Homework 3.2 – Ruby

Mandy Noto

Dr. Jozo Dujmović

19 May 2023

CSc 600.01

Contents

1 Iterator demo **1**

Listing 1 Source code for 1iteratordemo.rb 1

Figure 1 Screenshot output of executing *1iteratordemo.rb* 1

2 Recognizer **2**

Listing 2 Source code for 2recognizer.rb 2

Figure 2 Screenshot output of executing *2recognizer.rb* 2

3 Triangle **3**

Listing 3 Source code for 3triangle.rb 3

Listing 4 Source code for test-cases of 3triangle.rb 4

Figure 3 Screenshot output of executing *3triangle.rb* 4

4 Sphere **5**

Listing 5 Source code for 4sphere.rb 5

Figure 4 Screenshot output of executing *4sphere.rb* 5

5 References **6**

1 Iterator demo

Listing 1 The source code for *1iteratordemo.rb*.

```ruby
i=4; # initializes an empty integer for counting
names=["john","peter","job"] # initializes an array of names
abbreviations={a:"alpha", b:"beta", c:"charlie"} # is a hash of abbreviations

print"while:\t\t"    ; print "#{i-=1}" while i>1 # counts down from 3 to 1
print"\nuntil:\t\t"  ; until i>3 ;print i ; i+=1 end# counts up to 3
print"\nfor:\t\t"    ; for name in names; print"#{name} " end
print"\nupto:\t\t"   ; 4.upto(7){|n|print n}
print"\ndownto:\t\t" ; 7.downto(4){|n|print n}
print"\ntimes:\t\t"  ; 3.times{|n|print "thanks "}
print"\neach:\t\t"   ; abbreviations.each do |k,v| print"#{k}=#{v} " end
print"\nmap:\t\t"    ; [1,2,3,4].map{|x|print "#{x}.is.even " if x.even?}
print"\nstep:\t\t"   ; 0.step(10,2) do |x| print"*" end
print"\ncollect:\t"  ; squares=[2,3,4].collect{|x|x*x}; print squares
print"\nselect:\t\t"   ; divby9=[9,7,45,18,8].select{|x|x%9==0};print divby9
print"\nreject:\t\t"   ; notdivby9=[9,7,45,18,8].reject{|x|x%9==0};print notdivby9
```

```
~/school/2023spring/csc600-01/hw3-2-ruby/code at 19:07:43
❯ ruby 1iteratordemo.rb
while:          321
until:          123
for:            john peter job
upto:           4567
downto:         7654
times:          thanks thanks thanks
each:           a=alpha b=beta c=charlie
map:            2.is.even 4.is.even
step:           ******
collect:        [4, 9, 16]
select:         [9, 45, 18]
reject:         [7, 8]%

~/school/2023spring/csc600-01/hw3-2-ruby/code at 19:07:50
❯
```

Figure 1 Screenshot output of executing *1iteratordemo.rb*

2 Recognizer

Listing 2 The source code for *2recognizer.rb*.

```ruby
class Array
        #  Return true if amin <= a[i] <=amax for all values of i.
        def limited?(amin,amax)
                # Try to disprove the range
                for e in self do return false if amin>e||e>amax end
                return true # implies disapproval failed
        end
        # Return 0,-1,1 depending on how array is sorted.
        def sorted?
                inc,dec=true,true # assumes array is sorted in either direction
                0.upto self.length-2 do |i| # tries to disprove assumption
                        inc=false if self[i]>self[i+1]; dec=false if self[i]<self[i+1]
                end
                if inc then 1 elsif dec then -1 else 0 end # concludes assumption
        end
end

a=[2,4,6,8]; b=[1,2,3,1]
print a.limited?(2,4);puts; print a.limited?(1,9);puts
print b.limited?(1,2);puts; print b.limited?(1,3);puts
print [3,2,1,1].sorted?
print [0,0,0].sorted?; print [1,2,3].sorted?
print [3,2,1].sorted?; print [1,2,1].sorted?
```

Figure 2 Screenshot output of executing *2recognizer.rb*

3 Triangle

Listing 3 The source code for *3triangle.rb*.

```ruby
class Triangle
        def initialize(a,b,c) @a,@b,@c=a,b,c end # is the initializer.
        def sidea; @a end; def sideb; @b end; def sidec; @c end # are setters.
        def sidea=(a); @a=a end; def sideb=(b); @b=b end; def sideb=(c); @c=c end # are getters.

        def test # returns the type of this triangle.
                type=3 # (3) assumes this triangle is scalene, otherwise it classifies it as:
                if(@a==@b)and(@a==@c) then type=1 # (1) equilateral,
                elsif(@a==@b)or(@a==@c)or(@b==@c) then type=2 # (2) isosceles,
                elsif(@a**2==(@b**2+@c**2))or(@b**2==(@a**2+@c**2))or(@c**2==(@a**2+@b**2))
                        then type=4 # (4) right,
                elsif (@a>=(@b+@c))or(@b>=(@a+@c))or(@c>=(@b+@a)) then type=5 # or (5) invalid.
                end
                return type
        end

        def perimeter # returns the sum of all sides of this triangle.
                false if self.test==5;
                @a+@b+@c
        end

        def area; # returns calculated area of this triangle with heron's formula.
                false if self.test==5; s=self.perimeter/2.to_f
                Math.sqrt s*((s-@a)*(s-@b)*(s-@c))
        end

        def pp; print"(#{@a},#{@b},#{@c})" end # prints all sides of this triangle.
end

```

Please turn the page for test cases and its screenshot output. ↺

Listing 4 The source code for *3triangle.rb* at the test-cases part.

```
29
30 t1=Triangle.new(3,3,3); t2=Triangle.new(7,7,4); t3=Triangle.new(7,12,15)
31 t4=Triangle.new(3,4,5); t5=Triangle.new(3,6,2)
32 puts "#{t1.pp}\t  test=#{t1.test}"; puts "#{t2.pp}\t  test=#{t2.test}"
33 puts "#{t3.pp} test=#{t3.test}"; puts "#{t4.pp}\t  test=#{t4.test}"
34 puts "#{t5.pp}   test=#{t5.test}"
35 puts "#{t1.pp} perimeter=#{t1.perimeter} area=#{t1.area}"
36 puts "#{t2.pp} perimeter=#{t2.perimeter} area=#{t2.area}"
37 print "#{t1.pp}=>"; t1.sidea=5; print"#{t1.pp}:"
38 print "perimeter=#{t1.perimeter} area=#{t1.area}\n"
39 print "#{t2.pp}=>"; t2.sideb=9; print"#{t2.pp}:"
40 print "perimeter=#{t2.perimeter} area=#{t2.area}\n"
```

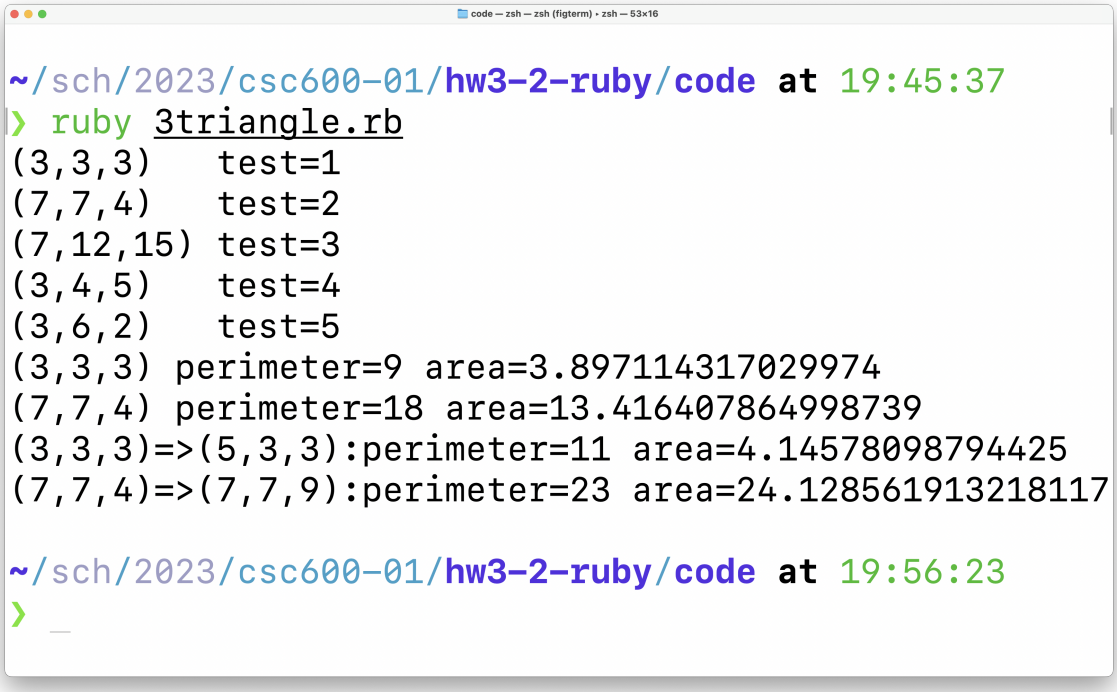

Figure 3 Screenshot output of executing *3triangle.rb*

4 Sphere

Listing 5 The source code for *4sphere.rb*.

```ruby
class Sphere
        def initialize(r); @radius=r end
        def area; 4*(@radius**2)*Math::PI end # returns this sphere's area
        def volume; (4*(@radius**3)*Math::PI)/3 end # returns this sphere's volume
end
class Ball < Sphere
        def initialize(r,c); super(r); @color=c end
end
class MyBall < Ball
        def initialize(r,c,o); super(r,c); @owner=o end
        # Show current and inherited instance variables
        def show; print "@radius=#{@radius}\n@color=#{@color}\n@owner=#{@owner}" end
end

mb=MyBall.new(5,"red","jon")
mb.show; puts
print "area=#{mb.area}\nvolume=#{mb.volume}"
```

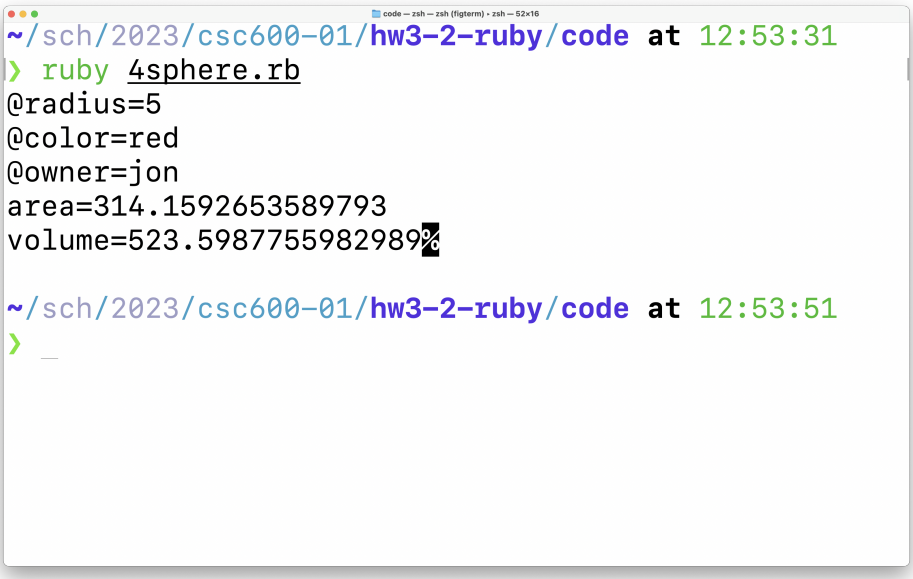

Figure 4 Screenshot output of executing *4sphere.rb*

5. References

[1] Saman MohamadiSaman Mohamadi et al. *Ruby add method to a class*. 1962. URL: https://stackoverflow.com/questions/34528649/ruby-add-method-to-a-class.

[2] URL: https://docs.ruby-lang.org/en/2.0.0/syntax/methods_rdoc.html#label-Method+Names.

[3] URL: https://www.wikihow.com/Determine-if-Three-Side-Lengths-Are-a-Triangle#:~:text=All%20you%20have%20to%20do.

[4] Dominik Czernia. *Area of a Sphere. Calculator — Formula*. URL: https://www.omnicalculator.com/math/area-of-sphere.

[5] URL: https://www.cuemath.com/measurement/volume-of-sphere/.