



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(национальный исследовательский университет)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

О т ч е т

по лабораторной работе № 7

Дисциплина: Языки Интернет-программирования.

Студент гр. ИУ6-33Б

(Подпись, дата)

Дасов Т.Д.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2019

Часть 1

Дан текстовый файл F. Записать в перевернутом виде строки файла F в файл G. Порядок строк в файле G должен быть обратным по отношению к порядку строк исходного файла F. Автоматический тест программы обязательно должен проверять работу с файлами.

Код основной программы:

```
# frozen_string_literal: true

# Reverse strings etc.
class FileChanger
  class << self
    def reverse_file(name, new_name)
      f = File.new(new_name, 'w')
      File.open(name).map { |l| l.chomp.reverse }.reverse.each { |l| f.puts(l) }
      rescue StandardError => e
        puts e
      end
    end
  end
end
```

Программа для взаимодействия с пользователем:

```
# frozen_string_literal: true

require_relative 'file_changer'

# Class which controls I/O
class IOController
  class << self
    def main
      FileGenerator.new.create_file(25, 20)
      FileChanger.reverse_file('F.txt', 'G.txt')
      puts 'Готово!'
    end
  end
end

# Generate random file with special string size
class FileGenerator
  def create_file(f_size, s_size)
    f = File.new('./F.txt', 'w')
    f_size.times { f.puts((0..s_size).map { ('a'..'z').to_a[rand(26)] }.join) }
    f.close
  end
end

IOController.main
```

Тесты

```
# frozen_string_literal: true

require 'test/unit'

# Class which test our program
class MyTest < Test::Unit::TestCase
  def setup; end

  def test_file_ex
    assert(File.file?('F.txt') && File.file?('G.txt'))
  end

  def test_files
    if File.file?('F.txt') && File.file?('G.txt')
      ar = TestFunc.rev(File.open('F.txt').map(&:chomp))
      assert_equal(ar, File.open('G.txt').map(&:chomp))
    else
      assert(false)
    end
  end
end

# Class with test functions
class TestFunc
  class << self
    def rev(arr)
      arr.map(&:reverse).reverse
    end
  end
end
```

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 точками на плоскости. Объект умеет выводить на экран значение своих полей и отвечать на запрос об этих значениях и вычислять площадь фигуры.

Объект — Треугольная призма. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их значения и площадь развертки.

В тестирующей программе обеспечить автоматическую проверку того, что созданные объекты действительно соответствуют заданной иерархии классов.

Код основной программы:

```
# frozen_string_literal: true

require 'matrix'

# Point class
class Point
  attr_accessor :x, :y, :z

  def initialize(args)
    @x = args[:x]
    @y = args[:y]
    @z = args[:z]
  end
end

# Parent class for shape instances
class Shape
  attr_reader :pts

  def initialize(args = nil)
    @pts = []
    args = Array.new(3) { Array.new(3, 0) } if args.nil?
    3.times do |i|
      pnt = Point.new(x: args[i][0], y: args[i][1], z: args[i][2])
      @pts << pnt
    end
  end

  # Abstract method
  def square
    raise 'Error'
  end

  protected

  def dis(pt1, pt2)
    Math.sqrt((pt2.x - pt1.x)**2 + (pt2.y - pt1.y)**2 + (pt2.z - pt1.z)**2)
  end
end

# Triangle class
class Triangle < Shape
  def initialize(args = nil)
    super(args)
  end

  # rubocop:disable Metrics/AbcSize
  def square
    pnt = @pts
    a = [pnt[0].x - pnt[2].x, pnt[0].y - pnt[2].y]
    b = [pnt[1].x - pnt[2].x, pnt[1].y - pnt[2].y]
    0.5 * Matrix[a, b].determinant
  end
  # rubocop:enable Metrics/AbcSize
end

# Prism class
class Prism < Triangle
  def initialize(args = nil)
    @pts = []
  end
end
```

```

    args = Array.new(6) { Array.new(3, 0) } if args.nil?
    6.times do |i|
      pnt = Point.new(x: args[i][0], y: args[i][1], z: args[i][2])
      @pts << pnt
    end
  end

  # rubocop:disable Metrics/AbcSize
  def square
    pnt = @pts
    a = super * 2 + dis(pnt[0], pnt[1]) * dis(pnt[0], pnt[3])
    a += dis(pnt[1], pnt[2]) * dis(pnt[1], pnt[4])
    a + dis(pnt[2], pnt[0]) * dis(pnt[2], pnt[5])
  end
  # rubocop:enable Metrics/AbcSize
end

```

Программа для взаимодействия с пользователем:

```

# frozen_string_literal: true

require_relative 'objects'

# IOController class
class IOController
  class << self
    def main
      tr = Triangle.new([[0, 0, 0], [2, 0, 0], [2, 3, 0]])
      puts "Площадь треугольника: #{tr.square}"
      pnts = [[3, 0, 0], [0, 3, 0], [0, 0, 0], [3, 0, 3], [0, 3, 3], [0, 0, 3]]
      prism = Prism.new(pnts)
      puts "Площадь призмы: #{prism.square}"
    end
  end
end

IOController.main

```

Тесты

```

# frozen_string_literal: true

require 'test/unit'
require_relative 'objects'

# Class for testing
class MyTest < Test::Unit::TestCase
  def setup; end

  def test1
    shape = Shape.new
    tr = Triangle.new
    pr = Prism.new
    assert(shape.class == tr.class.superclass)
    assert(tr.class == pr.class.superclass)
  end
end

```

```
tetovske@pop-os: ~/YAIP/laby/laba7/part1
tetovske@pop-os > ~/YAIP/laby/laba7/part1 > study ● ruby test.rb
Loaded suite test
Started
..
Finished in 0.000582992 seconds.
-----
2 tests, 2 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-----
3430.58 tests/s, 3430.58 assertions/s
tetovske@pop-os > ~/YAIP/laby/laba7/part1 > study ● ruby ../part2/test.rb
Loaded suite ../part2/test
Started
.
Finished in 0.000379308 seconds.
-----
1 tests, 2 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-----
2636.38 tests/s, 5272.76 assertions/s
tetovske@pop-os > ~/YAIP/laby/laba7/part1 > study ●
```

Рис. 1 (Тесты программы)

```
tetovske@pop-os > ~/YAIP/laby/laba7 > study ● rubocop *
Inspecting 6 files
.....
6 files inspected, no offenses detected
tetovske@pop-os > ~/YAIP/laby/laba7 > study ●
```

Рис. 2 (Отчёт Rubocop)

Вывод: в ходе выполнения лабораторной работы была изучена объектная модель, а также работа с файлами на языке программирования Ruby. Все программы были протестированы и проверены на соответствие стилю программой Rubocop.