|  |  |
| --- | --- |
| 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 **09.03.03 Прикладная информатика**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 7 |

**Дисциплина:** Языки интернет-программирования



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-35 Б |  | 27.10.2023 | И.А. Дулина |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Е.Ю. Гаврилова |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

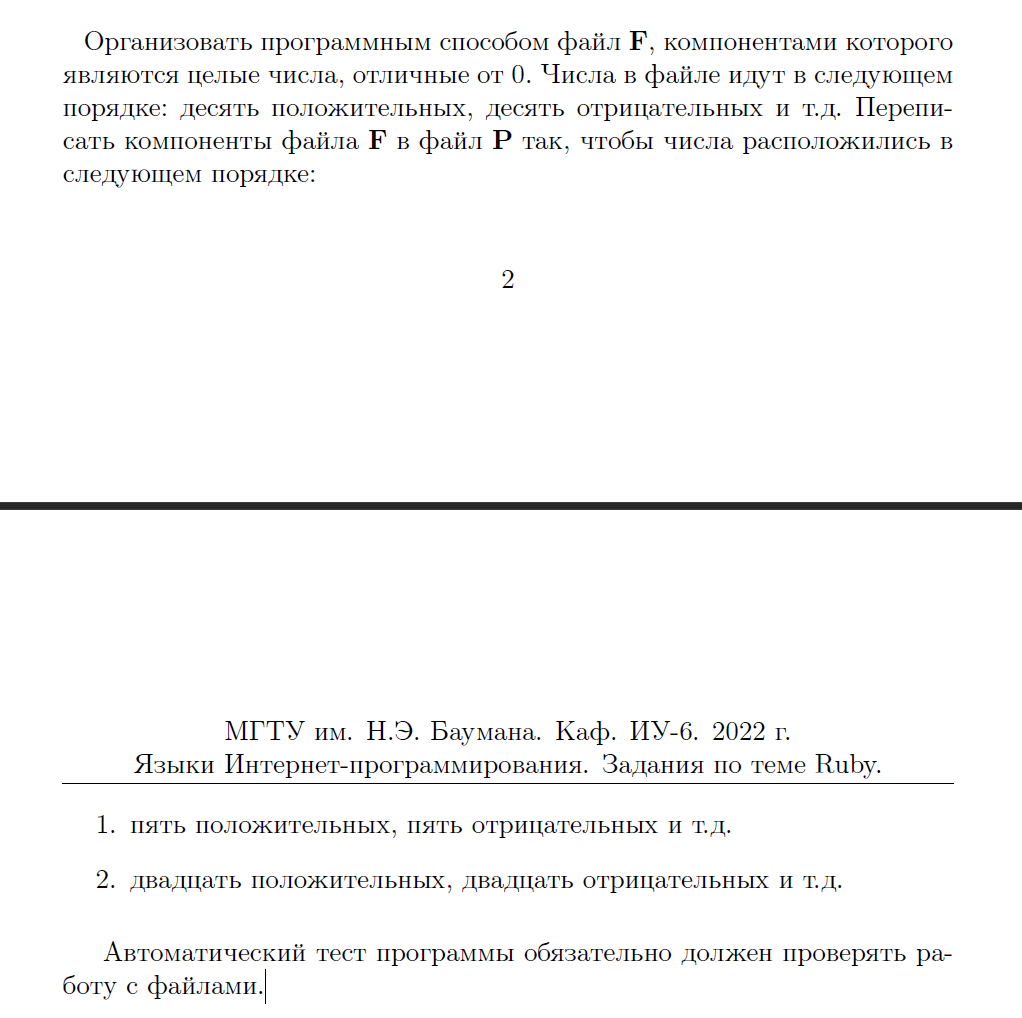
Москва, 2023

**Вариант 8**

**Цель работы:** научиться использовать язык программирования Ruby

**Часть 1**

**Задание:**



**Код программы:**

*Файл пользователя (client.rb):*

*# frozen\_string\_literal: true*require './main'  
  
puts('Введите количество чисел в файле F, кратное 40: ')  
*f* = false  
loop do  
 *count* = gets.chomp  
 if (*count*.to\_i.to\_s == *count*) && (*count*.to\_i % 40).zero?  
 *f* = true  
 *count* = *count*.to\_i  
 puts('Файл F:')  
 fill\_file\_f(*count*)  
 puts('Файл P:')  
 fill\_file\_p(5)  
 fill\_file\_p(20)  
 else  
 puts('Введено некорректное значение. Попробуйте ещё раз')  
 end  
 break if *f* == true  
end

*Файл основной программы (main.rb):*

*# frozen\_string\_literal: true*def fill\_file\_f(*count*)  
 *time* = *count* / 10  
 *file\_f* = File.open('f.txt', 'w')  
 *array* = []  
 *iter* = 0  
 *time*.times do  
 if *iter*.even?  
 10.times { *array* << Random.rand(1..100) }  
 else  
 10.times { *array* << Random.rand(-100..-1) }  
 end  
 *iter* += 1  
 end  
 *array*.length.times { |*i*| *file\_f*.write("#{*array*[*i*]} ") }  
 *file\_f*.close  
 printing(*file\_f*.path)  
end  
  
def fill\_file\_p(*number*)  
 *array* = File.readlines('f.txt')[0].to\_s.split.map(&:to\_i)  
 *array\_of\_pol* = []  
 *array\_of\_otr* = []  
 *# разделяем положительные и отрицательные элементы  
 array*.length.times do |*i*|  
 if (*array*[*i*]).positive?  
 *array\_of\_pol* << *array*[*i*]  
 else  
 *array\_of\_otr* << *array*[*i*]  
 end  
 end  
 *# записываем нужный порядок в новый массив  
 new\_array* = []  
 *iterpol* = 0  
 *iterotr* = 0  
 *array*.length.times do  
 *number*.times do  
 *new\_array* << *array\_of\_pol*[*iterpol*]  
 *iterpol* += 1  
 end  
 *number*.times do  
 *new\_array* << *array\_of\_otr*[*iterotr*]  
 *iterotr* += 1  
 end  
 end  
 *# переносим из нового массива с правильным порядком в файл  
 file\_p* = File.open('p.txt', 'w')  
 *new\_array*.length.times { |*i*| *file\_p*.write("#{*new\_array*[*i*]} ") }  
 *file\_p*.close  
 printing(*file\_p*.path)  
end  
  
def printing(*file\_name*)  
 *file* = File.open(*file\_name*, 'r')  
 puts *file*.readlines  
 *file*.close  
end

*Файл тестов (test.rb):*

*# frozen\_string\_literal: true*require 'test/unit'  
require './main'  
  
*# test*class *TestMinMax* < Test::Unit::TestCase  
 def test1  
 *file\_f* = File.open('f.txt', 'w')  
 *file\_f*.write('1 2 3 4 5 6 -1 -2 -3 -4 -5 -6')  
 *file\_f*.close  
 fill\_file\_p(3) *# по 3 пол и по 3 отриц* assert\_equal([1, 2, 3, -1, -2, -3, 4, 5, 6, -4, -5, -6],  
 File.readlines('p.txt')[0].to\_str.split.map(&:to\_i))  
 end  
end

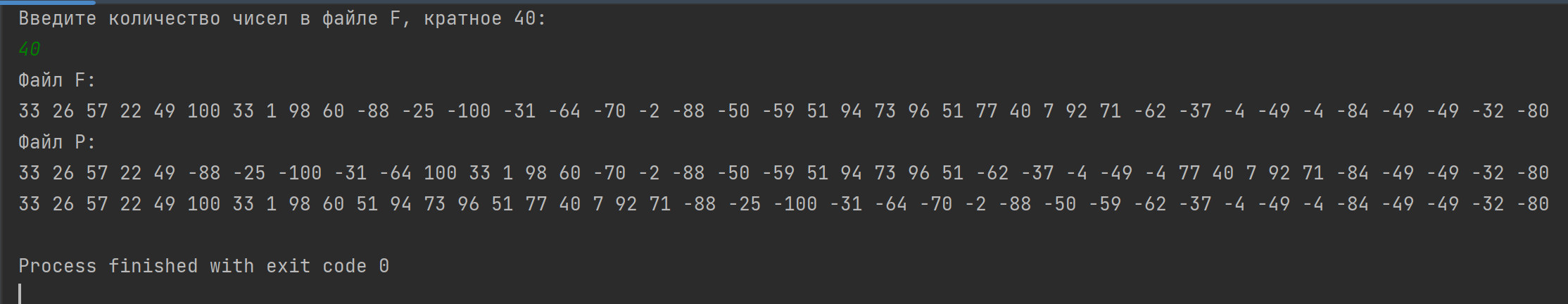


Рисунок 1.1 – результат работы программы

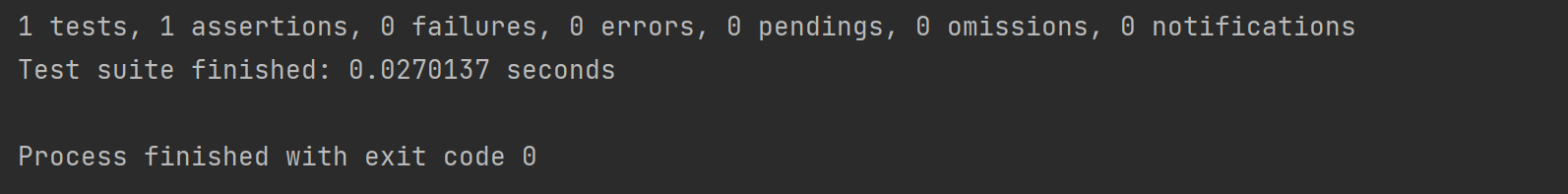
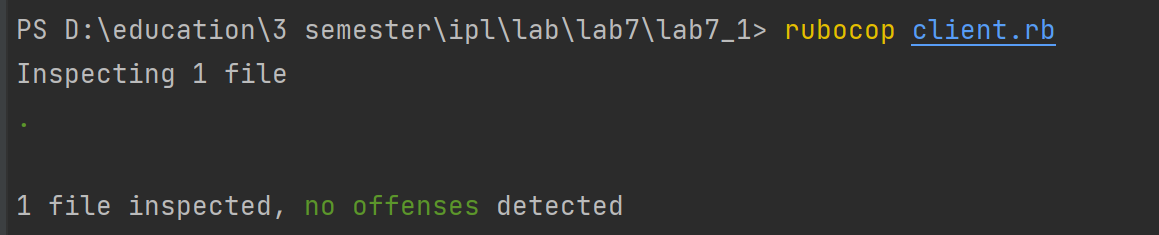


Рисунок 1.2 – результат работы тестов



1.3 – результат работы rubocop client.rb

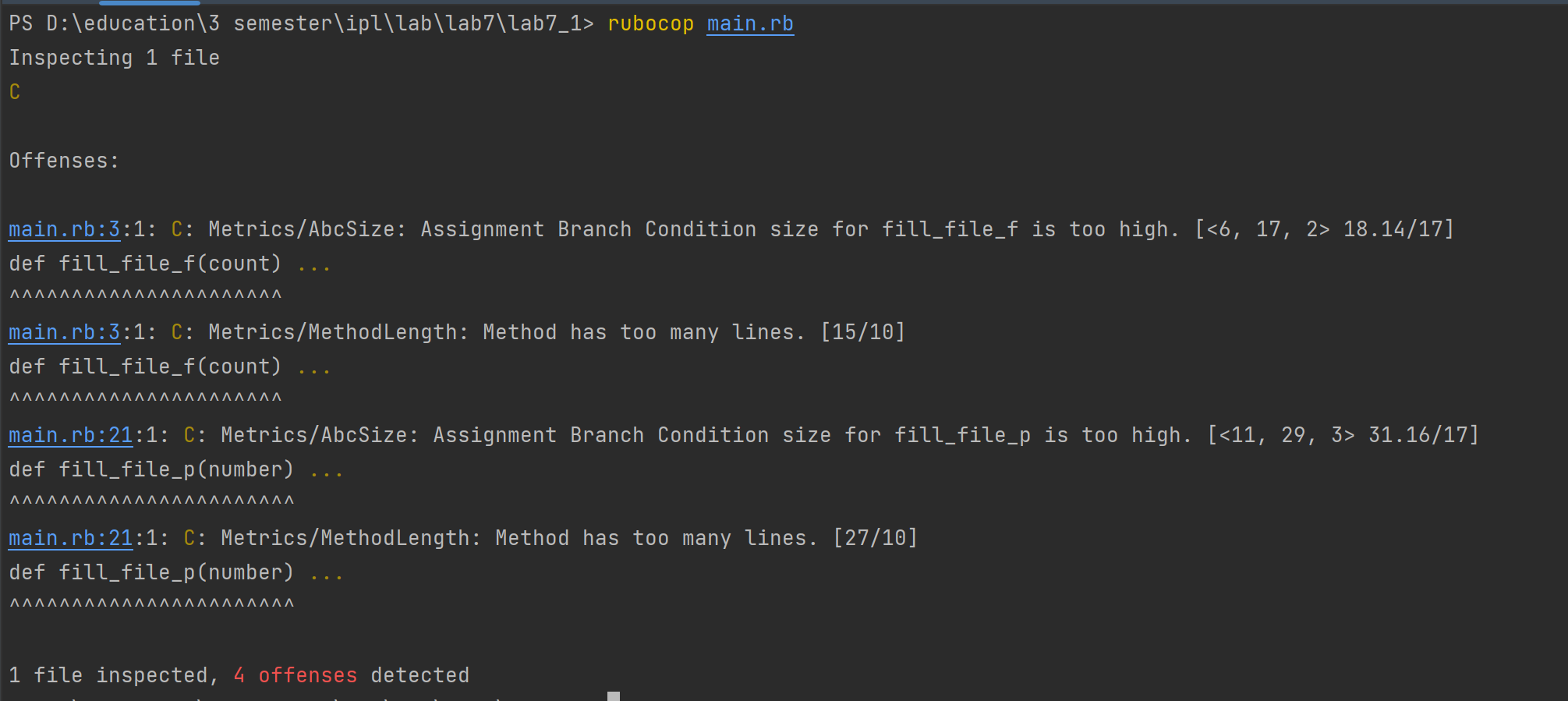


Рисунок 1.4 – результат работы main.rb

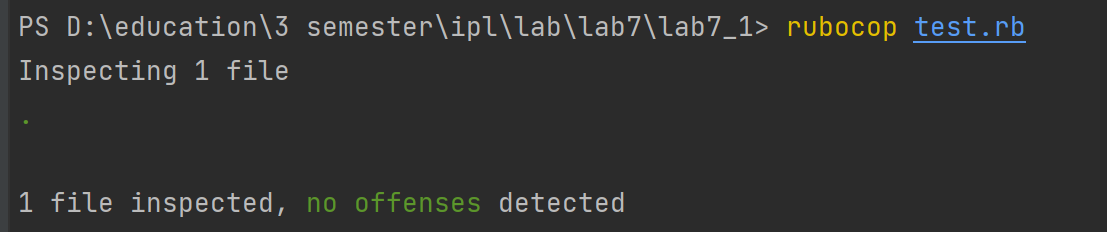
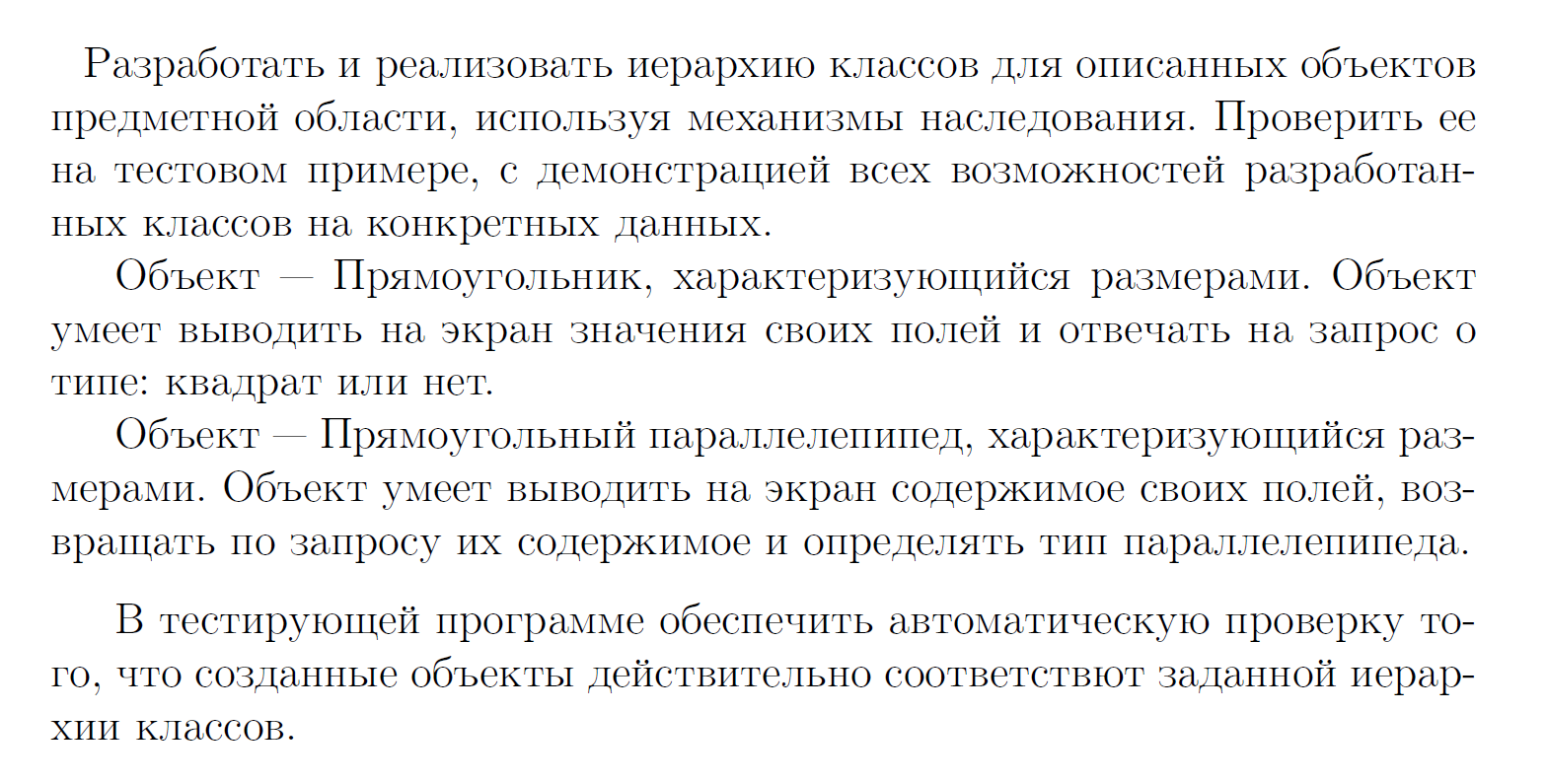


Рисунок 1.5 – результат работы test.rb

**Часть 2.**

**Задание:**

****

**Код программы:**

*Файл пользователя (client.rb):*

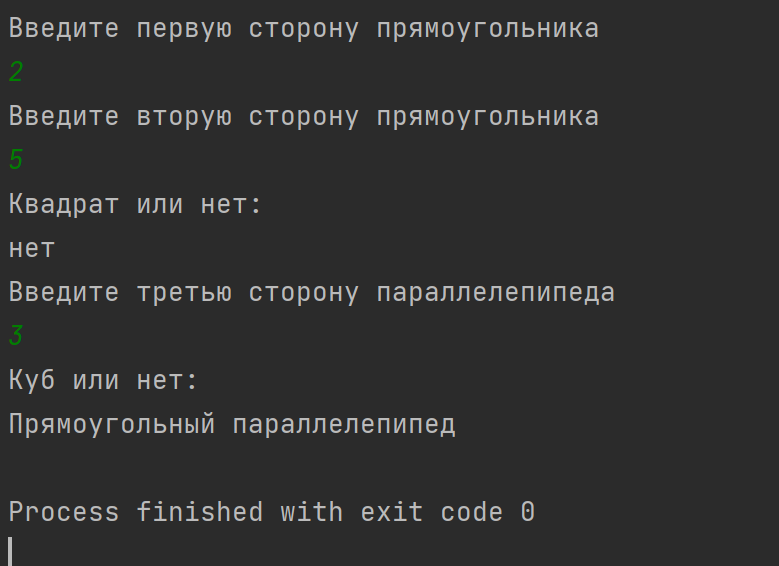
*# frozen\_string\_literal: true*require './main'  
  
puts('Введите первую сторону прямоугольника')  
*a* = gets.chomp.to\_f  
puts('Введите вторую сторону прямоугольника')  
*b* = gets.chomp.to\_f  
*pr* = Primoug.new(*a*, *b*)  
puts('Квадрат или нет: ')  
puts(*pr*.square)  
puts('Введите третью сторону параллелепипеда')  
*c* = gets.chomp.to\_f  
*par* = Paral.new(*a*, *b*, *c*)  
puts('Куб или нет: ')  
puts (*par*.cube)

*Файл основной программы (main.rb):*

*# frozen\_string\_literal: true  
  
# class Parent*class *Primoug* attr\_accessor :a, :b  
  
 def initialize(*first*, *second*)  
 @a = *first* @b = *second* end  
  
 def print  
 puts("Сторона 1: #{@a}")  
 puts("Сторона 2: #{@b}")  
 end  
  
 def square  
 if @a == @b  
 'да'  
 else  
 'нет'  
 end  
 end  
end  
  
*# class Child*class *Paral* < Primoug  
 attr\_accessor :h  
  
 def initialize(*first*, *second*, *third*)  
 super(*first*, *second*)  
 @h = *third* end  
  
 def print  
 Primoug.instance\_method(:print).bind(self).call  
 puts("Сторона 3: #{@h}")  
 end  
  
 def cube  
 if @a == @b && @b == @h  
 'Куб'  
 else 'Прямоугольный параллелепипед'  
 end  
 end  
end

*Файл тестов (test.rb):*

*# frozen\_string\_literal: true*require 'test/unit'  
require './main'  
  
*# test*class *TestMinMax* < Test::Unit::TestCase  
 def test1  
 *pr* = Primoug.new(2.5, 10)  
 assert\_equal('нет', *pr*.square)  
 *par* = Paral.new(3, 3, 3)  
 assert\_equal('Куб', *par*.cube)  
 assert\_equal par.class.superclass, Primoug  
 end  
end



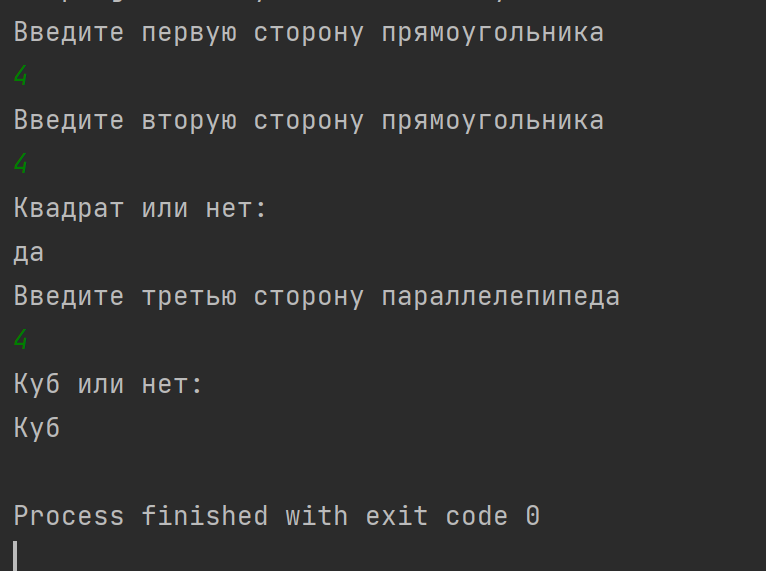


Рисунок 2.1 и 2.2 – результаты работы программы

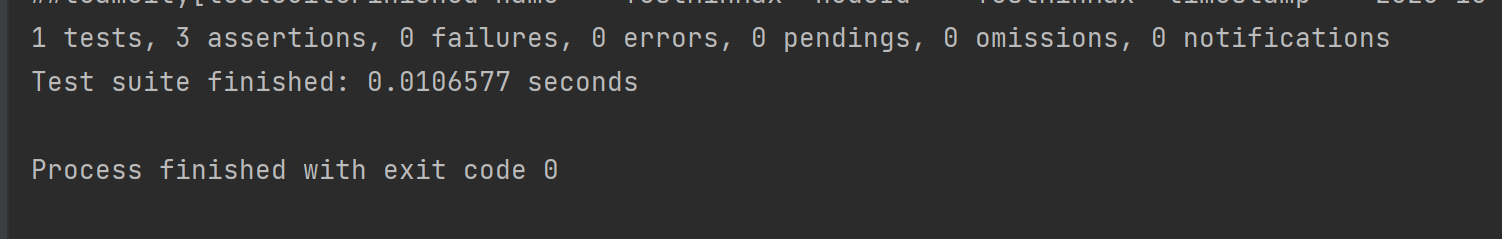
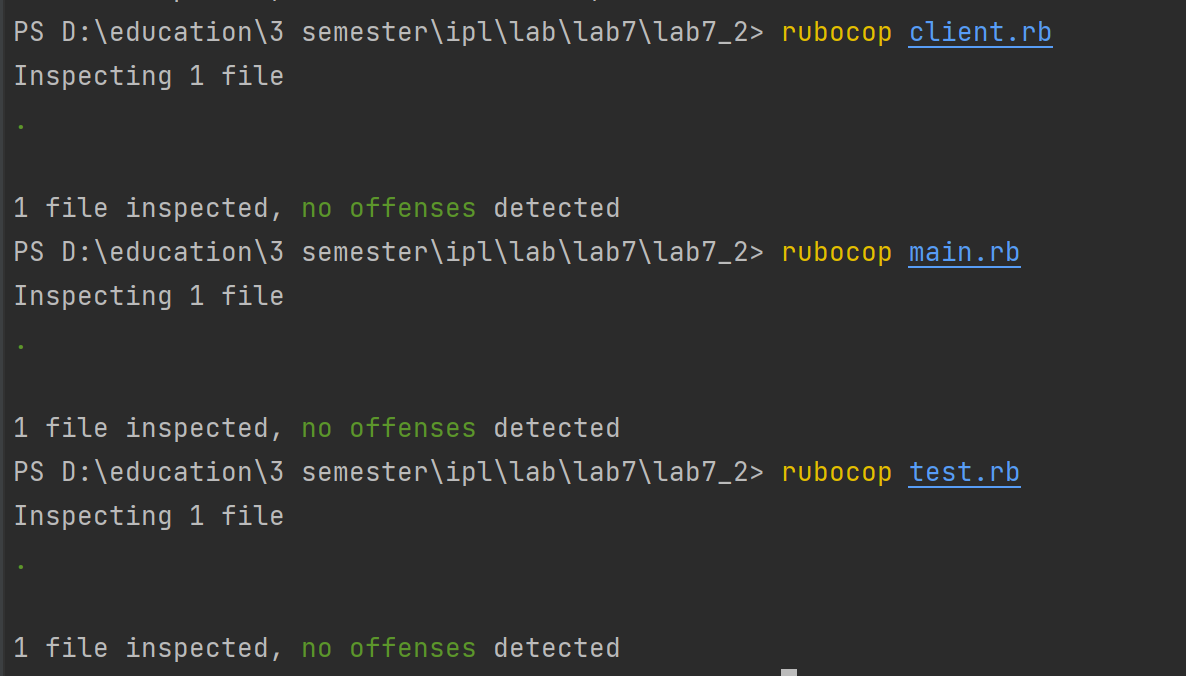


Рисунок 2.3 – результат работы тестов



2.4 – результат работы rubocop

**Вывод:** были изучены возможности работы с файлами на языке Ruby, а также создание классов, объектов класса и была сделана иерархия классов путём наследования