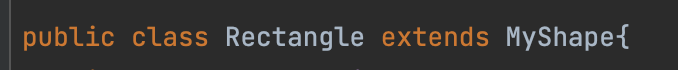
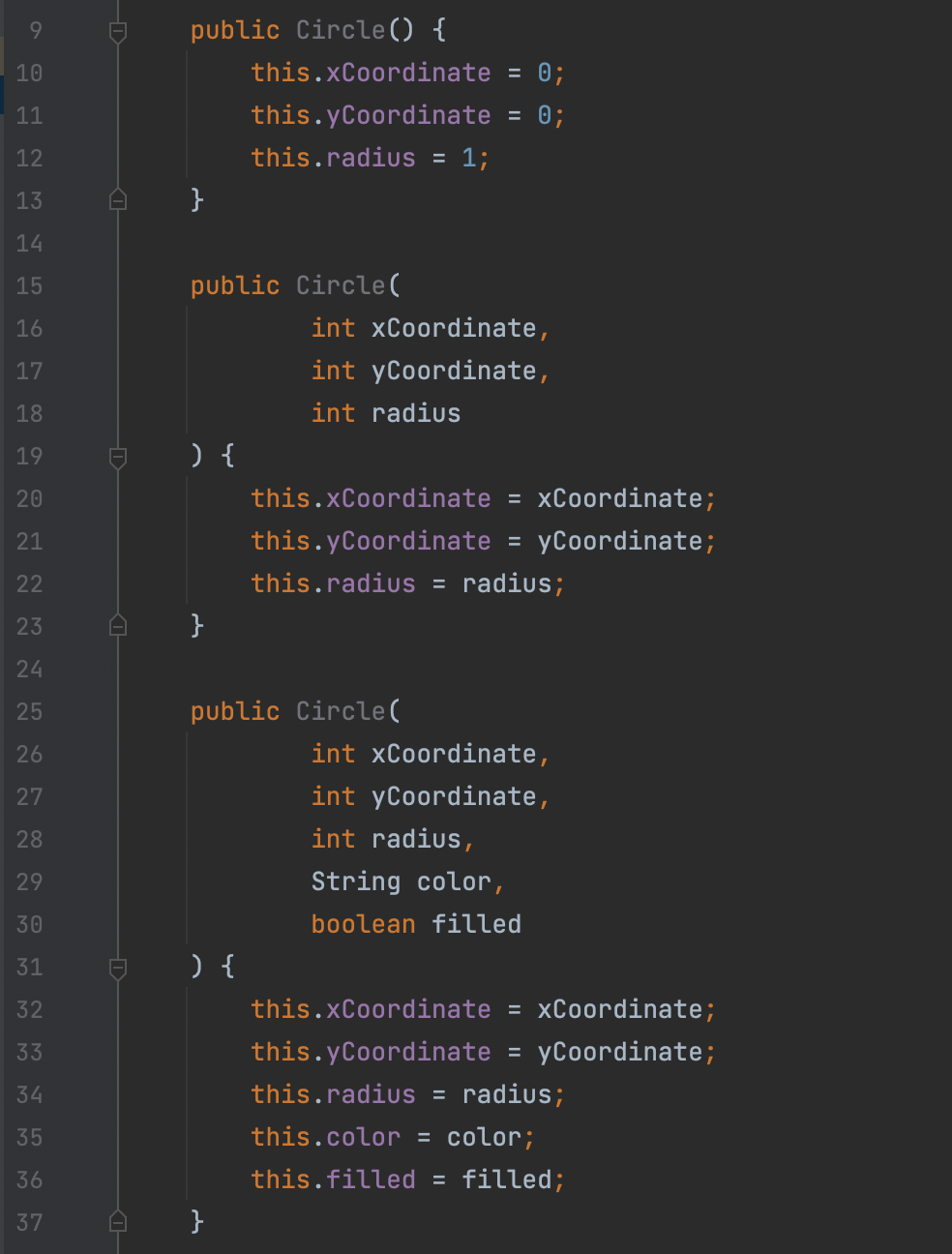
LAB 3

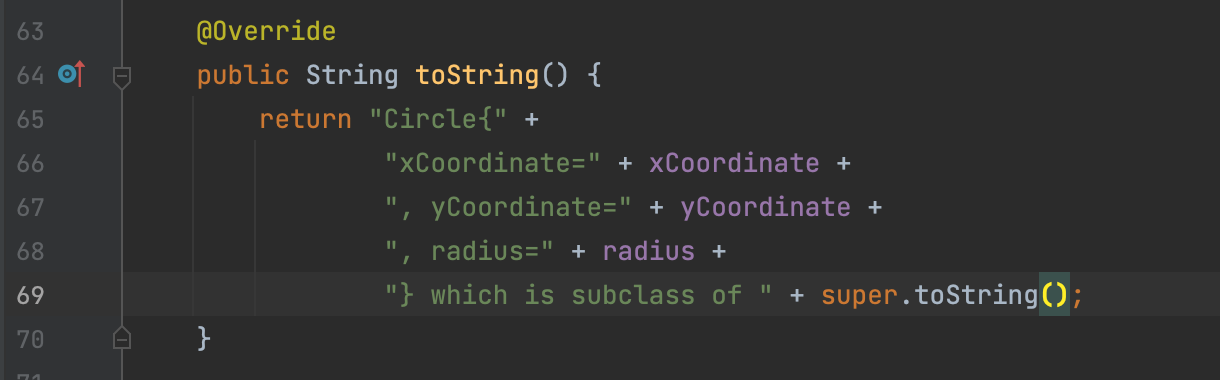
1. Создать класс MyShape:
   1. Создать параметры color (String) и filled (boolean);
   2. Написать два конструктора - один пустой с дефолтными параметрами color - red и filled - true, второй с параметрами color и filled;
   3. Написать геттеры и сеттеры для этих двух параметров (по конвенции Java языка, геттер для параметра с типом boolean начинается с isXXX (isFilled()) вместо getXXX как во всех других случаях);
   4. Переписать метод toString() чтобы возвращал текст с описанием цвета и заполнением ("MyShape{" + "color=" + color' + ", isFilled=" + filled + ”}”)
2. Указать классы Circle и Rectangle (из прошлой лабораторной) как потомки класса MyShape



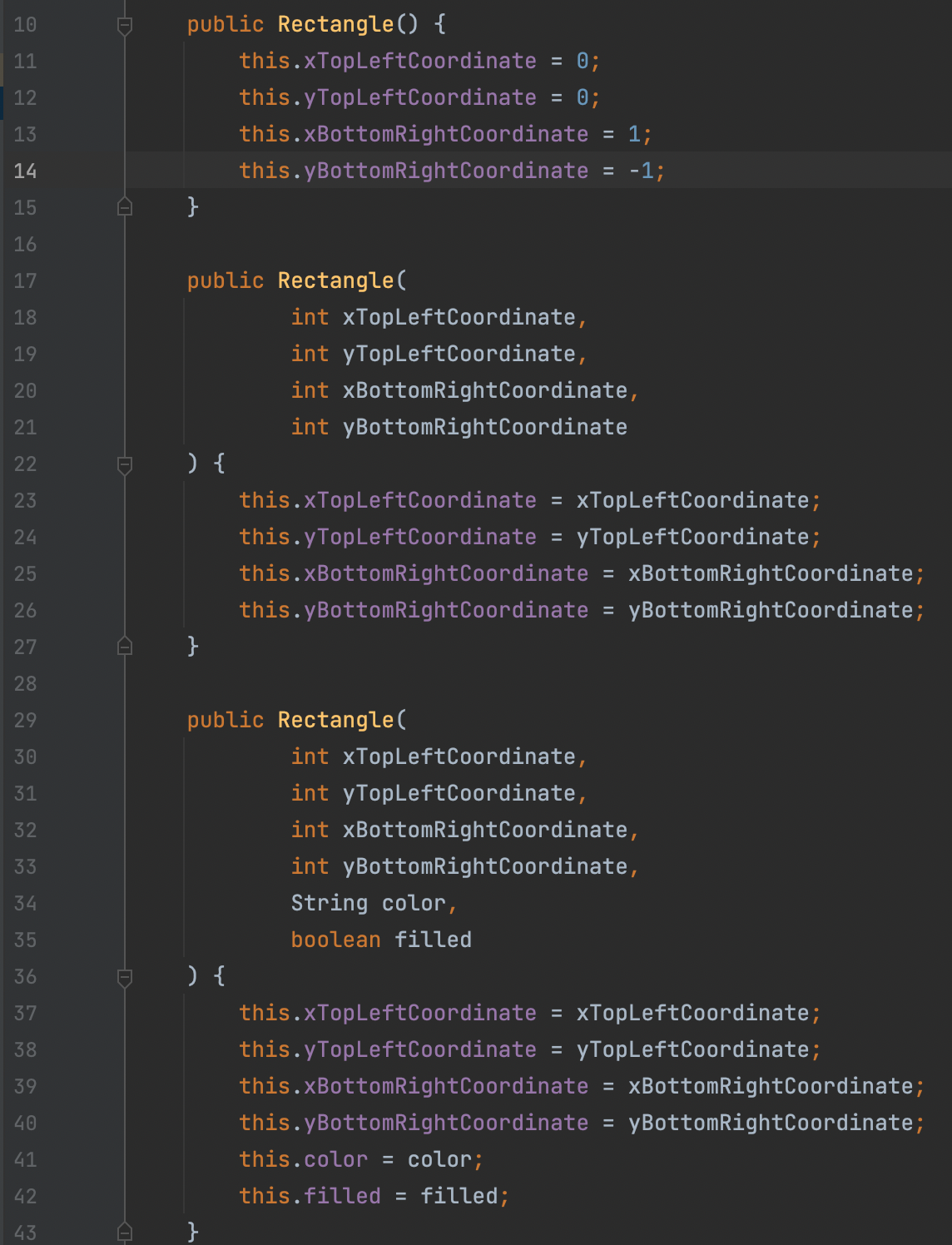
* 1. В Circle написать три конструктора: один пустой с дефолтными значениями для x=0, y=0, radius=1, второй с параметрами только класса Circle(int xCoordinate, int yCoordinate, int radius), //– обратите внимание что в случае вызова этого конструктора, вызывается дефолтный конструктор родительского класса –//, третий с дополнительными параметрами color и filled;



* 1. Дописать текст в методе toString() в Circle добавив туда “which is subclass of " + super.toString(), где вы вызываете дополнительно toString() родительского класса;

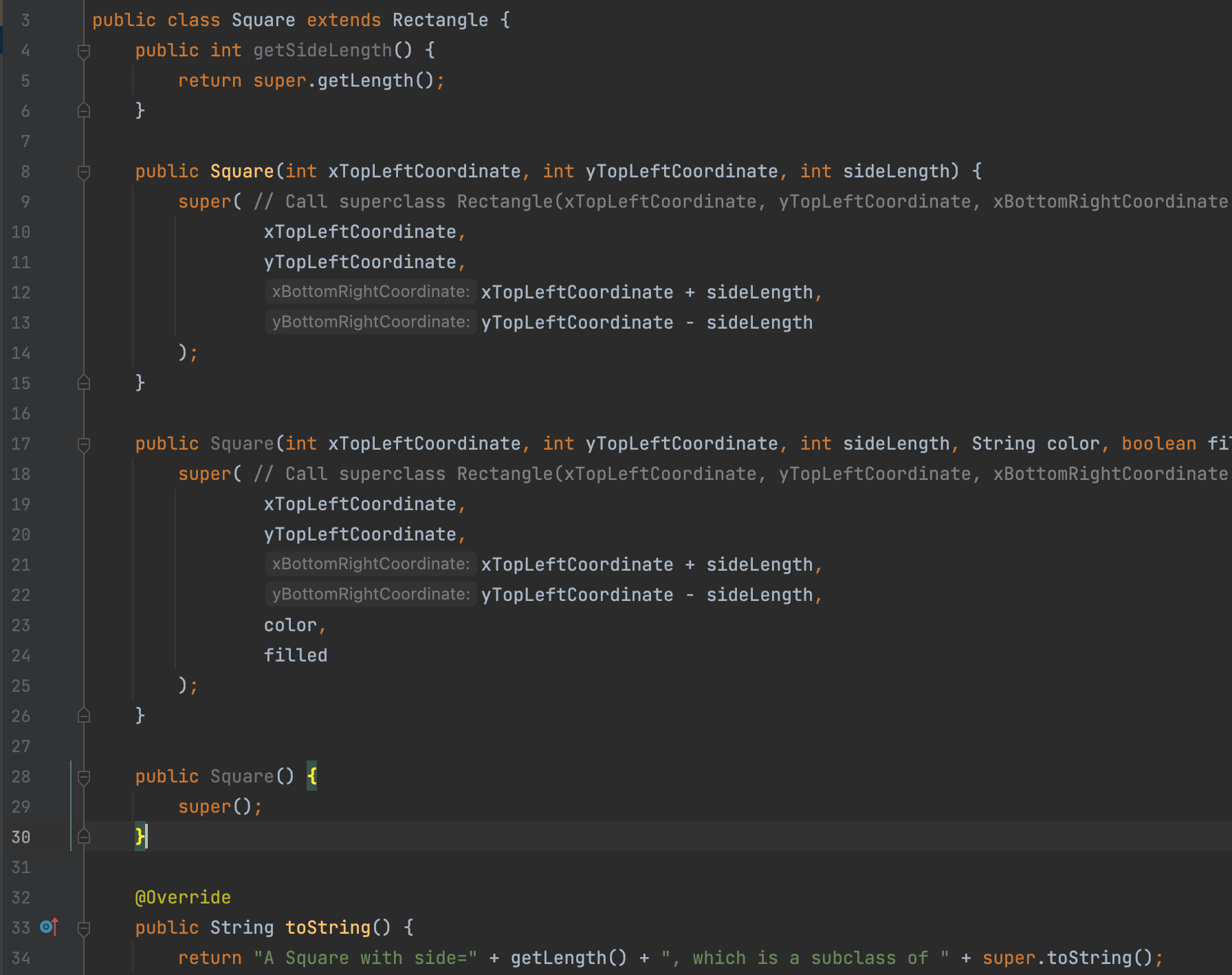


* 1. В Rectangle написать три конструктора: один пустой с дефолтными значениями для xTopLeftCoordinate=0, yTopLeftCoordinate=0, xBottomRightCoordinate=1, yBottomRightCoordinate=-1 второй с параметрами только класса Rectangle(int xTopLeftCoordinate, int yCyTopLeftCoordinateoordinate, int xBottomRightCoordinate, int yBottomRightCoordinate), //– обратите внимание что в случае вызова этого конструктора, вызывается дефолтный конструктор родительского класса –//, третий с дополнительными параметрами color и filled;



* 1. Написать методы getLength() расчет по оси X, getWidth() расчет по оси Y;
  2. Дописать текст в методе toString() в Rectangle добавив туда “which is subclass of " + super.toString(), где вы вызываете дополнительно toString() родительского класса;

1. Создать класс Square, который будет наследоваться от Rectangle
   1. Написать в нем три конструктора: пустой, со следующими параметрами - (int xTopLeftCoordinate, int yTopLeftCoordinate, int sideLength), и третий с дополнительными color и filled;
   2. Написать метод getSideLength (ПРИМЕЧАНИЕ - не создавать параметр sideLength, использовать только параметры и методы родительского Rectangle)



* 1. Дописать текст в методе toString() в Square добавив туда “which is subclass of " + super.toString(), где вы вызываете дополнительно toString() родительского класса;

1. Написать код в классе Main (в методе main()):
   1. Считать с консоли int xTopLeftCoordinate, int yTopLeftCoordinate, int sideLength;
   2. Создайте объект класса Square с названием square;
   3. Вывести текст:

System.out.println("Площадь: " + square.getArea());

System.out.println("Периметр: " + square.getCircumference());

System.out.println("Цвет: " + square.getColor());

System.out.println("Закрашен: " + square.isFilled());

* 1. Считать с консоли String newColor и измените на него параметр color в square;
  2. Вывести текст с новым цветом:

System.out.println("Цвет: " + square.getColor());