



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: Ruby

Дисциплина: Языки интернет программирования

Студент

ИУ6-53Б

(Группа)

(Подпись, дата)

Т.Р. Сапарбаев

(И.О. Фамилия)

Преподаватель

Р.С. Самарев

(Подпись, дата)

(И.О. Фамилия)

Москва, 2020

## Часть 1

Дан символьный файл **F**. Найти самое длинное слово среди слов, первая буква которых есть «а». Если слов с наибольшей длиной несколько, то найти последнее из них. Если таких слов нет вообще, то сообщить об этом.

Автоматический тест программы обязательно должен проверять работу с файлами.

### 1\_1.rb:

```
# frozen_string_literal: true

# Main Class
class Main
  def self.max_a
    words = File.open('./F.txt').read.split.select { |str| str.start_with?('a') }
    words.empty? ? 'No words found' : words.select { |str| str.length == words.max_by(&:length).length }.last
  end
end
```

### 1\_2.rb:

```
# frozen_string_literal: true

require_relative './1_1'
puts Main.max_a
```

### 1\_3.rb:

```
# frozen_string_literal: true

require_relative './1_1'
require 'test/unit'

# Class which test our program
class MyTest < Test::Unit::TestCase
  def test1_file
    if File.file?('./F.txt')
      assert_equal(Main.max_a, FuncTest.cal)
    else
      assert(false)
    end
  end
end

# Class with test functions
class FuncTest
```

```

def self.call
  words = File.open('./F.txt').read.split.select { |str| str.start_with?('a') }
  words.empty? ? 'No words found' : words.select { |str| str.length == words.max_by(&
:length).length }.last
end
end

```

Результат выполнения:

```

saparbi@saparbi-TM1703:~/Desktop/YAIP/LR7$ ruby 1_2.rb
alalalalalala

```

```

saparbi@saparbi-TM1703:~/Desktop/YAIP/LR7$ ruby 1_3.rb
Loaded suite 1_3
Started
.
Finished in 0.0005211 seconds.

-----
1 tests, 1 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-----
1919.02 tests/s, 1919.02 assertions/s

```

## 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 — больной. Параметры: фамилия, возраст. Методы: инициализирующий, вывода на экран фамилии и возраста.

Объект — больной. Параметры: фамилия, возраст, год последней диспансеризации. Методы: инициализирующий, определения года следующей диспансеризации, исходя из того, что диспансеризация должна проходиться каждые 3 года.

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

## 2\_1.rb:

```
# frozen_string_literal: true

# Person Class
class Person
  attr_reader :name, :age

  def initialize(name, age)
    @name = name
    @age = age
  end

  def print_person
    puts "Last name: #{name}"
    puts "Age: #{age}"
  end
end

# Sick Person Class
class Sick < Person
  attr_reader :last_visit

  def initialize(name, age, last_visit)
    super(name, age)
    @last_visit = last_visit
  end

  def next_visit
    puts "Year of the next medical examination: #{@last_visit + 3}"
  end
end
```

## 2\_2.rb:

```
# frozen_string_literal: true

require_relative './2_1'

puts 'Enter last name: '
l_name = gets
puts 'Enter age'
age = gets.to_i
p = Person.new(l_name, age)
p.print_person

s = Sick.new(l_name, age, rand(2018..2020))
s.print_person
s.next_visit
```

## 2\_3.rb:

```
# frozen_string_literal: true

require 'test/unit'
require_relative './2_1'

# Class for testing
class MyTest < Test::Unit::TestCase
  def test1_superclass
    person = Person.new('LastName', 18)
    sick = Sick.new('LastName', 18, 2020)
    assert(person.instance_of?(sick.class.superclass))
  end
end
```

## F.txt:

dmasdkqwe qwe  
s  
ada  
d  
SSFASD Aasdasdadada  
asdadasdasd  
asdase

## Результат выполнения:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR7$ ruby 2_2.rb
Enter last name:
Tamerlan
Enter age
19
Last name: Tamerlan
Age: 19
Last name: Tamerlan
Age: 19
Year of the next medical examination: 2021
```

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR7$ ruby 2_3.rb
Loaded suite 2_3
Started
.
Finished in 0.000463393 seconds.
-----
1 tests, 1 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
-----
2158.00 tests/s, 2158.00 assertions/s
```

## Rubocop:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR7$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.1.
warning: please see https://github.com/whitequark/parser#compatibility-with-ruby-mri.
Inspecting 6 files
.....

6 files inspected, no offenses detected
```

## Reek:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR7$ reek
Inspecting 6 file(s):
.....

0 total warnings
```

## Вывод:

В ходе работы были разработаны программы, соответствующие условиям задачи. На приведенных данных программы работают корректно. Необходимые тесты пройдены.