**geo\_types.py** — *типы объектов*

class Point:

a = [x, y] — координаты точки

class Line:

n = [x, y] — вектор нормали

c — свободный коэффициент в ур-нии прямой

v — вектор направления прямой

class Segment(Line):

n = [x, y]

c

v

end\_points = [[x1, y1], [x2, y2]] — концы отрезка

length — длина отрезка

class Ray(Line):

n = [x, y]

c

v

start\_point = [x, y] — начало луча

class Angle:

p = [x, y] — вершина угла

angle — величина угла в радианах [0 … 2pi)

v1 = [x1, y1] — вектор направления первой стороны

v2 = [x2, y2] — вектор направления второй стороны

start\_angle — угол вектора первой стороны

end\_angle — угол вектора второй стороны

r — радиус дуги, обозначающей угол

class Polygon:

points = [[x1, y1], [x2, y2], … ] — список вершин многоугольника

class Circle:

c = [x, y] — центр окружности

r — радиус окружности

r\_squared — квадрат радиуса окружности

class Arc(Circle):

c = [x, y]

r

r\_squared

angles = [a1, a2] — углы начала и конца дуги

class Vector:

end\_points = [[x1, y1], [x2, y2]] — вершины отрезка вектора

v = [x, y] — вектор

class Measure:

x — значение

dim — размерность (тип)

class AngleSize:

x — значение угла

class Boolean:

b — значение булевой величины

**commands.py** — *конструкторы, построения, вычисления и проверки, связанные с объектами*

**random\_constr.py** —

class Element:

data — данные элемента

label — надпись/имя элемента

class Command:

name — имя команды (построение, операция, проверка …)

input\_elements — элементы на входе

output\_elements — элементы на выходе

class ConstCommand:

datatype —

value —

element —

class Construction:

corners —

min\_border —

max\_border —

nc\_commands —

to\_prove —

element\_dict —

elements —

def command\_types\_name(name, params): # выдает сокращенное название функции в модуле

def parse\_command(line, element\_dict): # парсит строку и создает объект команды либо переменной

**read\_ggb.py** — *парсинг и конвертация файла .ggb --> .txt (краткий формат)*

class Element:

data — данные элемента

label — надпись/имя элемента

state — "to\_update" / "to\_keep" / "set"

class Command:

operation — имя команды (построение, операция, проверка …)

inputs — элементы на входе

outputs — элементы на выходе

class ConstCommand:

datatype — ?

value — ?

label — ?

def apply\_command(name, params): # применяет соответствующую функцию из commands.py и возвращает результат

def process\_ggb(filename): # обрабатывает файл .ggb и возвращает список команд

def tweak\_command\_name(name): # "AreCollinear" -> "are\_collinear"

def get\_element(label):

def get\_expr\_node(node, label=None):

def get\_expr(expr\_str, root\_label=None):