

Rubocop

- Leidžia pasalinti kodo klaidas, parasių rubocop, galim matyti kokios jos yra

```
Offenses:

array_task.rb:1:1: C: Style/FrozenStringLiteralComment: Missing frozen string literal comment.
# Klase
^
quadratic_task.rb:1:1: C: Style/FrozenStringLiteralComment: Missing frozen string literal comment.
quadratic_task.rb:2:1: C: Layout/LeadingEmptyLines: Unnecessary blank line at the beginning of the source.
# Klase
^^^^^^

3 files inspected, 3 offenses detected, 3 offenses auto-correctable

Tydis@DESKTOP-OHPBAT7 MINGW64 ~/OneDrive/Stalinis kompiuteris/RUBY
$
```

- Yra istaisomos visos klaidos su rubocop -a

```
$ rubocop -a
Inspecting 3 files
...

3 files inspected, no offenses detected
```

Kodo paleidimas

1 uždutis

```
1 # frozen_string_literal: true
2
3 # class
4 class ArrayTask
5   attr_accessor :arr1, :arr2
6   def test_array(array)
7     array_new = array.sort
8     array_int = 0
9
10    array_new.each_index do |index|
11      break if index == array_new.size - 1
12
13      array_int += array_new[index + 1].to_i - array_new[index].to_i - 1
14    end
15    array_int
16  end
17 end
18
19 array_object = ArrayTask.new
20 array_object.arr1 = [3, 1, 5, 6, 9]
21 array_object.arr2 = [0, 10]
22
23 puts(array_object.test_array(array_object.arr1))
24 puts(array_object.test_array(array_object.arr2))
25
26 # Mano 1 uždutis
27 # Atitinka Object-Oriented programavimo paradigma.
28 # Yra panaudojamas objektas kuris yra priskirtas klasei ArrayTask,
29 # klase priima kintamuosius, o juos priskiriu per savo array_object ir paduodu masyvus.
30 # Taip pat kintamieji turi tureti maziausiai 3 simbolius, kitaip neatitinka rubocopo reikalavimu ir
31 # jei yra naudojama 2 zodziai ar daugiau jie turi buti atskirti su apatiniu bruksniu ir zodis is mazuju raidziu.
32 # Dar yra naudojamas komentaras virs klases kuris yra butinas, nes kitaip mes klaida, kad nera dokumentacijos komentaro
33
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Inspecting 3 files

...

3 files inspected, no offenses detected

Tydis@DESKTOP-OHPBAT7 MINGW64 ~/OneDrive/Stalinis kompiuteris/RUBY

\$

Tydis@DESKTOP-OHPBAT7 MINGW64 ~/OneDrive/Stalinis kompiuteris/RUBY

\$ ruby array_task.rb

4

9

2 uzduotis

```
1  # frozen_string_literal: true
2
3  # class
4  class QuadraticTask
5    attr_accessor :aaa, :bbb, :ccc
6
7    def quadratic_equation(_aaa, _bbb, _ccc)
8      sum = suma(aaa, bbb, ccc)
9      if sum.zero?
10        if_true(aaa, bbb)
11      elsif sum.positive?
12        if_false(aaa, bbb)
13      else
14        puts 'Saknis neegzistuoja '
15      end
16    end
17
18    def suma(_aaa, _bbb, _ccc)
19      bbb * bbb - 4 * aaa * ccc
20    end
21
22    def if_true(_aaa, _bbb)
23      puts('Abidvi saknys yra lygios')
24      x1 = - bbb / (2.0 * aaa)
25      x2 = x1
26
27      puts('Pirma saknis: ' + x1.to_i.to_s)
28      puts('antra saknis: ' + x2.to_i.to_s)
29    end
30
31    def if_false(_aaa, _bbb)
32      puts('Abidvi saknys yra skirtingos')
33
34      x1 = calculation('+', aaa, bbb)
35      x2 = calculation('-', aaa, bbb)
36      puts('Pirma saknis: ' + x1.to_s)
37      puts('antra saknis: ' + x2.to_s)
38    end
39
40    def calculation(stringer, _bbb, _aaa)
```

```

41   if stringer == '-'
42     (-bbb - Math.sqrt(sum)) / (2 * aaa)
43   else
44     (-bbb + Math.sqrt(sum)) / (2 * aaa)
45   end
46 end
47 end
48
49 quadratic_object = QuadraticTask.new
50 puts 'Apskaiciuoti kvadratine lygti'
51 puts 'Iveskite pirmaja reiksme: '
52 quadratic_object.aaa = gets.to_i
53 puts 'Iveskite antra reiksme: '
54 quadratic_object.bbb = gets.to_i
55 puts 'Iveskite trecia reiksme: '
56 quadratic_object.ccc = gets.to_i
57 quadratic_object.quadratic_equation(quadratic_object.aaa, quadratic_object.bbb, quadratic_object.ccc)
58
59 # Mano 2 uzduotis
60 # principas kaip 1 uzduoties, tik cia viskas yra suskirstyta daugiau i metodus, kadangi pagal rubocop, yra
61 # neleidziama apkrauti viena metoda su daug veiksmu. Viska isskysciau i mazus metodus kur kiekvienas turi
62 # savo uzduoti.
63

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Abidvi saknys yra lygios
Pirma saknis: -1
antra saknis: -1

Ruby Unit Test

```
1 # frozen_string_literal: true
2
3 require 'test/unit'
4 require_relative 'array_task.rb'
5 # class
6 class ArrayTaskTest < Test::Unit::TestCase
7   def test_sorting
8     assert_equal(4, ArrayTask.new.test_array([3, 1, 5, 6, 9]))
9     assert_equal(9, ArrayTask.new.test_array([0, 10]))
10   end
11 end
12
13 # Nurodau kad naudosis test/unit
14 # Tada sukuriau sarysi su savo failu, kuriam noriu pritaikyti testa
15 # apsirasa klase
16 # sukuriau metoda kuris patikrina ar atitinka, ar lygus
17 # mano pirmasis parametras nurodo kokios tikiuosi, o kitas koki gaunu
18 # testas buna sekmingas jei jie abu sutampa
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

30.87 tests/s, 61.73 assertions/s

Tydis@DESKTOP-OHPBAT7 MINGW64 ~/OneDrive/Stalinis kompiuteris/RUBY

\$ ruby array_task_test.rb -n test_sorting

4

9

Loaded suite array_task_test

Started

!

Finished in 0.0020321 seconds.

1 tests, 2 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications

100% passed

492.10 tests/s, 984.20 assertions/s