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 _____ И.А. Нуруллаев
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ _____
(Подпись, дата) (И.О. Фамилия)

Москва, 2022

Часть 1

Организовать программным способом символьные файлы F и G. Переписать в файл H все начальные совпадающие компоненты файлов F и G. При возникновении непредвиденных ситуаций выдать соответствующие сообщения.

Автоматический тест программы обязательно должен проверять работу с файлами.

Решение:

user.rb

```ruby
require_relative 'main'

create_file('F.txt', 'i love cats')
create_file('G.txt', 'i love kittens')
write_matches_to_file('F.txt', 'G.txt', 'H.txt')

```

test.rb

```ruby
require 'minitest/autorun'
require_relative 'main'

# File tester
class Test < Minitest::Test
  def initialize(name)
    super name
    @first_file_name = 'F.txt'
    @second_file_name = 'G.txt'
    @output_file = 'H.txt'
  end

  def test_create
    create_file(@first_file_name, 'test string')
    assert_path_exists @first_file_name, 'Не получилось создать файл'
  end

  def test_first_variant
    create_file(@first_file_name, 'test string1')
    create_file(@second_file_name, 'test string2')
    write_matches_to_file(@first_file_name, @second_file_name, @output_file)

    assert_path_exists @output_file, 'Файл не был создан'
    result = File.read(@output_file)
    assert_equal 'test string', result, 'Программа отработала неверно'
  end

  def test_second_variant
    create_file(@first_file_name, 'test string')
    create_file(@second_file_name, 'string test')
    write_matches_to_file(@first_file_name, @second_file_name, @output_file)

    assert File.file?(@output_file), 'Файл не должно быть так как нет совпадающих начальных компонентов'
  end
end

```

main.rb

```ruby
def create_file(file_name, string)
  File.write(file_name, string)
end

def write_matches_to_file(first_file_name, second_file_name, output_file_name)
  first_enum = File.read(first_file_name).each_char
  second_enum = File.read(second_file_name).each_char
  matched = first_enum.zip(second_enum).take_while { |first_c, second_c| first_c == second_c }

  if matched.length.zero?
    puts('В файлах нет начальных совпадающих компонентов')
  else
    create_file(output_file_name, matched.transpose[0].join)
  end
end

```

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 55441

# Running:

.В файлах нет начальных совпадающих компонентов
..

Finished in 0.017004s, 176.4280 runs/s, 235.2374 assertions/s.
3 runs, 4 assertions, 0 failures, 0 errors, 0 skips
```

Файлы после отработки программы *user.rb*

F.txt

```
i love cats
```

G.txt

```
i love kittens
```

H.txt

```
i love
```

Часть 2

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработан ных 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.

Решение:

main.rb

```ruby
# Class patient
class Patient
  attr_reader :surname
  attr_accessor :age

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

  def print
    puts "Больной с фамилией - #{@surname}, возраст - #{age}"
  end
end

# Class patient with examination
class ExaminationPatient < Patient
  attr_accessor :last_examination_year

  def initialize(surname, age, last_examination_year)
    super surname, age
    @last_examination_year = last_examination_year
  end

  def next_examination_year
    @last_examination_year + 3
  end

  def print
    super
    puts "Год последней диспансеризации - #{@last_examination_year}, год следующей - #{next_examination_year}"
  end
end

```

user.rb

```ruby
require_relative 'main'

patient = ExaminationPatient.new 'Нуруллаев', 19, 2021
patient.print

```

test.rb

```ruby
require 'minitest/autorun'
require_relative 'main'

# Class tester
class Test < Minitest::Test
  def test_hierarchy
    examination_patient = ExaminationPatient.new 'Нуруллаев', 19, 2021
    assert_kind_of Patient, examination_patient
  end
end

```

Результат выполнения программ:

```
PS C:\Users\might\Desktop\WPL_bmstu\Lab7\Part 2> ruby .\user.rb
Больной с фамилией - Нуруллаев, возраст - 19
Год последней диспансеризации - 2021, год следующей - 2024
PS C:\Users\might\Desktop\WPL_bmstu\Lab7\Part 2> ruby .\test.rb
Run options: --seed 36447

# Running:

.

Finished in 0.011467s, 87.2068 runs/s, 87.2068 assertions/s.
1 runs, 1 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