



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ОТЧЕТ

по лабораторной работе № 7

Название:

Ruby - Файлы и объектная модель

Дисциплина: Языки Интернет-программирования

Студент

ИУ-326
(Группа)

(Подпись, дата)

С.В. Астахов
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Задание 1

Дан символьный файл **F**. Найти самое длинное слово среди слов, первая буква которых есть «а». Если слов с наибольшей длиной несколько, то найти последнее из них. Если таких слов нет вообще, то сообщить об этом.

Автоматический тест программы обязательно должен проверять работу с файлами.

Исходный код **Searcher.rb (логика)**

```
#!/usr/bin/ruby
# frozen_string_literal: true

# searches in file for the longest word starting with 'a'
module Searcher
  def self.search(path_arg)
    result = ""
    path = path_arg.strip
    if (File.exist? path) == false
      result = 'file doesnt exist'
    else
      IO.read(path).split.each { |word| result = word if (word[0] == 'a') &&
(word.length >= result.length) }
      # result = IO.read(path).split.select { |word| word.start_with?('a') }.max {|a1,
a2| a2.length <=> a1.length }
    end
    result = 'no target strings in file' if result.empty?
    result
  end
end
```

Src.rb (консольный интерфейс)

```
# frozen_string_literal: true

require_relative 'searcher'

puts "enter relative path to file"
path = gets
print Searcher.search(path)
```

Test.rb (автотесты)

```
# frozen_string_literal: true

require './searcher.rb'
require 'minitest/autorun'

# inherits default test class
class TestFunc < Minitest::Test
  def setup; end

  def test_file_generated
    File.open('generated.txt', 'w') { |file| file.write "app application zero \n standart all word" }
    result = Searcher.search('generated.txt')
    File.delete('generated.txt')
    assert_equal result, 'application'
  end

  def test_file_abrocadabra
    assert_equal Searcher.search('f.txt'), 'abrocadaboa'
  end

  def test_file_ambassador
    assert_equal Searcher.search('f2.txt'), 'ambassador'
  end

  def test_file_empty
    assert_equal Searcher.search('f3.txt'), 'no target strings in file'
  end

  def test_file_incorrect
    assert_equal Searcher.search('incorrect path'), 'file doesnt exist'
  end

  def teardown; end
end
```

Результат выполнения с произвольными входными данными

```
enter relative path to file
f.txt
abrocadaboa
```

Результаты автотестов

Run options: --seed 2756

Running:

.....

Finished in 0.023971s, 208.5862 runs/s, 208.5862 assertions/s.
5 runs, 5 assertions, 0 failures, 0 errors, 0 skips

Вывод Rubocop

Inspecting 3 files

...

3 files inspected, no offenses detected

Вывод Reek

Inspecting 3 file(s):

S..

searcher.rb -- 1 warning:

[6]:TooManyStatements: Searcher#self.search has approx 7 statements
[<https://github.com/troessner/reek/blob/v6.0.1/docs/Too-Many-Statements.md>]
1 total warning

Задание 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 — доска. Параметры: два целых числа — длина и ширина, Методы: инициализирующий, определения площади и возвращающие значения длины и ширины.

Объект — доска. Параметры: длина, ширина и толщина. Методы: инициализирующий, определения объема и вывода параметров и объема.

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

Исходный код Boards.rb (логика)

```
#!/usr/bin/ruby
# frozen_string_literal: true

# simplified board with only 2 coords
class FlatBoard
  def initialize(in_width, in_height)
    @width = in_width
    @height = in_height
  end

  def square
    @width * @height
  end
end

# sophisticated board with 3 coords
class VolumetricBoard < FlatBoard
  def initialize(in_width, in_height, in_depth)
    super(in_width, in_height)
    @depth = in_depth
  end

  def volume
    @depth * square
  end
end
```

```
end  
end
```

Src.rb (консольный интерфейс)

```
# frozen_string_literal: true  
  
require_relative 'boards'  
  
puts "\n\nInput x, y for flat board\n"  
width = gets.to_i  
height = gets.to_i  
board_s = FlatBoard.new(width, height)  
print "Square of flat board: #{board_s.square}"  
puts "\n\nInput x, y, z for flat board\n"  
width = gets.to_i  
height = gets.to_i  
depth = gets.to_i  
board_v = VolumetricBoard.new(width, height, depth)  
print "Square of volumetric board: #{board_v.square}\n"  
print "Volume of volumetric board: #{board_v.volume}"
```

Test.rb (автотесты)

```
# frozen_string_literal: true  
  
require './boards.rb'  
require 'minitest/autorun'  
  
# inherits default test class  
class TestFunc < Minitest::Test  
  def setup; end  
  
  def test_parent_instance  
    board = FlatBoard.new(1, 1)  
    assert board.instance_of? FlatBoard  
  end  
  
  def test_child_instance  
    board = VolumetricBoard.new(1, 1, 1)  
    assert board.instance_of? VolumetricBoard  
  end  
  
  def test_hierarchy  
    board = VolumetricBoard.new(1, 1, 1)  
    assert board.is_a? FlatBoard
```

```

end

def test_squares
  board_s = FlatBoard.new(2, 3)
  board_v = VolumetricBoard.new(2, 7, 4)
  assert (board_s.square == 6) && (board_v.square == 14)
end

def test_volume
  board_v = VolumetricBoard.new(2, 7, 4)
  assert_equal board_v.volume, 56 # 2*7*4
end

def teardown; end
end

```

Результат выполнения с произвольными входными данными

Input x, y for flat board

2

3

Square of flat board: 6

Input x, y, z for flat board

2

3

4

Square of volumetric board: 6

Volume of volumetric board: 24

Результаты автотестов

Run options: --seed 65466

Running:

.....

Finished in 0.003784s, 1321.4229 runs/s, 1321.4229 assertions/s.

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

Вывод Rubocop

Inspecting 3 files

...

3 files inspected, no offenses detected

Вывод Reek

Inspecting 3 file(s):

...

0 total warnings

Вывод

В ходе данной работы были изучены основы работы с файлами, основы объектной модели и ее тестирования в Ruby