



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

О Т Ч Е Т

по лабораторной работе № 7


Дисциплина: Языки интернет-программирования

Вариант: 23

Студент

ИУ6-33Б

(Группа)


(Подпись, дата)

Д. И. Мироненко

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В. Д. Шульман

(И.О. Фамилия)

Москва, 2023

Часть 1

Задание

Сформировать программным путем символьный файл F, содержащий слова. Определить количество слов в файле F, имеющих длину 5 символов.

Автоматический тест программы обязательно должен проверять работу с файлами.

Код программы

Файл “main.rb”

```
# frozen_string_literal: true

require_relative 'utils'

generate_words('words.txt')
puts "Words with a length of 5: #{count_words(5, 'words.txt')}"
```

Файл “utils.rb”

```
# frozen_string_literal: true

def count_words(target_len, source)
  count = 0

  File.open(source, 'r') do |file|
    file.each do |line|
      words = line.split(' ')
      words.each do |word|
        count += 1 if word.length == target_len
      end
    end
  end

  count
end

def generate_words(path)
  words = []
  letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'

  rand(1000).times do
    word = ""
    rand(20).times do
      word += letters[rand(letters.length)]
    end
    words << word
  end

  File.open(path, 'w') do |file|
    file.write(words.join(' '))
  end
end
```

Файл “test_utils.rb”

```
# frozen_string_literal: true

require 'minitest/autorun'
require_relative 'utils'

# Test 'count_words'
class TestCountWords < Minitest::Test
  @@path = './test_words.txt'

  def setup
    File.open(@@path, 'w') do |_|
      end
  end

  def teardown
    File.delete(@@path) do |_|
      end
  end

  def insert(content)
    File.open(@@path, 'w') do |file|
      file.write(content)
    end
  end

  def test_empty
    assert_equal 0, count_words(5, @@path)
  end

  def test_regular
    insert("asdf iur\nriwby\nnirbfa ebr")
    assert_equal 2, count_words(5, @@path)
  end
end
```

Работа программы

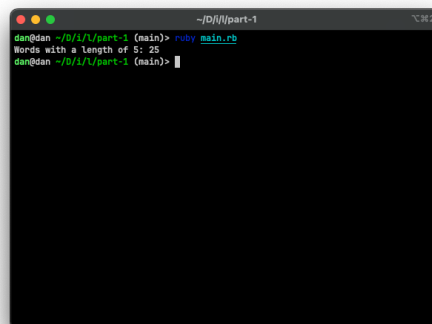


Рисунок 1 – работа программы

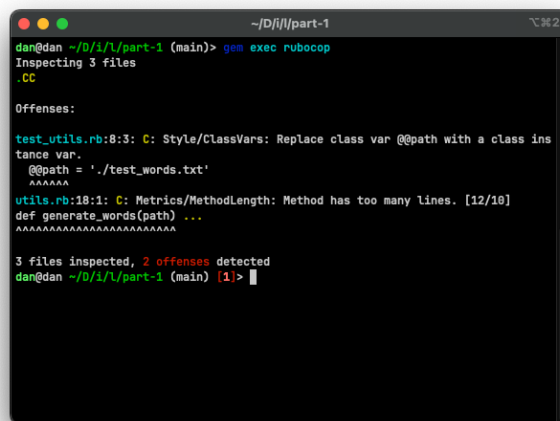
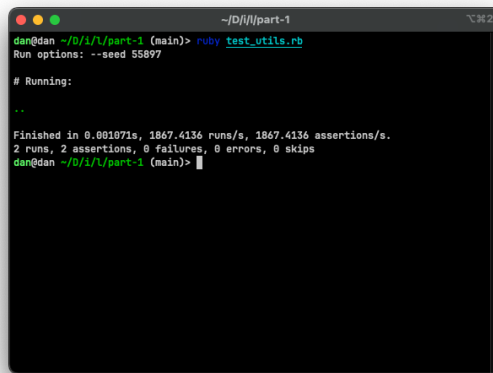


Рисунок 3 – проверка кода rubosor'ом

Часть 2

Задание

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 — окно, хранящее размеры и умеющее выводить на экран площадь.

Объект — окно, хранящее размеры и сведения о наличии жалюзи и умеющее выводить на экран площадь и признак наличия жалюзи.

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

Код программы

Файл “main.rb”

```
# frozen_string_literal: true

require_relative 'utils'

window = Window.new(5, 6)
puts "square? #{window.square}"

advanced_window = AdvancedWindow.new(5, 6, true)
puts "jalousie? #{advanced_window.jalousie?}"
```

Файл “utils.rb”

```
# frozen_string_literal: true

# window class
class Window
  attr_reader :width, :height

  def initialize(width, height)
    if !(width.is_a?(Integer) || width.is_a?(Float)) || !width.positive?
      raise TypeError, "'width' must be a positive integer!"
    end

    if !(height.is_a?(Integer) || height.is_a?(Float)) || !height.positive?
      raise TypeError, "'height' must be a positive integer!"
    end

    @width = width
    @height = height
  end

  def square
    width * height
  end
end

# advanced window class
class AdvancedWindow < Window
  attr_reader :jalousie

  def initialize(width, height, jalousie)
    raise TypeError, "'jalousie' must be a positive integer!" unless [true, false].include? jalousie

    super width, height
    @jalousie = jalousie
  end

  def jalousie?
    @jalousie
  end
end
```

Файл “test_utils.rb”

```

# frozen_string_literal: true

require 'minitest/autorun'
require_relative 'utils'

# Test 'Window'
class TestWindow < Minitest::Test
  def test_square
    window = Window.new(5, 6)
    assert_equal 30, window.square
  end

  def test_negative_width
    assert_raises TypeError do
      Window.new(4, -2)
    end
  end

  def test_negative_height
    assert_raises TypeError do
      Window.new(-4, 2)
    end
  end

  def test_zero
    assert_raises TypeError do
      Window.new(0, 4)
    end
  end

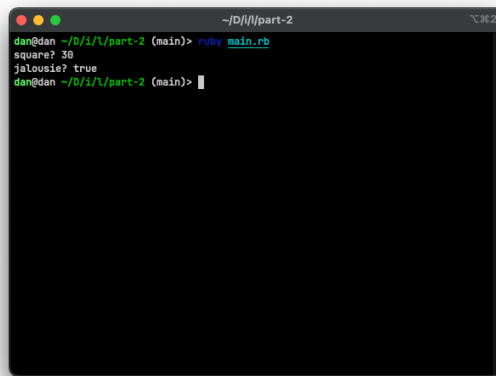
  def test_boolean
    assert_raises TypeError do
      Window.new(5, false)
    end
  end
end

# Test 'AdvancedWindow'
class TestAdvancedWindow < Minitest::Test
  def test_jalousie
    assert_equal true, AdvancedWindow.new(5, 6, true).jalousie?
  end

  def test_jalousie_invalid_type
    assert_raises TypeError do
      AdvancedWindow.new(5, 6, 1)
    end
  end
end

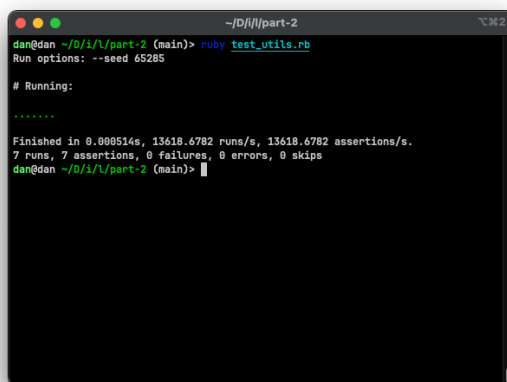
```

Работа программы



```
dan@dan ~/D/i/l/part-2 (main)> ruby main.rb
square? 36
jealousie? true
dan@dan ~/D/i/l/part-2 (main)>
```

Рисунок 4 – работа программы

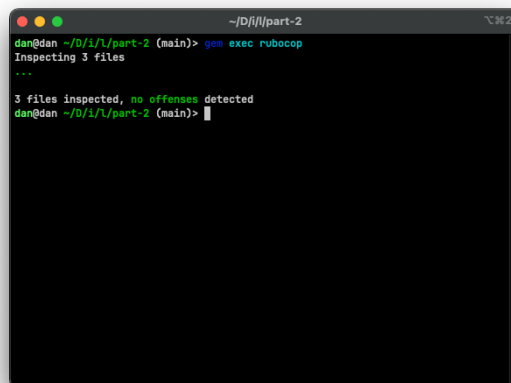


```
dan@dan ~/D/i/l/part-2 (main)> ruby test_utils.rb
Run options: --seed 65285

# Running:

*****
Finished in 0.008514s, 13618.6782 runs/s, 13618.6782 assertions/s.
7 runs, 7 assertions, 0 failures, 0 errors, 0 skips
dan@dan ~/D/i/l/part-2 (main)>
```

Рисунок 5 – выполнение unit-тестов



```
dan@dan ~/D/i/l/part-2 (main)> gem exec rubocop
Inspecting 3 files
...
3 files inspected, no offenses detected
dan@dan ~/D/i/l/part-2 (main)>
```

Рисунок 6 – проверка кода rubocop’ом

Вывод

Реализованы несколько программ на языке программирования ruby; функции всех программ проверена на unit-тестах; исходный код проверен линтером rubocop на наличие ошибок.