Модули arcpy

Модули ArcPy

ArcPy содержит модули, использующиеся в различных областях ArcGIS.

Модуль доступа к данным (arcpy.da),

модуль картографии (arcpy.mapping),
дополнительные модули

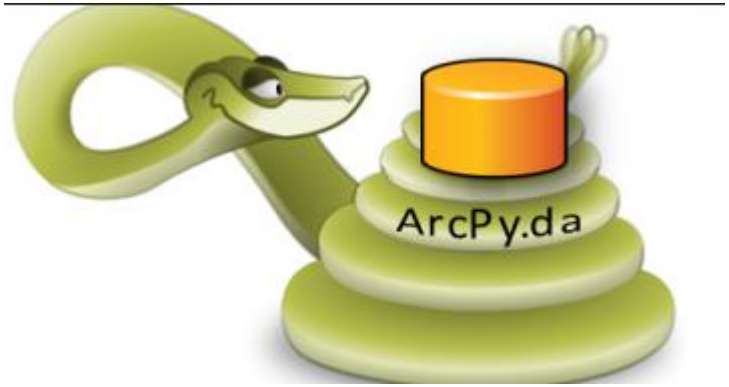
ArcGIS Spatial Analyst (arcpy.sa)

ArcGIS Network Analyst (arcpy.na).

Модуль Data Access

Модуль Data Access **arcpy.da -** модуль Python для работы с данными.

Позволяет контролировать сеанс редактирования, операции редактирования, поддерживает улучшенный курсор (включая улучшенную производительность), функции для конвертации таблиц и классов объектов в/из массивов NumPy, а также поддерживает версии, домены и подтипы рабочих процессов.



Работа с курсорами (arcpy.da)

Курсор – это объект доступа к данным, который может использоваться как для итерации по набору строк в таблице, так и для вставки новых строк в таблицу. Курсоры могут быть трех форм: поиска, вставки или обновления.

Курсор	Описание
`arcpy.InsertCursor(dataset, {spatial_reference})`	Вставляет строки
`arcpy.SearchCursor(dataset, {where_clause}, {spatial_reference}, {fields}, {sort_fields})`	Доступ только для чтения
`arcpy.UpdateCursor(dataset, {where_clause}, {spatial_reference}, {fields}, {sort_fields})`	Обновляет или удаляет строки

SearchCursor (arcpy.da)

`SearchCursor` устанавливает доступ только для чтения к записям, возвращенным из класса пространственных объектов или таблицы.

Синтаксис

SearchCursor (in_table, field_names, {where_clause}, {spatial_reference}, {explode_to_points}, {sql_clause})

Параметр	Объяснение	Тип данных
in_table	Входной класс объектов, слой, таблица или табличное представление.	String
field_names [field_names,...]	Список (или кортеж) имен полей. Для одного поля можно использовать строку вместо списка строк.	String
where_clause	Возвращается дополнительное выражение, которое ограничивает записи.	String
spatial_reference	Пространственная привязка класса объектов. Ее можно указать с объектом SpatialReference или строковым эквивалентом.	SpatialReference

SearchCursor (arcpy.da)

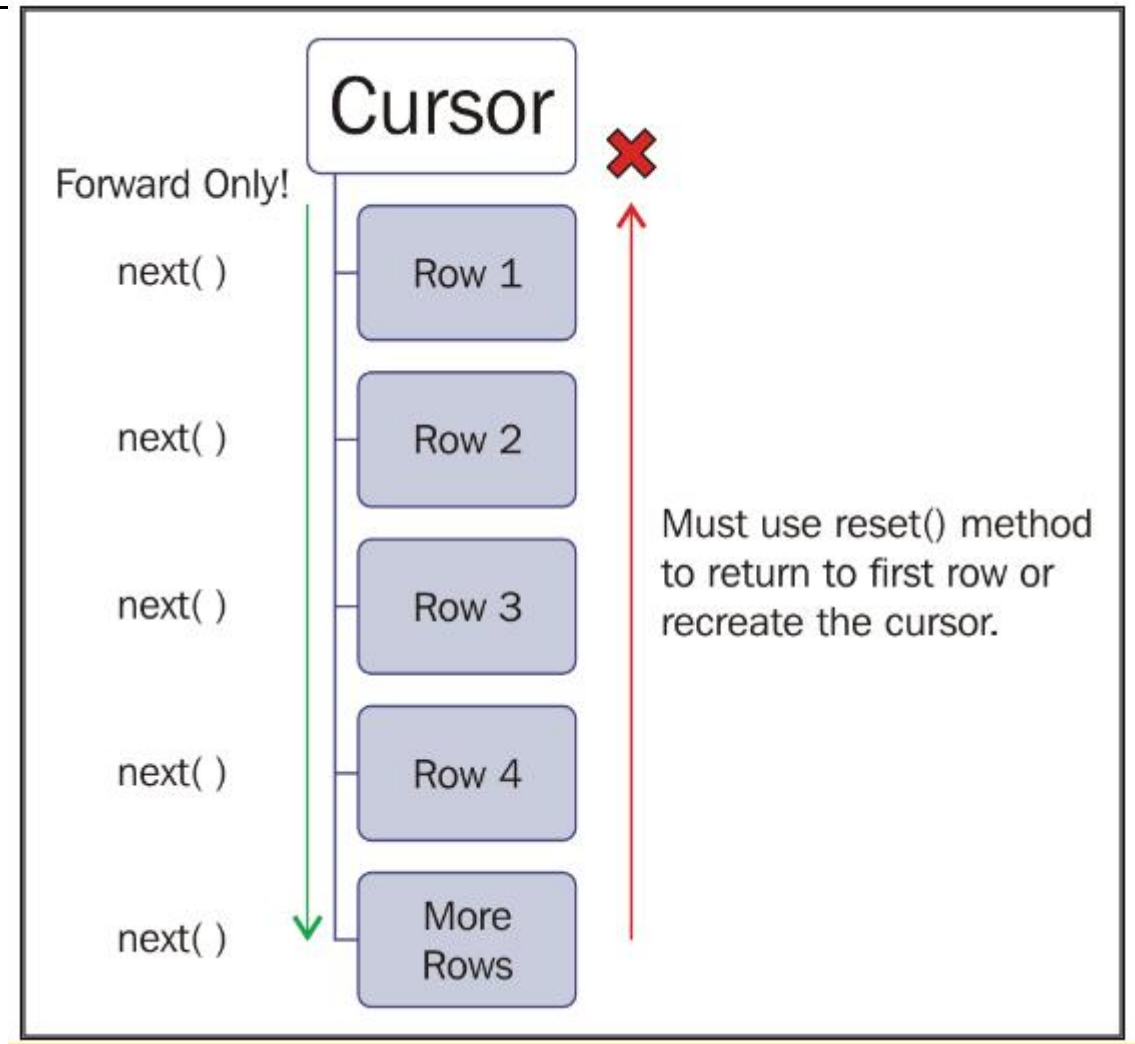
- SearchCursor
- Свойства

Свойство	Объяснение	Тип данных
fields (только чтение)	Список (или кортеж) имен полей, используемых курсором. В кортеж будут включены все поля (и токены), указанные аргументом **field_names**. Если аргумент **field_names** будет иметь значение "*", то в свойства полей будут включаться все поля, используемые курсором. Когда используется "*", значения геометрии будут выводиться в кортеже координат x и y (эквивалент токену **SHAPE@XY**).	tuple

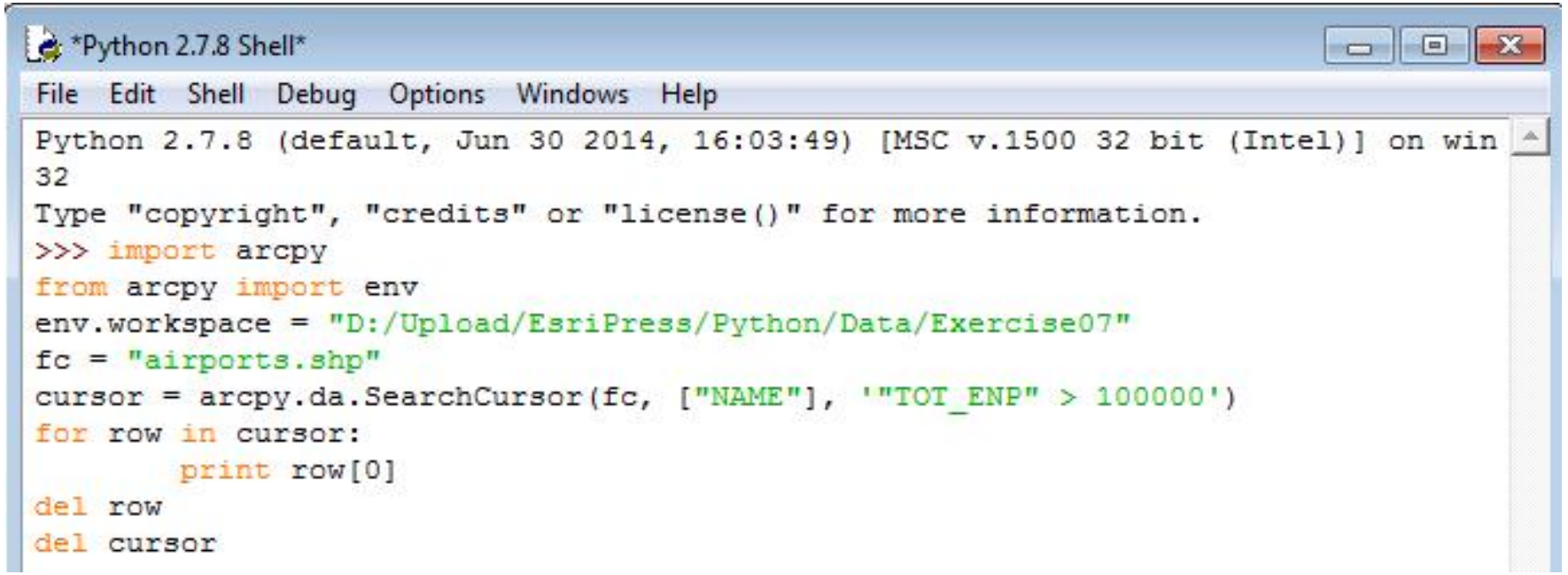
- Методы

Метод	Объяснение
next ()	Возвращает следующую строку в виде кортежа (набора взаимосвязанных величин). Порядок возвращаемых полей зависит от порядка, указанного при создании курсора.
reset ()	Сбрасывает курсор на первую строку.

Навигация по записям в курсоре



SearchCursor. Пример.



SQL-запросы (условие where)

- Имена полей заключаются в разделители.
- Разделители имени поля зависят от СУБД. При запросе к файловой базе геоданных, шейп-файлу, набору растровых данных в персональной базе геоданных имена полей заключаются в двойные кавычки:

  "area"

- При запросе к данным персональной базы геоданных, поля заключаются в квадратные скобки:

  [area]

SQL-запросы (условие where)

Строковые значения всегда заключаются в выражениях в одинарные кавычки.

SVCAREA = 'North'

Строки в выражениях чувствительны к регистру.

~~qry = "SVCAREA" = 'North‘~~

Кавычки должны быть экранированы косой чертой

qry = "SVCAREA" = \'North\'

Наконец, весь запрос должен быть заключен в кавычки:

qry = ' "SVCAREA" = \'North\' '

SQL-запросы (условие where)

- AddFieldDelimiters (dataSource, field)

Параметр	Объяснение	Тип данных
datasource	Разделители полей основываются на используемом источнике данных.	String
field	Имя поля, к которому будут добавлены разделители. Это поле может быть еще не создано.	String

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "Results/airports.shp"
delimfield = arcpy.AddFieldDelimiters(fc, "STATE")
cursor = arcpy.da.UpdateCursor(fc, ["STATE"], delimfield + " <> 'AK'")
for row in cursor:
    row[0] = "AK"
    cursor.updateRow(row)
del row
del cursor
```

SQL-запросы (условие where)

- <> не равно
- = равно
- > больше
- < меньше
- >= больше или равно
- <= меньше или равно
- BETWEEN (между)

CITY_NAME >= 'M'

SQL-запросы (условие where)

Групповые символы

При запросе к файловой базе геоданных, шейп-файлу

% - любое количество символов

_ - один символ

qry = '"SVCAREA" LIKE \'N%\'"

При запросе к персональной базе геоданных

* - любое количество символов

? - один символ

UpdateCursor (arcpy.da)

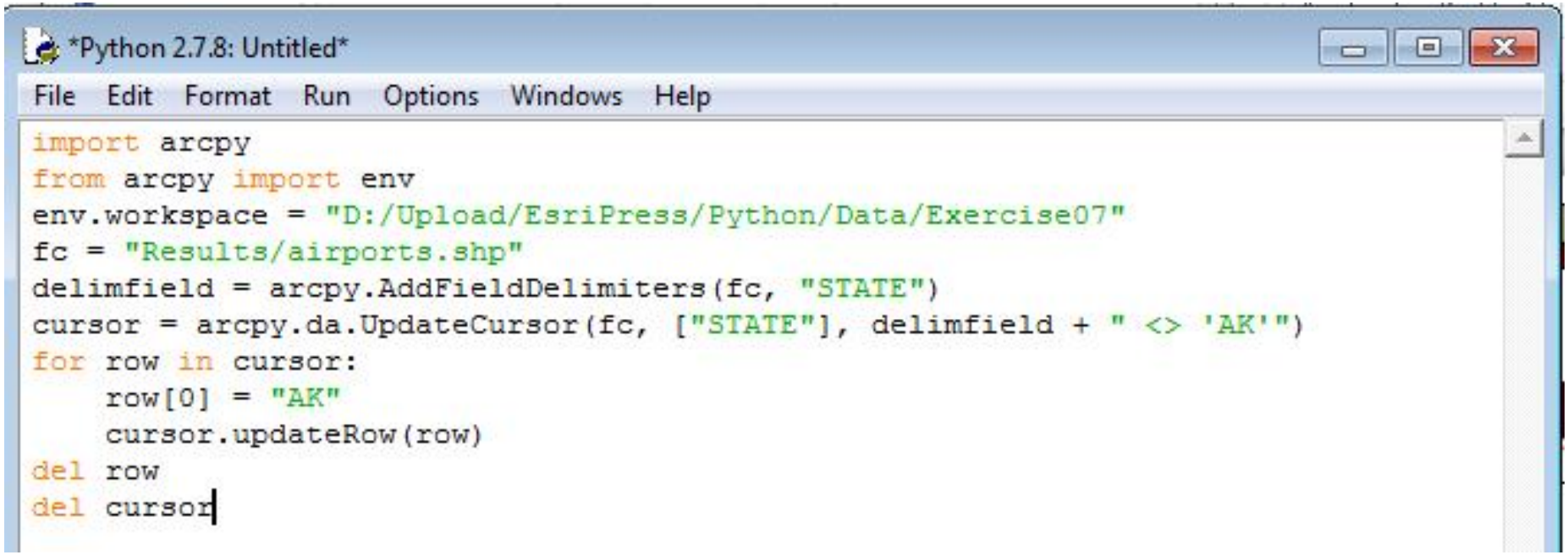
UpdateCursor устанавливает доступ для чтения и записи к записям, возвращаемым из класса объектов или таблицы.

Синтаксис

UpdateCursor (in_table, field_names, {where_clause}, {spatial_reference}, {explode_to_points}, {sql_clause})

Параметр	Объяснение	Тип данных
in_table	Входной класс объектов, слой, таблица или табличное представление.	String
field_names [field_names,...]	Список (или кортеж) имен полей. Для одного поля можно использовать строку вместо списка строк.	String
where_clause	Возвращается дополнительное выражение, которое ограничивает записи.	String
spatial_reference	Пространственную привязку класса пространственных объектов можно указать либо с объектом SpatialReference, либо строковым эквивалентом.	SpatialReference

UpdateCursor (arcpy.da). Пример

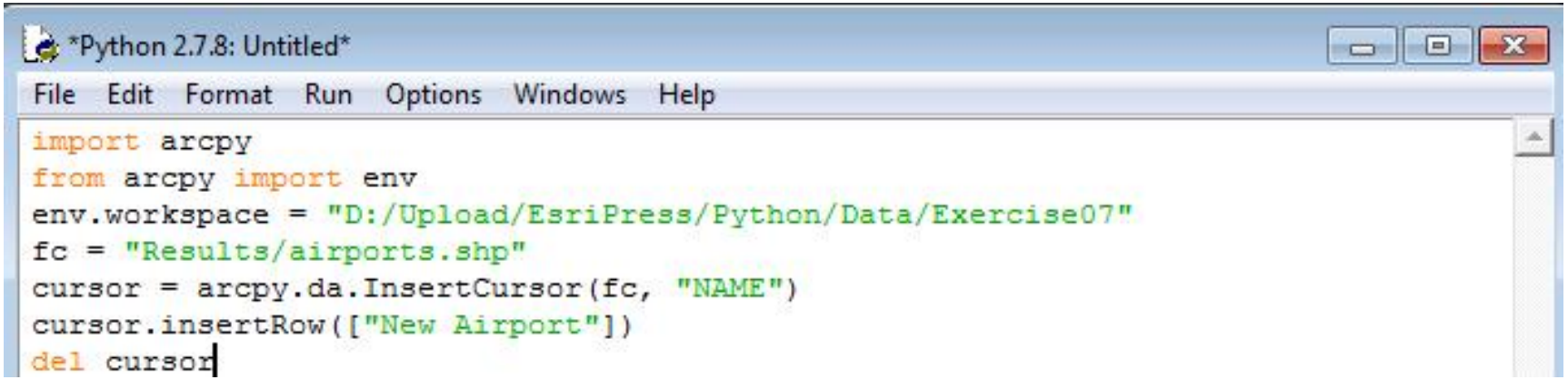


InsertCursor (arcpy.da)

InsertCursor создает курсор записи в классе объектов или таблице. InsertCursor можно использовать для добавления новых строк.

Синтаксис

InsertCursor (in_table, field_names)



InsertCursor (arcpy.da)


```
#Set current workspace
arcpy.env.workspace = "C:/Student/PYTH/Database/SanDiego.gdb"

#Create cursor on MajorAttractions feature class
curs = arcpy.InsertCursor("MajorAttractions")

# Create new row, update fields and insert row.
row = curs.newRow()
row.Name = "BLACK MOUNTAIN PARK"
row.Addr = "12115 BLACK MOUNTAIN RD"
curs.insertRow(row)
del curs, row
```

Insert/Update
и заблокированные данные

Insert/Update cursor не работают если данные заблокированы.

Класс пространственных объектов или таблица могут быть заблокированы если открыты в ArcMap или ArcCatalog.

Рекомендации.

1. Перед запуском скрипта, который создает или изменяет таблицы или классы пространственных объектов закройте все приложения ArcGIS.
2. Удалите объекты курсора когда закончите использовать их.

```
1  import arcpy
2
3  rows = arcpy.UpdateCursor("c:/data/base.gdb/roads")
4  for row in rows:
5      [redacted]
6      [redacted]
7
8  # Delete cursor and row objects to remove locks on the data
9  del row
10 del rows
```

Работа с текстовыми файлами

Текстовые и бинарные файлы в Python – это две большие разницы.

- **Как открыть файл**

```
open (name, {mode}, {buffering})
```

```
>>> f=open("C:/Data/sample.txt", "w")
```

Режимы (mode) функции **open**

"r"	открывает для чтения (по умолчанию)
"w""	открывает для записи
"+"	открывает для обновления (чтения или записи)
"a"	открывает файл для добавления информации в файл
"rb"	открывает файл для чтения в двоичном формате
"wb"	Открывает файл для записи в двоичном формате

Методы работы с файлами: чтение

- Метод `read()`

```
>>> f=open("D:/Upload/some.txt", "r")
>>> print f.read()
ArcGIS
MapInfo
QGIS
gvSIG
```

```
>>> f=open("D:/Upload/some.txt", "r")
>>> print f.read(5)
ArcGI
```

Читает не более чем 5 символов из файла

Методы работы с файлами: чтение

- Метод `readline` () читает одну строчку

```
>>> f=open("D:/Upload/some.txt", "r")
>>> print f.readline()
ArcGIS

>>> print f.readline()
 MapInfo
```

- `readlines()` читает строки и помещает их в список.

```
>>> f=open("D:/Upload/some.txt", "r")
>>> f.readlines()
['ArcGIS \n', ' MapInfo \n', ' QGIS \n', ' gvSIG']
```

Методы работы с файлами: запись

- Метод `write()` записывает строку в открытый файл.

  ! не добавляет символ переноса строки ('\n') в конец файла.

  Синтаксис метода write() → `file.write(string)`

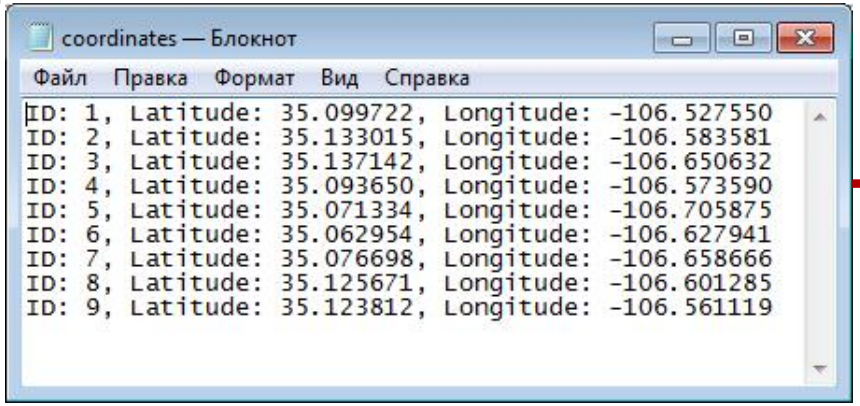
```
>>> f=open("D:/Upload/GIS.txt", "w")
>>> f.write("ArcGIS\nMapinfo\nQGIS\ngvSIG\nПанорама")
>>> f.close()
```

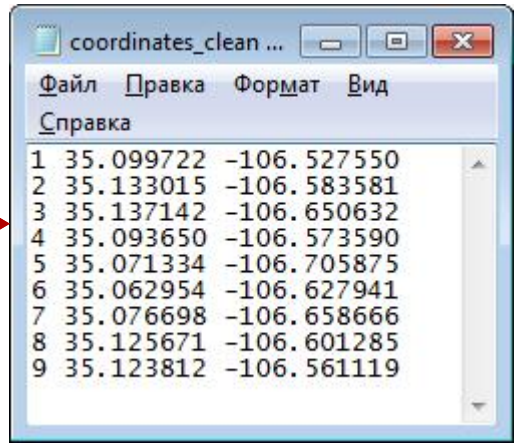
Чтение файла с помощью цикла for

```
>>> f = file("somefile.txt",'r')
>>> for line in f:
...     print(line)
...
Here's a line.

Here's another line.
```

```
inp =open("D:/Upload/EsriPress/Python/Data/Exercise07/coordinates.txt", "r")
output = open("D:/Upload/EsriPress/Python/Data/Exercise07/coordinates_clean.txt", "w")
for line in inp:
        str = line.replace("ID: ","")
        str = str.replace(", Latitude:", "")
        str = str.replace(", Longitude:", "")
        output.write(str)
inp.close()
output.close()
```





Инструкция With…. as

Еще один способ открыть и прочитать файл (нет необходимости файл закрывать)

```
>>> with open("D:/Upload/some_text.txt“, “w") as opened_file:
                opened_file.write("Minsk")
```

Эквивалентен

```
>>> file = open("D:/Upload/some_text.txt“, “w")
>>> try:
        opened_file("Minsk")
finally:
        file.close()
```

Инструкция With…. as

arcpy.da.Cursors поддерживают инструкцию `with`

```
With arcpy.da.SearchCursor(table,
field) as cursor:
    for row in cursor:
        print row[0]
```

Работа с геометрией

- Классы пространственных объектов имеют поле, хранящиеся как тип *Geometry* (поле Shape)
- Поле Geometry возвращает объект `Geometry`
- Объекты `Geometry` имеют **свойства**

  (area, length, isMultipart, partcount, pointcount….)

  и методы

  (contains (), clip (), crosses ()…)

```
import arcpy
from arcpy import env
env.workspace="D:/Upload/Lab1_data"
cursor = arcpy.da.SearchCursor("Brest_railway.shp",["SHAPE@"])
length = 0
for row in cursor:
    length+=row[0].length
del cursor
print length
```

Работа с геометрией

- Иерархия
  - Класс пространственных объектов состоит из объектов
  - Объекты состоят из частей
  - Часть создается точками
- Иерархия в терминах Python
  - Точка = [X, Y]
  - Часть = [[Point, Point, Point…]]
  - Составной полигон = [[Point, Point, Point…], [Point, Point, Point…]]
  - Полигон с отверстием = [[Point, Point, Point, ,Point, Point, Point…]]

Использование маркеров геометрии

Токен	Описание
SHAPE@	Объект geometry (геометрия) для пространственного объекта.
SHAPE@XY	Кортеж x, y координат центроида объекта.
SHAPE@TRUECENTROID	Кортеж x, y координат истинного центроида объекта.
SHAPE@X	Значение двойной точности координаты x объекта.
SHAPE@Y	Значение двойной точности координаты y объекта.
SHAPE@Z	Значение двойной точности координаты z объекта.
SHAPE@M	Значение двойной точности m для объекта.
SHAPE@AREA	Значение двойной точности для площади объекта.
SHAPE@LENGTH	Значение двойной точности для длины объекта.

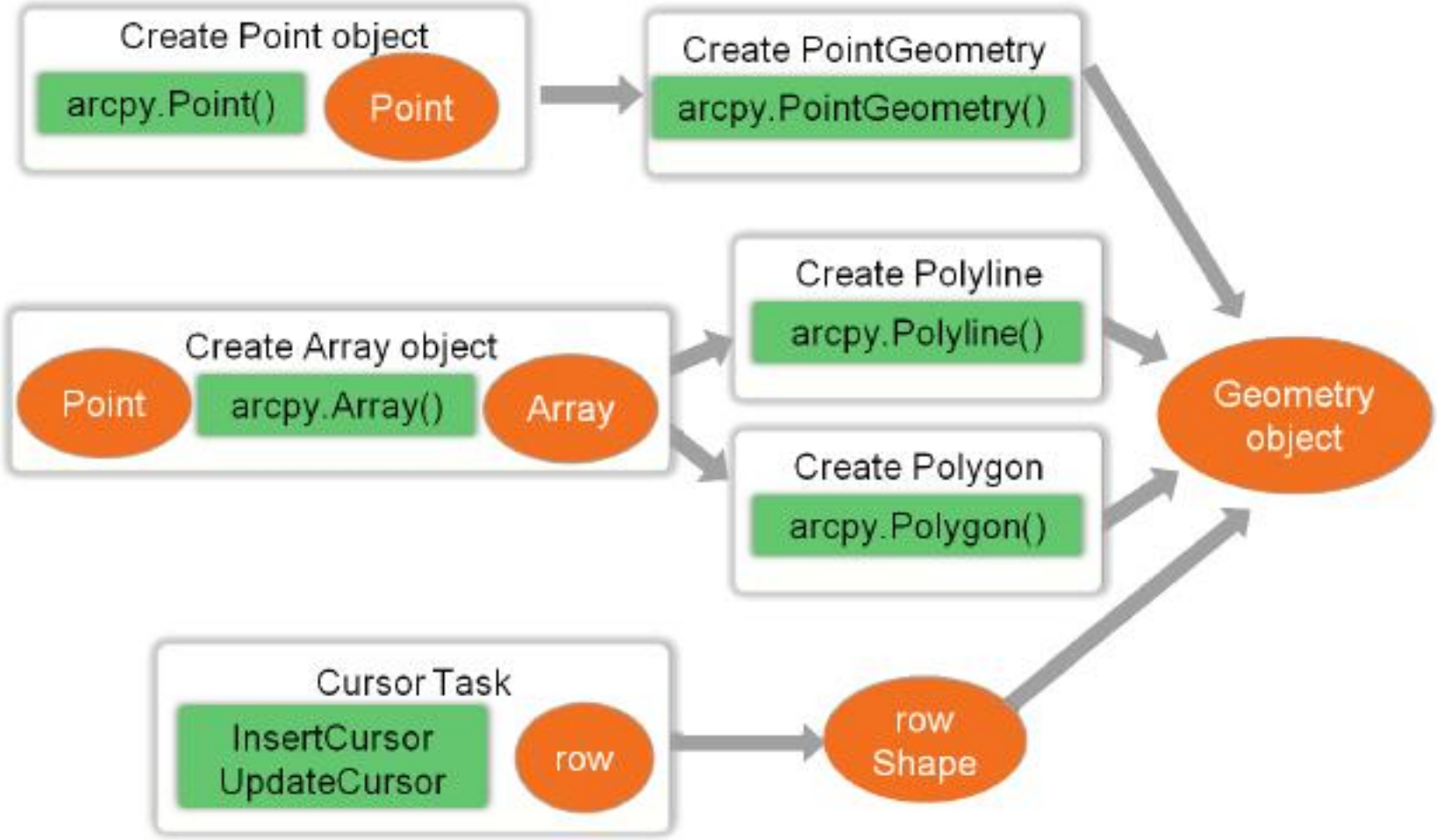
Работа с геометрией

- Типы объектов геометрии
  - `arcpy.PointGeometry`
  - `arcpy.Polyline`
  - `arcpy.Polygon`
  - `arcpy.Multipoint`

Схожие свойства и методы

- Объекты `Point` и `Array`
  - Используются для создания объектов

Работа с геометрией



Классы Point и PointGeometry

- Синтаксис

```
Point ({X}, {Y}, {Z}, {M}, {ID})
```


```
import arcpy
pnt = arcpy.Point()
pnt.X=-98.36
pnt.Y=29.56
print " X: {}, Y: {}". format(pnt.X, pnt.Y)
```

Свойства:

X	Горизонтальная координата точки.
Y	Вертикальная координата точки.
Z	Значение высоты точки.
M	Значение измерения в точке
ID	Целое число, используемое для уникальной идентификации точки

Классы Point и PointGeometry

- Синтаксис

```
PointGeometry (inputs, {spatial_reference}, {has_z}, {has_m})
```

`Inputs` - Координаты, используемые для создания объекта. В качестве данных можно использовать объекты Point или Array.


```
import arcpy
pointList=[[1,2],[3,5],[7,3]]
point=arcpy.Point()
pointGeometryList=[]
for pt in pointList:
    point.X=pt[0]
    point.Y=pt[1]
    pointGeometry=arcpy.PointGeometry(point)
    pointGeometryList.append(pointGeometry)
arcpy.CopyFeatures_management(pointGeometryList, "D:/Upload/points.shp")
```

Классы ArcPy: Array

Объект-массив **используется для построения геометрических объектов**, может содержать точки и массивы

- Синтаксис

```
Array ({items})
```

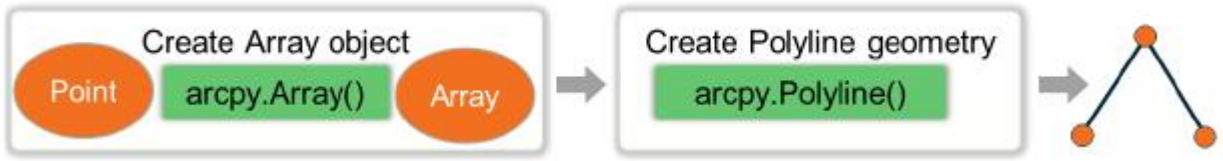
`items` - может представлять собой список, объект Point или другой объект Array.

```
import arcpy
from arcpy import env
env.workspace="D:/Upload/Lab1_data"
fc="Brest_railway.shp"
incurs=arcpy.da.InsertCursor(fc, ["SHAPE@"])
array=arcpy.Array([arcpy.Point(327889.46,5812812.23), arcpy.Point(322840.29, 5833732.85)
polyline = arcpy.Polyline(array)
incurs.insertRow([polyline])
del incurs
```

Классы ArcPy: Polyline

`Polyline (inputs, {spatial_reference}, {has_z}, {has_m})`

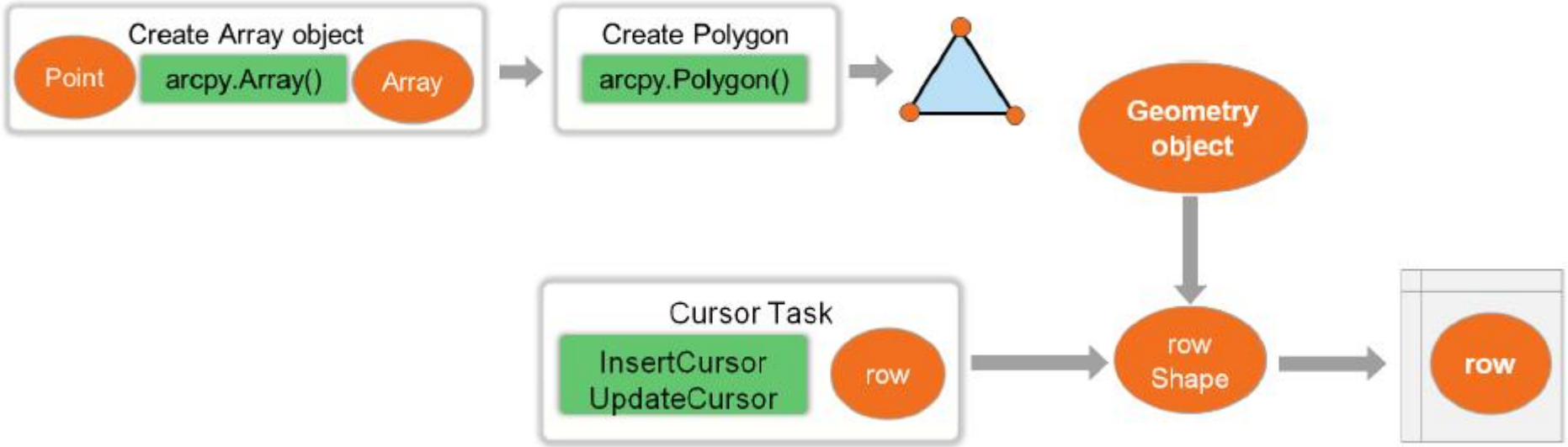
Точечный объект, заполнив его свойства и поместив его в массив. Затем массив можно использовать для задания геометрии пространственного объекта, используя классы геометрии Полигон (Polygon), Полилиния (Polyline), PointGeometry или Мультиточка (Multipoint).




```
import arcpy
fc = "d:/Upload/EsriPress/Python/Data/Exercise08/rivers.shp"
cursor = arcpy.da.InsertCursor(fc, ["SHAPE@"])
array = arcpy.Array([arcpy.Point(5997611.48964, 2069897.7022),
                     arcpy.Point(5997577.46097, 2069905.81145)])
polyline = arcpy.Polyline(array)
cursor.insertRow([polyline])
del cursor
```

Классы ArcPy: Polygon

```
Polygon (inputs, {spatial_reference}, {has_z}, {has_m})
```




```
cursor = arcpy.da.InsertCursor(featClass, "SHAPE@")
ary = arcpy.Array([arcpy.Point(9,11),
                   arcpy.Point(9,8),
                   arcpy.Point(6,8),
                   arcpy.Point(6, 11)])
polygon = arcpy.Polygon(array)
cursor.insertRow([polygon])
del cursor
```

Классы ArcPy: Geometry

```
Geometry (geometry, inputs, {spatial_reference},
          {has_z}, {has_m})
```

Свойства	Методы
area	boundary ()
centroid	buffer (distance)
isMultipart	clip (envelope)
length	contains (second_geometry)
partCount	crosses (second_geometry)
pointCount	cut (cutter)
spatialReference	distanceTo (other)
type	equals (second_geometry)
…….	……

```
point.within(polygon)
```

Выборка

`MakeFeatureLayer_management (in_features, out_layer, {where_clause}, {workspace}, {field_info})`

Параметр	Объяснение	Тип данных
in_features	Входной класс объектов или слой, из которого создается новый слой.	Feature Layer
out_layer	Имя создаваемого векторного слоя.	Feature Layer
where_clause	SQL-выражение, использованное для выбора поднабора пространственных объектов..	SQL Expression
workspace	Входная рабочая область используется для проверки имен полей.	Workspace;Feature Dataset
field_info	Может использоваться для просмотра и изменения имен полей….	Field Info

```
import arcpy
from arcpy import env
env.workspace="D:/Upload/EsriPress/Python/Data/Exercise06"
arcpy.MakeFeatureLayer_management("counties.shp", "counties_lyr")
```

Выборка

```
SelectLayerByLocation_management (in_layer,
{overlap_type}, {select_features}, {search_distance},
{selection_type}, {invert_spatial_relationship})
```

INTERSECT
WITHIN_A_DISTANCE
CONTAINS
COMPLETELY_CONTAINS
WITHIN
COMPLETELY_WITHIN
BOUNDARY_TOUCHES
HAVE_THEIR_CENTER_IN
……

```
import arcpy
from arcpy import env
env.workspace="D:/Upload/EsriPress/Python/Data/Exercise06"
arcpy.MakeFeatureLayer_management("counties.shp", "counties_lyr")
arcpy.SelectLayerByLocation_management("counties_lyr", "CONTAINS",
                                       "amtrak_stations.shp")
arcpy.CopyFeatures_management("counties_lyr", "counties_station.shp")
```