



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 № 7

Название: Работа с файлами в Ruby

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-33Б

(Группа)

(Подпись, дата)

И.А. Нуруллаев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

Часть 1

Организовать программным способом символьный файл *F*. Записать в файл *G* строки файла *F* в обратном порядке, удалив символы *e, E, r, t, U*. При возникновении непредвиденных ситуаций выдать соответствующие сообщения. Автоматический тест программы обязательно должен проверять работу с файлами.

Решение:

```
WPL_bmstu - test.rb

1  require 'minitest/autorun'
2  require_relative 'main'
3
4  # File tester
5  class Test < Minitest::Test
6    def initialize(name)
7      super name
8      @input_file_name = 'F.txt'
9      @output_file_name = 'G.txt'
10   end
11
12   def test_create
13     create_file(@input_file_name, 'test string')
14     assert_path_exists @input_file_name, 'Не получилось создать файл'
15   end
16
17   def self.create_string
18     (0...(rand(10..20))).map { ('0'..'z').to_a[rand(62)] }.join
19   end
20
21   def self.check_result(string)
22     %w[e E r t U].all? { |char| !string.include? char }
23   end
24
25   def test_first
26     create_file('F.txt', (4..10).map { Test.create_string }.join("\n"))
27     process_file(@input_file_name, @output_file_name)
28     assert_path_exists @output_file_name, 'Файл не был создан'
29     result = File.read(@output_file_name)
30     assert Test.check_result(result), 'Программа отработала неверно'
31   end
32 end
33
```

```
WPL_bmstu - main.rb

1  def create_file(file_name, string)
2    File.write(file_name, string)
3  end
4
5  def process_file(input_file_name, output_file_name)
6    if File.exist?(input_file_name)
7      input_enum = File.readlines(input_file_name)
8      create_file(output_file_name, input_enum.reverse.map { |string| string.tr('eErtU', '') }.join)
9    else
10     puts('Такого файла нет')
11   end
12 end
13
```

```

WPL_bmstu - user.rb

1  require_relative 'main'
2
3  create_file('F.txt', "Lorem ipsum dolor sit amet, consectetur adipiscing elit.
4  Nam eleifend erat non metus auctor bibendum.
5  Morbi ultricies justo id posuere tristique.
6  Donec elementum mauris nec erat commodo, sed sagittis dolor elementum.
7  Ut commodo augue in arcu aliquam, id dapibus tellus suscipit.
8  Aenean commodo justo quis luctus viverra.
9  Ut ac tortor eget lorem efficitur sagittis.
10 ")
11
12 process_file('F.txt', 'G.txt')
13

```

Результат выполнения программ:

```

PS C:\Users\might\Desktop\WPL_bmstu\Lab7\Part 1> ruby .\user.rb
PS C:\Users\might\Desktop\WPL_bmstu\Lab7\Part 1> ruby .\test.rb
Run options: --seed 30040

# Running:

..

Finished in 0.019088s, 104.7790 runs/s, 157.1685 assertions/s.
2 runs, 3 assertions, 0 failures, 0 errors, 0 skips

```

Файлы после отработки программы *user.rb*

```

WPL_bmstu - F.txt

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nam eleifend erat non metus auctor bibendum.
Morbi ultricies justo id posuere tristique.
Donec elementum mauris nec erat commodo, sed sagittis dolor elementum.
Ut commodo augue in arcu aliquam, id dapibus tellus suscipit.
Aenean commodo justo quis luctus viverra.
Ut ac tortor eget lorem efficitur sagittis.

```

```

WPL_bmstu - G.txt

ac oo g lom fficiu sagiis.
Anan commodo juso quis lucus viva.
commodo augu in acu aliquam, id dapibus llus suscipi.
Donc lmnum mauis nc a commodo, sd sagiis dolo lmnum.
Mobi ulicis juso id posu isiqu.
Nam lifnd a non mus auco bibndum.
Lom ipsum dolo si am, conscu adipiscing li.

```

Часть 2

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере с демонстрацией всех возможностей разработанных классов на конкретных данных. Объект — треугольник, заданный длинами сторон. Объект умеет выводить на экран значение своих полей и отвечать на запрос о площади фигуры. Объект — четырехугольник, заданный длинами сторон и диагонали. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их значения и площадь. В тестирующей программе обеспечить автоматическую проверку того, что созданные объекты действительно соответствуют заданной иерархии классов

Решение:

```
WPL_bmstu - main.rb

1 # Class triangle
2 class Triangle
3   attr_reader :side_a, :side_b, :side_c
4
5   def initialize(side_a, side_b, side_c)
6     @side_a = side_a
7     @side_b = side_b
8     @side_c = side_c
9   end
10
11   def print
12     puts "Треугольник со сторонами #{@side_a}, #{@side_b}, #{@side_c}"
13   end
14
15   def area
16     p = (@side_a + @side_b + @side_c) / 2
17     Math.sqrt(p * (p - @side_a) * (p - @side_b) * (p - @side_c))
18   end
19 end
20
21 # Class quadrilateral
22 class Quadrilateral < Triangle
23   attr_reader :side_d, :diagonal
24
25   def initialize(side_a, side_b, side_c, side_d, diagonal)
26     super side_a, side_b, side_c
27     @side_d = side_d
28     @diagonal = diagonal
29   end
30
31   def print
32     puts "Четырехугольник со сторонами #{@side_a}, #{@side_b}, #{@side_c}, #{@side_d}
33 , и диагональю #{@diagonal}"
34   end
35
36   def area
37     p1 = (@side_a + @side_b + @diagonal) / 2
38     area1 = Math.sqrt(p1 * (p1 - @side_a) * (p1 - @side_b) * (p1 - @diagonal))
39     p2 = (@side_c + @side_d + @diagonal) / 2
40     area2 = Math.sqrt(p2 * (p2 - @side_c) * (p2 - @side_d) * (p2 - @diagonal))
41     area1 + area2
42   end
43 end
```

```

WPL_bmstu - test.rb

1 require 'minitest/autorun'
2 require_relative 'main'
3
4 # Class tester
5 class Test < Minitest::Test
6   def test_hierarchy
7     quadrilateral = Quadrilateral.new(3, 4, 3, 4, 5)
8     assert_kind_of Triangle, quadrilateral
9   end
10
11   def test_area_triangle
12     triangle = Triangle.new(3, 4, 5)
13     assert_in_epsilon triangle.area, 6.0, 0.001, 'Площадь треугольника вычислена неправильно'
14   end
15
16   def test_area_quadrilateral
17     quadrilateral = Quadrilateral.new(3, 4, 3, 4, 5)
18     assert_in_epsilon quadrilateral.area, 12.0, 0.001, 'Площадь четырехугольника вычислена неправильно'
19   end
20 end
21

```

```

WPL_bmstu - user.rb

1 require_relative 'main'
2
3 triangle = Triangle.new(3, 4, 5)
4 quadrilateral = Quadrilateral.new(3, 4, 3, 4, 5)
5
6 triangle.print
7 quadrilateral.print
8
9 puts("Площадь треугольника - #{triangle.area}")
10 puts("Площадь четырехугольника - #{quadrilateral.area}")
11

```

Результат выполнения программ:

```

PS C:\Users\might\Desktop\WPL_bmstu\Lab7\Part 2> ruby .\user.rb
Треугольник со сторонами 3, 4, 5
Четырехугольник со сторонами 3, 4, 3, 4, и диагональю 5
Площадь треугольника - 6.0
Площадь четырехугольника - 12.0
PS C:\Users\might\Desktop\WPL_bmstu\Lab7\Part 2> ruby .\test.rb
Run options: --seed 28422

# Running:

...

Finished in 0.015437s, 194.3345 runs/s, 194.3345 assertions/s.
3 runs, 3 assertions, 0 failures, 0 errors, 0 skips

```

Проверка кода при помощи *rubocop*

```

PS C:\Users\might\Desktop\WPL_bmstu\Lab7> rubocop
Inspecting 6 files
.....

6 files inspected, no offenses detected

```

Итоговый код данной лабораторной работы доступен по ссылке:

https://github.com/tenessinum/WPL_bmstu/tree/main/Lab7