



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 09.03.03 Прикладная информатика

О Т Ч Е Т

по лабораторной работе № 5

Название: Ruby

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-35Б

(Группа)

18.10.2022

(Подпись, дата)

В.К.Полубояров

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

Цель работы

Изучить основной синтаксис языка Ruby.

Часть 1

Вычислить:
$$a = 1 + |y - x| + \frac{(y - x)^2}{2} + \frac{|y - x|^3}{3}.$$

Код программы

console_app.rb

```
# frozen string_literal: true

include Equation
require_relative 'calc_a'

puts 'Input x'
x = gets.chomp.to_f
puts 'Input y'
y = gets.chomp.to_f
p "The result is #{calc_a(x, y)}"
```

calc_a.rb

```
# frozen_string_literal: true

# calc the equation
module Equation
  def calc_a(first_input, second_input)
    mod = (second_input - first_input)

    1 + mod.abs + ((mod.abs**2) / 2) + ((mod.abs**3) / 3)
  end
end
```

equation_spec.rb

```
# frozen_string_literal: true

require_relative 'calc_a'
RSpec.describe Equation do
  include Equation
  describe '#calc_a' do
    context 'when parameters are equal' do
      let!(:first_parameter) { rand(100) }
      let!(:second_parameter) { first_parameter }
    end
  end
end
```

```

    it 'result is 1' do
      expect(calc_a(first_parameter, second_parameter)).to eq(1)
    end
  end
end
end
end

```

Результаты работы программы

```

Input x
5
Input y
4
"The result is 2.8333333333333335"
radicalovich@MacBook-Air-Valeriy 1 % rspec '/Users/radicalovich/Documents/LIP/LAB/5/1/equation_spec.rb'
.
Finished in 0.00206 seconds (files took 0.07608 seconds to load)
1 example, 0 failures

```

Часть 2

Ведомость на зарплату представлена как два массива. Один содержит фамилии работников цеха, а второй – их зарплату за текущий месяц. Найдите фамилию работника, зарплата которого наименее отклоняется от средней зарплате всех работников за текущий месяц. Найдите фамилии двух работников с наибольшей зарплатой. Удалите из ведомости на зарплату сведения о работнике, зарплата которого минимальна.

Код программы console.rb

```

# frozen_string_literal: true

require_relative 'functions'

name = []
salary = []

def input(name, salary)
  print "\nДля остановки ввода вместо ввода фамилии сотрудника нажмите Enter"

  loop do
    print "\nВведите фамилию сотрудника: "
    name_input = gets.chomp
    break if name_input == ''

    name << name_input

    print "\nВведите зарплату этого сотрудника: "
    salary_input = gets.chomp.to_i
    salary << salary_input
  end
end

```

```

end

end

def result_output(name, salary)
  print "\n Фамилия сотрудника, зарплата которого наименее отклоняется от средней: "
  #{Employers.almost_average(name, salary)}"
  print "\n Фамилии двух работников с наибольшей зарплатой: "
  Employers.find_2_max(name, salary).each { |n| print "#{n}, " }
  Employers.delete_min!(name, salary)
  print "\nФамилии сотрудников, после удаления: "
  name.each { |n| print "#{n} " }
  print "\nЗарплаты сотрудников, после удаления: "
  salary.each { |n| print "#{n} " }
end

input(name, salary)
print "\nФамилии сотрудников: "
name.each { |n| print "#{n} " }
print "\nЗарплаты сотрудников: "
salary.each { |n| print "#{n} " }
result_output(name, salary)

```

functions.rb

```

# frozen_string_literal:true
# all stuff

class Employers
  def self.find_2_max(name, salary)
    array_of_2_max = salary.each_with_index.max(2)
    [name[array_of_2_max[0][1]], name[array_of_2_max[1][1]]]
  end

  def self.almost_average(name, salary)
    average_salary = salary.sum / salary.size
    res_index = salary.index(salary.min_by { |n| (n - average_salary).abs })
    name[res_index]
  end

  def self.delete_min!(name, salary)
    delete_index = salary.index(salary.min)
    salary.delete_at(delete_index)
    name.delete_at(delete_index)
  end
end

```

```

end
end

employers_spec.rb

# frozen_string_literal: true

require_relative 'functions'
RSpec.describe Employers do
  describe '#self_find_2_max' do
    context 'length is 2' do
      let!(:names) { ['Ягмуров', 'Полубояров'] }
      let!(:salaries) { [1000, 1600] }
      it 'result is array' do
        expect(described_class.find_2_max(names, salaries)).to eq(["Полубояров", "Ягмуров"])
      end
    end
  end
end
end
end

```

Результаты работы программы

Фамилии сотрудников: Полубояров Ягмуров Спасскова Курицын

Зарплаты сотрудников: 500 700 1300 789

Фамилия сотрудника, зарплата которого наименее отклоняется от средней:
Курицын

Фамилии двух работников с наибольшей зарплатой: Спасскова, Курицын,

Фамилии сотрудников, после удаления: Ягмуров Спасскова Курицын

Зарплаты сотрудников, после удаления: 700 1300 789 %

Часть 3

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами, в конце — точка. Слова в строке образуют пары: каждое первое слово — заменяемое, каждое второе — замещающее. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в замене всех заменяемых слов замещающими. Вывести на печать исходную и скорректированную последовательности строк.

Автоматический тест программы обязательно должен генерировать случайные строки в соответствии с правилами, перечисленными в задании.

Код программы console.rb

```

# frozen_string_literal: true

```

```

require_relative 'functions'

strings = []

def input(string_array)
  loop do
    print "\nВведите строку, оканчивающуюся точкой или нажмите Enter для остановки ввода:\n"
    user_input = gets.chomp
    break unless /[.]/.match?(user_input)

    string_array << user_input
  end
end

def result_output(string_array)
  print "Результат ввода:\n"
  string_array.each { |n| print "#{n}\n" }
  print "Результат вывода:\n"
  string_array.each { |n| puts ChangePositionsOfWords.change_words(n) }
end

input(strings)
result_output(strings)

```

change_positions_of_the_words_spec.rb

```

# frozen_string_literal: true

require 'faker'
require_relative 'functions'

RSpec.describe ChangePositionsOfWords do
  describe '#self_change.words' do
    context '3 words line' do
      let!(:first_word) { Faker::Lorem.word }
      let!(:second_word) { Faker::Lorem.word }
      let!(:third_word) { Faker::Lorem.word }
      let!(:sentence) { "#{first_word} #{second_word} #{third_word}" }
      let!(:result_sentence) { "#{second_word} #{second_word} #{third_word}" }

      it 'then the result is' do
        expect(described_class.change_words(sentence)).to eq(result_sentence)
      end
    end
  end
end

```

functions.rb

```
# frozen_string_literal: true

# change position
class ChangePositionsOfWords
  def self.change_words(string)
    result_string = ''
    temp = ''
    string.gsub(/\w+/).with_index(1) do |n, i|
      temp = n
      if i.even?
        result_string += "#{temp} #{temp} "
        temp = ''
      end
    end
    (result_string + temp).strip
  end
end
```

Результаты работы программы

Введите строку, оканчивающуюся точкой или нажмите Enter для остановки ввода:
Hello world.

Введите строку, оканчивающуюся точкой или нажмите Enter для остановки ввода:
I am fine.

Введите строку, оканчивающуюся точкой или нажмите Enter для остановки ввода:

Результат ввода:

Hello world.

I am fine.

Результат вывода:

world world

am am fine

Вывод

В ходе выполнения лабораторной работы был изучен основной синтаксис языка Ruby, освоены основные методы работы с различными видами объектов (массивы, строки и тд).