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

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

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № \_\_4\_\_**

**Дисциплина: \_\_Языки Интернет-программирования\_\_\_\_\_**

**Название лабораторной работы:** \_\_\_\_Javascript\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент гр. \_ИУ6-33Б**\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Яныгин Д.С.\_\_\_**

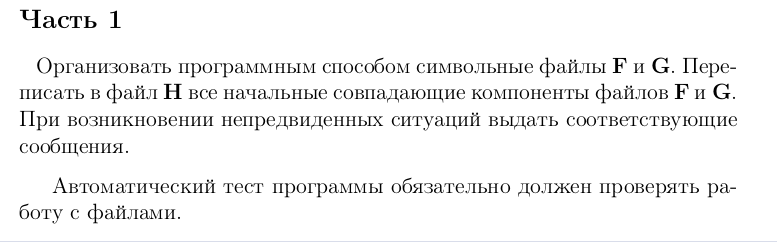
(Подпись, дата) (И.О. Фамилия)

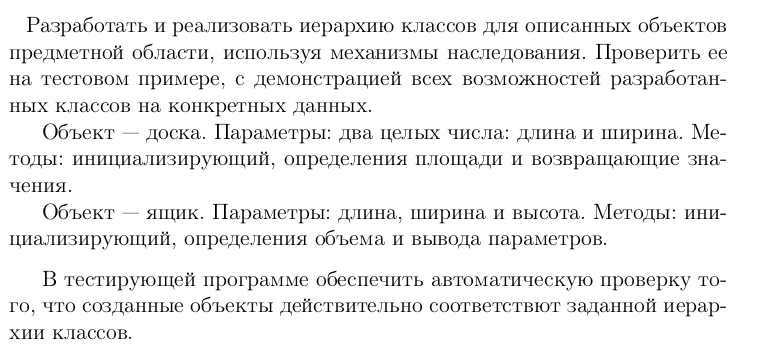
Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

**Задание:**

Часть 2



**Часть 1**

**LR7\_1\_func.rb**

def create\_file(file\_name, str)

File.open(file\_name, 'w') do |f|

f.puts(str)

end

end

def find\_equal\_start(filename1, filename2)

str1 = File.read filename1

str2 = File.read filename2

components1 = str1.split

components2 = str2.split

sorted\_array = components1.zip(components2).take\_while { |elem1, elem2| elem1 == elem2 }

result = sorted\_array.reduce('') do |str, elem|

str + elem[0] + ' '

end

result

end

**LR7\_1\_test.rb**

load 'LR7\_1\_func.rb'

require 'minitest/autorun'

def generate\_text(num\_of\_components)

# words = %w[accommodate arm amount promote ring underline attack convict shop]

words = %w[right left]

random\_file\_text1 = ''

random\_file\_text2 = ''

output = ''

is\_first = true

num\_of\_components.times do

word1 = words[rand(0..words.size - 1)]

word2 = words[rand(0..words.size - 1)]

random\_file\_text1 += word1 + ' '

random\_file\_text2 += word2 + ' '

if (word1 == word2) && (is\_first)

output += word1 + ' '

else is\_first = false

end

end

[random\_file\_text1, random\_file\_text2, output]

end

# Class for setting any tests on logic methods

class Tests < MiniTest::Test

def setup; end

def test\_files

100.times do

random\_strings = generate\_text(5)

create\_file 'files/F.txt', random\_strings[0]

create\_file 'files/G.txt', random\_strings[1]

File.open('files/H.txt', 'w') do |f|

f.puts(find\_equal\_start('files/F.txt', 'files/G.txt'))

end

str\_in\_file = (File.read 'files/H.txt')

output\_str = random\_strings[2]

str\_in\_file = str\_in\_file[0...-1] # truncate \n

assert\_equal(str\_in\_file, output\_str)

end

end

end

**LR7\_1\_user.rb**

load 'LR7\_1\_func.rb'

puts 'Input text for file F.txt'

arg1 = gets.chomp

puts 'Input text for file G.txt'

arg2 = gets.chomp

create\_file 'files/F.txt', arg1

create\_file 'files/G.txt', arg2

File.open('files/H.txt', 'w') do |f|

f.puts(find\_equal\_start('files/F.txt', 'files/G.txt'))

end

**Результат:**

Рис. 1

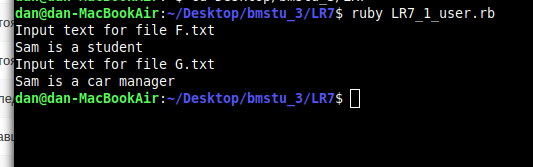
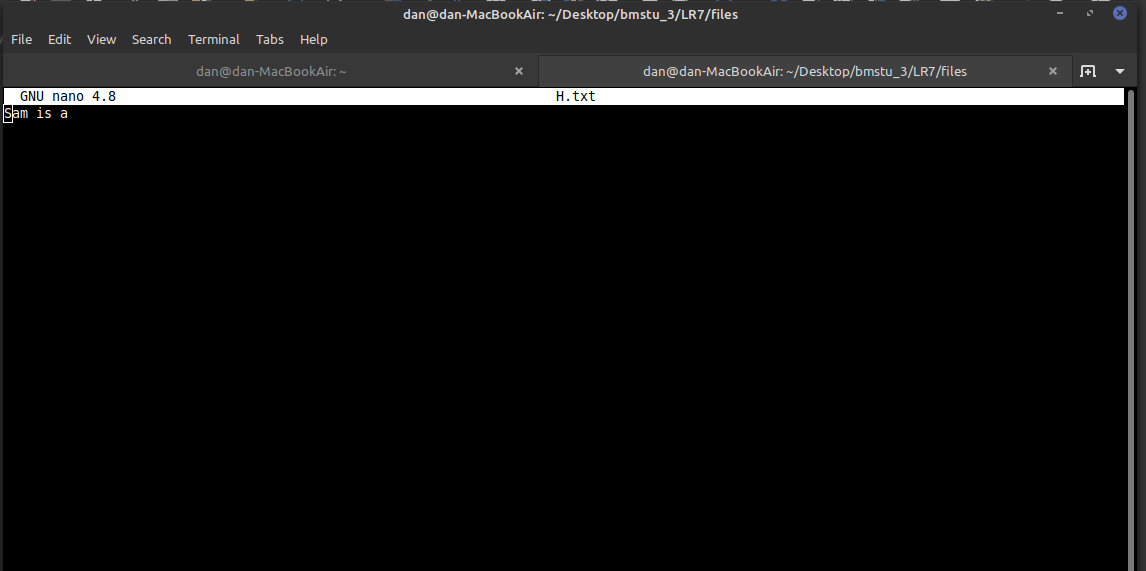


Рис. 2 результат



**Часть 2**

**LR7\_2\_func.rb**

class Board

attr\_reader :length, :width

def initialize(length, width)

@length = length

@width = width

end

def area

length \* width

end

end

class Box < Board

attr\_reader :height

def initialize(length, width, height)

super length, width

@height = height

end

def volume

area \* height

end

def output

puts 'Длина - ' + length.to\_s

puts 'Ширина - ' + width.to\_s

puts 'Высота - ' + height.to\_s

puts 'Объем - ' + volume.to\_s

end

end

**LR7\_2\_test.rb**

load 'LR7\_2\_func.rb'

require 'minitest/autorun'

class TestRectangle < Minitest::Test

def test\_square\_returns\_true\_if\_length\_eq\_to\_width

assert\_equal 6, Board.new(2, 3).area

end

def test\_type\_returns\_cube\_if\_all\_dims\_equal

assert\_equal 6, Box.new(1, 2, 3).volume

end

end

**LR7\_2\_user.rb**

фload 'LR7\_2\_func.rb'

puts 'Введите длину, ширину и высоту через пробел'

params = gets.chomp.split.map { |x| Float x, exception: false }.compact

unless params.size == 3

puts 'Некорректный ввод'

exit 1

end

length, width, height = params

board = Board.new length, width

box = Box.new length, width, height

puts 'Параметры прямоугольника:'

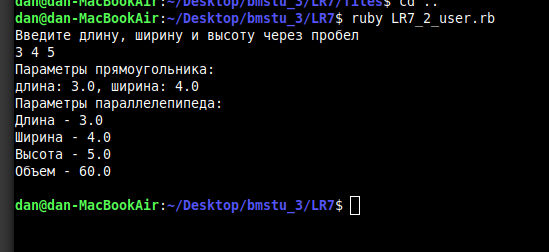
puts "длина: #{board.length}, ширина: #{board.width}"

puts 'Параметры параллелепипеда:'

puts box.output

**Результат:**

Рис. 3



**Вывод:**

Была проведена работа с файлами и классами в ruby