



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети(ИУ6)

О т ч е т

по лабораторной работе № 6

Дисциплина: Языки интернет-программирования

Название лабораторной работы: _____

Студент гр. ИУ6-32Б _____ Д.В.Лабзунова
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____
(Подпись, дата) (И.О. Фамилия)

Задание

Часть 1

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.

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод minmax, отыскивающую $x \in [a, b]$, для которого функция $y = f(x)$ принимает максимальное и минимальное значение с точностью 0,01. В основной программе использовать этот метод для математических функций $y = \frac{x-1}{x+2}$; $x \in [0, 2]$ и $y = \sin(\frac{x}{2} - 1)$, $x \in [-1, 1]$.

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

Код программы

1 часть:

main.rb:

```
# frozen_string_literal: true
```

```
require_relative 'integral'
```

```
# client class
```

```
class Main
```

```
  integral = Integral.new
```

```
  puts 'integral with 10^-3 precision', integral.count(0.001)
```

```
  puts 'integral with 10^-4 precision', integral.count(0.0001)
```

```
end
```

integral.rb:

```
# frozen_string_literal: true
```

```
# :reek:UtilityFunction
```

```
# integral counting
```

```
class Integral
```

```
  def count(precision)
```

```
    arg = 1
```

```

    result = 0
    while arg <= 2
      result += precision * Math.log(arg)
      arg += precision
    end
    result
  end
end
end

```

test.rb:

```
# frozen_string_literal: true
```

```
require 'minitest/autorun'
require_relative 'integral'
```

```
# :reek:TooManyStatements
# testing math logic
```

```
class TestMath < Minitest::Test
  def setup; end
```

```

  def test_math
    truth = 0.3862943611199
    integral = Integral.new
    min_precision = integral.count(0.001)
    max_precision = integral.count(0.0001)
    assert(min_precision - 0.001 < truth) && (min_precision + 0.001 > truth)
    assert(max_precision - 0.0001 < truth) && (max_precision + 0.0001 > truth)
  end
end

```

2 часть:

main.rb:

```
# frozen_string_literal: true
```

```
require_relative 'integral_enum'
```

```
# client class
```

```

class Main
  puts 'integral with 10^-3 precision', count(0.001.to_f)
  puts 'integral with 10^-4 precision', count(0.0001.to_f)
end

```

integral_enum.rb:

```
# frozen_string_literal: true
```

```
# :reek:TooManyStatements
```

```
# :reek:UncommunicativeVariableName
```

```
# :reek:FeatureEnvy
```

```
# class integral
```

```
class Integral
```

```
  include Enumerable
```

```
  def each
```

```
    sum = 0
```

```
    step = 0.01
```

```
    loop do
```

```
      yield sum
```

```
      sum = 0
```

```
      (1.0..2.0).step(step) { |x| sum += step * Math.log(x) }
```

```
      step /= 10
```

```

    end
  end
end

# :reek:UtilityFunction
def count(precision)
  real_value = 0.3862943611199
  @minmax.new.find { |sum| (sum - real_value).abs < precision }
end

```

test.rb:

```

# frozen_string_literal: true

require 'minitest/autorun'
require_relative 'integral_enum'

# testing math logic
class TestMath < Minitest::Test
  def setup; end

  def test_math
    truth = 0.3862943611199
    min_precision = count(0.001.to_f)
    max_precision = count(0.0001.to_f)
    assert(min_precision - 0.001 < truth) && (min_precision + 0.001 > truth)
    assert(max_precision - 0.0001 < truth) && (max_precision + 0.0001 > truth)
  end
end

```

3 часть:

main.rb:

```

# frozen_string_literal: true

require_relative 'extrems'
# client class
class Main
  res = []
  puts 'hyperbola or sin'
  case gets.chomp
  when 'sin'
    left = -1
    right = 1
    func = ->(arg) { @minmax.new.sin(arg / 2 - 1) }
  when 'hyperbola'
    left = 0
    right = 2
    func = ->(arg) { (arg - 1) / (arg + 2) }
  else puts 'invalid'
  end

  puts 'block or lambda'
  case gets.chomp
  when 'block'
    res = @minmax.new.minmax_block(left, right) do |arg|
      (arg - 1) / (arg + 2)
    end
  when 'lambda'
    res = @minmax.new.minmax_func(left, right, func)
  end
end

```

```

else puts 'invalid'
end
puts 'min', res[0]
puts 'max', res[1]
end

```

extrems.rb:

```
# frozen_string_literal: true
```

```
# minmax class
```

```
class Extrems
```

```
  # :reek:TooManyStatements
```

```
  # :reek:UtilityFunction
```

```
  def minmax_func(left, right, func)
```

```
    min = func.call(0)
```

```
    max = min
```

```
    left.step(right, 0.01) do |arg|
```

```
      value = func.call(arg)
```

```
      min = value if value < min
```

```
      max = value if value > max
```

```
    end
```

```
    [min, max]
```

```
end
```

```
  # :reek:TooManyStatements
```

```
  # :reek:UtilityFunction
```

```
  def minmax_block(left, right)
```

```
    min = yield(0)
```

```
    max = min
```

```
    left.step(right, 0.01) do |arg|
```

```
      min = yield(arg) if yield(arg) < min
```

```
      max = yield(arg) if yield(arg) > max
```

```
    end
```

```
    [min, max]
```

```
end
```

```
end
```

test.rb:

```
# frozen_string_literal: true
```

```
require 'minitest/autorun'
```

```
require_relative 'extrems'
```

```
# Extrems logic testing
```

```
class Test < Minitest::Test
```

```
  def setup; end
```

```
  def sin_check_lambda
```

```
    res = minmax_func(0, 2, ->(arg) { Math.sin(arg / 2 - 1) })
```

```
    assert_equal(res[0], -0.9974949866040544)
```

```
    assert_equal(res[1], -0.479425538604203)
```

```
  end
```

```
  def sin_check_block
```

```
    res = Extrems.new.minmax_block(left, right) do |arg|
```

```
      Math.sin(arg / 2 - 1)
```

```
    end
```

```
    assert_equal(res[0], -0.9974949866040544)
```

```
    assert_equal(res[1], -0.479425538604203)
```

```
  end
```

```
def hyperbola_check_lambda
  res = minmax_func(0, 2, ->(arg) { (arg - 1) / (arg + 2) })
  assert_equal(res[0], -1)
  assert_equal(res[1], 0.25)
end
```

```
def hyperbola_check_block
  res = .new.minmax_block(left, right) do |arg|
    (arg - 1) / (arg + 2)
  end
  assert_equal(res[0], -1)
  assert_equal(res[1], 0.25)
end
```

Результаты выполнения:

1 часть:

Основная программа:

```
main x
/home/labzunova/.rvm/rubies/ruby-2.3.5/bin/ruby "/home/labzunova/Рабочий стол/labs BMSTU/LR6/6.1/main.rb"
integral with 10^-3 precision
0.3866408930434727
integral with 10^-4 precision
0.3863290180622198
Process finished with exit code 0
```

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR6/6.1$ ruby test.rb
Run options: --seed 24382

# Running:

.

Finished in 0.001720s, 581.2622 runs/s, 1162.5244 assertions/s.

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

Тесты:

2 часть:

Основная программа:

```
/home/labzunova/.rvm/rubies/ruby-2.3.5/bin/ruby "/home/labzunova/Рабочий стол/labs BMSTU/LR6/6.2/main.rb"
integral with 10^-3 precision
0.3866408930435061
integral with 10^-4 precision
0.3863290180622514
|
Process finished with exit code 0
```

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR6/6.2$ ruby test.rb
Run options: --seed 59353

# Running:

.

Finished in 0.001765s, 566.6987 runs/s, 1133.3975 assertions/s.

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

Тесты:

3 часть:

Основная программа:

```
/home/labzunova/.rvm/rubies/ruby-2.3.5/bin/ruby "/home/labzunova/Рабочий стол/labs BMSTU/LR6/6.3/main.rb"
hyperbola or sin
hyperbola
block or lambda
block
min
-1
max
0.25

Process finished with exit code 0
```

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR6/6.3$ ruby test.rb
Run options: --seed 18070

# Running:

Finished in 0.000489s, 0.0000 runs/s, 0.0000 assertions/s.

0 runs, 0 assertions, 0 failures, 0 errors, 0 skips
```

Тесты:

Проверка rubocop и reek:

Rubocop:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR6$ rubocop
NOTE: Gem.gunzip is deprecated; use Gem::Util.gunzip instead. It will be removed on or after 2018-12-01.
Gem.gunzip called from /usr/lib/ruby/vendor_ruby/unicode/display_width/index.rb:5.
Inspecting 9 files
*****
9 files inspected, no offenses detected
```

:

Reek:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR6$ reek
Inspecting 9 file(s):
.....

0 total warnings
```