|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | **5** |

**Название:** Создание консольного приложения Ruby.

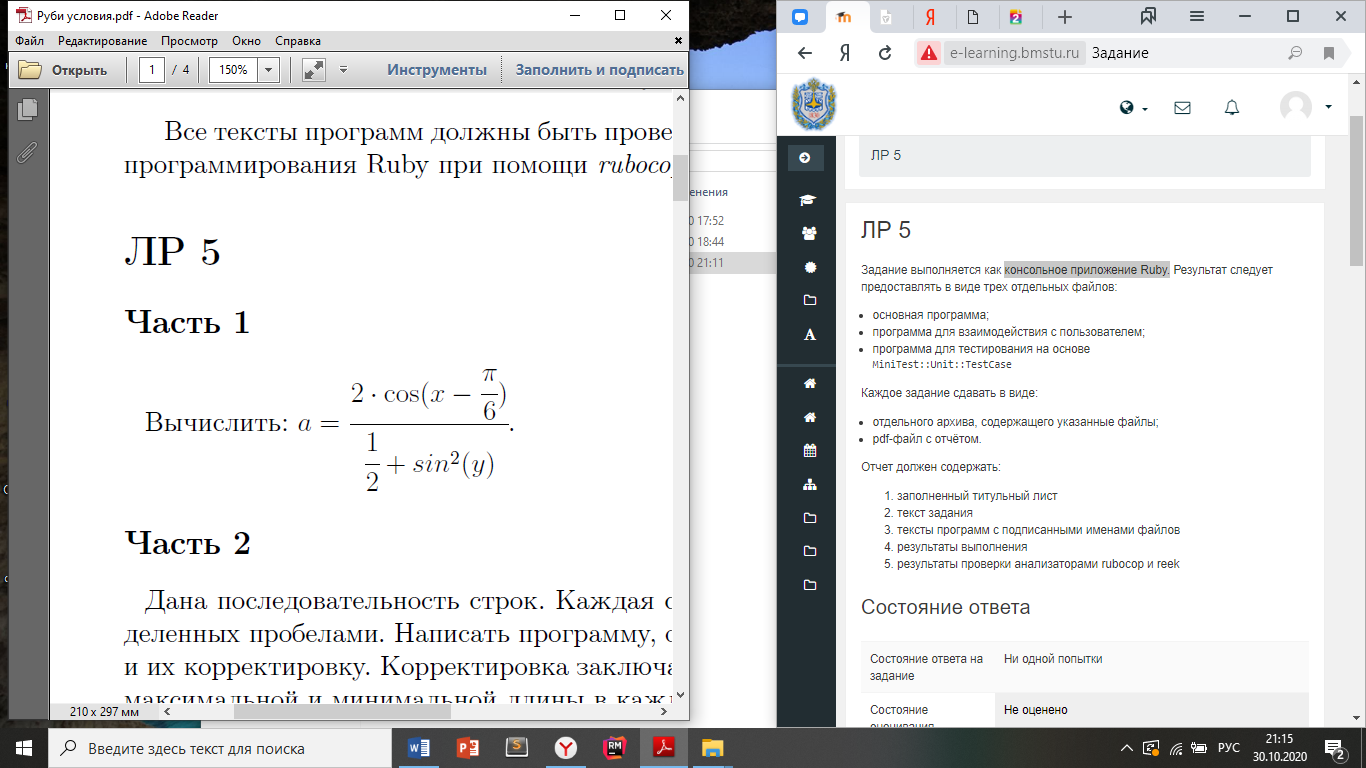
**Дисциплина:** Языки интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6 |  |  |  |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**Лабораторная работа №4**

**Часть 1:**



**Код в console.rb:**

*# frozen\_string\_literal: true  
  
require\_relative* 'main'  
print 'x: '  
*x\_num* = gets.to\_f  
print 'y: '  
*y\_num* = gets.to\_f  
puts "Result: #{maths *x\_num*, *y\_num*}"

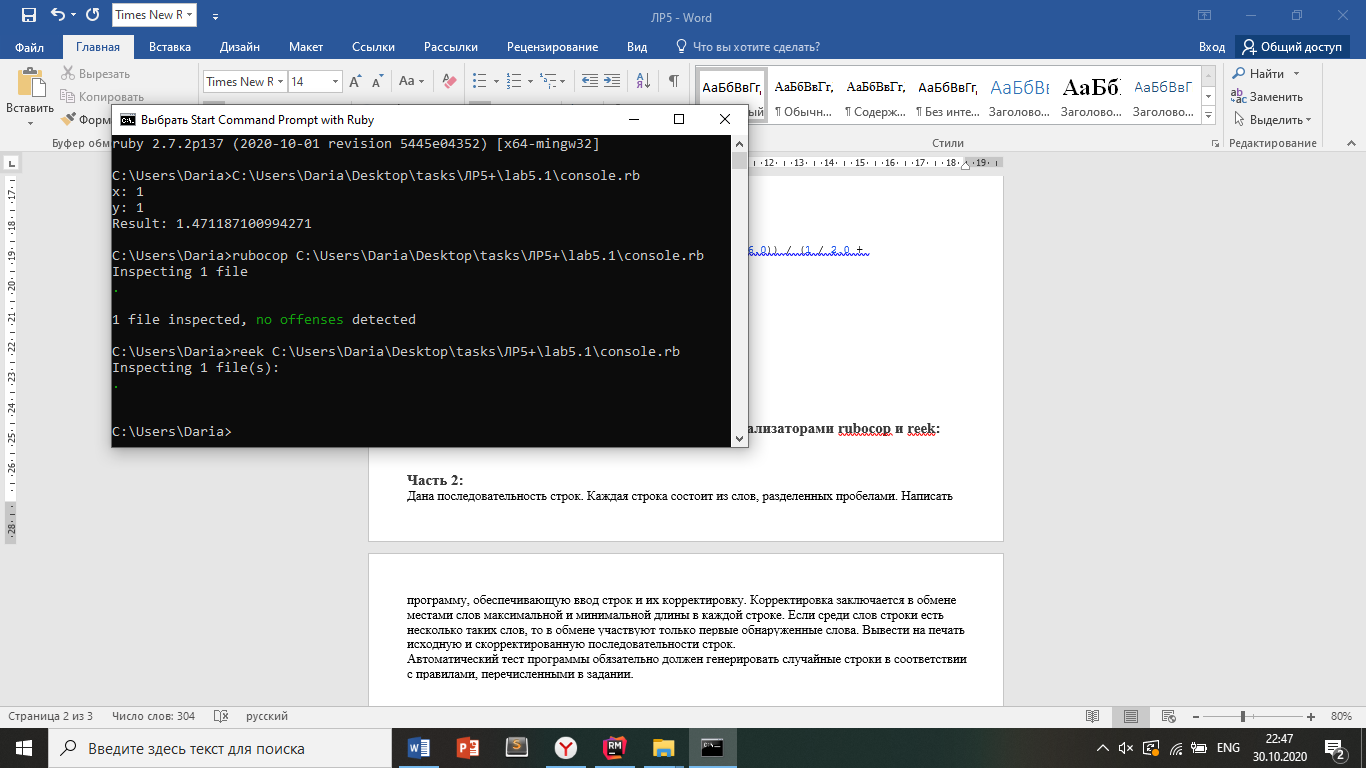
**Код в main.rb:**

*# frozen\_string\_literal: true*def maths(x\_num, y\_num)  
 (2 \* *Math*.cos(x\_num - *Math*::*PI* / 6.0)) / (1 / 2.0 + *Math*.sin(y\_num)\*\*2)  
end

**Код в test.rb:**

*# frozen\_string\_literal: true  
  
require* 'minitest/autorun'  
*require* 'minitest/assertions'  
*require\_relative* 'main'  
  
*#* ***class*** *with test*class *Testing* < *Minitest*::*Test* def *setup* super  
 @del = 0.001  
 @fail = 'Error'  
 end  
  
 def *test\_1  
 x\_num* = rand(-10..10)  
 *y\_num* = rand(-10..10)  
 assert\_in\_delta((2 \* *Math*.cos(*x\_num* - *Math*::*PI* / 6.0)) / (1 / 2.0 + *Math*.sin(*y\_num*)\*\*2), maths(*x\_num*, *y\_num*),  
 @del, @fail + '2')  
 end  
  
 def *test\_2  
 x\_num* = 1  
 *y\_num* = 1  
 *res* = 1.471187100994271  
 assert\_in\_delta(*res*, maths(*x\_num*, *y\_num*),  
 @del, @fail + ' 1')  
 end  
end

**Результат выполнения и результаты проверки анализаторами rubocop и reek:**



**Часть 2:**  
Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в обмене местами слов максимальной и минимальной длины в каждой строке. Если среди слов строки есть несколько таких слов, то в обмене участвуют только первые обнаруженные слова. Вывести на печать исходную и скорректированную последовательности строк.  
Автоматический тест программы обязательно должен генерировать случайные строки в соответствии с правилами, перечисленными в задании.

**Код в console.rb:**

*# frozen\_string\_literal: true  
  
require\_relative* 'main'  
*ARGF*.each do |*line*|  
 puts output(*line*.chomp)  
end

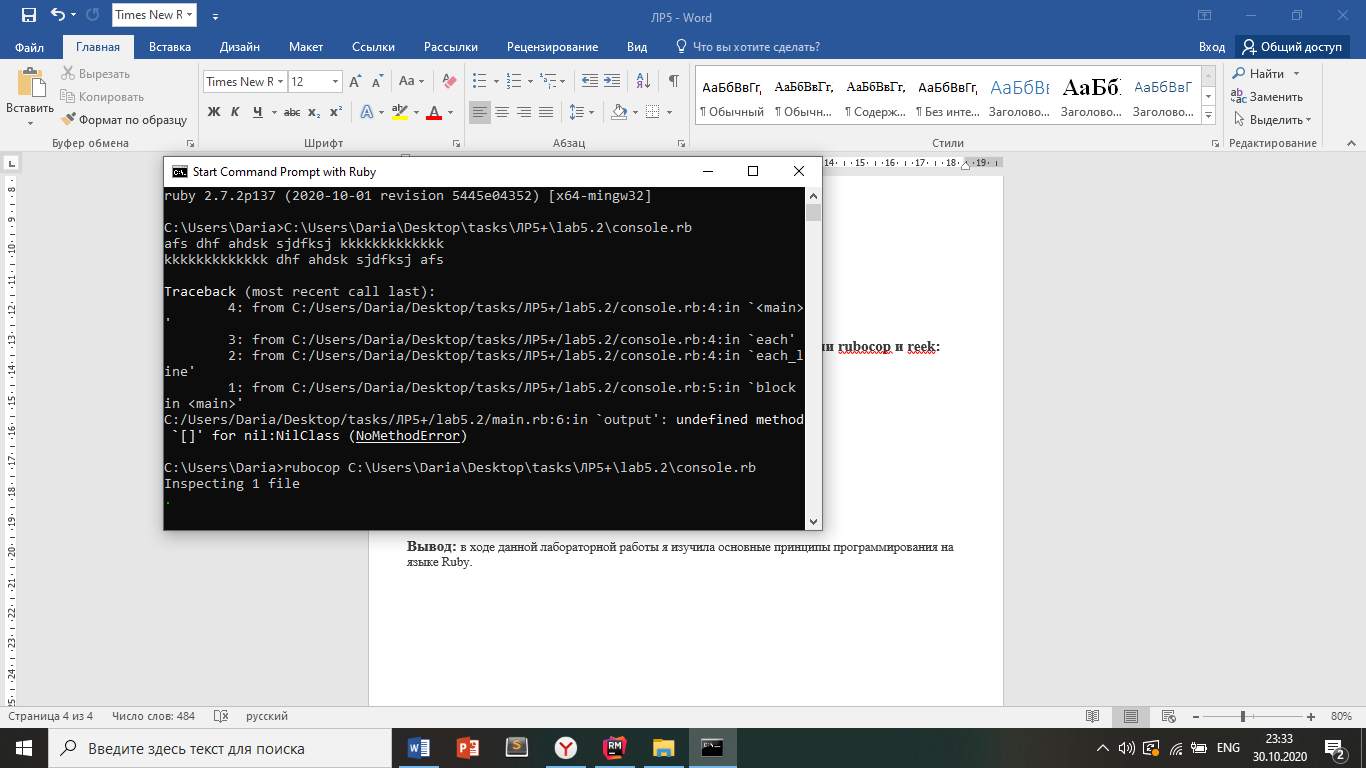
**Код в main.rb:**

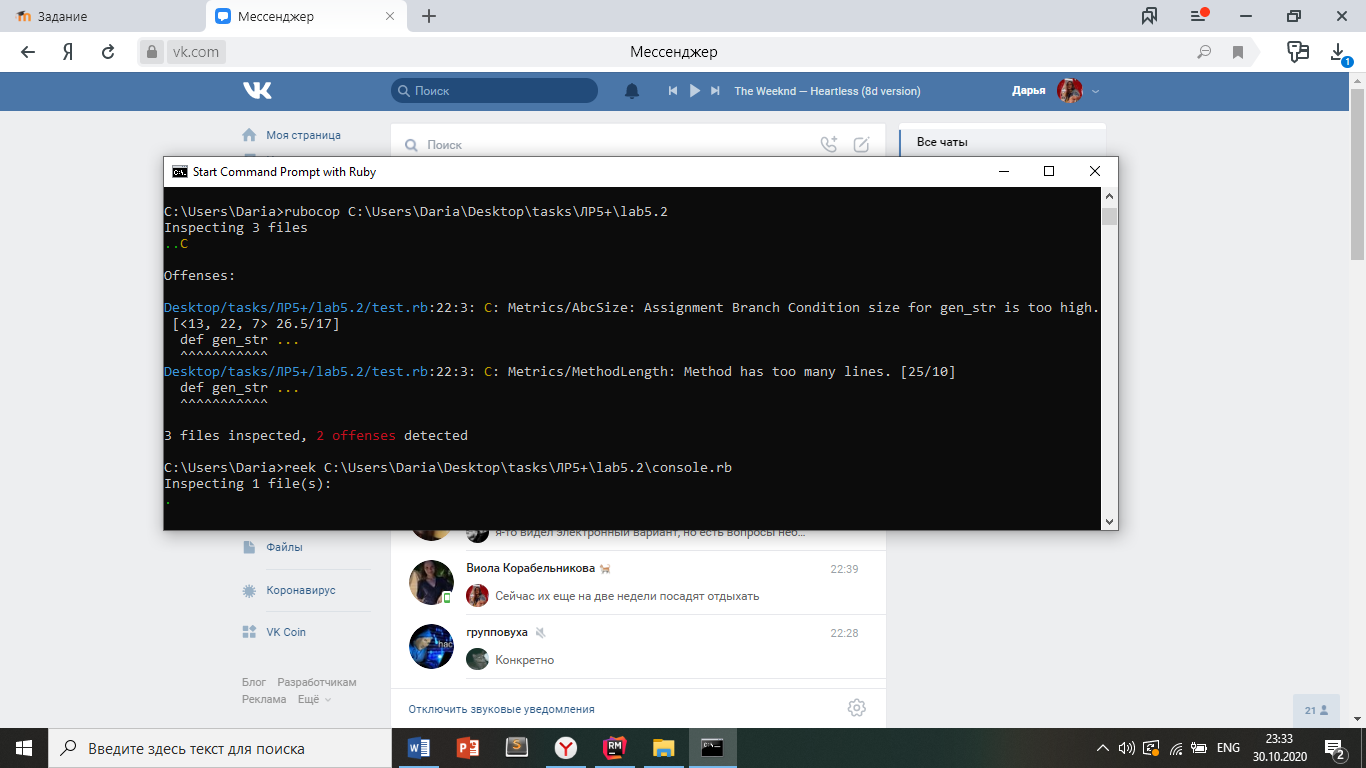
*# frozen\_string\_literal: true*def *output*(*str*)  
 *arr* = *str*.split  
 *minmax* = *arr*.each\_with\_index.minmax\_by { |*word*, *\_index*| *word*.length }  
 *arr*[*minmax*[0][1]] = *minmax*[1][0]  
 *arr*[*minmax*[1][1]] = *minmax*[0][0]  
 *arr*.join ' '  
end

**Код в test.rb:**

*# frozen\_string\_literal: true  
  
require* 'minitest/autorun'  
*require* 'minitest/assertions'  
*require\_relative* 'main'  
  
*#* ***class*** *with test*class *Testing* < *Minitest*::*Test  
 CHARS* = ('а'..'я').to\_a  
  
 def *test\_1* 10.times do  
 *str* = gen\_str  
 assert\_equal *str*[1], output(*str*[0])  
 end  
 end  
  
 def *gen\_word*(*len\_word*)  
 (*CHARS*.sample *len\_word*).join ''  
 end  
  
 def *gen\_str  
 max\_size* = rand(7..10)  
 *min\_size* = rand(2..6)  
 *max\_word* = gen\_word *max\_size  
 min\_word* = gen\_word *min\_size  
 str\_length* = rand(3..10)  
 *bool1* = false  
 *bool2* = false  
 *arr\_input* = []  
 *arr\_result* = []  
 *str\_length*.times do |*i*|  
 if *i* == rand(*i*...*str\_length* - 1) && !*bool2  
 arr\_input* << [*min\_word*]  
 *arr\_result* << [*max\_word*]  
 *bool2* = true  
 elsif *i* == rand(*i*...*str\_length*) && !*bool1  
 arr\_input* << [*max\_word*]  
 *arr\_result* << [*min\_word*]  
 *bool1* = true  
 else  
 *word* = gen\_word rand((*min\_size* + 1)...*max\_size*)  
 *arr\_input* << *word  
 arr\_result* << *word* end  
 end  
 [*arr\_input*.join(' '), *arr\_result*.join(' ')]  
 end  
end

**Результат выполнения и результаты проверки анализаторами rubocop и reek:**





**Вывод:** в ходе данной лабораторной работы я изучила основные принципы программирования на языке Ruby.