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 **09.03.03 Прикладная информатика**

О Т Ч Е Т

по лабораторной работе № 7

Дисциплина: Языки интернет-программирования

Студент	ИУ6-35 Б	27.10.2023	И.А. Дулина
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			Е.Ю. Гаврилова
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2023

Вариант 8.

Часть 1.

Задание:

Организовать программным способом файл **F**, компонентами которого являются целые числа, отличные от 0. Числа в файле идут в следующем порядке: десять положительных, десять отрицательных и т.д. Переписать компоненты файла **F** в файл **P** так, чтобы числа расположились в следующем порядке:

---

1. пять положительных, пять отрицательных и т.д.
2. двадцать положительных, двадцать отрицательных и т.д.

Автоматический тест программы обязательно должен проверять работу с файлами.

Код программы:

Файл пользователя (client.rb):

```
# frozen_string_literal: true

require './main'

puts('Введите количество чисел в файле F, кратное 40: ')
f = false
loop do
  count = gets.chomp
  if (count.to_i.to_s == count) && (count.to_i % 40).zero?
    f = true
```

```
    count = count.to_i
    puts('Файл F:')
    fill_file_f(count)
    puts('Файл P:')
    fill_file_p(5)
    fill_file_p(20)
  else
    puts('Введено некорректное значение. Попробуйте ещё раз')
  end
  break if f == true
end
```

Файл основной программы (main.rb):

```
# frozen_string_literal: true

def fill_file_f(count)
  time = count / 10
  file_f = File.open('f.txt', 'w')
  array = []
  iter = 0
  time.times do
    if iter.even?
      10.times { array << Random.rand(1..100) }
    else
      10.times { array << Random.rand(-100..-1) }
    end
    iter += 1
  end
  array.length.times { |i| file_f.write("#{array[i]} ") }
  file_f.close
  printing(file_f.path)
```

```
end

def fill_file_p(number)
  array = File.readlines('f.txt')[0].to_s.split.map(&:to_i)
  array_of_pol = []
  array_of_otr = []
  # разделяем положительные и отрицательные элементы
  array.length.times do |i|
    if (array[i]).positive?
      array_of_pol << array[i]
    else
      array_of_otr << array[i]
    end
  end
  # записываем нужный порядок в новый массив
  new_array = []
  iterpol = 0
  iterotr = 0
  array.length.times do
    number.times do
      new_array << array_of_pol[iterpol]
      iterpol += 1
    end
    number.times do
      new_array << array_of_otr[iterotr]
      iterotr += 1
    end
  end
  # переносим из нового массива с правильным порядком в файл
  file_p = File.open('p.txt', 'w')
  new_array.length.times { |i| file_p.write("#{new_array[i]} ") }
```

```
  file_p.close
  printing(file_p.path)
end

def printing(file_name)
  file = File.open(file_name, 'r')
  puts file.readlines
  file.close
end
```

Файл тестов (test.rb):

```
# frozen_string_literal: true

require 'test/unit'
require './main'

# test
class TestMinMax < Test::Unit::TestCase
  def test1
    file_f = File.open('f.txt', 'w')
    file_f.write('1 2 3 4 5 6 -1 -2 -3 -4 -5 -6')
    file_f.close
    fill_file_p(3) # по 3 пол и по 3 отриц
    assert_equal([1, 2, 3, -1, -2, -3, 4, 5, 6, -4, -5, -6],
                 File.readlines('p.txt')[0].to_str.split.map(&:to_i))
  end
end
```

Результат работы программы:

```
Введите количество чисел в файле F, кратное 40:
40
Файл F:
33 26 57 22 49 100 33 1 98 60 -88 -25 -100 -31 -64 -70 -2 -88 -50 -59 51 94 73 96 51 77 40 7 92 71 -62 -37 -4 -49 -4 -84 -49 -49 -32 -80
Файл P:
33 26 57 22 49 -88 -25 -100 -31 -64 100 33 1 98 60 -70 -2 -88 -50 -59 51 94 73 96 51 -62 -37 -4 -49 -4 77 40 7 92 71 -84 -49 -49 -32 -80
33 26 57 22 49 100 33 1 98 60 51 94 73 96 51 77 40 7 92 71 -88 -25 -100 -31 -64 -70 -2 -88 -50 -59 -62 -37 -4 -49 -4 -84 -49 -49 -32 -80

Process finished with exit code 0
```

Результат работы тестов:

```
1 tests, 1 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
Test suite finished: 0.0270137 seconds

Process finished with exit code 0
```

Результат работы rubocop:

```
PS D:\education\3 semester\ipl\lab\lab7\lab7_1> rubocop client.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
```

```
PS D:\education\3 semester\ipl\lab\lab7\lab7_1> rubocop main.rb
Inspecting 1 file
C

Offenses:

main.rb:3:1: C: Metrics/AbcSize: Assignment Branch Condition size for fill_file_f is too high. [<6, 17, 2> 18.14/17]
def fill_file_f(count) ...
^^^^^^^^^^^^^^^^^^^^^^
main.rb:3:1: C: Metrics/MethodLength: Method has too many lines. [15/10]
def fill_file_f(count) ...
^^^^^^^^^^^^^^^^^^^^^^
main.rb:21:1: C: Metrics/AbcSize: Assignment Branch Condition size for fill_file_p is too high. [<11, 29, 3> 31.16/17]
def fill_file_p(number) ...
^^^^^^^^^^^^^^^^^^^^^^^
main.rb:21:1: C: Metrics/MethodLength: Method has too many lines. [27/10]
def fill_file_p(number) ...
^^^^^^^^^^^^^^^^^^^^^^^

1 file inspected, 4 offenses detected
```

```
PS D:\education\3 semester\ipl\lab\lab7\lab7_1> rubocop test.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
```

Часть 2.

Задание:

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 — Прямоугольник, характеризующийся размерами. Объект умеет выводить на экран значения своих полей и отвечать на запрос о типе: квадрат или нет.

Объект — Прямоугольный параллелепипед, характеризующийся размерами. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их содержимое и определять тип параллелепипеда.

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

Код программы:

Файл пользователя (client.rb):

```
# frozen_string_literal: true

require './main'


puts('Введите первую сторону прямоугольника')
a = gets.chomp.to_f
puts('Введите вторую сторону прямоугольника')
b = gets.chomp.to_f
pr = Primoug.new(a, b)
puts('Квадрат или нет: ')
puts(pr.square)
puts('Введите третью сторону параллелепипеда')
c = gets.chomp.to_f
par = Paral.new(a, b, c)
puts('Куб или нет: ')
puts (par.cube)
```

Файл основной программы (main.rb):

```
# frozen_string_literal: true

# class Parent
class Primoug
  attr_accessor :a, :b

  def initialize(first, second)
    @a = first
    @b = second
  end

  def print
    puts("Сторона 1: #{@a}")
    puts("Сторона 2: #{@b}")
  end

  def square
    if @a == @b
      'да'
    else
      'нет'
    end
  end
end

# class Child
class Paral < Primoug
  attr_accessor :h
```

```
  def initialize(first, second, third)
    super(first, second)
    @h = third
  end

  def print
    Primoug.instance_method(:print).bind(self).call
    puts("Сторона 3: #{@h}")
  end

  def cube
    if @a == @b && @b == @h
      'Куб'
    else 'Прямоугольный параллелепипед'
    end
  end
end
```

Файл тестов (test.rb):

```
# frozen_string_literal: true

require 'test/unit'
require './main'

# test
class TestMinMax < Test::Unit::TestCase
  def test1
    pr = Primoug.new(2.5, 10)
    assert_equal('нет', pr.square)
    par = Paral.new(3, 3, 3)
    assert_equal('Куб', par.cube)
```

```
    assert_equal par.class.superclass, Primoug
  end
end
```

Результат работы программы:

```
Введите первую сторону прямоугольника
2
Введите вторую сторону прямоугольника
5
Квадрат или нет:
нет
Введите третью сторону параллелепипеда
3
Куб или нет:
Прямоугольный параллелепипед

Process finished with exit code 0
```

```
Введите первую сторону прямоугольника
4
Введите вторую сторону прямоугольника
4
Квадрат или нет:
да
Введите третью сторону параллелепипеда
4
Куб или нет:
Куб

Process finished with exit code 0
```

Результат работы тестов:

```
1 tests, 3 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
Test suite finished: 0.0106577 seconds

Process finished with exit code 0
```

Результат работы rubocop:

```
PS D:\education\3 semester\ipl\lab\lab7\lab7_2> rubocop client.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
PS D:\education\3 semester\ipl\lab\lab7\lab7_2> rubocop main.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
PS D:\education\3 semester\ipl\lab\lab7\lab7_2> rubocop test.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
```

Вывод: мы научились работать с файлами на языке Ruby, а также создавать классы, объекты класса и реализовывать иерархию классов путём наследования