

J*. Geometry

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

https://gitlab.com/ibr11/cpp_psami_base/-/tree/main/geometry

В этой задаче необходимо реализовать набор классов для решения геометрических задач на плоскости. Все координаты предполагаются целочисленными.

Vector

Реализовать класс Vector для вектора на плоскости с необходимыми арифметическими операциями (унарные и бинарные $+$ / $-$, умножение/деление на скаляр, присваивающие версии операций, сравнение на равенство).

Shape

Создать набор классов-фигур, которые наследуются от абстрактного базового класса IShape для работы с двумерными геометрическими примитивами:

- Point (точка);
- Segment (отрезок);
- Line (линия);
- Ray (луч);
- Polygon (простой многоугольник - часть плоскости, ограниченная замкнутой ломаной без самопересечений);
- Circle (окружность).

В базовом классе IShape предусмотреть чисто виртуальные методы:

- Move(const Vector&) - сдвига на заданный вектор, метод должен изменять состояние объекта и возвращать ссылку на сам объект;
- ContainsPoint(const Point&) - проверка (true/false) содержит ли фигура (внутренность фигуры) точку;
- CrossesSegment(const Segment&) проверка (true/false) пересекается ли фигура (граница фигуры) с отрезком;
- Clone() - копирование объекта (необходимо вернуть умный или обычный указатель на копию фигуры);
- ToString() - строковое представление фигуры (формат см. в примерах).

В производных классах - реализовать эти методы. Разность двух точек должна возвращать вектор перемещения одной точки в другую.

Детали

Все классы должны располагаться в пространстве имен `geometry`.

С публичным интерфейсом классов, который используется в задаче, подробнее можно ознакомиться в файле `geometry_main.cpp`. В частности, `Point` должен уметь конструироваться от двух целых чисел, `Segment`, `Line` и `Ray` - от двух `Point`, многоугольник - от `std::vector<Point>`, окружность - от центра (`Point`) и радиуса (`int`).

Объявления классов расположите в соответствующих `.h` файлах, а необходимые файлы реализации положите в папку `src`. После этого вы можете запустить `geometry_public_test` (по обычной инструкции - через терминал, либо с помощью IDE) и протестировать на различных входных данных. Вы можете прочитать, что принимает программа на вход из файла `geometry_main.cpp` и найти примеры корректной работы на странице задачи в контексте.

Где ошибка? Тесты:

- 1-6 совпадают с примером
- 7-15 *Point*
- 16-46 *Segment*
- 47-55 *Ray*
- 56-60 *Line*
- 61-71 *Polygon*
- 72-80 *Circle*

Формат ввода

В первой строчке задается тип геометрического примитива: `<<point>>`, `<<segment>>`, `<<ray>>`, `<<line>>`, `<<circle>>` или `<<polygon>>`. Далее вводится сам примитив.

После чего вводится две точки A и B , которые используются в *CheckFunctions*. Все числа целочисленные и не превосходят 10000 по модулю.

Для `Point` выводятся ее координаты; `Segment` - два конца (точки); `Ray` - начало (точка) и направляющий вектор; `Line` - коэффициенты уравнения прямой ($ax + by + c$) с точностью до некоторого множителя (по идее система должна принимать и `Line(1, -1, 1)`, и `Line(-2, 2, -2)`); `Polygon` - последовательность вершин; `Circle` - центр и радиус.

Пример 1

Ввод 

```
point
0 0
-1 -1 1 1
```

Вывод 

```
Given shape does not contain point A
Given shape crosses segment AB
Point(2, 2)
```

Пример 2

Ввод 

```
segment
0 -1 0 1
0 0 0 1
```

Вывод 

```
Given shape contains point A
Given shape crosses segment AB
Segment(Point(0, 0), Point(0, 2))
```

Пример 3

Ввод 

```
ray
0 0 1 1
2 2 3 2
```

Вывод 

```
Given shape contains point A
Given shape crosses segment AB
Ray(Point(1, 0), Vector(1, 1))
```

Пример 4

Ввод 

```
line
0 0 0 1
1 1 2 1
```

Вывод 

```
Given shape does not contain point A
Given shape does not cross segment AB
Line(1, 0, -1)
```

Пример 5

Ввод 

```
polygon
4
0 0 3 0 3 3 0 3
1 1 2 2
```

Вывод 

```
Given shape contains point A
Given shape does not cross segment AB
Polygon(Point(1, 1), Point(4, 1), Poin
```



Пример 6

Ввод 

```
circle
0 0 5
0 0 5 5
```

Вывод 

```
Given shape contains point A
Given shape crosses segment AB
Circle(Point(5, 5), 5)
```