|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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**

**Отчет**

**по лабораторной работе 7**

**Дисциплина: языки интернет-программирования**

**Вариант №15**

Студент гр. **ИУ6-33Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** В.К. Залыгин

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** В.Д. Шульман

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**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 c 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

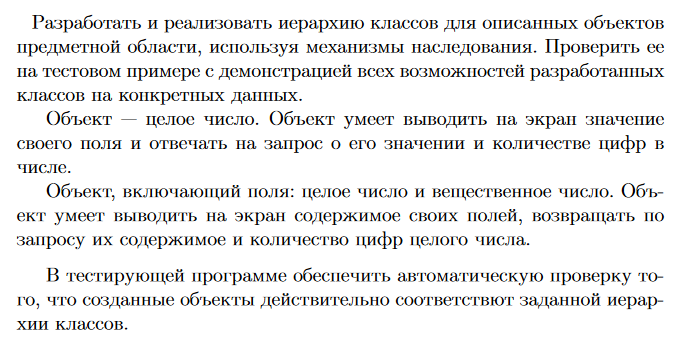
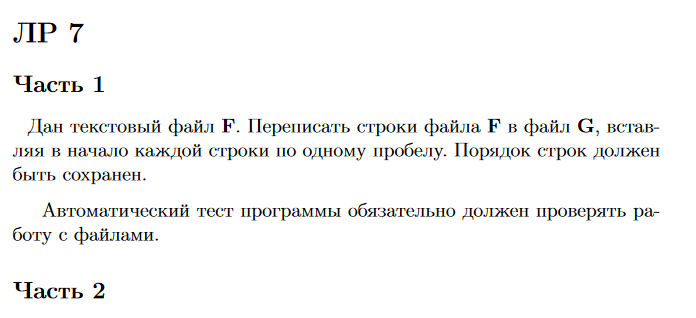


Рисунок 1 - задание для 15 варианта

**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

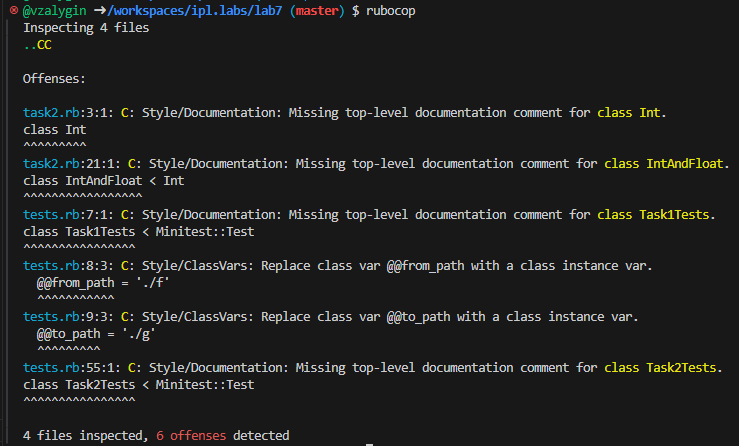


Рисунок 2 - прогон рубокопа

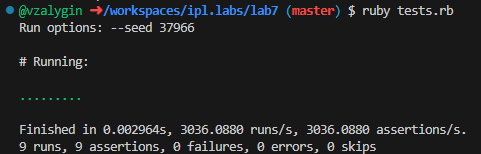


Рисунок 3 - прогон тестов

**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('')

    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\ncd\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

  def number\_of\_digits

    @i\_value.abs.to\_s.length

  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

    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.