## ООП в функциональном стиле

Создайте базовый класс **Figure**, который будет хранить координаты (x, y) и цвет фигуры. На базе класса **Figure** создайте три класса – **Line**, **Rect**, **Circle**, каждый из которых создает соответствующую фигуру. Пример создания экземпляров каждого класса и параметры фигур:

|  |
| --- |
| var line = new Line(50, 250, 200, 200, 'red'); // x1, y1, x2, y2, color  var circle = new Circle(120, 120, 50, 'green'); // x, y, r, color  var rect = new Rect(260, 130, 60, 120, 'blue'); // x, y, w, h, color |

Все три класса-наследника имеют метод **draw** для рисования фигуры с соответствующими параметрами (координаты, размеры, цвет).

Фигуры рисуются на [Canvas](http://bfy.tw/A5t0). Для рисования на канвасе создайте еще один класс – **Canvas**, в котором инициализируется элемент <canvas> из DOM. Класс **Canvas** – final, он не наследуется. В этом классе есть метод **add**, который и отображает созданные вами фигуры на странице. Обратите внимание, добавлять фигуры на канвас можно как по отдельности, так и списком. Также у класса есть свойства width и height, которые можно передать при вызове конструктора и которые меняют размер элемента canvas в html. Если в конструктор не передать значения для ширины/высоты, то элементу canvas устанавливаются значения по умолчанию

|  |
| --- |
| var drawArea = new Canvas('canvasID');  drawArea.add(line);  drawArea.add(circle, rect); |

|  |
| --- |
| <canvas id="canvasID"></canvas> |

Творчество не ограничено. Базовые возможности вашей программы:

