



**«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**


ФАКУЛЬТЕТ _____ ИНФОРМАТИКИ И СИСТЕМ УПРАВЛЕНИЯ
КАФЕДРА _____ КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ 6)

О т ч е т
по лабораторной работе № 7

Дисциплина: Языки программирования

Название лабораторной работы: Консольное приложение Ruby

Студент гр. ИУ6-34



(Подпись, дата)

Габолаев Г.К.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Изория В.Ф.

(И.О. Фамилия)

Москва, 2017

Часть 1:

Коды программ:

```
> cat src.rb
1. def create_files
2.   File.open('F.txt', 'w') do |f|
3.     rand(10_000).times do
4.       f.write("#{rand(1000)} \n")
5.     end
6.   end
7.
8.   File.open('F.txt') do |f|
9.     g = File.open('G.txt', 'w')
10.    h = File.open('H.txt', 'w')
11.    f.readlines.each { |line| line.to_i.even? ? g.write(line) :
      h.write(line) }
12.  end
13.end
```

```
> cat user.rb
1. require_relative '../part1/src'
2.
3. if $PROGRAM_NAME == __FILE__
4.   puts('Press ENTER for start...')
5.   gets
6.   create_files
7.   puts('Files F.txt, G.txt, H.txt has been created.')
8. end
```

```
> cat src_test.rb
1. require 'minitest/autorun'
2.
3. # really useful class
4. class MyTest < MiniTest::Test
5.   # I sincerely tried to come up with tests for this job, but alas
6. end
```

Результат выполнения:

```
> ruby user.rb
Press ENTER for start...

Files F.txt, G.txt, H.txt has been created.
```

Г	Д	В
283	380	283
483	822	483
380	726	573
573	82	731
822	820	199
731	720	321
199	184	169
321	562	259
726	126	317
82	824	249
820	624	877
и т. д.	и т. д.	и т. д.

Результат тестирования:

```
> ruby src_test.rb
Run options: --seed 7064

# Running:

Finished in 0.000917s, 0.0000 runs/s, 0.0000 assertions/s.

0 runs, 0 assertions, 0 failures, 0 errors, 0 skips
```

Rubocop:

```
> rubocop *
Inspecting 3 files
...

3 files inspected, no offenses detected
```

Часть 2

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработанных классов на конкретных данных.

Объект — точка на плоскости, заданная радиальными координатами, умеющая выводить их на экран и возвращать в ответ на запрос в декартовых координатах.

Объект — окружность, заданная центром и радиусом окружности, умеющая выводить их на экран и возвращать в ответ на запрос данные о координатах центра и радиусе.

В тестирующей программе обеспечить автоматическую проверку того, что созданные объекты действительно соответствуют заданной иерархии классов.

Коды программ:

```
> cat src.rb
1. # Expanded standard class
2. class Numeric
3.   def degrees
4.     self * Math::PI / 180
5.   end
6. end
7.
8. # Point class definition
9. class Point
10.  def initialize(input_rad, input_phi)
11.    @rad = input_rad.to_f
12.    @phi = input_phi.to_f
13.  end
14.
15.  def print_coords
16.    puts("Rad = #{@rad}, Degree = #{@phi}")
17.  end
18.
19.  def cartesian_x
20.    (@rad * Math.cos(@phi.degrees)).round(2)
21.  end
22.
23.  def cartesian_y
24.    (@rad * Math.sin(@phi.degrees)).round(2)
25.  end
26.  attr_reader :rad, :phi
27.  private :rad, :phi
28. end
29.
30. # Circle class definition.
31. class Circle
32.  def initialize(center_rad, center_deg, circle_rad)
33.    @circle_center = Point.new(center_rad, center_deg)
34.    @circle_rad = circle_rad
35.  end
36.
37.  def print_info
38.    puts 'Circle center polar coordinates:'
39.    puts @circle_center.print_coords
40.    puts "Circle radius is a #{@circle_rad}"
41.  end
42.
43.  def center_x
44.    @circle_center.cartesian_x
45.  end
46.
47.  def center_y
48.    @circle_center.cartesian_y
49.  end
50.
51.  def center_rad
52.    @circle_rad
```

```

53. end
54. attr_reader :circle_rad, :circle_center
55. private :circle_rad
56.end

```

```

> cat user.rb
1. require_relative '../part2/src'
2.
3. if $PROGRAM_NAME == __FILE__
4.   print('Enter the radius of the center of the circle: ')
5.   rad = gets
6.   print('Enter the angle of the center of the circle: ')
7.   degree = gets
8.   print('Enter the radius of the circle: ')
9.   radius = gets
10.  [rad, degree, radius].each do |item|
11.    raise(TypeError, 'Wrong input value type') unless item =~ /
    ^[0-9]+$/
12.  end
13.  circle = Circle.new(rad, degree, radius)
14.  circle.print_info
15.  puts 'Circle center cartesian coordinates:'
16.  puts "x = #{circle.center_x}"
17.  puts "y = #{circle.center_y}"
18.end

```

```

> cat src_test.rb
1. require 'minitest/autorun'
2. require_relative '../part2/src'
3.
4. # tests of relatives and values
5. class MyTest < MiniTest::Test
6.   def test_relatives
7.     assert_kind_of Point, Circle.new(10, 30, 10).circle_center
8.   end
9.
10.  def test_value_rad
11.    assert_equal 10, Circle.new(10, 30, 10).center_rad
12.  end
13.
14.  def test_value_x
15.    assert_equal 8.66, Circle.new(10, 30, 10).center_x
16.  end
17.
18.  def test_value_y
19.    assert_equal 5.0, Circle.new(10, 30, 10).center_y
20.  end
21.
22.  def test_fail_privacy
23.    assert_raises NoMethodError do
24.      Circle.new(10, 30, 10).circle_center.rad
25.    end
26.  end
27.end

```

Результат выполнения:

```
> ruby user.rb
Enter the radius of the center of the circle: 30
Enter the angle of the center of the circle: 60
Enter the radius of the circle: 10
Circle center polar coordinates:
Rad = 30.0, Degree = 60.0

Circle radius is a 10
Circle center cartesian coordinates:
x = 15.0
y = 25.98
```

Результат тестирования:

```
> ruby src_test.rb
Run options: --seed 60978

# Running:

.....

Finished in 0.002054s, 2434.2745 runs/s, 2434.2745 assertions/s.

5 runs, 5 assertions, 0 failures, 0 errors, 0 skips
```

Rubocop:

```
> rubocop *
Inspecting 3 files
...

3 files inspected, no offenses detected
```