



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 № 6

Название: Создание консольного приложения Ruby.

Дисциплина: Языки интернет-программирования

Студент

ИУ6

(Группа)

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2020

# Лабораторная работа №6

## Часть 1:

Решить задачу, организовав итерационный цикл. Вычислить сумму ряда:  $S = \sum_{k=1}^{\infty} \frac{1}{k(k+1)}$  расчет продолжать до тех пор, пока приращение  $R$  будет больше заданной точности  $\xi < 10^{-3}, 10^{-4}$ . Определить, как изменится число итераций при изменении точности.

### Код в console.rb:

```
# frozen_string_literal: true

require_relative 'logic'

puts "eps = 1e-3: res = #{find(1e-3)}"
puts "eps = 1e-4: res = #{find(1e-4)}"
```

### Код в logic.rb:

```
# frozen_string_literal: true

def find(eps = 1e-4)
  steps = 0
  sum1 = 0
  sum2 = 0
  loop do
    steps += 1
    sum2 = sum1 + func(steps)
    break if (sum2 - sum1).abs <= eps

    sum1 = sum2
  end
  [sum2, steps]
end

# :reek:UtilityFunction
def func(k_num)
  (1.0 / (k_num * (k_num + 1))).to_f
end
```

### Код в test.rb:

```
# frozen_string_literal: true

require 'minitest/autorun'
require 'minitest/assertions'
require_relative 'logic'

# class for test
class MyTest < Minitest::Test
  def test_first
    result = find(1e-3)
    assert_in_delta(0.9696969696969695, result[0], 1e-3)
    assert_equal(32, result[1])
  end

  def test_second
    result = find(1e-4)
    assert_in_delta(0.9900990099009898, result[0], 1e-4)
  end
end
```

## Результат выполнения и результаты проверки анализаторами rubocop и reek:

```
Terminal: Local x +
C:\Users\Daria\Desktop\lab6.1>ruby console.rb
eps = 1e-3: res = [0.9696969696969695, 32]
eps = 1e-4: res = [0.9900990099009898, 100]

C:\Users\Daria\Desktop\lab6.1>ruby test.rb
Run options: --seed 60118

# Running:

..

Finished in 0.003854s, 518.9683 runs/s, 778.4524 assertions/s.
2 runs, 3 assertions, 0 failures, 0 errors, 0 skips

C:\Users\Daria\Desktop\lab6.1>rubocop
Inspecting 3 files
...

3 files inspected, no offenses detected

C:\Users\Daria\Desktop\lab6.1>reek
Inspecting 3 file(s):
.S.

logic.rb -- 3 warnings:
[3]:TooManyStatements: find has approx 9 statements [https://github.com/troessner/reek/blob/v6.0.2/docs/Too-Many-Statements.md]
[5, 12]:UncommunicativeVariableName: find has the variable name 'sum1' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.md]
[6, 9]:UncommunicativeVariableName: find has the variable name 'sum2' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.md]
3 total warnings
```

## Часть 2:

Решить предыдущее задание с помощью Enumerator и Enumerable

### Код в console.rb:

```
# frozen_string_literal: true

require_relative 'logic'

puts "eps = 1e-3: res = #{answer(1e-3)[0]}"
puts "eps = 1e-4: res = #{answer(1e-4)[0]}"
```

### Код в logic.rb:

```
# frozen_string_literal: true

def find_series
  Enumerator.new do |yielder|
    steps = 0
    sum1 = 0
    sum2 = 0
    loop do
      steps += 1
      sum2 += func(steps)
      yielder << [sum2, sum1]
      sum1 = sum2
    end
  end
end

# :reek:UtilityFunction
def func(k_num)
  (1.0 / (k_num * (k_num + 1))).to_f
end

def answer(eps)
  find_series.find { |s2, s1| (s2 - s1).abs <= eps }
end
```

## Код в test.rb:

```
# frozen_string_literal: true

require 'minitest/autorun'
require 'minitest/assertions'
require_relative 'logic'

# class for test
class MyTest < Minitest::Test
  def test_first
    result = answer(1e-3)
    assert_in_delta(0.9696969696969695, result[0], 1e-3)
  end

  def test_second
    result = answer(1e-4)
    assert_in_delta(0.9900990099009898, result[0], 1e-4)
  end
end
```

## Результат выполнения и результаты проверки анализаторами rubocop и reek:

The screenshot shows a terminal window with the following content:

```
Terminal: Local x +
C:\Users\Daria\Desktop\lab6.2>ruby console.rb
eps = 1e-3: res = 0.9696969696969695
eps = 1e-4: res = 0.9900990099009898

C:\Users\Daria\Desktop\lab6.2>ruby test.rb
Run options: --seed 39711

# Running:

..

Finished in 0.006018s, 332.3529 runs/s, 332.3529 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
```

Below the terminal window, the results of static analysis are shown:

```
Terminal: Local x +
C:\Users\Daria\Desktop\lab6.2>rubocop
Inspecting 3 files
..C.

Offenses:

logic.rb:3:1: C: Metrics/MethodLength: Method has too many lines. [11/10]
def find_series ...
^^^^^^^^^^^^^^^^

3 files inspected, 1 offense detected

C:\Users\Daria\Desktop\lab6.2>reek
Inspecting 3 file(s):
.S.

logic.rb -- 5 warnings:
[3]:TooManyStatements: find_series has approx 9 statements [https://github.com/troessner/reek/blob/v6.0.2/docs/Too-Many-Statements.md]
[23]:UncommunicativeVariableName: answer has the variable name 's1' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.md]
[23]:UncommunicativeVariableName: answer has the variable name 's2' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.md]
[6, 12]:UncommunicativeVariableName: find_series has the variable name 'sum1' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.m
d]
[7, 10]:UncommunicativeVariableName: find_series has the variable name 'sum2' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.m
d]
5 total warnings
```

## Часть 3:

Составить метод `intg` вычисления определенного интеграла по формуле прямоугольников:

$$S = \frac{b-a}{n} \sum_{i=1}^n f(x_i), \text{ где } n - \text{ количество отрезков разбиения. В основной}$$

программе использовать метод `intg` для вычисления интегралов:  $\int_{0,1}^1 \frac{\sin x}{x} dx$

$$\text{и } \int_1^2 \frac{\operatorname{tg}(x+1)}{x+1} dx.$$

Реализовать вызов метода двумя способами: в виде передаваемого `lambda`-выражения и в виде блока.

### Код в `console.rb`:

```
# frozen_string_literal: true

require_relative 'logic'

puts 'Choose a function:'
puts '1) sin(x) / x'
puts '2) tg(x + 1) / (x + 1)'
code = gets.to_i

puts 'Input step'
steps = gets.to_i
func = ->(x) { Math.sin(x) / x }
case code
when 1
  puts intg(steps, 1, 0.1, func)
when 2
  puts intg(steps, 2, 1) { |x| Math.tan(x + 1) / (x + 1) }
else puts 'error'
end
```

### Код в `logic.rb`:

```
# frozen_string_literal: true

def intg(step, top, bottom, func = false)
  square = 0
  d_x = (top - bottom) / step.to_f
  (bottom..top).step(d_x) { |cur_pos| square += func ? func.call(cur_pos) :
yield(cur_pos) }
  square * d_x
end
```

### Код в `test.rb`:

```
# frozen_string_literal: true

require 'minitest/autorun'
require_relative 'logic'

# class with test
class TestMath < Minitest::Test
  DEL = 1e-2
  def test_first
    assert_in_delta(1.1201, intg(3, 1, 0.1) { |x| Math.sin(x) / x }, DEL)
  end

  def test_second
    assert_in_delta(-0.593827, intg(3, 2, 1) { |x| Math.tan(x + 1) / (x + 1) }, DEL)
  end
end
```

## Результат выполнения и результаты проверки анализаторами rubocop и reek:

```
Terminal: Local × +
C:\Users\Daria\Desktop\tasks\ЛР6\lab6.3>ruby console.rb
Choose a function:
1) sin(x) / x
2) tg(x + 1) / (x + 1)
1
Input step
3
1.1200985966447803

C:\Users\Daria\Desktop\tasks\ЛР6\lab6.3>ruby test.rb
Run options: --seed 20694

# Running:

..
Finished in 0.006386s, 313.1704 runs/s, 313.1704 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips

Terminal: Local × +
C:\Users\Daria\Desktop\tasks\ЛР6\lab6.3>reek
Inspecting 3 file(s):
.SS

test.rb -- 3 warnings:
[14, 14]:DuplicateMethodCall: TestMath#test_second calls 'x + 1' 2 times [https://github.com/troessner/reek/blob/v6.0.2/docs/Duplicate-Method-Call.md]
[10]:UncommunicativeVariableName: TestMath#test_first has the variable name 'x' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.md]
[14]:UncommunicativeVariableName: TestMath#test_second has the variable name 'x' [https://github.com/troessner/reek/blob/v6.0.2/docs/Uncommunicative-Variable-Name.md]
```

**Вывод:** в ходе данной лабораторной работы я изучила основные принципы программирования на языке Ruby.