

## fill.rb

A 12. feladat (fill) megoldására egy javaslat (még b?vítend? az alakzat konvexitását ellen?rz? kóddal)

```
class AngleException < Exception; end
class NonConvexException < Exception; end

class Vertex
  attr_accessor :x, :y
  def initialize(x=0, y=0)
    self.x = x
    self.y = y
  end
  def +(v2)
    Vertex.new(self.x + v2.x, self.y + v2.y)
  end
  def -(v2)
    Vertex.new(self.x - v2.x, self.y - v2.y)
  end
  def ==(v2)
    self.x == v2.x && self.y == v2.y
  end
  def to_s()
    "http://wiki.math.bme.hu("http://wiki.math.bme.hu + self.x.to_s + "http://wiki.math.bme.hu, "h
  end
end

class Polygon
  attr_accessor :width, :height, :vertices, :canvas

  def initialize(dim, vertices)
    self.width = dim[0].to_i
    self.height = dim[1].to_i
    self.vertices = vertices
    self.canvas = Array.new(self.height).map!{ Array.new(self.width) }
    # check 45 degrees
    vertices.each_index do |i|
      v1 = vertices[i]
      v2 = vertices[(i + 1) % vertices.length]
      if(v1.x != v2.x && v1.y != v2.y && v1.x - v2.x != v1.y - v2.y && v1.x - v2.x != v2.y - v1.y)
        raise AngleException
      end
    end
    # TODO: check if polygon is convex
  end

  # emit (already rendered) picture in PBM format
  def emit()
    print "http://wiki.math.bme.huP1 "http://wiki.math.bme.hu + self.width.to_s + "http://wiki.math
    self.canvas.reverse_each {|row|
      row.each {|pixel|
        print pixel.to_s + "http://wiki.math.bme.hu "http://wiki.math.bme.hu
      }
      print "http://wiki.math.bme.hu\n"http://wiki.math.bme.hu
    }
  end

  # render picture to canvas buffer
  def render()
    # clear canvas
    canvas.map! {|row| row.fill(0)}
    # draw polygon
  end
end
```

```

vertices.each_index do |i|
  v1 = vertices[i]
  v2 = vertices[(i + 1) % vertices.length]
  e = Vertex.new((v2.x-v1.x)!=0 ? (v2.x-v1.x)/(v2.x-v1.x).abs : 0, (v2.y-v1.y)!=0 ? (v2.y-v1.y)/(v2.y-v1.y).abs : 0)
  v = v1 - e
  # print "http://wiki.math.bme.hu/v1="http://wiki.math.bme.hu + v1.to_s + "http://wiki.math.bme.hu/v2="http://wiki.math.bme.hu + v2.to_s + "http://wiki.math.bme.hu/e="http://wiki.math.bme.hu + e.to_s + "http://wiki.math.bme.hu/v="http://wiki.math.bme.hu + v.to_s + "http://wiki.math.bme.hu"
  while (v != v2) do
    v += e
    # print "http://wiki.math.bme.hu/tv="http://wiki.math.bme.hu + v.to_s + "http://wiki.math.bme.hu"
    self.canvas[v.y][v.x] = 1
  end
end
end
# fill polygon
self.canvas.each {|row|
  min = row.index(1)
  max = row.rindex(1)
  if(min && max) then
    for x in min..max do row[x] = 1 end
  end
}
end
end

dim=[]
vertices=[]
STDIN.each_line { |s|
  begin
    if(dim.empty?)
      dim=s.split(/\s+/)
    else
      vertex = s.split(/\s+/)
      if(vertex[0].to_i == -1 and vertex[1].to_i == -1)
        poly = Polygon.new(dim, vertices)
        poly.render()
        poly.emit()
        dim=[]
        vertices=[]
      else
        v = Vertex.new(vertex[0].to_i, vertex[1].to_i)
        vertices.push(v)
      end
    end
  end
rescue AngleException
  print "http://wiki.math.bme.hu/ANGLE\n"http://wiki.math.bme.hu
rescue NonConvexException
  print "http://wiki.math.bme.hu/NOT_CONVEX\n"http://wiki.math.bme.hu
end
}

```