*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 № 7**

**Дисциплина:** Языки интернет-программирования

**Название лабораторной работы:** Консольное приложение ruby

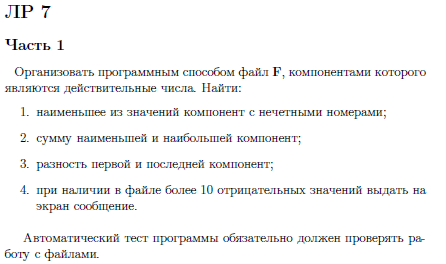
Студент гр. ИУ6-33Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Захаров.К.А.**

(Подпись, дата) (И.О. Фамилия)

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

(Подпись, дата) (И.О. Фамилия)

Москва, 2019



**main.rb**

# frozen\_string\_literal: true

File.open('F.dot', 'w') do |f|

rand(10..20).times do

f.puts rand(-100.0..100.00)

end

end

def get\_results(filename)

nums = []

File.open(filename, 'r') do |f|

while (line = f.gets)

nums << line.to\_f

end

end

mn = nums.each.with\_index.select { |\_v, i| i.even? }.min

minmax\_diff = nums.min + nums.max

last\_first = nums.last - nums.first

min\_count = nums.select { |n| n < 0 }.count

{

'min\_odd' => mn[0],

'minmax\_diff' => minmax\_diff,

'last\_first' => last\_first,

'min\_count' => min\_count

}

end

res = get\_results('F.dot')

p res

puts "Min element with odd index: #{res['min\_odd']}"

puts "Min + max: #{res['minmax\_diff']}"

puts "first - last: #{res['last\_first']}"

puts 'More than 10 negative' if

res['min\_count'] > 10

**test.rb**

#!/usr/bin/env ruby

# frozen\_string\_literal: true

require 'test/unit'

# require 'minitest/autorun'

require '/Users/tenzy/Desktop/lab7/main1'

# Test for part1

class Test1 < Test::Unit::TestCase

def test\_1

low = rand(-99..0)

high = rand(100)

file = File.open('F1.dot', 'w') do |f|

f.puts(low)

rand(10..100).times do

f.puts rand(low..high)

end

f.puts(high)

end

nums = []

File.open('F1.dot', 'r') do |f|

while (line = f.gets)

nums << line.to\_f

end

min\_c = nums.select { |n| n < 0 }.count

exp = {

'min\_odd' => low,

'minmax\_diff' => high + low,

'last\_first' => high - low,

'min\_count' => min\_c.to\_f

}

if min\_c > 10

puts 'There are more than 10 negative numbers in array!

= ' + min\_c.to\_s

else puts 'Less than 10 negative numbers= ' + min\_c.to\_s

end

assert\_equal(exp, get\_results('F1.dot'))

end

end

end  
  
**Результат:**C:\Users\Keller\Desktop\Печать\7>ruby main.rb

{"min\_odd"=>-82.21355120418622, "minmax\_diff"=>14.05834875261121, "last\_first"=>2.8819341493525314, "min\_count"=>1}

Min element with odd index: -82.21355120418622

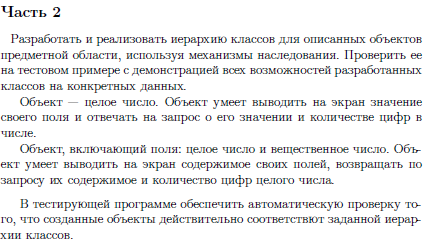
Min + max: 14.05834875261121

first - last: 2.8819341493525314  
**Проверка rubocop**

C:\Users\Keller\Desktop\Печать\7>rubocop \*.rb

Inspecting 3 files

.

3 files inspected, no offenses detected   
  
  
  


**main.rb**

class Integers

attr :ch

def initialize(ch)

@ch = ch

end

def put

puts "Вывод значения числа в классе 1: #{@ch}"

end

def len

i = 0

ch1=@ch

while ch1%10!=0

i+=1

ch1/=10

end

print "Вывод количества цифр целового числа: #{i}"

i

end

end

class Numbers < Integers

attr :real

def initialize(ch,real)

super(ch)

@real = real

end

def put

puts "Вывод содержимого класса 2: #{@real} #{@ch}"

end

end

obj2 = Numbers.new(12,15.0)

obj2.put

obj2.len

**test.rb**

#!/usr/bin/env ruby

# frozen\_string\_literal: true

require 'test/unit'

require './main.rb'

# Test for part2

class Test2 < Test::Unit::TestCase

def setup

@numbers = Numbers.new(12, 10.0)

end

def test\_1

assert @numbers.is\_a? Integers

end

def test\_2

assert\_equal(2, @numbers.len)

end

end

**Результат:**C:\Users\Keller\Desktop\Печать\7\2>ruby main.rb

Вывод содержимого класса 2: 15.0 12

Вывод количества цифр целового числа: 2  
**Проверка rubocop**C:\Users\Keller\Desktop\Печать\7\2>rubocop \*.rb

Inspecting 2 files

.

2 files inspected, no offenses detected