



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: Вычисления с заданной точностью в Ruby

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-33Б

(Группа)

(Подпись, дата)

И.А. Нуруллаев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

Часть 1

Решить задачу с точностью $\xi = 10^{-3}, 10^{-4}$, организовав итерационный цикл. Вычислить значение определенного интеграла методом прямоугольников: $\int_1^2 \ln x \, dx$. Считать точным значением: 0,3862943611199. Определить, как изменяется число итераций при изменении точности.

Решение:

```
WPL_bmstu - main.rb

1 def calc_int(accuracy)
2   sum = 0.0
3   left = 1.0
4   right = 2.0
5   k = 0
6
7   loop do
8     sum += Math.log(left)
9     left += accuracy
10    k += 1
11    break if left >= right
12  end
13
14  [sum / k, k]
15 end
16
```

```
WPL_bmstu - user.rb

1 require_relative 'main'
2
3 result1 = calc_int 0.001
4 result2 = calc_int 0.0001
5
6 puts("Вычисление с точностью 0.001 дало результат #{result1[0]}
7   , потребовалось итераций - #{result1[1]}")
8 puts("Вычисление с точностью 0.0001 дало результат #{result2[0]}
9   , потребовалось итераций - #{result2[1]}")
10
```

```
WPL_bmstu - test.rb

1 require 'minitest/autorun'
2 require_relative 'main'
3
4 # Integral tester
5 class Test < Minitest::Test
6   def test_first
7     result = calc_int 0.001
8     assert_in_delta 0.3862943611199, result[0], 0.001, 'Неправильный результат'
9   end
10
11   def test_second
12     result = calc_int 0.0001
13     assert_in_delta 0.3862943611199, result[0], 0.0001, 'Неправильный результат'
14   end
15 end
16
```

Результат выполнения программ:

```
PS C:\Users\might\Desktop\WPL_bmstu\Lab6\Part 1> ruby .\user.rb
Вычисление с точностью 0.001 дало результат 0.38625463840506785, потребовалось итераций - 1001
Вычисление с точностью 0.0001 дало результат 0.38629038902331586, потребовалось итераций - 10001
PS C:\Users\might\Desktop\WPL_bmstu\Lab6\Part 1> ruby .\test.rb
Run options: --seed 44033

# Running:

..

Finished in 0.014497s, 137.9558 runs/s, 137.9558 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
```

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator

Решение:

```
WPL_bmstu - main.rb

1 def calc_int(accuracy)
2   series(accuracy).take_while { |left, right| left < right }.
  last[2]
3 end
4
5 def series(accuracy)
6   Enumerator.new do |yielder|
7     sum = 0.0
8     left = 1.0
9     right = 2.0
10    k = 0
11
12    loop do
13      k += 1
14      sum += Math.log(left)
15      left += accuracy
16      yielder.yield left, right, sum / k
17    end
18  end
19 end
20
```

```
WPL_bmstu - user.rb

1 require_relative 'main'
2
3 puts("Вычисление с точностью 0.001 дало результат #{calc_int 0.001}")
4 puts("Вычисление с точностью 0.0001 дало результат #{calc_int 0.0001}")
5
```

```
WPL_bmstu - test.rb

1 require 'minitest/autorun'
2 require_relative 'main'
3
4 # Sum tester
5 class Test < Minitest::Test
6   def test_first
7     result = calc_int 0.001
8     assert_in_delta 0.3862943611199, result, 0.001, 'Неправильный результат'
9   end
10
11   def test_second
12     result = calc_int 0.0001
13     assert_in_delta 0.3862943611199, result, 0.0001, 'Неправильный результат'
14   end
15 end
16
```

Результат выполнения программ:

```
PS C:\Users\might\Desktop\WPL_bmstu\Lab6\Part 2> ruby .\user.rb
Вычисление с точностью 0.001 дало результат 0.385947745862913
Вычисление с точностью 0.0001 дало результат 0.38625970334416226
PS C:\Users\might\Desktop\WPL_bmstu\Lab6\Part 2> ruby .\test.rb
Run options: --seed 14485

# Running:

. .

Finished in 0.017405s, 114.9075 runs/s, 114.9075 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
```

Часть 3

Составить метод `intg` вычисления определенного интеграла по формуле прямоугольников: $S = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$, где n – количество отрезков разбиения, $[a, b]$ – отрезок интегрирования. В основной программе использовать метод `intg` для вычисления интегралов: $\int_{0,1}^1 \frac{\ln(x)}{x} dx$ и $\int_0^2 \sin x \cos x dx$

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

Решение:

```
WPL_bmstu - main.rb

1 def intg(x_a, x_b, n_divisions, f_lambda = nil, &f_block)
2   func = f_lambda || f_block
3   range = (x_b - x_a).to_f
4   dx = range / n_divisions
5   sum = (x_a..x_b).step(dx).reduce do |acc, x|
6     acc + func.call(x)
7   end
8   dx * (((func.call(x_a) + func.call(x_b)) / 2) + sum)
9 end
10
```

```
WPL_bmstu - user.rb

1 require_relative 'main'
2
3 f1 = ->(x) { Math.log(x) / x }
4 puts("Результат вычисления интеграла первой функции - #{intg(0.1, 1, 10_000, f1)}")
5 puts("Результат вычисления интеграла второй функции - #{intg(0, 2, 10_000) { |x| Math.sin(x) * Math.cos(x) }}")
6
```

```
WPL_bmstu - test.rb

1 require 'minitest/autorun'
2 require_relative 'main'
3
4 # Integral testing
5 class Test < Minitest::Test
6   def test_f1_lambda
7     f1 = ->(x) { Math.log(x) / x }
8     assert_in_epsilon(-2.65095, intg(0.1, 1, 10_000, f1), 1e-3)
9   end
10
11   def test_f1_block
12     assert_in_epsilon(-2.65095, intg(0.1, 1, 10_000) { |x| Math.log(x) / x }, 1e-3)
13   end
14
15   def test_f2_lambda
16     f2 = ->(x) { Math.sin(x) * Math.cos(x) }
17     assert_in_epsilon 0.413411, intg(0, 2, 10_000, f2), 1e-3
18   end
19
20   def test_f2_block
21     assert_in_epsilon 0.413411, intg(0, 2, 10_000) { |x| Math.sin(x) * Math.cos(x) }, 1e-3
22   end
23 end
24
```

Результат выполнения программ:

```
PS C:\Users\might\Desktop\WPL_bmstu\Lab6\Part 3> ruby .\user.rb
Результат вычисления интеграла первой функции - -2.6509402774886635
Результат вычисления интеграла второй функции - 0.4134108997037556
PS C:\Users\might\Desktop\WPL_bmstu\Lab6\Part 3> ruby .\test.rb
Run options: --seed 20280

# Running:

.....

Finished in 0.021376s, 187.1240 runs/s, 187.1240 assertions/s.
4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
```

Проверка кода при помощи *rubocop*

```
PS C:\Users\might\Desktop\WPL_bmstu\Lab6> rubocop
Inspecting 9 files
.....

9 files inspected, no offenses detected
```

Итоговый код данной лабораторной работы доступен по ссылке:

https://github.com/tenessinum/WPL_bmstu/tree/main/Lab6