



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 № 5

Название: ЛР 5

Дисциплина: Языки интернет-программирования

Студент

ИУ6-32Б

(Группа)

(Подпись, дата)

Джафаров Э.Э.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Петрова Я.С.

(И.О. Фамилия)

Москва, 2022

Цель работы:

Целью работы является ознакомление с языком Ruby и написании программ и тестов

Задание:

Часть 1

Вычислить: $y = e^x / \operatorname{tg}(x^3 - 5) + x^2$

Часть 2

Дана строка, состоящая из дат в ISO-формате, разделённых одним пробелом. Найти самую близкую дату к сегодняшнему дню. (2022-01-11 2022-01-24 . . .)

Часть 3

Дана последовательность строк. Строки содержат слова, разделенные пробелом. Используя цифровой шифр, например 31206, зашифровать каждую строку по следующей методике: 31206 312063 12 063

Пирог сгорел до тла

Ткток фкррло ер есг

То есть, к каждой букве применяют соответствующую цифру для определения смещения этой буквы, с целью получения буквы шифра. Написать программу, обеспечивающую ввод строк, шифровку и дешифровку.

Вывести на печать зашифрованную и подвергнутую дешифровке последовательности строк.

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

Решение:

Часть 1

1. Основная программа

```
# frozen_string_literal: true

# Calculator
class Calculate
  include Math

  def initialize(number)
    @x = number
  end

  def calc
    @ex = exp(@x)
    @tang = tan(@x**3 - 5)
    @y = @ex / @tang + @x**2
  end

  def get
    @y
  end

  def prt
    puts "x: #{@x}"
    puts "exponenta: #{@ex}"
    puts "tangent: #{@tang}"
    puts "Finally: #{@y.round(2)}"
  end
end
```

2. Программа для взаимодействия с пользователем через консоль

```
# frozen_string_literal: true

require './main'

x = gets.to_f
a = Calculate.new(x)
a.calc
a.prt
```

3. Тест

```
# frozen_string_literal: true

# test for Part 1
require './main'
require './get_input'

RSpec.describe Calculate do
  before(:example) do
    include Math
    @x = rand(10)
  end

  it 'should return right values with 3' do
    a = Calculate.new(3)
    a.calc
    expect(a.get.round(2)).to eq(2278.13)
  end

  it 'should return right values with 0' do
    b = Calculate.new(0)
    b.calc
    expect(b.get.round(1)).to eq(0.3)
  end

  it 'should return right values with 10' do
    c = Calculate.new(10)
    c.calc
    expect(c.get.round(2)).to eq(-17_928.85)
  end
end
```

4. Результат

1.1 Основной программы

```
x: 5.0
exponenta: 148.4131591025766
tangent: 0.7131230097859091
Finally: 233.1171930592067
```

1.2 Теста

```
...
Finished in 0.06067 seconds (files took 0.47517 seconds to load)
3 examples, 0 failures
```

5. Проверка Rubocop:

```
m@dznet12k@m-dznet12k-VivoBook-A505Laptop:
Inspecting 3 files
...
3 files inspected, no offenses detected
```

Часть 2

1. Основная программа

```
# frozen_string_literal: true

# Errors
class DayError < StandardError; end
class MonthError < StandardError; end

# Dates
class Dates
  def initialize(arr)
    @arr = arr
    @now = Time.now
    @now_days = @now.year * 365 + @now.month * 30 + @now.day
    @index = 0
    @maxi = 0
  end

  def find_nearest
    @arr.each.with_index do |x, index|
      year, month, day = x.split('-').map(&:to_i)
      if check(month, day)
        days = year * 365 + month * 30 + day
        if (@now_days >= days) && (days > @maxi)
          @maxi = days
          @index = index
        end
      end
    end
  end

  def get
    @arr[@index]
  end
end

def check(month, day)
  raise DayError, 'Too much days' if day > 30 || day <= 0
  raise MonthError, 'Too much months' if month > 12 || month <= 0

  1
rescue DayError
  puts "\nDay should be between 1 and 30"
  0
rescue MonthError
  puts "\nMonth should be between 1 and 12"
  0
end
```

2. Программа для взаимодействия с пользователем через консоль

```
# frozen_string_literal: true

# get input
require './dates'

puts 'year-month-day:'
date = gets
a = Dates.new(date.split)
a.find_nearest
puts a.get
```

3. Тест

```
# frozen_string_literal: true

require './dates'

RSpec.describe Dates do
  it 'should return right values' do
    a = Dates.new(%w[2020-12-12 2021-12-12 2022-10-23])
    a.find_nearest
    expect(a.get).to eq('2022-10-23')
  end

  it 'should return right values through MonthError' do
    a = Dates.new(%w[2020-13-12 2021-12-12 2022-10-23])
    a.find_nearest
    expect(a.get).to eq('2022-10-23')
  end

  it 'should return right values through DayError' do
    a = Dates.new(%w[2020-10-33 2021-12-12 2022-10-23])
    a.find_nearest
    expect(a.get).to eq('2022-10-23')
  end

  it 'should return right values through future day' do
    a = Dates.new(%w[2020-10-29 2021-12-12 2022-10-23])
    a.find_nearest
    expect(a.get).to eq('2022-10-23')
  end
end
```

4. Результат

1.1 Основной программы

```
2022-09-09 2022-02-04 2019-10-17 2010-8-9
["2022-09-09", "2022-02-04", "2019-10-17", "2010-8-9"]
2022-09-09
```

1.2 Теста

```
.
Month should be between 1 and 12
.
Day should be between 1 and 30
..

Finished in 0.00853 seconds (files took 0.26172 seconds to load)
4 examples, 0 failures
```

5. Проверка Rubocop:

```
Inspecting 3 files
...

3 files inspected, no offenses detected
```

Часть 3

1. Основная программа

```
# frozen_string_literal: true

# class
class Converter
  def initialize(str, num)
    @string = str
    @numbers = num
  end

  def convert
    @string.each_with_index do |word, index|
      word.each_char.with_index do |elem, j|
        code = elem.ord
        modified = code + @numbers[index][j].to_i
        modified_symbol = modified.chr('UTF-8')
        word[j] = modified_symbol
      end
    end
  end

  def get
    @string
  end
end
```


2. Программа для взаимодействия с пользователем через консоль

```
# frozen_string_literal: true

require './converter'

print 'Enter a string: '
string = gets.split
print 'Enter a numbers: '
number = gets.split
a = Converter.new(string, number)
a.convert
puts a.get
```

3. Тест

```
# frozen_string_literal: true

require './converter'

RSpec.describe Converter do
  before(:example) do
    @words = []
    @numbers = []
    rand(1..5).times do
      @words.append(create_string)
    end
    create_number(@words, @numbers)
  end

  it 'should return inintially value' do
    a = Converter.new(@words, @numbers)
    a.convert
    expect(@words).to eq(unconvert(a.get, @numbers))
  end
end
```

4. Результат

1.1 Основной программы

```
Enter a string: пирог сгорел до тла
Enter a numbers: 12345 61232 90 123
ркути
чдрузл
но
унг
```


1.2 Теста

```
mpdznotik@mpdznotik-VivoBook-ASUSLaptop-X512DK-X512DK:~/User/Dzn
.
Finished in 0.0082 seconds (files took 0.22783 seconds to load)
1 example, 0 failures
```

5. Проверка Rubocop:

```
Inspecting 3 files
...
3 files inspected, no offenses detected
```

6. Проверка Reek:

6.1 Первой части:

```
rk/IPL-Laboratory5$ reek Part\ 1/main.rb
Inspecting 1 file(s):
.
```

6.2 Второй части:

```
rk/IPL-Laboratory5$ reek Part\ 2/dates.rb
Inspecting 1 file(s):
.
```

6.3 Третьей части:

```
rk/IPL-Laboratory5$ reek Part\ 3/converter.rb
Inspecting 1 file(s):
.
```

Вывод:

Я познакомился с языком Ruby и попрактиковался в написании программ и тестов