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 № 5

Название: Введение в язык Ruby

Дисциплина: Языки Интернет-программирования

Студент ИУ-32б _____ С.В. Астахов
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ _____
(Подпись, дата) (И.О. Фамилия)

Москва, 2020

Задание 1

Вычислить: $b = \dfrac{1 + \cos(y - 2)}{\dfrac{x^4}{2} + \sin^2(z)}$.

Исходный код
Count.rb (логика)

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

# counts given function
module Counter
  def self.count(x, y, z)
        (1 + Math.cos(y.to_f - 2)) / ((x.to_f**4) / 2 + (Math.sin(z.to_f)**2))
  end
end
```

Src.rb (консольный интерфейс)

```
# frozen_string_literal: true

require_relative 'count'
require_relative 'count'

print "Input x, y, z to calculate function \n"
x1 = gets
y1 = gets
z1 = gets
printf('result: %<res>3.2f', res: Counter.count(x1, y1, z1))
```

Test.rb (автотесты)

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

require './count.rb'
require 'minitest/autorun'

# inherits default test class
```

```
class TestFunc < Minitest::Test
  def setup; end

  def test_default
        assert_in_delta Counter.count(1, 1, 1), 1.28, 0.01, 'Test 1,1,1'
  end

  def test_infinity
        assert_equal Counter.count(0, 1, 0), Float::INFINITY, 'Test 0,1,0 -> Inf'
  end

  def test_sqrt_two
        assert_in_delta Counter.count(Math.sqrt(2), 2, 0), 1, 0.01, 'Test sqrt2, 2, 0 -> 1'
  end

  def test_cos_pi
        assert_in_delta Counter.count(1, Math::PI + 2, 1), 0, 0.01, 'Test 1, pi + 2, 1 -> 0'
  end

  def teardown; end
end
```

Результат выполнения с произвольными входными данными

```
Input x, y, z to calculate function
1
1
1
result: 1.28
```

Результаты автотестов

```
Run options: --seed 44693

# Running:

....

Finished in 0.002547s, 1570.3518 runs/s, 1570.3518 assertions/s.
4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
```

Вывод Rubocop

Offenses:

count.rb:6:18: C: Naming/MethodParameterName: Method parameter must be at least 3 characters long.
  def self.count(x, y, z)
                 ^
count.rb:6:21: C: Naming/MethodParameterName: Method parameter must be at least 3 characters long.
  def self.count(x, y, z)
                 ^
count.rb:6:24: C: Naming/MethodParameterName: Method parameter must be at least 3 characters long.
  def self.count(x, y, z)

Вывод Reek

count.rb -- 3 warnings:
  [6]:UncommunicativeParameterName: Counter#self.count has the parameter name 'x' [https://github.com/troessner/reek/blob/v6.0.1/docs/Uncommunicative-Parameter-Name.md]
  [6]:UncommunicativeParameterName: Counter#self.count has the parameter name 'y' [https://github.com/troessner/reek/blob/v6.0.1/docs/Uncommunicative-Parameter-Name.md]
  [6]:UncommunicativeParameterName: Counter#self.count has the parameter name 'z' [https://github.com/troessner/reek/blob/v6.0.1/docs/Uncommunicative-Parameter-Name.md]

Пояснение

Обе программы выдают предупреждение, что имена параметров функции Count - x, y, z не несут в себе достаточной смысловой нагрузки. Однако так как исходное задание не предоставляет дополнительной информации касательно их реального смысла, возможно лишь обойти ограничения reek и rubocop сменив название, но не повысив этим реальную читаемость исходного кода.

Задание 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в удалении лишних пробелов и слов, состоящих из одного символа. Лишними считаются пробелы в начале и конце строки, а также более одного пробела между словами. Вывести на печать исходную и скорректированную последовательности строк.

Автоматический тест программы обязательно должен генерировать случайные строки в соответствии с правилами, перечисленными в задании.

Исходный код

Parser.rb (логика)

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

# Class that modifies string
module Parser
  def self.parse(input_s)
        input_s.split.reject { |elem| elem.to_s.length == 1 }.join(' ')
  end
end
```

Src.rb (консольный интерфейс)

```
# frozen_string_literal: true

require_relative 'parser'
my_string = 'placeholder'
until my_string.strip.empty?
  print "\n\nInput string to remove spaces and single letters (OR EMPTY TO EXIT)\n"
  my_string = gets
  if my_string.strip.empty? == false
        print "echo: #{my_string}"
        print "formatted: #{Parser.parse(my_string)}"
  end
end
```

Test.rb (автотесты)

```
# frozen_string_literal: true

require './parser.rb'
require 'minitest/autorun'

# Testing Parser module
class TestFunc < Minitest::Test
  def setup
        # use regex variable
        # o = [('a'..'z'), ('A'..'Z'), (' '..' ')].map(&:to_a).flatten
        arg = ''
        alphabet = ('a'..'z').to_a
        alphabet.push([' '] * 10 + ['  '] * 5)

        arg = alphabet.sample(rand(14..18)).to_s while
arg.scan(/([\s]{2,})|(\s\w\s)/m).empty?

        @test_string = arg
  end

  def test_random
        assert Parser.parse(@test_string).scan(/([\s]{2,})|(\s\w\s)/m).size.zero?
  end

  def test_bad_string
        assert Parser.parse('a string   full o f mis
takes').scan(/([\s]{2,})|(\s\w\s)/m).size.zero?
  end

  def test_good_string
        assert_equal Parser.parse('nice string with nothing to cut'), 'nice string with
nothing to cut'
  end

  def teardown; end
end
```

Результат выполнения с произвольными входными данными

Input string to remove spaces and single letters (OR EMPTY TO EXIT)
the re is o o not a good one g
echo: the re is o o not a good one g
formatted: the re is not good one

Input string to remove spaces and single letters (OR EMPTY TO EXIT)

(выход)

Результаты автотестов

Run options: --seed 6145

Running:

...

Finished in 0.003542s, 847.0269 runs/s, 847.0269 assertions/s.
3 runs, 3 assertions, 0 failures, 0 errors, 0 skips

Вывод Rubocop

Inspecting 3 files
...

3 files inspected, no offenses detected

Вывод Reek

test.rb -- 1 warning:
  [7]:InstanceVariableAssumption: TestFunc assumes too much for instance variable '@test_string'
[https://github.com/troessner/reek/blob/v6.0.1/docs/Instance-Variable-Assumption.md]
1 total warning

Вывод

В ходе данной работы были изучены основные конструкции языка Ruby, освоены базовые навыки написания автоматических тестов и применения средств для проверки исходного кода на соответствие стандартам и читаемость (reek и rubocop)