



**«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**


ФАКУЛЬТЕТ _____ ИНФОРМАТИКИ И СИСТЕМ УПРАВЛЕНИЯ
КАФЕДРА _____ КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ 6)

О т ч е т
по лабораторной работе № 6

Дисциплина: Языки программирования

Название лабораторной работы: Консольное приложение Ruby

Студент гр. ИУ6-34



(Подпись, дата)

Габолаев Г.К.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Изория В.Ф.

(И.О. Фамилия)

Москва, 2017

Часть 1:

Решить задачу, организовав итерационный цикл с точностью $\xi = 10^{-4}, 10^{-5}$.
Вычислить сумму ряда:

$$S = 1 + \sum_{k=1}^{\infty} \frac{(-1)^k}{k!}, \text{ точное значение равно } \frac{1}{e}.$$

Коды программ:

```
> cat src.rb
1. def next_val(k)
2.   ((-1)**k) / (1..k).inject(:*).to_f
3. end
4.
5. def find_with_precision_no_enum(precision)
6.   raise(TypeError) if precision.is_a? String
7.   precision = precision.to_f
8.   now_val = 1.0
9.   k = 0
10.  now_val += next_val k += 1 while (now_val - 1 / Math::E).abs >
    precision
11.  now_val
12. end
13.
```

```
> cat user.rb
1. require './src'
2.
3. if $PROGRAM_NAME == __FILE__
4.   puts 'Введите точность 1e-4 или 1e-5'
5.   input = gets
6.   p 'Ввели', input
7.   number = input.chomp.to_f
8.   if [1e-4, 1e-5].include? number
9.     puts 'Убрали \r\n', number
10.    puts 'Результат операции: ',
        find_with_precision_no_enum(number)
11.  else
12.    puts 'Введите одно из предложенных выше чисел'
13.  end
14. end
15.
```

```

> cat src_test.rb
1. require 'minitest/autorun'
2. require './src'
3.
4. # fuck
5. class MyTest < MiniTest::Test
6.   def test_ok
7.     assert_equal 0.3678571428571429,
       find_with_precision_no_enum(10e-5)
8.   end
9.
10.  def test_fail
11.    assert_raises TypeError do
12.      find_with_precision_no_enum 'energo'
13.    end
14.  end
15.end

```

Результат выполнения:

```

Введите точность 1e-4 или 1e-5
1e-4
Ввели
"1e-4\n"
Убрали \r\n
0.0001
Результат операции:
0.3678571428571429

```

Результат тестирования:

```

> ruby src_test.rb
Run options: --seed 64278
# Running:
..
Finished in 0.001171s, 1707.9419 runs/s, 1707.9419 assertions/s.

2 runs, 2 assertions, 0 failures, 0 errors, 0 skips

```

Rubocop:

```

> rubocop *
Inspecting 3 files
...

3 files inspected, no offenses detected

```

Часть 2:

Решить предыдущее задание с помощью Enumerator.

Коды программ:

```
> cat src.rb
1. def next_val(k)
2.   ((-1)**k) / (1..k).inject(:*).to_f
3. end
4.
5. def find_with_precision_enum(precision)
6.   precision.is_a?(String) ? raise(TypeError) : (precision =
   precision.to_f)
7.   global = 0
8.   list = Enumerator.new do |yield_er|
9.     sum = 1.0
10.    counter = 0.0
11.    loop do
12.      yield_er.yield sum
13.      sum += next_val counter += 1
14.      global = sum
15.    end
16.  end
17.  list.take_while { |sum| (1 / Math::E - sum).abs > precision }
18.  global
19.end
20.

> cat user.rb
1. require './src'
2.
3. if $PROGRAM_NAME == __FILE__
4.   puts 'Введите точность 1e-4 или 1e-5'
5.   input = gets
6.   p 'Ввели', input
7.   number = input.chomp.to_f
8.   if [1e-4, 1e-5].include? number
9.     puts 'Убрали \r\n', number
10.    puts 'Результат операции: ',
        find_with_precision_enum(number)
11.  else
12.    puts 'Введите одно из предложенных выше чисел'
13.  end
14.end
15.
```

```

> cat src_test.rb
1. require 'minitest/autorun'
2. require './src'
3.
4. # i hate you rubocop
5. class MyTest < MiniTest::Test
6.   def test_ok
7.     assert_equal 0.3678571428571429,
       find_with_precision_enum(10e-5)
8.   end
9.
10.  def test_fail
11.    assert_raises TypeError do
12.      find_with_precision_enum 'energo'
13.    end
14.  end
15.end

```

Результат выполнения:

```

Введите точность 1e-4 или 1e-5
1e-5
Ввели
"1e-5\n"
Убрали \r\n
1.0e-05
Результат операции:
0.3678819444444445

```

Результат тестирования:

```

> ruby src_test.rb
Run options: --seed 26890
# Running:
..
Finished in 0.001300s, 1538.4615 runs/s, 1538.4615 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips

```

Rubocop:

```

> rubocop *
Inspecting 3 files
C..
Offenses:
src.rb:5:1: C: Assignment Branch Condition size for
find_with_precision_enum is too high. [15.65/15]
def find_with_precision_enum(precision)
^^^

src.rb:5:1: C: Method has too many lines. [13/10]
def find_with_precision_enum(precision) ...
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

3 files inspected, 2 offenses detected

```

Часть 3:

Составить метод `intprg` вычисления определенного интеграла по формуле прямоугольников: $S = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$, где n – количество отрезков разбиения. В основной программе использовать метод `intprg` для вычисления интегралов: $\int_0^1 \frac{e^x}{x+1} dx$ и $\int_0^2 x(x-1) dx$.

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

Коды программ:

```
> cat src.rb
1. def calc_integral(a, b, procedure = nil, n = 10_000)
2.   a, b, n = [a, b, n].map!(&:to_f)
3.   result = 0
4.   h = (b - a) / n
5.   (1..n).each do |index|
6.     result += if block_given?
7.                 yield(a + h * (index + 0.5))
8.             else
9.                 procedure.call(a + h * (index + 0.5))
10.            end
11.   end; result * h
12.end

> cat user.rb
1. require '../part3/src'
2.
3. if $PROGRAM_NAME == __FILE__
4.   puts 'Press ENTER for magic'
5.   gets
6.   puts "\nLambda: "
7.   p calc_integral(0, 1, ->(x) { Math.exp(x) / (x + 1) })
8.   p calc_integral(0, 2, ->(x) { x * (x - 1) })
9.   puts "\nBlock: "
10.  p calc_integral(0, 1) { |x| Math.exp(x) / (x + 1) }
11.  p calc_integral(0, 2) { |x| x * (x - 1) }
12.end
```

```

> cat src_test.rb
1. require 'minitest/autorun'
2. require_relative '../part3/src'
3. # 1488
4. class Test < MiniTest::Test
5.   def test_ok
6.     assert_equal 1.1254220002893505, calc_integral(0, 1) { |x|
7.       Math.exp(x) / (x + 1)
8.     }
9.     assert_equal 0.6670667399999997, calc_integral(0, 2, ->(x) {
    x * (x - 1) })
10.  end
11.
12.  def test_fail
13.    assert_raises NoMethodError do
14.      calc_integral('a', 'b')
15.    end
16.  end
17.end

```

Результат выполнения:

```

1. Press ENTER for magic
2.
3.
4. Lambda:
5. 1.1257455354272916
6. 0.6706740000000002
7.
8. Block:
9. 1.1257455354272916
10.0.6706740000000002

```

Результат тестирования:

```

> ruby src_test.rb
Run options: --seed 64327
# Running:
..
Finished in 0.007836s, 255.2323 runs/s, 382.8484 assertions/s.
2 runs, 3 assertions, 0 failures, 0 errors, 0 skips

```

Rubocop:

```

Inspecting 3 files
C..
Offenses:
src.rb:1:1: C: Assignment Branch Condition size for calc_integral
is too high. [15.3/15]
def calc_integral(a, b, procedure = nil, n = 10_000)
^^^
3 files inspected, 1 offense detected

```