



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

Дисциплина: Языки интернет-программирования

Студент

ИУ6-31Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Лазарев Е.С.

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Малахов Д.В.

(И.О. Фамилия)

Москва, 2022

Цель работы: изучить принцип работы с файлами в языке Ruby

## Часть 1

Организовать программным способом символьный файл **F**. Получить файл **G**, образованный из файла **F** заменой всех его прописных (больших) букв одноименными строчными (малыми).

Автоматический тест программы обязательно должен проверять работу с файлами.

Основная программа:

```
def laba71 st
  File.open("F.txt", "w") do |f|
    f.write(st)
  end
  array = File.readlines 'F.txt'
  str = array[0]
  str = str.downcase
  File.open("G.txt", "w") do |f|
    f.write(str)
  end
end
```

Программа для взаимодействия с пользователем через консоль:

```
require_relative 'main71.rb'
st = gets.to_s
laba71 st
```

Программа для автоматического тестирования на основе RSpec:

```
require_relative 'main71.rb'
RSpec.describe "Supermain" do
  it 'should check that file F is opened' do
    laba71("Hi, Bro!")
    str_1 = File.new('F.txt', 'r').read
    expect("Hi, Bro!").to eq str_1
  end
  it 'should check that file G is opened' do
    str_2 = File.new('G.txt', 'r').read
    expect("hi, bro!").to eq str_2
  end
end
```

```
• yulitylight@yulitylight-HLYL-WXX9:~/Документы/Лабы по ЯИП/Лаба 7$ rspec test71.rb
..

Finished in 0.00201 seconds (files took 0.06037 seconds to load)
2 examples, 0 failures
```

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 — дом, хранящий и умеющий выводить на экран год постройки, общий метраж и стоимость.

Объект — дом, хранящий и умеющий выводить на экран год постройки, общий метраж, стоимость и цену одного метра.

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

Основная программа:

```
class HouseA
  attr_accessor :year, :square, :price
  def initialize(year, square, price)
    @year = year
    @square = square
    @price = price
  end
end
```

```
class HouseB < HouseA
  attr_accessor :oneprice
  def initialize(year, square, price)
    super(year, square, price)
    @oneprice = price / square
  end
end
```

Программа для взаимодействия с пользователем через консоль:

```
require_relative 'main72.rb'
a = HouseA.new(2006, 100, 2200000)
b = HouseB.new(1992, 230, 5000000)
puts b.oneprice
```

Программа для автоматического тестирования на основе RSpec:

```
require_relative 'main72.rb'
RSpec.describe do
  it 'should return true' do
    puts HouseB.ancestors.include? HouseA
    expect(HouseB.ancestors.include? HouseA).to eq true
  end
end
```

```
yulitylight@yulitylight-HLYL-WXX9:~/Документы/Лабы по ЯИП/Лаба 7$ rspec test72.rb
true
.
Finished in 0.00182 seconds (files took 0.06047 seconds to load)
1 example, 0 failures
```

Вывод: я изучил принцип работы с файлами в языке Ruby