



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**

**Дисциплина: языки интернет-программирования**

**Вариант №15**

Студент гр. ИУ6-33Б

\_\_\_\_\_

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

## Цель

Изучить работу с файлами в руби и систему объектно-ориентированные аспекты языка.

## Задание

Задание выполняется как консольное приложение Ruby. Результат следует предоставлять в виде трех отдельных файлов:

- основная программа;
- программа для взаимодействия с пользователем;
- программа для тестирования на основе

```
MiniTest::Unit::TestCase
```

Каждое задание сдавать в виде:

- отдельного архива, содержащего указанные файлы;
- pdf-файл с отчётом.

Отчет должен содержать:

1. заполненный титульный лист
2. текст задания
3. тексты программ с подписанными именами файлов
4. результаты выполнения
5. результаты проверки анализаторами rubocop и reek

## ЛР 7

### Часть 1

Дан текстовый файл **F**. Переписать строки файла **F** в файл **G**, вставляя в начало каждой строки по одному пробелу. Порядок строк должен быть сохранен.

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 — целое число. Объект умеет выводить на экран значение своего поля и отвечать на запрос о его значении и количестве цифр в числе.

Объект, включающий поля: целое число и вещественное число. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их содержимое и количество цифр целого числа.

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

Рисунок 1 - задание для 15 варианта

### Выполнение

Main.rb

```
# frozen_string_literal: true

require_relative 'task1'
require_relative 'task2'

def nav_to_task
  puts 'select task number'
  case gets.to_i
  when 1
    io_task1
  when 2
    io_task2
  else
    puts 'unknown task number'
  end
end
```

```

@vzalygin →/workspaces/ipl.labs/lab7 (master) $ rubocop
Inspecting 4 files
..CC

Offenses:

task2.rb:3:1: C: Style/Documentation: Missing top-level documentation comment for class Int.
class Int
^^^^^^^^

task2.rb:21:1: C: Style/Documentation: Missing top-level documentation comment for class IntAndFloat.
class IntAndFloat < Int
^^^^^^^^^^^^^^^^^^^^

tests.rb:7:1: C: Style/Documentation: Missing top-level documentation comment for class Task1Tests.
class Task1Tests < Minitest::Test
^^^^^^^^^^^^^^^^^^^^

tests.rb:8:3: C: Style/ClassVars: Replace class var @@from_path with a class instance var.
  @@from_path = './f'
  ^^^^^^^^^^^

tests.rb:9:3: C: Style/ClassVars: Replace class var @@to_path with a class instance var.
  @@to_path = './g'
  ^^^^^^^^^

tests.rb:55:1: C: Style/Documentation: Missing top-level documentation comment for class Task2Tests.
class Task2Tests < Minitest::Test
^^^^^^^^^^^^^^^^^^^^

4 files inspected, 6 offenses detected

```

Рисунок 2 - прогон рубокопа

```

@vzalygin →/workspaces/ipl.labs/lab7 (master) $ ruby tests.rb
Run options: --seed 37966

# Running:

.....

Finished in 0.002964s, 3036.0880 runs/s, 3036.0880 assertions/s.
9 runs, 9 assertions, 0 failures, 0 errors, 0 skips

```

Рисунок 3 - прогон тестов

## Часть 1

Task1.rb

```

# frozen_string_literal: true

def reverse(from, to)
  from = File.open(from, 'r')
  to = File.open(to, 'w')
  to.write(
    from.readlines
      .map { |line| " #{line.strip}" }
      .inject('') { |acc, x| acc + "#{x}\n" }
  )
  from.close

```

```
    to.close
end

def io_task1
  reverse('./f', './g')
  puts 'done'
end
```

Тестирование:

```
# frozen_string_literal: true

require 'minitest/autorun'
require_relative 'task1'
require_relative 'task2'

class Task1Tests < Minitest::Test
  @@from_path = './f'
  @@to_path = './g'

  def setup
    File.open(@@from_path, 'w+') do |file|
    end
    File.open(@@to_path, 'w+') do |file|
    end
  end

  def teardown
    File.delete(@@from_path)
    File.delete(@@to_path)
  end

  def insert_content(text)
    File.open(@@from_path, 'w') do |file|
      file.write(text)
    end
  end

  def result
    file = File.open(@@to_path, 'r')
    res = file.readlines.join('')
  end
```

```

    file.close
    res
end

def test_empty
  insert_content('')
  expected = ''

  reverse(@@from_path, @@to_path)

  assert_equal(expected, result)
end

def test_regular
  insert_content("ab\n cd\n")
  expected = " ab\n cd\n"

  reverse(@@from_path, @@to_path)

  assert_equal(expected, result)
end
end

```

## Часть2

Task2.rb

```

# frozen_string_literal: true

class Int
  def initialize(value)
    @i_value = value.to_i
  end

  def to_s
    @i_value.to_s
  end

  def print
    puts self
  end
end

```

```

    def number_of_digits
      @i_value.abs.to_s.length
    end
  end
end

class IntAndFloat < Int
  def initialize(ivalue, fvalue)
    super ivalue
    @f_value = fvalue
  end

  def to_s
    "#{super} #{@f_value}"
  end

  def print
    puts self
  end
end

```

Тестирование:

```

class Task2Tests < Minitest::Test
  def test_int_to_s
    obj = Int.new(32)
    expected = '32'

    assert_equal(expected, obj.to_s)
  end

  def test_int_number_of_digits
    obj = Int.new(-42)
    expected = 2

    assert_equal(expected, obj.number_of_digits)
  end

  def test_int_number_of_digits2
    obj = Int.new(434)
    expected = 3
  end
end

```

```
    assert_equal(expected, obj.number_of_digits)
end

def test_intfloat_to_s
  obj = IntAndFloat.new(42, -4.2)
  expected = '42 -4.2'

  assert_equal(expected, obj.to_s)
end

def test_intfloat_number_of_digits
  obj = IntAndFloat.new(42, -4.2)
  expected = 2

  assert_equal(expected, obj.number_of_digits)
end

def test_inheritance_intfloat
  obj = IntAndFloat.new(42, -4.3)

  assert_kind_of(Int, obj)
end

def test_inheritance_int
  obj = Int.new(42)

  assert_instance_of(Int, obj)
end
end
```

## Вывод

Изучена работа с файлами в руби и система объектно-ориентированных аспектов языка.