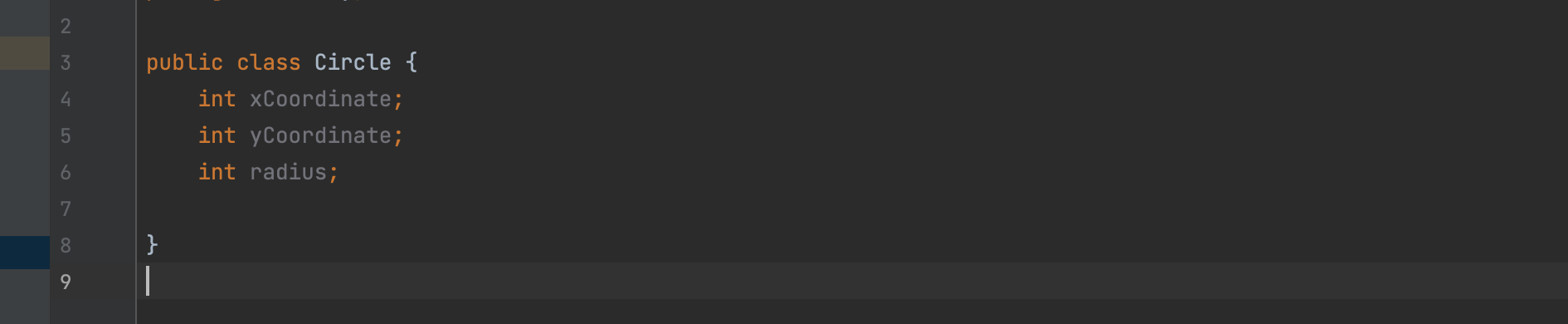
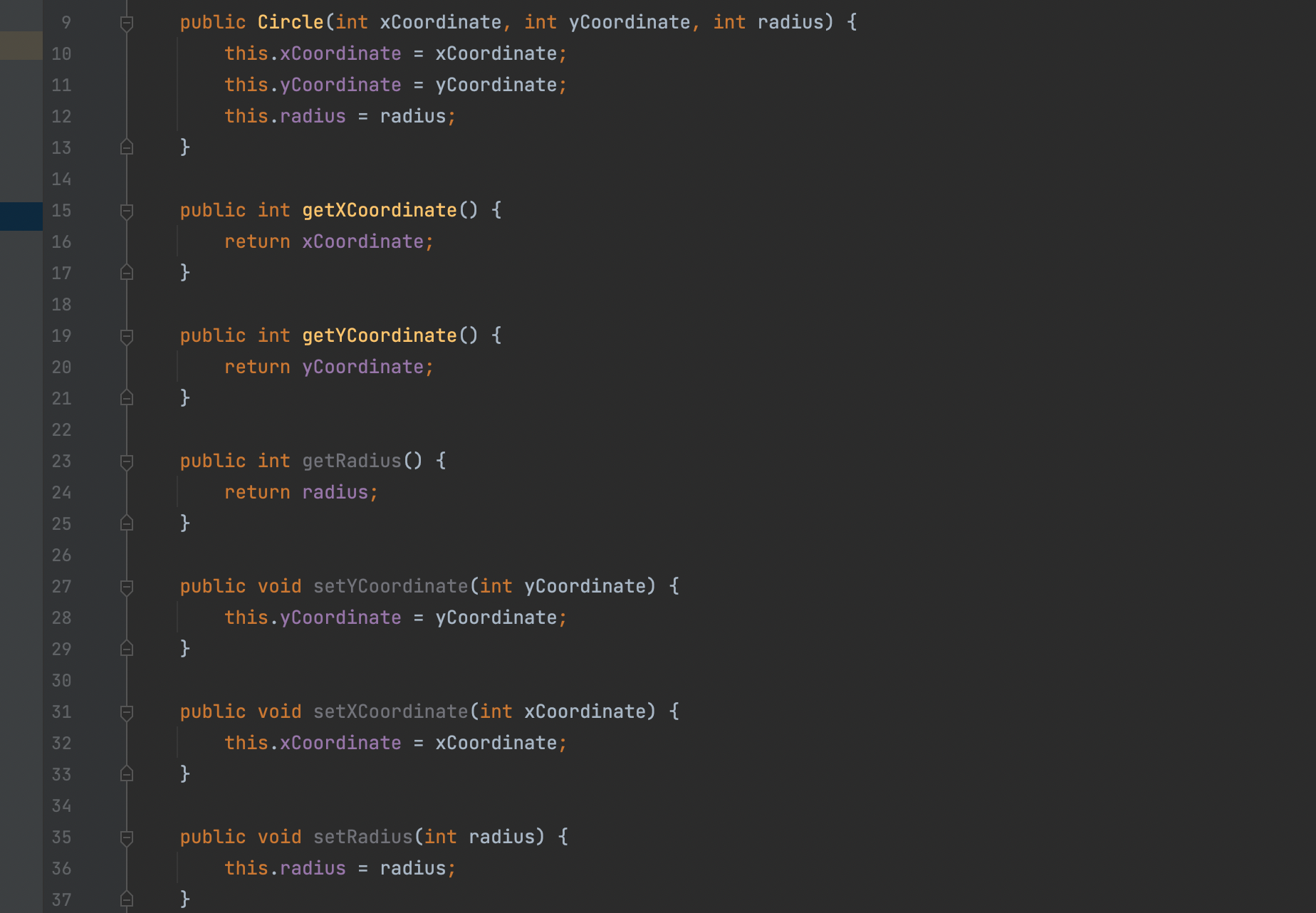
Lab work 2

1 - Написать класс Circle с параметрами: xCoordinate, yCoordinate, radius

(Создать новый файл рядом с файлом Main и назвать его Circle)

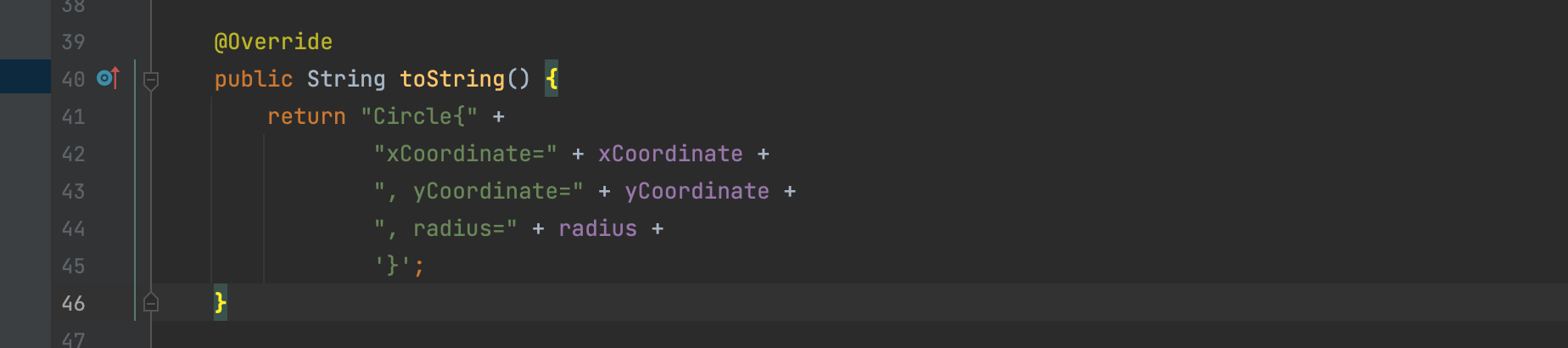


2 - затем добавить в этом классе конструктор, геттеры и сеттеры



3 - переопределить в этом классе метод toString()

-- так как все классы в Java неявно наследуются от класса Object, а класс Object уже имеет метод toString(), мы используем @Override аннотацию чтобы при вызове toString() нашего класса вызвалась именно наша реализация а не реализация класса Object



4 - считать с консоли ввод координат для центра и радиус

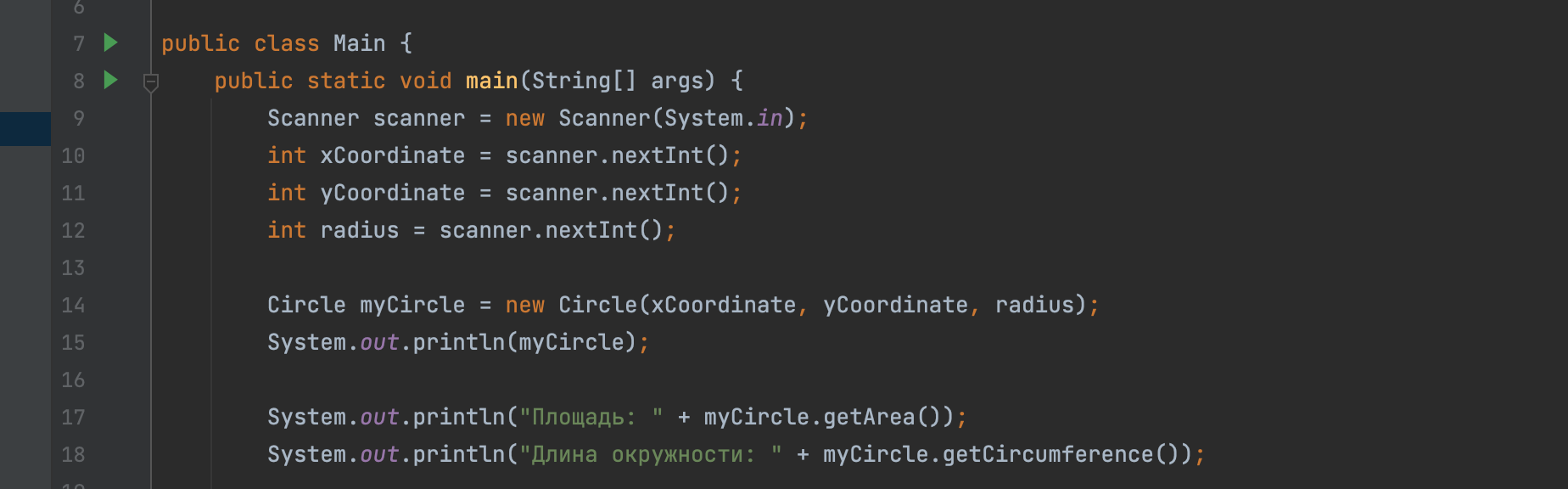
5 - создать объект myCircle класса Circle вызовом его конструктора и указанием параметров xCoordinate, yCoordinate, radius

6 - вывести объект myCircle в консоль

7 - написать реализацию методов "Double getArea()" (расчет площади круга) и "Double getCircumference()" (расчет длины окружности)

8 - вывести площадь в консоль

9 - вывести длину окружности в консоль



10 - написать реализацию метода "Double getDistanceTo(Circle anotherCircle)" (расчет расстояния от центра окружности до центра второй окружности

11 - считать с консоли новые параметры для нового объекта secondCircle

12 - вывести расстояние от центра myCircle до secondCircle

ADVANCED

- Дополнительно создать класс Point и использовать его в качестве центра в классе AdvancedCircle

- написать реализацию метода "Double getDistance(Point point)" внутри класса Point

- при вызове метода Double getDistanceTo(AdvancedCircle anotherCircle) внутри AdvancedCircle вызывать реализацию "getDistance(Point point)" внутренних объектов класса Point

Домашняя работа

В этом же проекте написать класс Rectangle (прямоугольник) описанного уже с помощью двух точек (верхней левой и правой нижней) и расписать реализации методов getArea(), getPerimeter() и getDistanceToTheCenterOf(Rectangle anotherRectangle) учитывая что xCenter = (x1 + x2) / 2 и yCenter = (y1 + y2) / 2), где x1 это левая координата по оси X и x2 это правая координата по оси X (y1 - нижняя и y2 - верхняя)

ADVANCED

- Дополнительно создать класс Point и использовать его в качестве точек (верхней левой и нижней правой) в классе AdvancedRectangle