



**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**

**Название:** Работа с файлами. ООП

**Дисциплина:** Языки Интернет-программирования

Студент

ИУ6-35Б

(Группа)

27.10.2023

(Подпись, дата)

В. И. Мамыкин

(И.О. Фамилия)

Преподаватель

Е.Ю. Гаврилова

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

## Задание:

### Часть 1

Дан текстовый файл **F**. Записать в перевернутом виде строки файла **F** в файл **G**. Порядок строк в файле **G** должен совпадать с порядком исходных строк в файле **F**.

Автоматический тест программы обязательно должен проверять работу с файлами.

### 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 символ, умеющий выводить на экран свое значение и отвечать на запрос о значении своего символьного поля и числовом коде символа.

Объект запись, состоящая из полей: символ и целое число — количество повторений символа при выводе. Объект умеет выводить на экран строку из указанного количества символов, возвращать по запросу содержимое своих полей и код символа.

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

**Цель:** реализовать консольные приложения на Ruby и написать для них тесты.

Научиться работать с файлами и классами.

#### Часть 1:

##### **1.rb**

```
# frozen_string_literal: true
```

```
require_relative 'functions'
```

```
fill_the_files
```

## **functions.rb**

```
require 'securerandom'
```

```
def generate_random_string(length)
```

```
  SecureRandom.alphanumeric(length)
```

```
end
```

```
def fill_the_files
```

```
  f_file = File.open('F.txt', 'w')
```

```
  rand(1..10).times do
```

```
    random_string = generate_random_string(rand(5..10))
```

```
    f_file.puts random_string
```

```
  end
```

```
  f_file.close
```

```
  f_file = File.open('F.txt', 'r')
```

```
  g_file = File.open('G.txt', 'w')
```

```
  f_file.each_line do |line|
```

```
    reversed_line = line.chomp.reverse
```

```
    g_file.puts reversed_line
```

```
  end
```

```
  f_file.close
```

```
  g_file.close
```

```
  true
```

end

def proverka

f\_file = File.open('F.txt', 'r')

g\_file = File.open('G.txt', 'r')

f\_file.each\_line do |line|

reserved\_line = line.chomp.reverse

line\_from\_g\_file = g\_file.gets.chomp

return false if reserved\_line != line\_from\_g\_file

end

true

end

**tests.rb**

# frozen\_string\_literal: true

require 'minitest/autorun'

require\_relative '1'

class TestFunc < Minitest::Test

def test\_func1

assert\_equal(true, File.exist?('F.txt'))

assert\_equal(proverka, fill\_the\_files)

end

def test\_func2

  assert\_equal(true, File.exist?('F.txt'))

  assert\_equal(proverka, fill\_the\_files)

end

def test\_func3

  assert\_equal(true, File.exist?('F.txt'))

  assert\_equal(proverka, fill\_the\_files)

end

end

Часть 2:

**1.rb**

# frozen\_string\_literal: true

require\_relative 'functions'

symbol\_a = Symbols.new('A')

record\_b = Record.new('B', 4)

# ВЫЗОВ МЕТОДОВ

symbol\_a.display\_value

puts symbol\_a.symbol\_field

```
puts symbol_a.symbol_code  
  
record_b.display_value  
  
puts record_b.symbol_field  
  
puts record_b.symbol_code  
  
puts record_b.repeat_count  
  
record_b.display_string
```

### **functions.rb**

```
class Symbols  
  
  def initialize(value)  
  
    @value = value  
  
  end  
  
  
  def display_value  
  
    puts @value  
  
  end  
  
  
  def symbol_field  
  
    @value  
  
  end  
  
  
  def symbol_code  
  
    @value.ord  
  
  end  
  
end
```

```

end

end

class Record < Symbols

  def initialize(value, repeat_count)

    super(value)

    @repeat_count = repeat_count

  end

  def display_string

    puts @value * @repeat_count

  end

  attr_reader :repeat_count

end

def proverka

  symbol_a = Symbols.new('A')

  record_b = Record.new('B', 4)

  symbol_a.is_a?(Symbols)      &&      record_b.is_a?(Symbols)      &&
  record_b.is_a?(Record)

end

tests.rb

# frozen_string_literal: true

```

```
require 'minitest/autorun'

require_relative '1'

class TestFunc < Minitest::Test

  def test_func1

    assert_equal(proverka, true)

  end

end
```

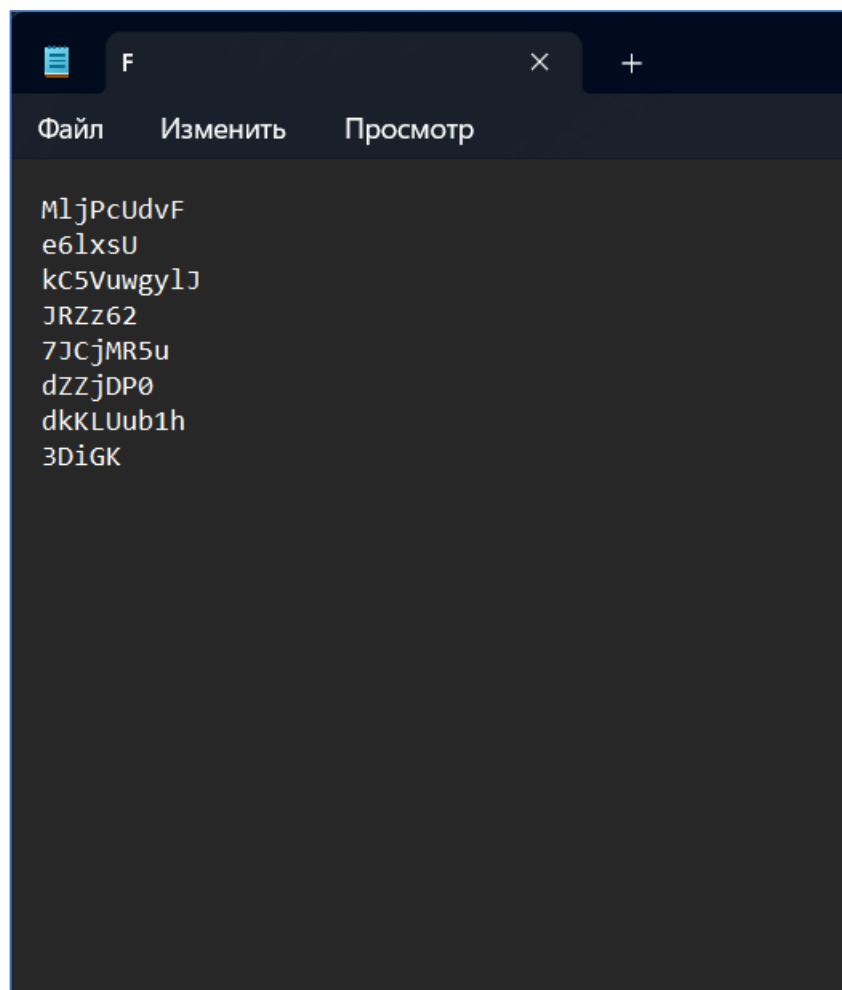


Рисунок 1 – Файл F



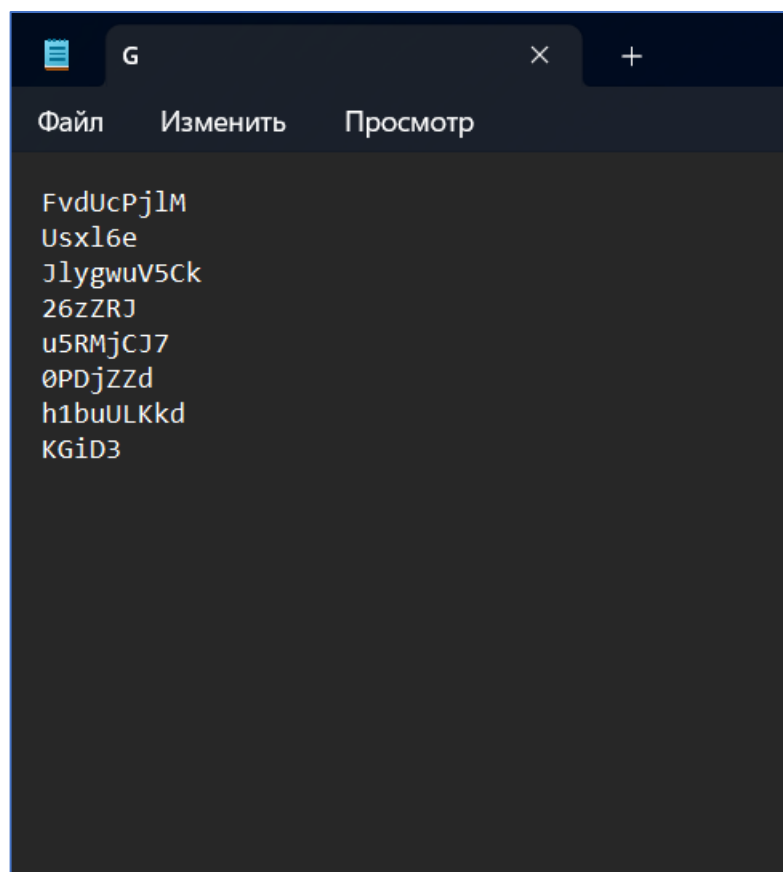


Рисунок 2 – Файл G

```
C:\Ruby32-x64\bin\ruby.exe D:/Documents/GitHub/BMSTU/3semestr/YAIP/lab7/2/1.rb
A
A
65
B
B
66
4
BBBB

Process finished with exit code 0
```

Рисунок 3 – Результат выполнения 2 программы

```
# Running:

...

Finished in 0.063493s, 47.2497 runs/s, 94.4994 assertions/s.
3 runs, 6 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 4 – Результат выполнения тестов 1 части

```
# Running:

.

Finished in 0.014186s, 70.4895 runs/s, 70.4895 assertions/s.
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 5 – Результат выполнения тестов 2 части

```
Inspecting 3 files
...

3 files inspected, no offenses detected
```

Рисунок 6 – Результат работы rubocop (*rubocop --config rubocop.yml*) (аналогично для всех частей задания)

```
Inspecting 3 file(s):
...

0 total warnings
```

Рисунок 7 – Результат работы reek (аналогично для всех частей задания)

**Вывод:** были сделаны консольные приложения на Ruby и написаны для них тесты. Было изучено взаимодействие с файлами и классами.