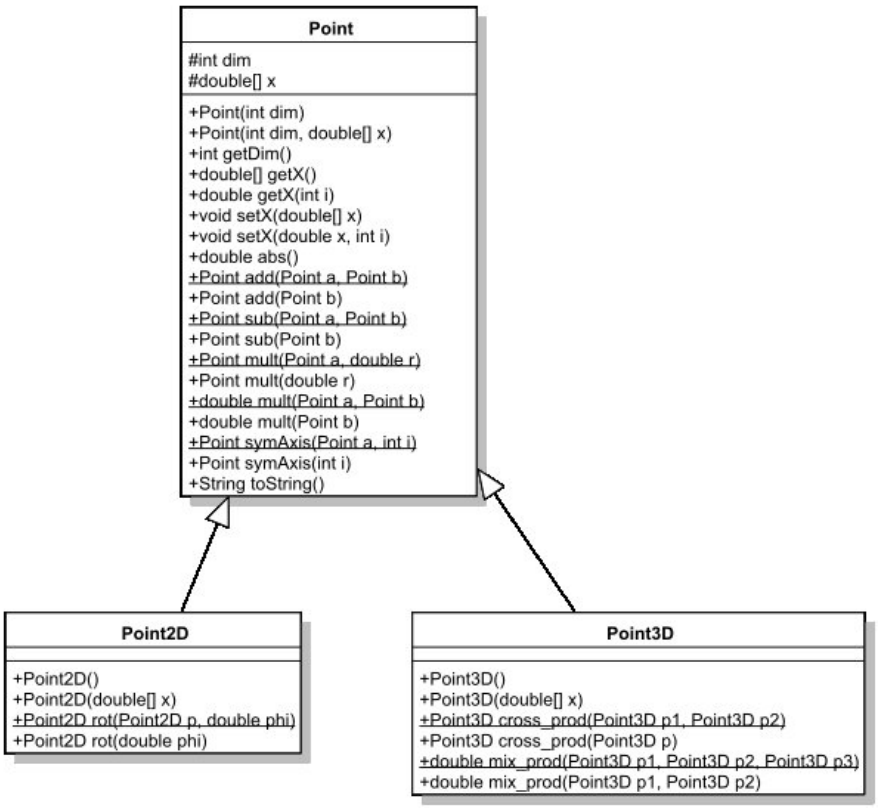
Общие замечания:

1. Нужно переопределить toString для всех классов, в которых указан, чтобы выводил все внутренние параметры.
2. В конструкторах необходимо выполнить проверку на корректность входных данных (Количество координат совпадает с указанной размерностью, в прямоугольнике смежные стороны под прямым углом, …) через соответствующие исключения(Exception).
3. В схемах, представленных ниже подчёркнуты статические функции (static).
4. Для каждого внутреннего параметра должен быть реализован get-метод.
5. Для каждого внутреннего параметра, кроме количества точек и размерности пространства, должен быть реализован set-метод.
6. Если внутренний параметр массив, то get/set-методы должны быть реализованы в двух вариантах:
   a. Работа с цельным массивом.
   b. Работа с элементом по номеру.

Классы, описывающие точку в пространстве:



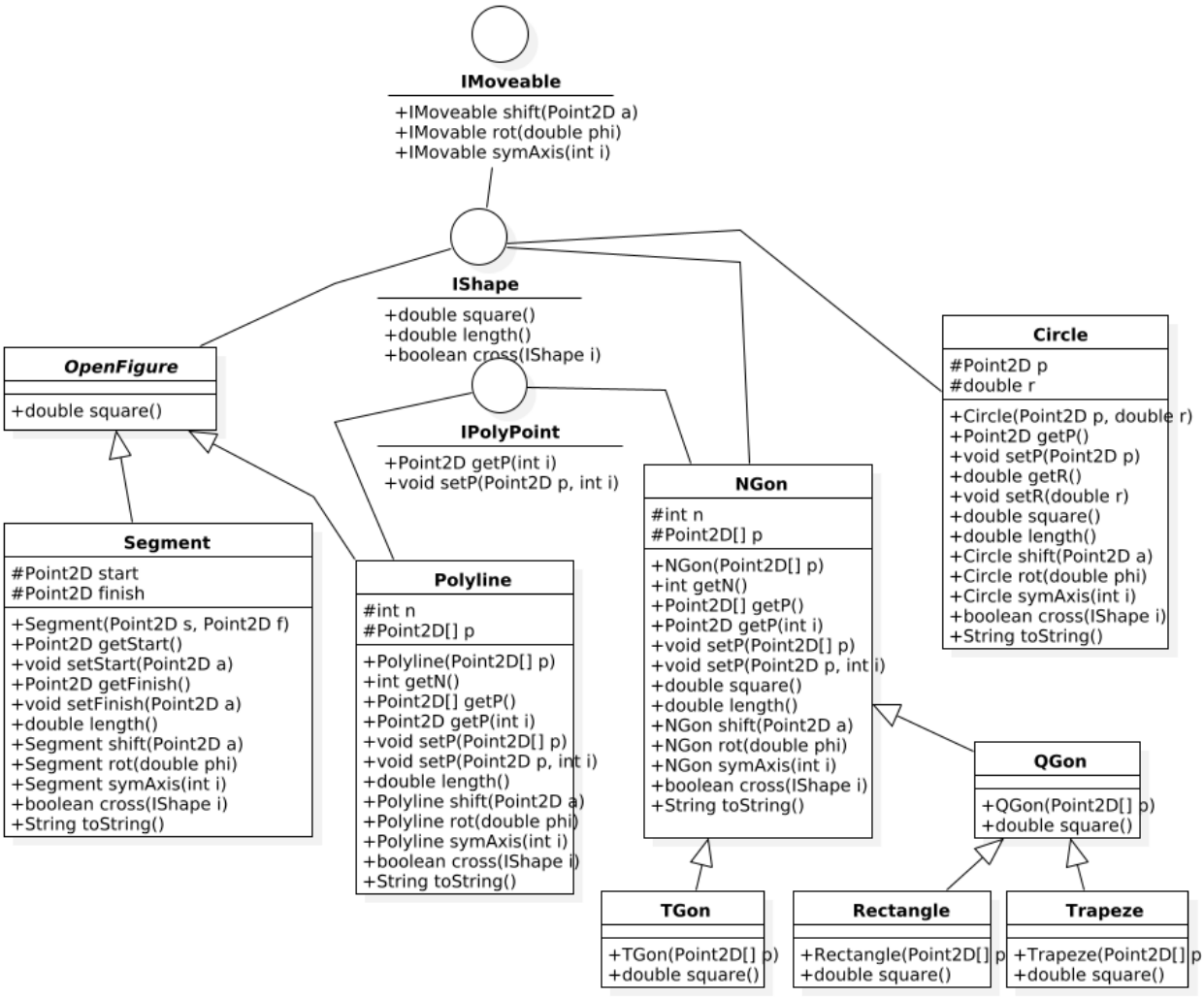
1. Point – описывает точку в N-мерном пространстве. Имеет следующие внутренние параметры:
   a. dim – размерность пространства.
   b. x – массив координат размерности dim.

   Имеет следующие функции:

a. Point – конструктор, описывающий инициализацию объекта. Существует в двух вариантах:
    i. Одно целое число – размерность пространства. Создаёт начало координат в пространстве указанной размерности.
    ii. Целое число и массив дробных чисел – создаёт точку в пространстве указанной размерности, значения координат берёт из массива дробных чисел.
b. abs – возвращает модуль точки (расстояние до начала координат).
c. add – складывает по координатам две точки.
d. sub – находит разность по координатам двух точек.
e. mult – умножение точки на:
    i. Дробное число – произведение по координатам.
    ii. Точка – скалярное произведение двух точек.
f. symAxis – симметрия точки относительно оси под заданным номером (0 – Ox, 1 – Oy, 2 – Oz, …).

2. Point2D – описывает точку на плоскости. Дополнительных внутренних параметров нет. Имеет следующие дополнительные функции:
    a. Point2D – конструктор, описывающий инициализацию объекта. Существует в двух вариантах:
        i. Без аргументов – создаёт начало координат на плоскости.
        ii. Массив дробных чисел – создаёт точку на плоскости, значения координат берёт из массива дробных чисел.
    b. rot – функция поворота точки на угол phi в радианах относительно начала координат. Положительное направление – против часовой стрелки.
3. Point3D – описывает точку в 3-мерном пространстве. Дополнительных внутренних параметров нет. Имеет следующие дополнительные функции:
    a. Point3D – конструктор, описывающий инициализацию объекта. Существует в двух вариантах:
        i. Без аргументов – создаёт начало координат на плоскости.
        ii. Массив дробных чисел – создаёт точку на плоскости, значения координат берёт из массива дробных чисел.
    b. cross_prod – векторное произведение двух точек.
    c. mix_prod – смешанное произведение трёх точек

Классы, описывающие фигуры на плоскости:



1. <u>IShape – интерфейс, описывающий поведение фигур на плоскости</u>. Содержит следующие функции:
   a. square – возвращает площадь фигуры.
   b. length – возвращает длину/периметр фигуры.
   c. shift – сдвиг фигуры на вектор заданный объектом класса Point2D (вектор от начала координат до указанной точки)
   d. rot – поворот фигуры на угол phi в радианах относительно начала координат. Положительное направление – против часовой стрелки.
   d. symAxis – симметрия точки относительно оси под заданным номером (0 – Ox, 1 – Oy, 2 – Oz, …).
   e. cross – проверяет пересекаются ли фигуры (true – пересекаются, false – нет).
2. <u>OpenFigure – абстрактный класс незамкнутых фигур</u>. Реализована только функция square из интерфейса.
4. <u>Segment – отрезок</u>. Имеет следующие внутренние параметры:
   a. start – начальная точка отрезка.

b. finish – конечная точка отрезка.

Реализованы все функции из интерфейса, кроме square. Имеет следующие дополнительные функции:

a. Segment – конструктор, описывающий инициализацию объекта. На вход принимает две точки – начало и конец.

5. Polyline – ломаная линия. Имеет следующие внутренние параметры:
   a. n – количество точек.
   b. p – массив двумерных точек.

   Реализованы все функции из интерфейса, кроме square. Имеет следующие дополнительные функции:
   a. Polyline – конструктор, описывающий инициализацию объекта. На вход принимает массив точек в порядке обхода по ломаной.
6. Circle – окружность. Имеет следующие внутренние параметры:
   a. r – радиус окружности.
   b. p – точка, центр окружности.

   Реализованы все функции из интерфейса. Имеет следующие дополнительные функции:
   a. Circle – конструктор, описывающий инициализацию объекта. На вход принимает центр окружности и его радиус.
7. NGon – N-угольник. Имеет следующие внутренние параметры:
   a. n – количество точек.
   b. p – массив двумерных точек.

   Реализованы все функции из интерфейса. Имеет следующие дополнительные функции:
   a. NGon – конструктор, описывающий инициализацию объекта. На вход принимает массив точек в порядке обхода по N-угольнику.
8. TGon – треугольник. Дополнительных внутренних параметров нет. Переопределяет функцию square. Имеет следующие дополнительные функции:
   a. TGon – конструктор, описывающий инициализацию объекта. На вход принимает массив точек в порядке обхода по треугольнику.
9. QGon – четырёхугольник. Дополнительных внутренних параметров нет. Переопределяет функцию square. Имеет следующие дополнительные функции:
   a. QGon – конструктор, описывающий инициализацию объекта. На вход принимает массив точек в порядке обхода по четырёхугольнику.
10. Rectangle – прямоугольник. Дополнительных внутренних параметров нет. Переопределяет функцию square. Имеет следующие дополнительные функции:

   a. Rectangle – конструктор, описывающий инициализацию объекта. На вход принимает массив точек в порядке обхода по прямоугольнику.
11. Trapeze – трапеция. Дополнительных внутренних параметров нет. Переопределяет функцию square. Имеет следующие дополнительные функции:
   a. Trapeze – конструктор, описывающий инициализацию объекта. На вход принимает массив точек в порядке обхода по трапеции.

Main.java:

1. Создать переменную типа List<IShape>.
2. Ввести количество фигур.
3. Для каждой фигуры:
   a. Ввести её тип (окружность, треугольник, отрезок, …).
   b. Ввести все её параметры (точки, радиус, …).
   c. Сохранить в списке.
4. Вывести на экран:
   a. Суммарная площадь всех фигур.
   b. Суммарная длина всех фигур.
   c. Средняя площадь по всем фигурам.
5. Второй цикл размером по количеству фигур:
   a. Создаёте новую фигуру:
      i. Тип вводимой фигуры соответствует типу фигуры из списка под тем же номером.
      ii. Ввести все её параметры (точки, радиус, …).
   b. Вывести пересекаются ли введённая фигура с фигурой из списка.
   c. Определить движение введённой фигуры:
      i. Ввести тип движения (поворот, сдвиг, симметрия).
      ii. Ввести параметры движения (угол, вектор, …).
   d. Вывести пересекаются ли введённая фигура с фигурой из списка.