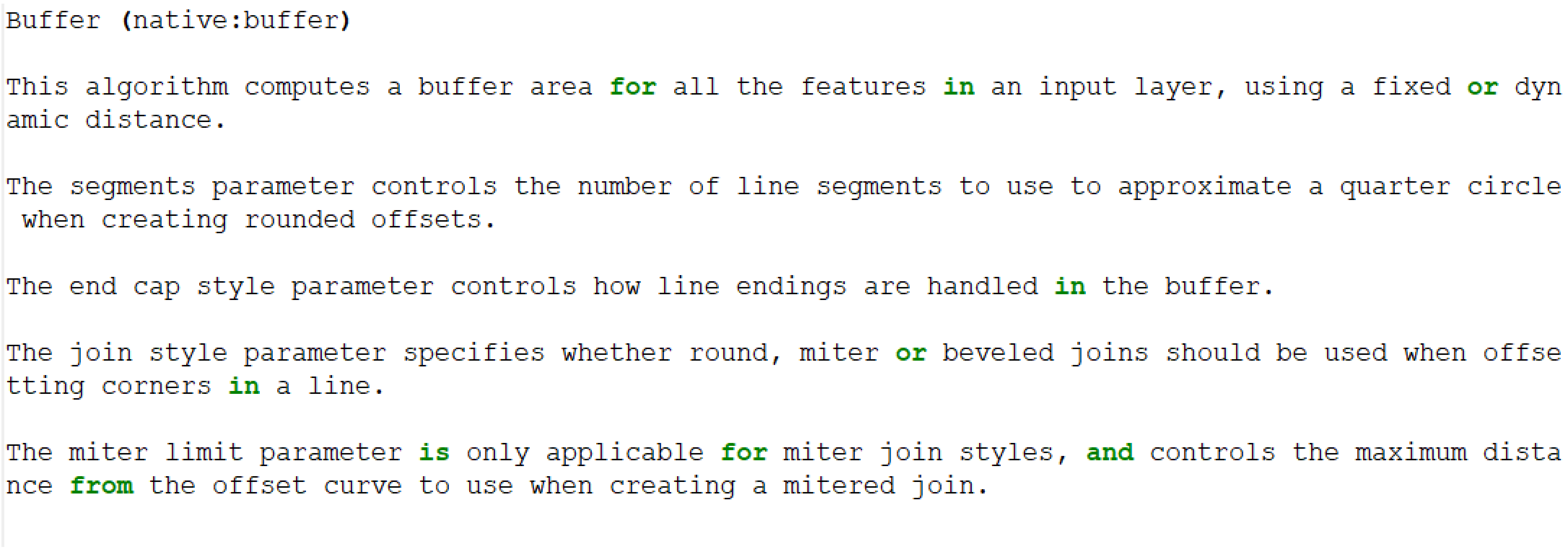
Help danego algorytmu



```
>>> processing.algorithmHelp("qgis:buffer")
```

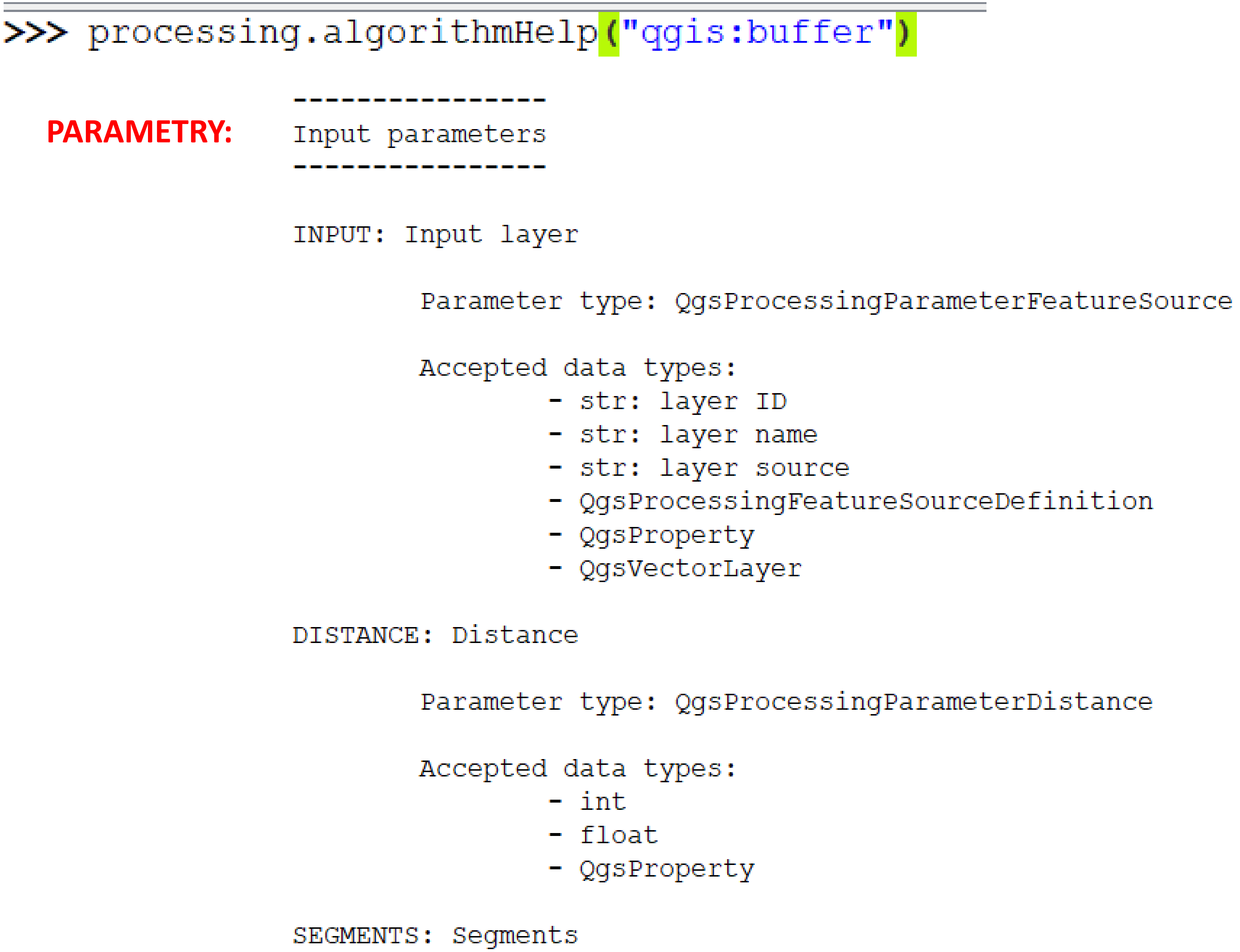
małe litery

OPIS:



```
Buffer (native:buffer)

This algorithm computes a buffer area for all the features in an input layer, using a fixed or dyn
amic distance.

The segments parameter controls the number of line segments to use to approximate a quarter circle
 when creating rounded offsets.

The end cap style parameter controls how line endings are handled in the buffer.

The join style parameter specifies whether round, miter or beveled joins should be used when offse
tting corners in a line.

The miter limit parameter is only applicable for miter join styles, and controls the maximum dista
nce from the offset curve to use when creating a mitered join.
```

Help danego algorytmu

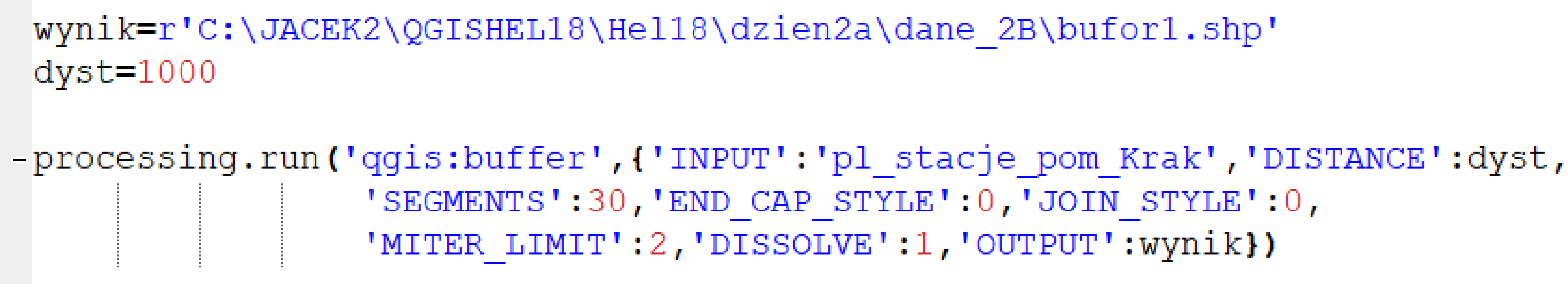


```
>>> processing.algorithmHelp("qgis:buffer")
```

PARAMETRY:

```
----------------
Input parameters
----------------

INPUT: Input layer

        Parameter type: QgsProcessingParameterFeatureSource

        Accepted data types:
                - str: layer ID
                - str: layer name
                - str: layer source
                - QgsProcessingFeatureSourceDefinition
                - QgsProperty
                - QgsVectorLayer

DISTANCE: Distance

        Parameter type: QgsProcessingParameterDistance

        Accepted data types:
                - int
                - float
                - QgsProperty

SEGMENTS: Segments
```

WYKORZYSTANIE:

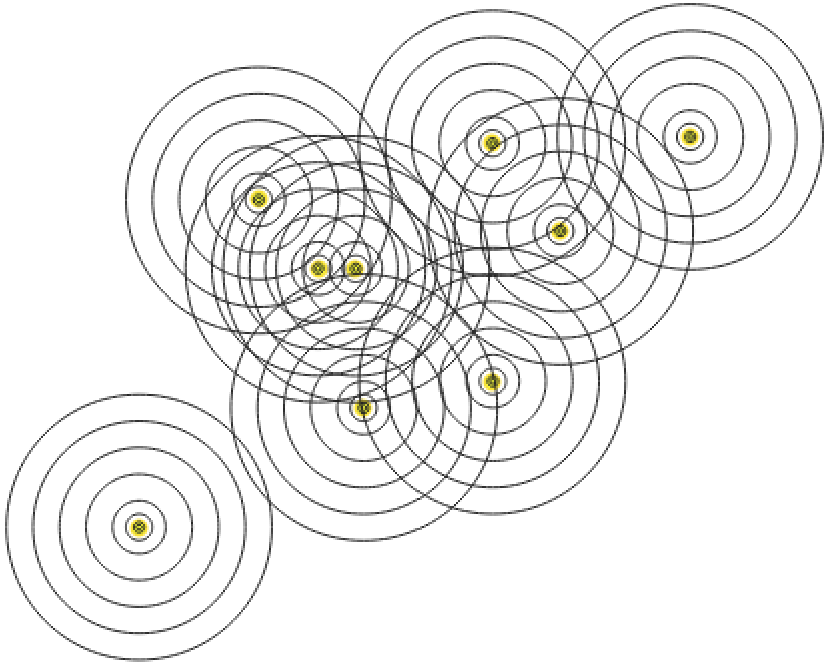


```
wynik=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\bufor1.shp'
dyst=1000

processing.run('qgis:buffer',{'INPUT':'pl_stacje_pom_Krak','DISTANCE':dyst,
                'SEGMENTS':30,'END_CAP_STYLE':0,'JOIN_STYLE':0,
                'MITER_LIMIT':2,'DISSOLVE':1,'OUTPUT':wynik})
```

Zadanie 1

Utworzyć warstwę buforów otaczających punkty pomiarowe dla dystansów : 100,200,500,1000,2000,3000,4000,5000

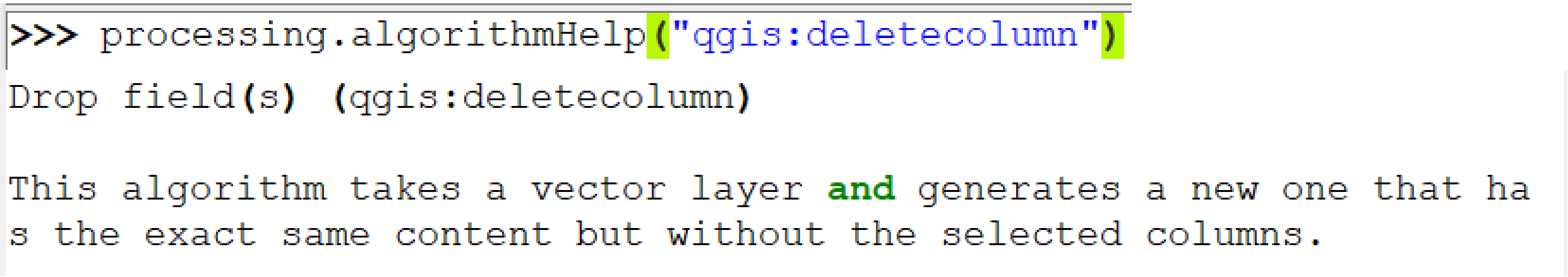
Bufory powinny mieć pola opisujące numer punktu i dystansu



a) Wykonanie ręczne dla 100 m – zaplanowanie sekwencji funkcji

Buffer
Drop Fields
Field Calculator

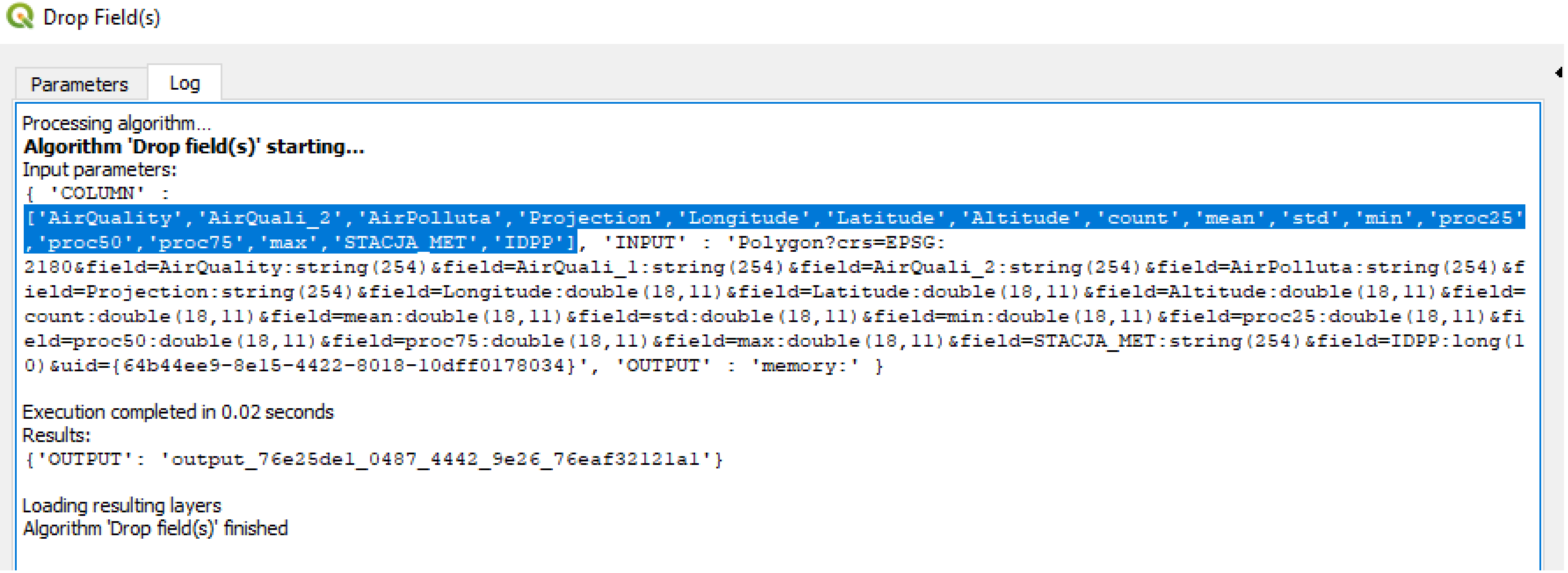
b) Wykonanie tego samego jako programu dla 100m



```
>>> processing.algorithmHelp("qgis:deletecolumn")
Drop field(s) (qgis:deletecolumn)

This algorithm takes a vector layer and generates a new one that ha
s the exact same content but without the selected columns.
```

Podpowiedź:

Parametry możemy sprawdzić w Log narzędzia processingu



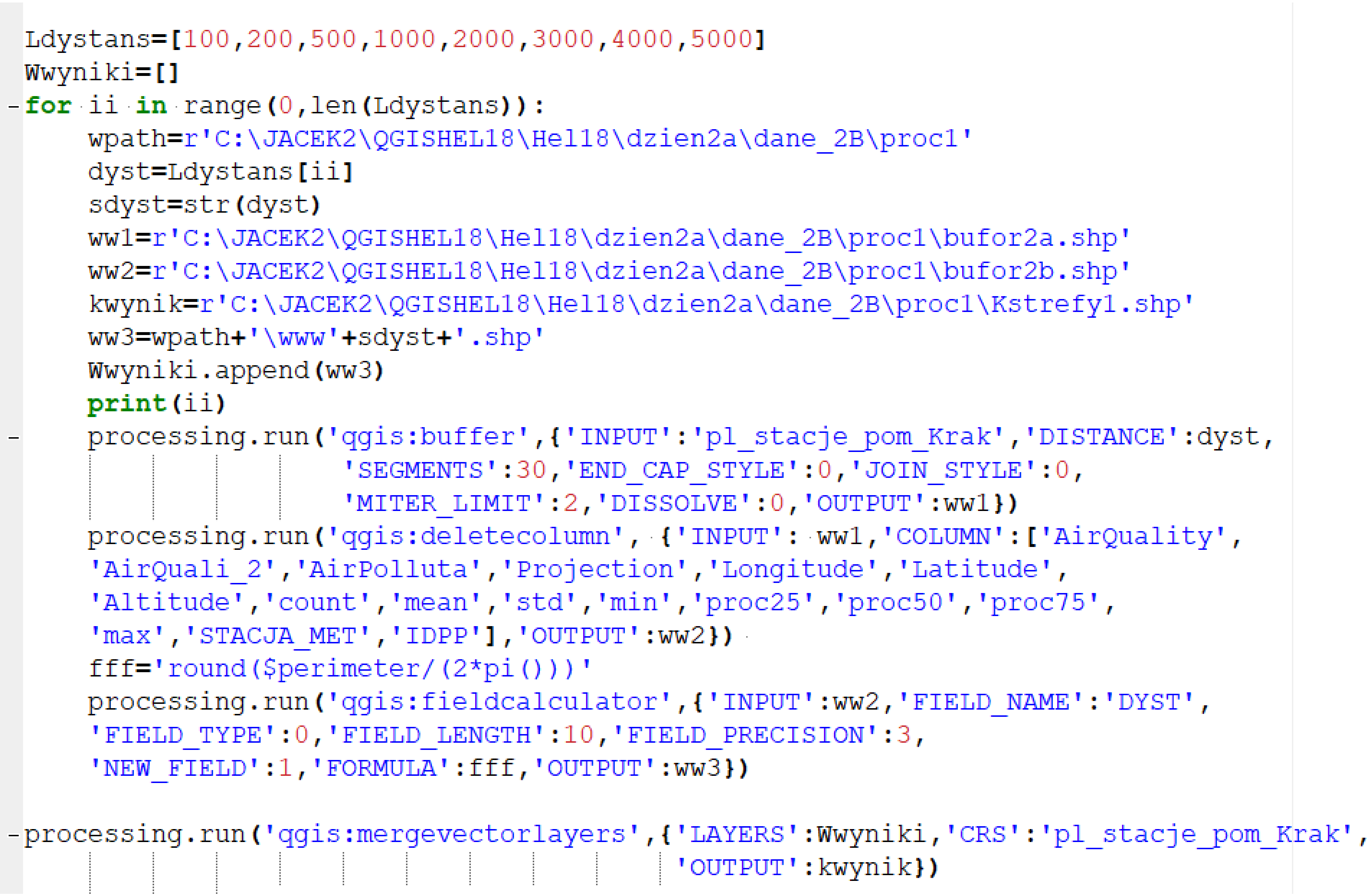
```
Processing algorithm...
Algorithm 'Drop field(s)' starting...
Input parameters:
{ 'COLUMN' :
['AirQuality','AirQuali_2','AirPolluta','Projection','Longitude','Latitude','Altitude','count','mean','std','min','proc25'
,'proc50','proc75','max','STACJA_MET','IDPP'], 'INPUT' : 'Polygon?crs=EPSG:
2180&field=AirQuality:string(254)&field=AirQuali_1:string(254)&field=AirQuali_2:string(254)&field=AirPolluta:string(254)&f
ield=Projection:string(254)&field=Longitude:double(18,11)&field=Latitude:double(18,11)&field=Altitude:double(18,11)&field=
count:double(18,11)&field=mean:double(18,11)&field=std:double(18,11)&field=min:double(18,11)&field=proc25:double(18,11)&fi
eld=proc50:double(18,11)&field=proc75:double(18,11)&field=max:double(18,11)&field=STACJA_MET:string(254)&field=IDPP:long(1
0)&uid={64b44ee9-8e15-4422-8018-10dff0178034}', 'OUTPUT' : 'memory:' }

Execution completed in 0.02 seconds
Results:
{'OUTPUT': 'output_76e25de1_0487_4442_9e26_76eaf32121a1'}

Loading resulting layers
Algorithm 'Drop field(s)' finished
```

Wyrażenie na wpisanie dystansu będzie miało postać,

```
round($perimeter/(2*pi()))
```

Program dla wszystkich i połączenie do jednego pliku

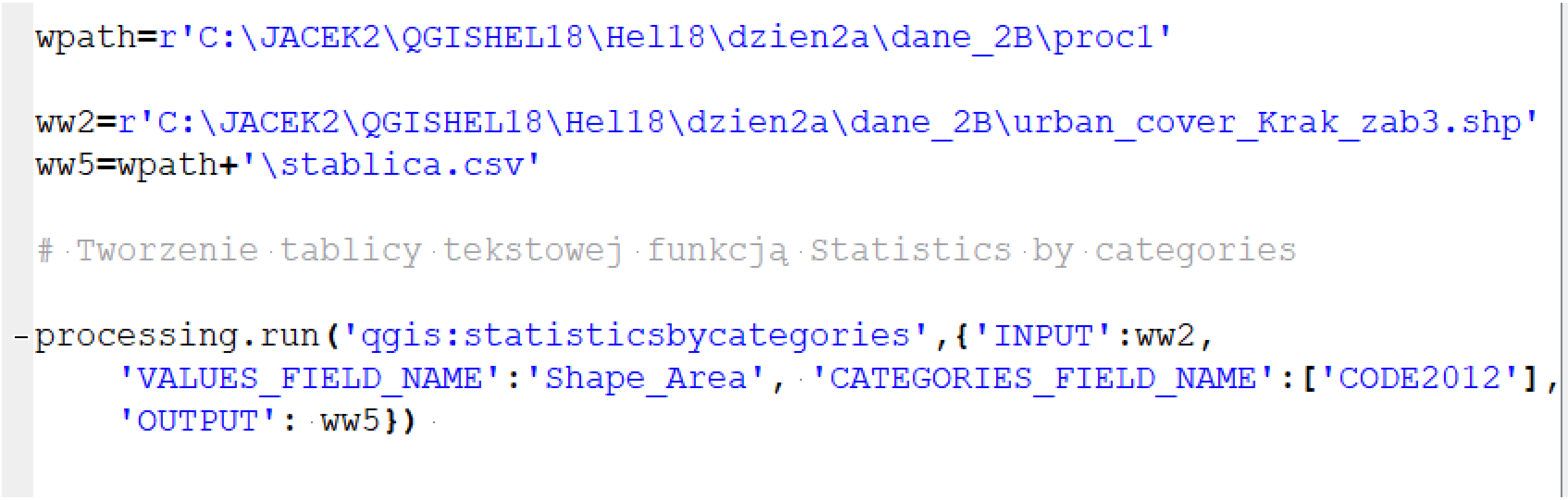
processing.algorithmHelp('qgis:mergevectorlayers')



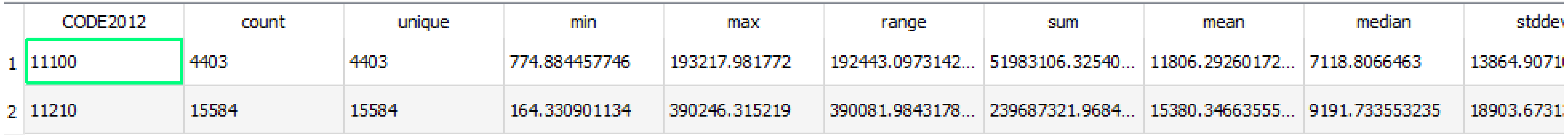
```
Ldystans=[100,200,500,1000,2000,3000,4000,5000]
Wwyniki=[]
for ii in range(0,len(Ldystans)):
    wpath=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\proc1'
    dyst=Ldystans[ii]
    sdyst=str(dyst)
    ww1=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\proc1\bufor2a.shp'
    ww2=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\proc1\bufor2b.shp'
    kwynik=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\proc1\Kstrefy1.shp'
    ww3=wpath+'\www'+sdyst+'.shp'
    Wwyniki.append(ww3)
    print(ii)
    processing.run('qgis:buffer',{'INPUT':'pl_stacje_pom_Krak','DISTANCE':dyst,
                    'SEGMENTS':30,'END_CAP_STYLE':0,'JOIN_STYLE':0,
                    'MITER_LIMIT':2,'DISSOLVE':0,'OUTPUT':ww1})
    processing.run('qgis:deletecolumn', {'INPUT': ww1,'COLUMN':['AirQuality',
    'AirQuali_2','AirPolluta','Projection','Longitude','Latitude',
    'Altitude','count','mean','std','min','proc25','proc50','proc75',
    'max','STACJA_MET','IDPP'],'OUTPUT':ww2})
    fff='round($perimeter/(2*pi()))'
    processing.run('qgis:fieldcalculator',{'INPUT':ww2,'FIELD_NAME':'DYST',
    'FIELD_TYPE':0,'FIELD_LENGTH':10,'FIELD_PRECISION':3,
    'NEW_FIELD':1,'FORMULA':fff,'OUTPUT':ww3})

processing.run('qgis:mergevectorlayers',{'LAYERS':Wwyniki,'CRS':'pl_stacje_pom_Krak',
                                         'OUTPUT':kwynik})
```

Zadanie 2

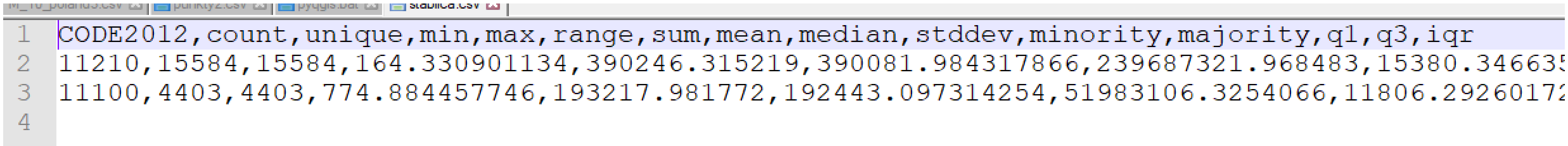
Napisz program, który wydrukuje (w konsoli) średnie powierzchnie obiektów dwóch rodzajów zabudowy w m2 (kody 11100 i 11210).



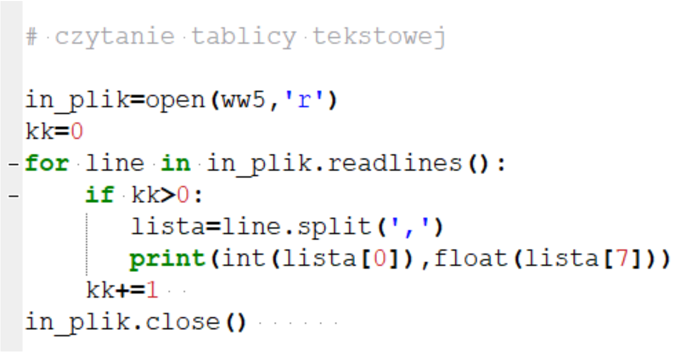
```
wpath=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\proc1'

ww2=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\urban_cover_Krak_zab3.shp'
ww5=wpath+'\stablica.csv'

# Tworzenie tablicy tekstowej funkcją Statistics by categories

processing.run('qgis:statisticsbycategories',{'INPUT':ww2,
    'VALUES_FIELD_NAME':'Shape_Area', 'CATEGORIES_FIELD_NAME':['CODE2012'],
    'OUTPUT': ww5})
```



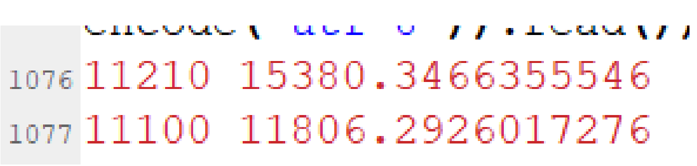
	CODE2012	count	unique	min	max	range	sum	mean	median	stddev
1	11100	4403	4403	774.884457746	193217.981772	192443.0973142...	51983106.32540...	11806.29260172...	7118.8066463	13864.90710
2	11210	15584	15584	164.330901134	390246.315219	390081.9843178...	239687321.9684...	15380.34663555...	9191.733553235	18903.6731



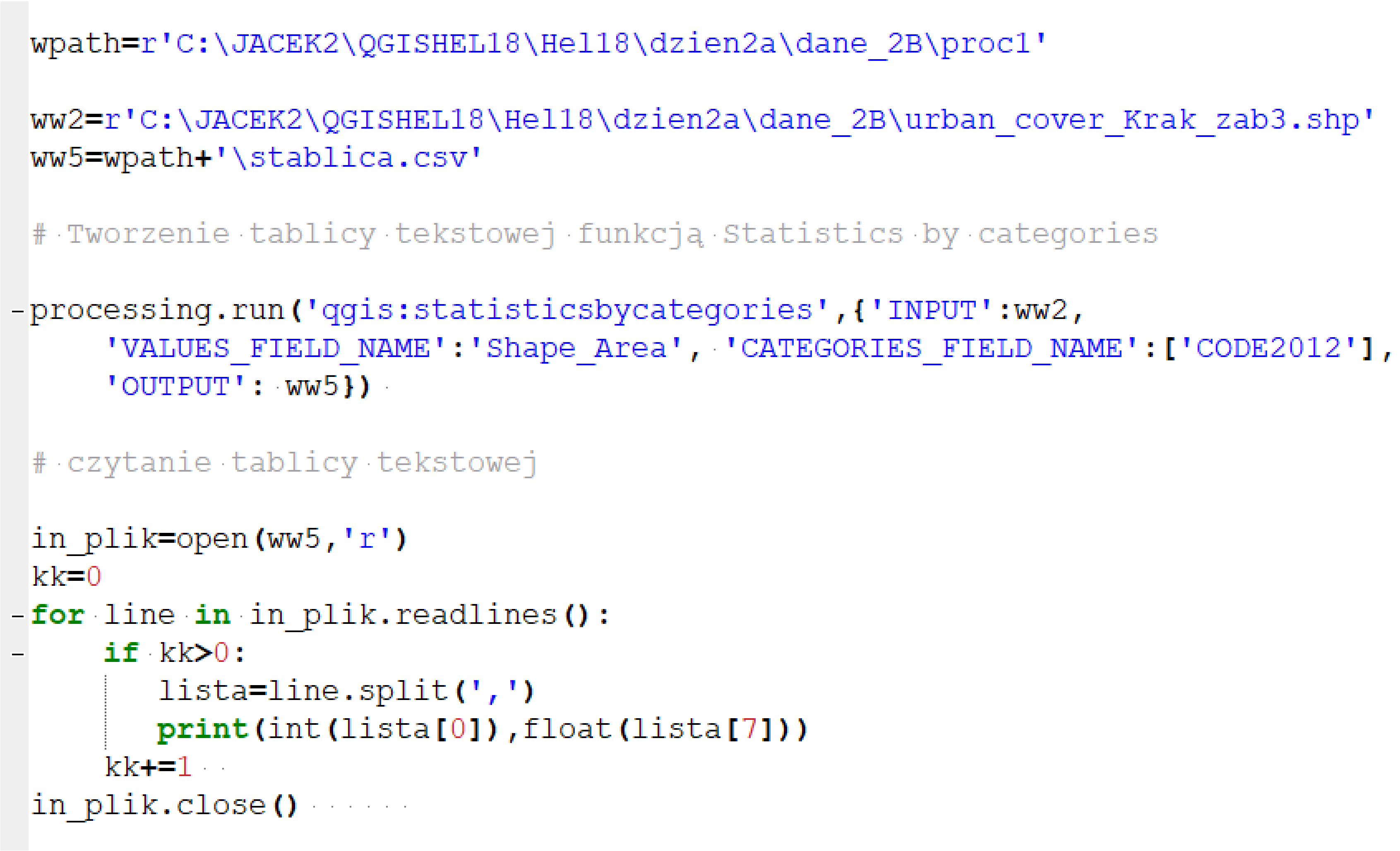
```
CODE2012,count,unique,min,max,range,sum,mean,median,stddev,minority,majority,q1,q3,iqr
11210,15584,15584,164.330901134,390246.315219,390081.984317866,239687321.968483,15380.346635
11100,4403,4403,774.884457746,193217.981772,192443.097314254,51983106.3254066,11806.29260172

```



```
# czytanie tablicy tekstowej

in_plik=open(ww5,'r')
kk=0
for line in in_plik.readlines():
    if kk>0:
        lista=line.split(',')
        print(int(lista[0]),float(lista[7]))
    kk+=1
in_plik.close()
```



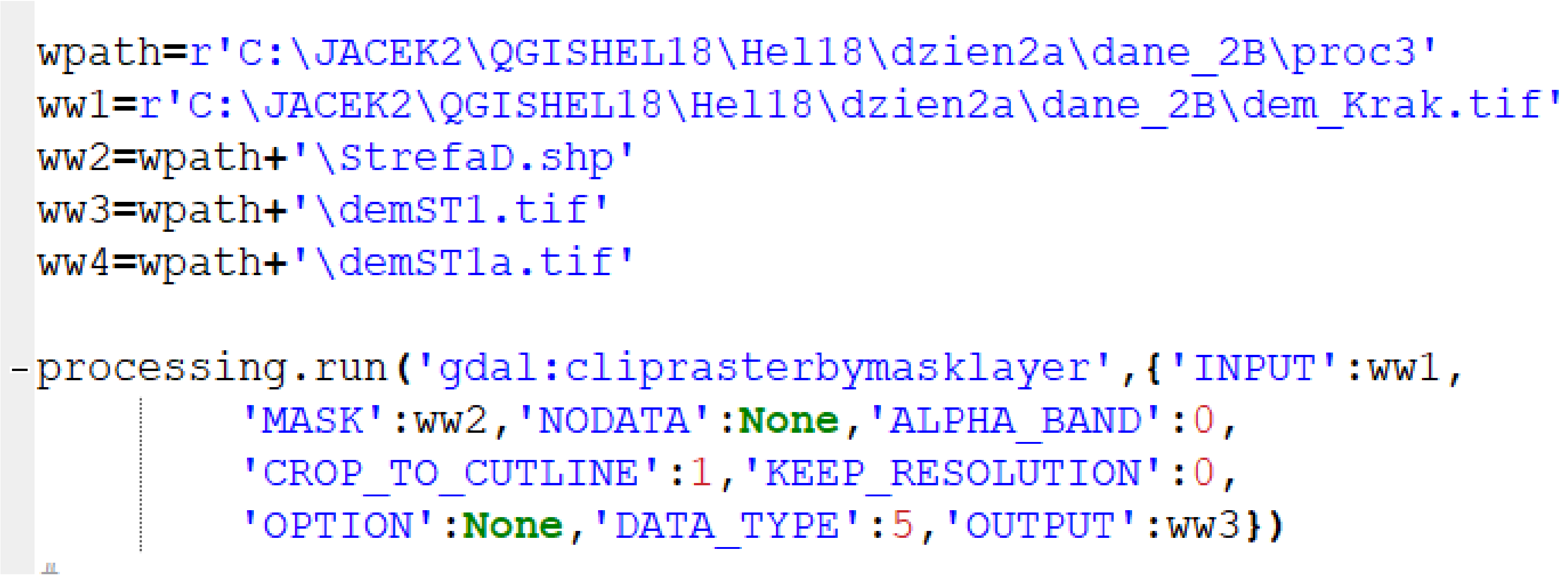
```
encode('utf-8')).read())
11210 15380.3466355546
11100 11806.2926017276
```



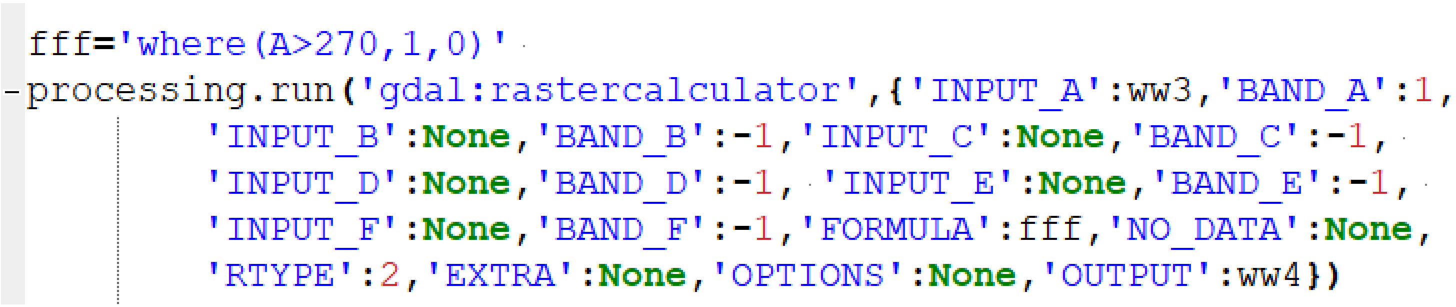
```
wpath=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\proc1'

ww2=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\urban_cover_Krak_zab3.shp'
ww5=wpath+'\stablica.csv'

# Tworzenie tablicy tekstowej funkcją Statistics by categories

processing.run('qgis:statisticsbycategories',{'INPUT':ww2,
    'VALUES_FIELD_NAME':'Shape_Area', 'CATEGORIES_FIELD_NAME':['CODE2012'],
    'OUTPUT': ww5})

# czytanie tablicy tekstowej

in_plik=open(ww5,'r')
kk=0
for line in in_plik.readlines():
    if kk>0:
        lista=line.split(',')
        print(int(lista[0]),float(lista[7]))
    kk+=1
in_plik.close()
```

Zadanie 3

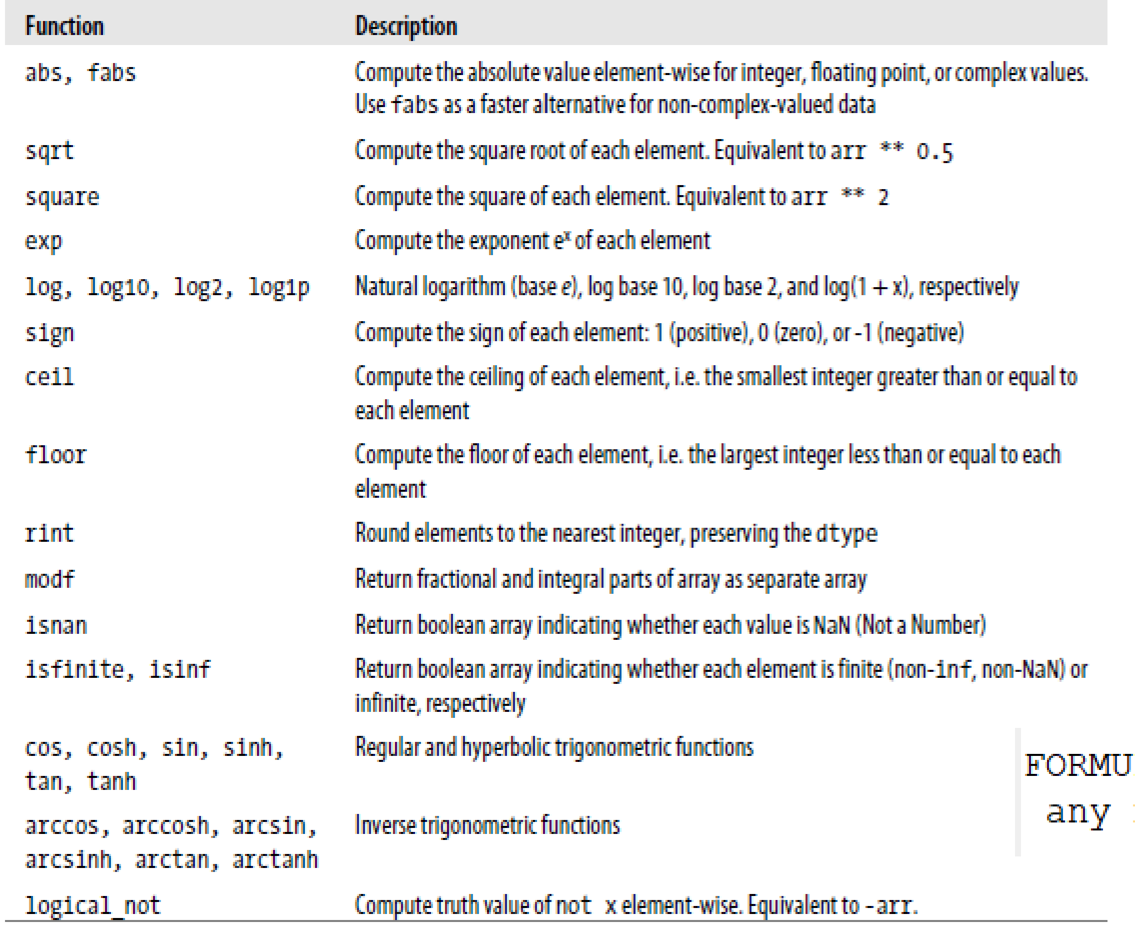
Za pomocą dowolnego bufora 5 km wytnij z warstwy dem_Krak warstwę wysokości i zreklasyfikuj ją
tak aby wartości > 270 = 1 a reszta 0



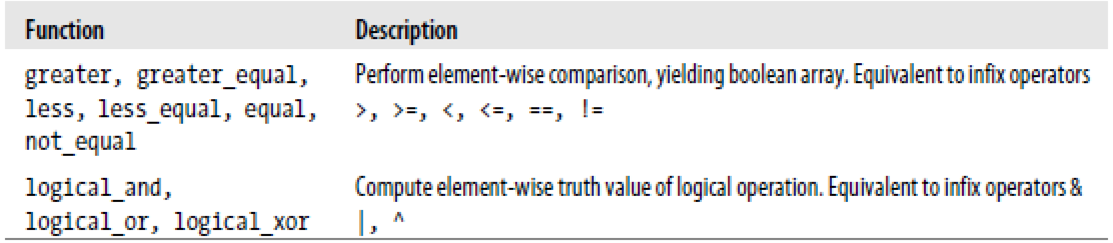
```
wpath=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\proc3'
ww1=r'C:\JACEK2\QGISHEL18\Hel18\dzien2a\dane_2B\dem_Krak.tif'
ww2=wpath+'\StrefaD.shp'
ww3=wpath+'\demST1.tif'
ww4=wpath+'\demST1a.tif'

processing.run('gdal:cliprasterbymasklayer',{'INPUT':ww1,
        'MASK':ww2,'NODATA':None,'ALPHA_BAND':0,
        'CROP_TO_CUTLINE':1,'KEEP_RESOLUTION':0,
        'OPTION':None,'DATA_TYPE':5,'OUTPUT':ww3})
```



```
fff='where(A>270,1,0)'
processing.run('gdal:rastercalculator',{'INPUT_A':ww3,'BAND_A':1,
        'INPUT_B':None,'BAND_B':-1,'INPUT_C':None,'BAND_C':-1,
        'INPUT_D':None,'BAND_D':-1, 'INPUT_E':None,'BAND_E':-1,
        'INPUT_F':None,'BAND_F':-1,'FORMULA':fff,'NO_DATA':None,
        'RTYPE':2,'EXTRA':None,'OPTIONS':None,'OUTPUT':ww4})
```



Function	Description
abs, fabs	Compute the absolute value element-wise for integer, floating point, or complex values. Use fabs as a faster alternative for non-complex-valued data
sqrt	Compute the square root of each element. Equivalent to arr ** 0.5
square	Compute the square of each element. Equivalent to arr ** 2
exp	Compute the exponent e^x of each element
log, log10, log2, log1p	Natural logarithm (base *e*), log base 10, log base 2, and log(1 + x), respectively
sign	Compute the sign of each element: 1 (positive), 0 (zero), or -1 (negative)
ceil	Compute the ceiling of each element, i.e. the smallest integer greater than or equal to each element
floor	Compute the floor of each element, i.e. the largest integer less than or equal to each element
rint	Round elements to the nearest integer, preserving the dtype
modf	Return fractional and integral parts of array as separate array
isnan	Return boolean array indicating whether each value is NaN (Not a Number)
isfinite, isinf	Return boolean array indicating whether each element is finite (non-inf, non-NaN) or infinite, respectively
cos, cosh, sin, sinh, tan, tanh	Regular and hyperbolic trigonometric functions
arccos, arccosh, arcsin, arcsinh, arctan, arctanh	Inverse trigonometric functions
logical_not	Compute truth value of not x element-wise. Equivalent to -arr.



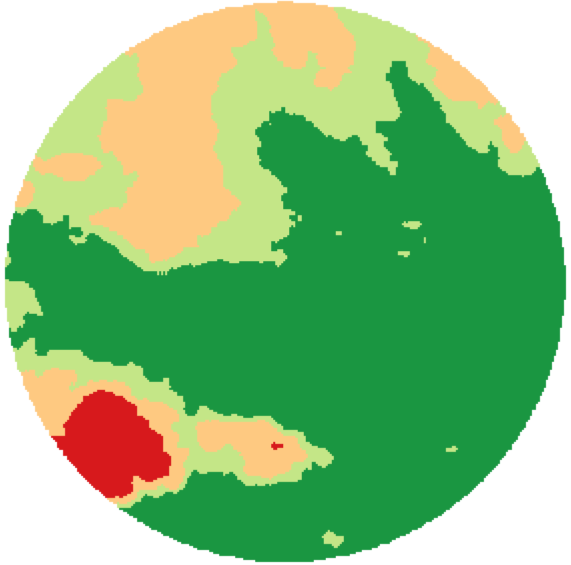
Function	Description
greater, greater_equal, less, less_equal, equal, not_equal	Perform element-wise comparison, yielding boolean array. Equivalent to infix operators >, >=, <, <=, ==, !=
logical_and, logical_or, logical_xor	Compute element-wise truth value of logical operation. Equivalent to infix operators & |, ^

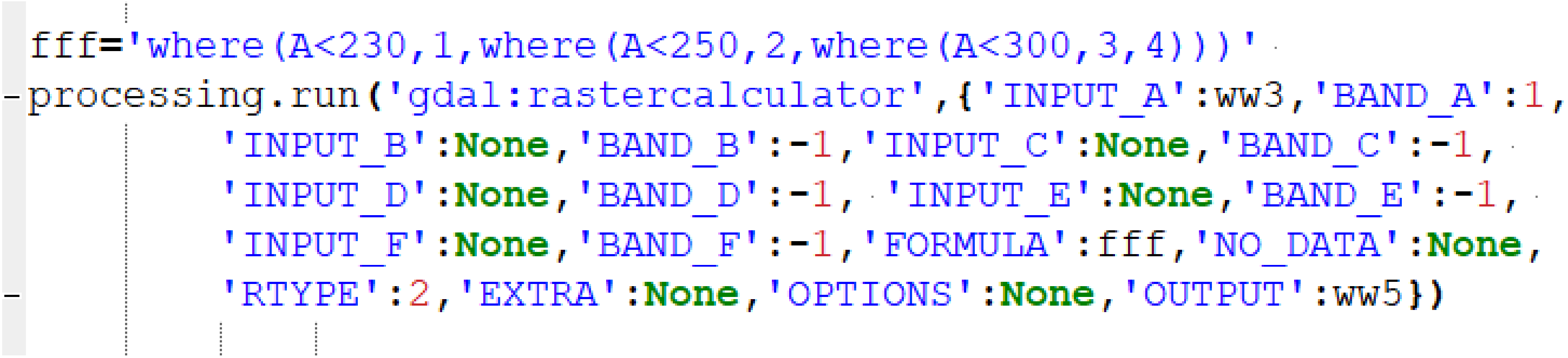
```
FORMULA: Calculation in gdalnumeric syntax using +-/* or
 any numpy array functions (i.e. logical_and())
```

Zadanie 4

Za pomocą operatora where zreklasyfikuj wycięty DEM na 4 klasy:

1 <230
2 230 – 250
3 250 - 300
4 > 300

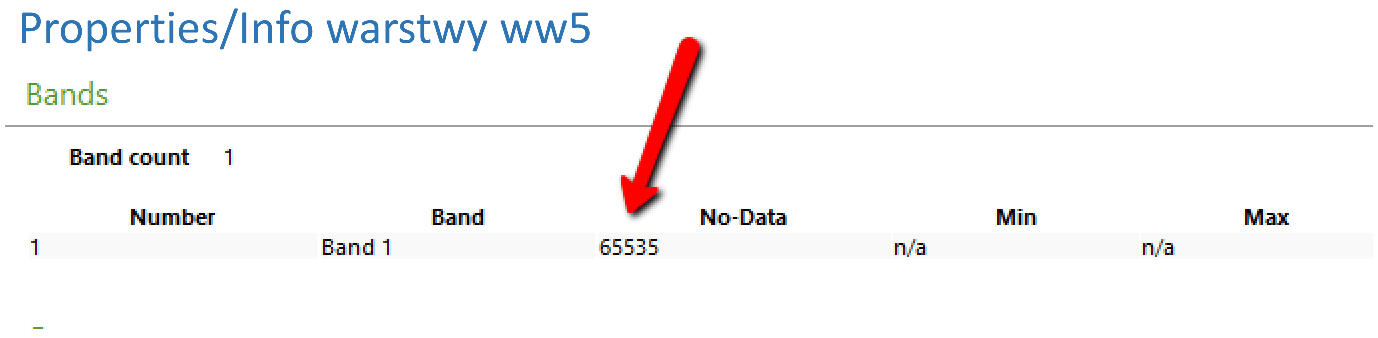




```
fff='where(A<230,1,where(A<250,2,where(A<300,3,4)))'
processing.run('gdal:rastercalculator',{'INPUT_A':ww3,'BAND_A':1,
        'INPUT_B':None,'BAND_B':-1,'INPUT_C':None,'BAND_C':-1,
        'INPUT_D':None,'BAND_D':-1, 'INPUT_E':None,'BAND_E':-1,
        'INPUT_F':None,'BAND_F':-1,'FORMULA':fff,'NO_DATA':None,
        'RTYPE':2,'EXTRA':None,'OPTIONS':None,'OUTPUT':ww5})
```

Zadanie 5

Klasę 1 i 2 w poprzednim zadaniu zmień na NoData.

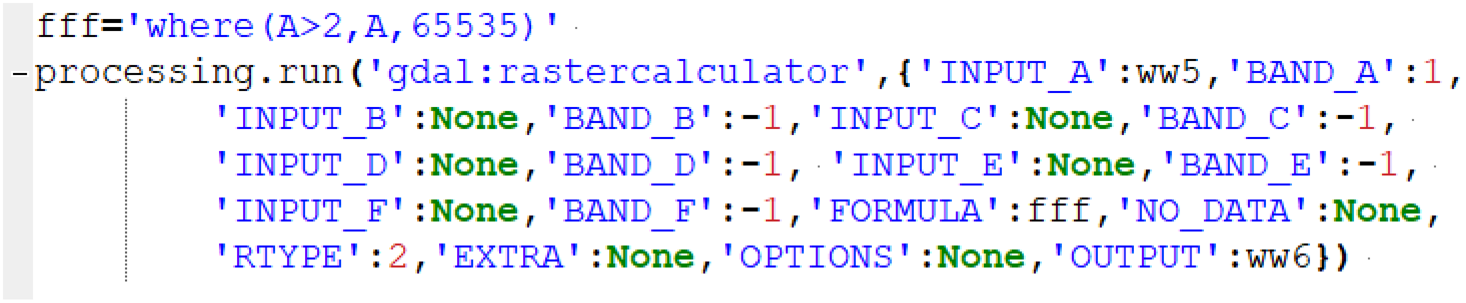


Properties/Info warstwy ww5

Bands

Band count 1

Number	Band	No-Data	Min	Max
1	Band 1	65535	n/a	n/a



```
fff='where(A>2,A,65535)'
processing.run('gdal:rastercalculator',{'INPUT_A':ww5,'BAND_A':1,
        'INPUT_B':None,'BAND_B':-1,'INPUT_C':None,'BAND_C':-1,
        'INPUT_D':None,'BAND_D':-1, 'INPUT_E':None,'BAND_E':-1,
        'INPUT_F':None,'BAND_F':-1,'FORMULA':fff,'NO_DATA':None,
        'RTYPE':2,'EXTRA':None,'OPTIONS':None,'OUTPUT':ww6})
```