



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 09.03.03 «Прикладная информатика»

## О Т Ч Е Т

по лабораторной работе № 7

Название: Ruby. Классы, модули, примеси.

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-34Б

(Группа)

30.10.2022

(Подпись, дата)

С. А. Рахманов

(И.О. Фамилия)

Преподаватель

Д. В. Малахов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

**Цель работы:** получение навыков программирования на языке Ruby с подключением файлов и взаимодействием с ними, работа со средствами проверки соответствия стиля программирования.

### **Задание:**

Все консольные приложения Ruby следует реализовывать в виде трех отдельных файлов:

1. основная программа;
2. программа для взаимодействия с пользователем через консоль;
3. программа для автоматического тестирования на основе `MiniTest::Unit` или `RSpec`. Везде, где это возможно, данные для проверки должны формироваться автоматически по правилам, указанным в задании.

При реализации программ везде, где это возможно, следует избегать использования циклов `for`, `do`, `while`. Вместо них используйте методы из примеси `Enumerable`.

Все тексты программ должны быть проверены на соответствие стилю программирования Ruby при помощи `rubocop.ru` или `reek`.

## **Часть 1**

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.

## **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 — 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.

## **Тексты программ**

## Часть 1

### interface.rb

```
# frozen_string_literal: true

require './main'

loop do
  print 'Start? [y/n]'
  start = gets.chomp
  if start == 'y'
    print "words with 5 letters in file = #{calc_words5(true)}\n"
  else
    exit
  end
end
```

### main.rb

```
# frozen_string_literal: true

def generate_word
  (1..(rand(10))).map { rand(97..122).chr }.join
end

def generate_str
  str = ''
  rand(1..10).times do
    str += "#{generate_word} "
  end
  str += "\n"
end

def generate_text
  text = ''
  rand(1..10).times do
    text += generate_str
  end
  text
end

def generate_file(text)
  f = File.open('F.txt', 'w')
  f.write(text)
  f.close
end

def calc_words5(file_creation)
  if file_creation
    some_text = generate_text
    generate_file(some_text)
  end
  f = File.open('F.txt', 'r')
  arr = []
  f.each { |str| arr << str.split } # ; p str.chomp
  f.close
  arr.flatten!
  arr.each.inject(0) { |acc, word| word.length == 5 ? acc + 1 : acc }
end
```

### test.rb

```
# frozen_string_literal: true

require 'minitest/autorun'
require './main'
```

```

# Test Class
class TestTree < Minitest::Test
  # first test
  def test_1
    assert_equal(true, File.exist?('F.txt'), "File don't exist")
  end

  # second test
  def test_2
    data = ['Hello my name is John', 'Hello, how are you? Everything is okay
or not',
           "If no block is given, \n a new Enumerator is returned that
includes the index "]
    data.each do |d|
      generate_file(d)
      f = File.open('F.txt', 'r')
      assert_equal(d, f.read, 'Incorrect generate of file')
    end
  end

  # third test
  def test_3
    data = ['Hello my name is John', 'Hello, how are you? Everything is okay
or not',
           "If no block is given, \n a new Enumerator is returned that
includes the index "]
    ans = [1, 0, 2]
    3.times do |ind|
      generate_file(data[ind])
      assert_equal(ans[ind], calc_words5(false), 'Incorrect words number')
    end
  end
end

```

## Часть 2

### interface.rb

```

# frozen_string_literal: true

require_relative 'main'

print "Choose 'window' or 'blind' >>> "
obj = gets.chomp

if obj.downcase == 'window'
  print 'Enter length and width >>> '
  len, wid = gets.split(' ')
  my_window = Window.new(len.to_i, wid.to_i)

  puts "\nSquare: #{my_window.square}"
else
  print 'Enter length, width and blindness >>> '
  len, wid, bld = gets.split(' ')
  my_blind = Blind.new(len.to_i, wid.to_i, bld)

  puts "\nSquare: #{my_blind.square}"
  puts "Is Blind: #{my_blind.blind?}"
end

```

### main.rb

```

# frozen_string_literal: true

```

```

# Class for an ordinary window
class Window
  def initialize(leng, width)
    @len = leng
    @wid = width
  end

  def square
    @len * @wid
  end
end

# Class for an professional window
class Blind < Window
  def initialize(leng, width, bld)
    @bld = true?(bld)
    super(leng, width)
  end

  def true?(str)
    str.downcase == 'true'
  end

  def blind?
    @bld ? true : false
  end
end

```

### test.rb

```

# frozen_string_literal: true

require 'minitest/autorun'
require_relative 'main'

# Class Test
class Test < Minitest::Test
  def setup
    @window = Window.new(10, 5)
    @blind1 = Blind.new(15, 2, 'false')
  end

  def test_1
    assert_equal(true, @window.is_a?(Window))
    assert_equal(true, @blind1.is_a?(Blind))
  end

  def test_2
    assert_equal(false, @window.is_a?(Blind))
    assert_equal(true, @blind1.is_a?(Window))
  end

  def test_3
    assert_equal(false, @blind1.instance_of?(Window))
  end

  def test_4
    res = [Blind, Window, Object, Minitest::Expectations, Kernel,
BasicObject]
    assert_equal(res, @blind1.class.ancestors)
  end
end

```

### **Результаты выполнения**

## Часть 1

```
Терминал
s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab7/part1$ ruby interface.rb
Start? [y/n]y
words with 5 letters in file = 4
Start? [y/n]n
s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab7/part1$ ruby test.rb
Run options: --seed 19659

# Running:

Finished in 0.002746s, 0.0000 runs/s, 0.0000 assertions/s.
0 runs, 0 assertions, 0 failures, 0 errors, 0 skips
s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab7/part1$
```

## Часть 2

```
Терминал
s3r6anita@s3r6anita-Modern-14-B10MW://home/s3r6anita/LoIP/labs/lab7/part2$ ruby interface.rb
Choose 'window' or 'blind' >>> window
Enter length and width >>> 1 3

Square: 3
s3r6anita@s3r6anita-Modern-14-B10MW://home/s3r6anita/LoIP/labs/lab7/part2$ ruby test.rb
Run options: --seed 577

# Running:

Finished in 0.002994s, 0.0000 runs/s, 0.0000 assertions/s.
0 runs, 0 assertions, 0 failures, 0 errors, 0 skips
s3r6anita@s3r6anita-Modern-14-B10MW://home/s3r6anita/LoIP/labs/lab7/part2$
```

## Результаты проверки анализатором rubosor

### Часть 1

```
Терминал
s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab7/part1$ rubocop main.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab7/part1$ rubocop interface.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab7/part1$ rubocop test.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab7/part1$
```

## Часть 2

```
Терминал
s3r6anita@s3r6anita-Modern-14-B10MW://home/s3r6anita/LoIP/labs/lab7/part2$ rubocop main.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
s3r6anita@s3r6anita-Modern-14-B10MW://home/s3r6anita/LoIP/labs/lab7/part2$ rubocop interface.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
s3r6anita@s3r6anita-Modern-14-B10MW://home/s3r6anita/LoIP/labs/lab7/part2$ rubocop test.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
s3r6anita@s3r6anita-Modern-14-B10MW://home/s3r6anita/LoIP/labs/lab7/part2$
```

**Вывод:** получил навыки программирования на языке Ruby с подключением файлов и взаимодействием с ними, также получил опыт работы со средствами проверки соответствия стиля программирования.