

Sınıf Metodları

```
#!/usr/bin/ruby

class Point
  attr_reader :x, :y

  ORIGIN = 0, 0

  # FIXME: Sınıf niteliği yerine başka bir şey kullanılabilir mi?
  @@population = 0

  def initialize(x, y)
    @x, @y = x, y
    @@population += 1
  end

  def >(other)
    @x > other.x
  end

  def <(other)
    @x < other.x
  end

  def >=(other)
    @x > other.x || @x == other.x
  end

  def <=(other)
    @x < other.x || @x == other.x
  end
end
```

```
def ==(other)
  @x == other.x && @y == other.y
end

# + işlecini overload ederek sınırsız sayıda noktayı da
# toplama olanağına kavuştuk.
def +(other)
  # Nesne metodundan sınıf metodu (origin) nasıl çağrılır?
  # Point.origin olabilir ama DRY değil, cevap: self.class
  other ||= self.class.origin
  self.class.new @x + other.x, @y + other.y
end

def move(x, y)
  @x += x
  @y += y
  self
end

def to_s
  "(#{@x},#{@y})"
end

# Sınıf metodları

# Her seferinde self. yazmak yerine sınıfı yarıp metodu
# içine koyalım. Böylesi daha DRY

# Burada self ne değer alır? Point
class << self
  def origin
    # new metodunun alıcısı kim? self
    # self burada ne değer alır? Point (yani sınıf)
    new(*ORIGIN)
  end
end
```

```
    def population
      @@population
    end
  end
end

def main
  _, _, _ = Point.new(3, 5), Point.new(9, 7), Point.new(-2, 5)
  puts "Nokta sayısı: #{Point.population}"
end

main if __FILE__ == $PROGRAM_NAME
```