*Федеральное государственное бюджетное образовательное учреждение* *высшего профессионального образования*

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ *Информатика и системы управления*

КАФЕДРА *Компьютерные системы и сети*

**Отчет**

**по лабораторной работе № 6**

**Дисциплина:** Языки интернет-программирования

**Название лабораторной работы:** Консольное приложение ruby

Студент гр. ИУ6-33Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Захаров.К.А.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Самарев Р.С.**

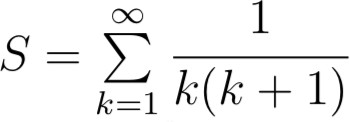
(Подпись, дата) (И.О. Фамилия)

Москва, 2019

# Часть 1

**Задание**

Решить задачу, организовав итерационный цикл с точностью ξ = 10−4, 10−5. Точное значение равно 1. Определить, как изменяется число итераций при изменении точности.



# Код

# main1.rb

def count(prec)

sum = 0.0

n = 1.0

prev = 1.0

while (prev - sum).abs >= 10\*\*prec

prev = sum

sum += 1 / (n \* (n + 1))

n += 1

end

sum

end

**input1.rb**

require './main1.rb'

puts count(-4)

puts count(-5)

# test1.rb

# require './main1.rb'

# require 'minitest/autorun'

# # Test class

class Test < MiniTest::Unit::TestCase

# def test\_1

# assert\_in\_delta(0.9900, count(-4), 1e-4)

# end

# def test\_2

# assert\_in\_delta(0.99684, count(-5), 1e-5)

# end

# end

# Результат

C:\Users\Keller\Desktop\Печать\6\1>ruby input1.rb

0.9900990099009898

0.9968454258675077

**Проверка rubocop**

C:\Users\Keller\Desktop\Печать\6\3>rubocop \*.rb

Inspecting 3 files0 files inspected, no offenses detected

# Часть 2

**Задание**

Решить предыдущее задание с помощью Enumerator.

**Код**

# main2.rb

def enumerator

Enumerator.new do |yielder|

sum = 0.0

prev = 0.5

(1..Float::INFINITY).lazy.each do |counter|

yielder.yield sum, prev

prev = sum

sum += (1 / (counter.to\_f \* (counter + 1)))

end

end end

def count(prec)

arr = enumerator.take\_while { |sum, prev| (prev - sum).abs >= 10\*\*prec }

arr.last[0]

end

# input2.rb

require './main2.rb'

puts count(-4)

puts count(-5)

# test2.rb

require './main2.rb' require 'minitest/autorun'

# Test class

class Test < MiniTest::Unit::TestCase

def test\_1

assert\_in\_delta(0.9900, count(-4), 1e-4)

end

def test\_2

assert\_in\_delta(0.99684, count(-5), 1e-5)

end

end

**Результат**

C:\Users\Keller\Desktop\Печать\6\2>ruby input2.rb

0.9900990099009898

0.9968454258675077

**Проверка rubocop**

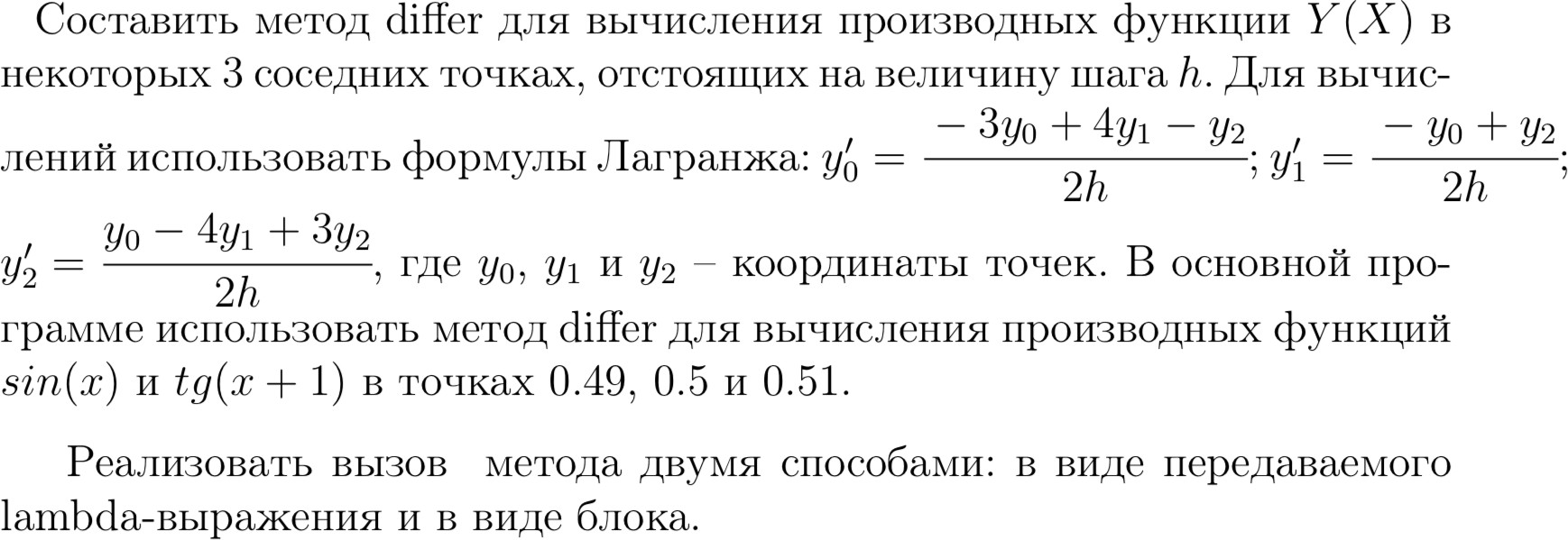
C:\Users\Keller\Desktop\Печать\6\2>rubocop \*.rb

Inspecting 3 files

3 files inspected, no offenses detected

# Часть 3

**Задание**



**Код**

# main3.rb

# frozen\_string\_literal: true

def differ(xi0, xi1, xi2, lam = nil)

y = if block\_given?

call\_proc(xi0, xi1, xi2, &Proc.new)

else

call\_lam(xi0, xi1, xi2, lam)

end

y\_res = result(y)

y\_res

end

def call\_lam(xi0, xi1, xi2, lam)

y = []

y[0] = lam.call(xi0)

y[1] = lam.call(xi1)

y[2] = lam.call(xi2)

y

end

def call\_proc(xi0, xi1, xi2)

y = []

y[0] = yield xi0

y[1] = yield xi1

y[2] = yield xi2

y

end

def result(yi)

res = [(-3 \* yi[0] + 4 \* yi[1] - yi[2]) / 0.02,

(-yi[0] + yi[2]) / 0.02,

(yi[0] - 4 \* yi[1] + 3 \* yi[2]) / 0.02]

res

end

# input3.rb

# frozen\_string\_literal: true

require './main3.rb'

puts ' ' \* 27 + 'sin(x)'

lambda\_sin = ->(x) { Math.sin(x) }

p differ(0.49, 0.50, 0.51) { |x| Math.sin(x) }

p differ(0.49, 0.50, 0.51, lambda\_sin)

puts ' ' \* 27 + 'tag(x)'

lambda\_tan = ->(x) { Math.tan(x + 1) }

p differ(0.49, 0.50, 0.51) { |x| Math.tan(x + 1) }

p differ(0.49, 0.50, 0.51, lambda\_tan)

# test3.rb

# frozen\_string\_literal: true

require './main3.rb'

require 'minitest/autorun'

# Test Class

class Test < MiniTest::Unit::TestCase

def test\_1

expected = [0.8823621510215207, 0.8775679355874755, 0.8727737201534302]

output = differ(0.49, 0.50, 0.51) { |x| Math.sin(x) }

end

def test\_2

expected = [146.40093799620715, 203.9117631129767, 261.42258822974645]

output = differ(0.49, 0.50, 0.51) { |x| Math.tan(x + 1) }

end end

# Результат

# C:\Users\Keller\Desktop\Печать\6\3>ruby input3.rb

# sin(x)

# [0.8823621510215235, 0.8775679355874727, 0.8727737201534191]

# [0.8823621510215235, 0.8775679355874727, 0.8727737201534191]

# tag(x)

# [146.40093799620715, 203.91176311297681, 261.42258822974645]

# [146.40093799620715, 203.91176311297681, 261.42258822974645]

# Проверка rubocop

# C:\Users\Keller\Desktop\Печать\6\3>rubocop \*.rb

# Inspecting 3 files

# .3 files inspected, no offenses detected

# Вывод

Были изучены методы работы с классом Enumerator и модулем Enumerable. Все программы проходят анализатор Rubocop, запускаются и работают корректно