



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: Ruby

Дисциплина: Языки интернет программирования

Студент

ИУ6-53Б
(Группа)

(Подпись, дата)

Т.Р. Сапарбаев
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Р.С. Самарев
(И.О. Фамилия)

Москва, 2020

Часть 1

Решить задачу, организовав итерационный цикл с точностью $\xi = 10^{-3}, 10^{-4}$.

Вычислить определенный интеграл методом трапеций: $\int_0^1 x\sqrt{1-x^2} dx$.

Точное значение интеграла равно $\frac{1}{3}$. Определить, как изменяется число отрезков деления при изменении точности.

1_1.rb:

```
# frozen_string_literal: true

ANS = 1.0 / 3
# Integral Class
class Integral
  attr_reader :eps, :a_limit, :b_limit

  def initialize(eps)
    @a_limit = 0
    @b_limit = 1
    @eps = eps
  end

  def self.func(argument)
    argument * Math.sqrt(1 - argument * argument)
  end

  def sum_result(iter)
    result = 0
    (@a_limit + iter).step(@b_limit -
iter, iter) { |x_val| result += Integral.func(x_val) }
    result += (Integral.func(@a_limit) + Integral.func(@b_limit)) / 2
    result * iter
  end

  def eval
    iter = 10.0
    result = 0
    while (result - ANS).abs > @eps
      iter /= 10
      result = sum_result(iter)
    end
    result
  end
end
```

1_2.rb:

```
# frozen_string_literal: true
```

```
require_relative './1_1'
puts 'Enter the accuracy:'
eps = gets.to_f
puts Integral.new(eps).eval
```

1_3.rb:

```
# frozen_string_literal: true

require 'test/unit'
require_relative './1_1'

# main test
class MyTest < Test::Unit::TestCase
  def test1_delta
    [0.001, 0.0001].each { |eps| assert_in_delta(ANS, Integral.new(eps).eval, eps) }
  end
end
```

Результат выполнения:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ ruby 1_2.rb
Enter the accuracy:
0.0001
0.33332395447013247
```

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ ruby 1_3.rb
Loaded suite 1_3
Started
.
Finished in 0.000934966 seconds.

-----
1 tests, 2 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-----
1069.56 tests/s, 2139.12 assertions/s
```

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

2_1.rb:

```
# frozen_string_literal: true

ANS = 1.0 / 3
# Integral Class
class Integral
  include Enumerable

  attr_reader :a_limit, :b_limit

  def initialize(a_limit = 0, b_limit = 1)
    @a_limit = a_limit
    @b_limit = b_limit
  end

  def self.func(argument)
    argument * Math.sqrt(1 - argument * argument)
  end

  def sum_result(iter)
    result = (a_limit + iter).step(b_limit -
iter, iter).map { |x_val| Integral.func(x_val) }.sum
    result += (Integral.func(@a_limit) + Integral.func(@b_limit)) / 2
    result * iter
  end

  def each
    iter = 10.0
    loop do
      iter /= 10
      result = sum_result(iter)
      yield result
    end
  end

  def eval(eps)
    each_with_index.find { |result, _| (result - ANS).abs < eps }
  end

  class << self
    def eval(eps)
      new.eval(eps)
    end
  end
end
```

2_2.rb:

```
# frozen_string_literal: true

require_relative './2_1'
puts 'Enter the accuracy:'
p Integral.eval(gets.to_f)
```

2_3.rb:

```
# frozen_string_literal: true

require 'test/unit'
require_relative './2_1'

# main test
class MyTest < Test::Unit::TestCase
  def test1_delta
    [0.01, 0.001].each do |eps|
      assert_in_delta(ANS, Integral.eval(eps).first, eps)
    end
  end
end
```

Результат выполнения:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ ruby 2_2.rb
Enter the accuracy:
0.000000001
[0.33333333303925455, 6]
```

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ ruby 2_3.rb
Loaded suite 2_3
Started
.
Finished in 0.000743887 seconds.
-----
1 tests, 2 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-----
1344.29 tests/s, 2688.58 assertions/s
```

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.

3_1.rb:

```
# frozen_string_literal: true

# Class Eval
class EvalClass
  def self.eval(a_limit, b_limit, dif, &block)
    iter = (b_limit - a_limit).to_f / dif
    a_limit.step(b_limit, iter).map(&block).sum * iter
  end
end
```

3_2.rb:

```
# frozen_string_literal: true

require_relative './3_1'

puts 'Enter the difference:'
dif = gets.to_f
p EvalClass.eval(0, 1, dif, &->(x) { Math.exp(x) / (x + 1) })
p EvalClass.eval(0, 2, dif) { |x| x * (x - 1) }
```

3_3.rb:

```
# frozen_string_literal: true

require 'test/unit'
require_relative './3_1'

ANS1 = 1.12538608308327
ANS2 = 0.6666666666666667

# main test
class MyTest < Test::Unit::TestCase
  # :reek:FeatureEnvy
```

```

def test1_lambda
  assert_in_delta(ANS1, EvalClass.eval(0, 1, 1000, &-
>(x_val) { Math.exp(x_val) / (x_val + 1) })), 0.1)
  assert_in_delta(ANS2, EvalClass.eval(0, 2, 1000, &->(x_val) { x_val * (x_val -
1) })), 0.1)
end

# :reek:FeatureEnvy
def test2_block
  assert_in_delta(ANS1, EvalClass.eval(0, 1, 1000) { |x_val| Math.exp(x_val) / (x_val
+ 1) }, 0.1)
  assert_in_delta(ANS2, EvalClass.eval(0, 2, 1000) { |x_val| x_val * (x_val -
1) }, 0.1)
end
end

```

Результат выполнения:

```

saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ ruby 3_2.rb
Enter the difference:
10000
1.12550404069529
0.6668666800000002

```

```

saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ ruby 3_3.rb
Loaded suite 3_3
Started
..
Finished in 0.001919849 seconds.

-----
2 tests, 4 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-----
1041.75 tests/s, 2083.50 assertions/s

```

Rubocop:

```

saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.1.
warning: please see https://github.com/whitequark/parser#compatibility-with-ruby-mri.
Inspecting 9 files
.....

9 files inspected, no offenses detected

```

Reek:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR6$ reek
Inspecting 9 file(s):
.....

0 total warnings
```

Вывод:

В ходе работы были разработаны программы, соответствующие условиям задачи. На приведенных данных программы работают корректно. Необходимые тесты пройдены.