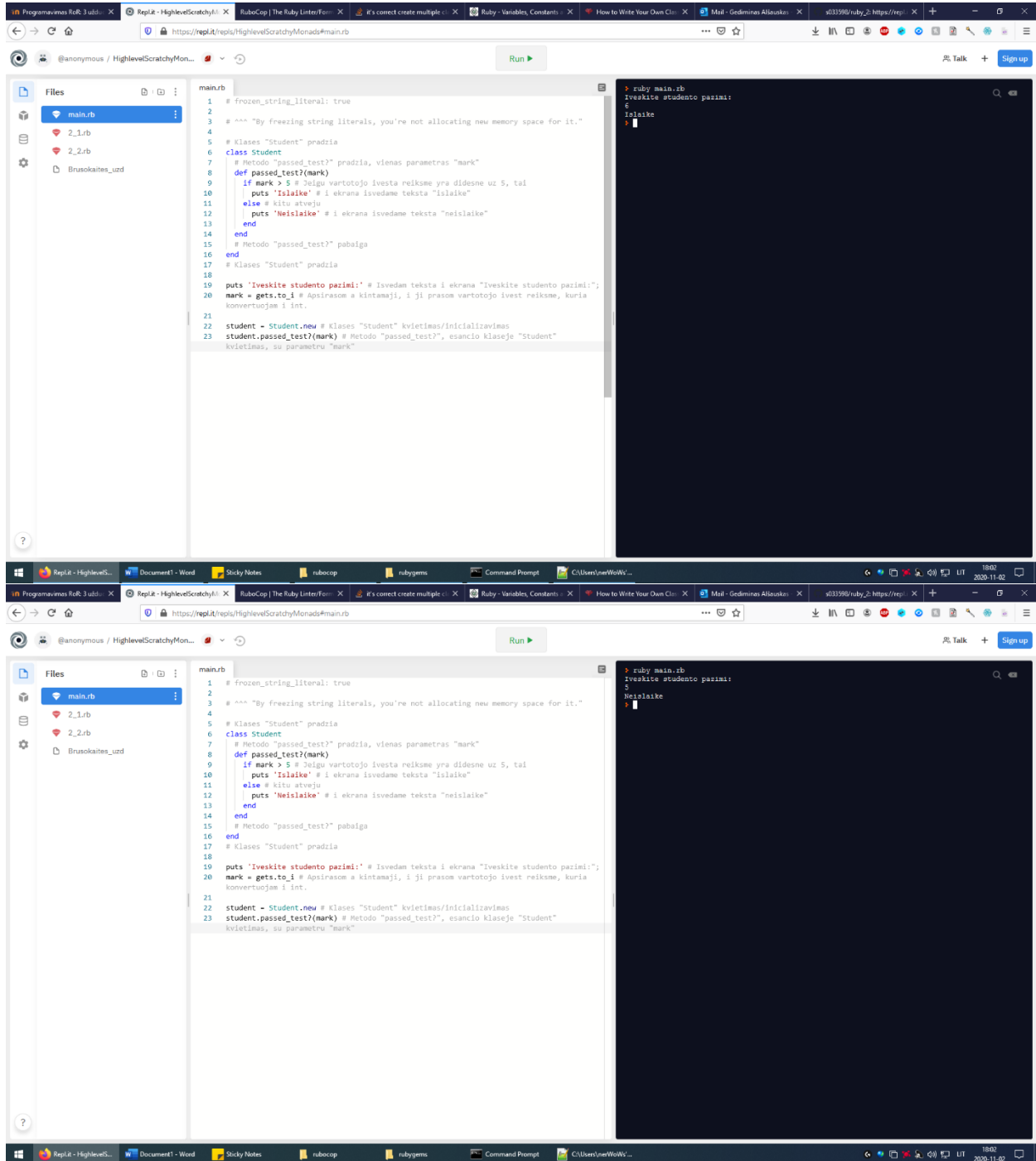


# PIRMA UŽDUOTIS



```
1 # frozen_string_literal: true
2
3 # ^^^ "By freezing string literals, you're not allocating new memory space for it."
4
5 # Klases "Student" pradzia
6 class Student
7   # Metodo "passed_test?" pradzia, vienas parametras "mark"
8   def passed_test?(mark)
9     if mark > 5 # Jaigu vartotojo ivesta reiksme yra didesne uz 5, tai
10       puts "Islaikė" # i ekrana isvedame teksta "Islaikė"
11     else # kitu atveju
12       puts "Neislaikė" # i ekrana isvedame teksta "neislaikė"
13     end
14   end
15   # Metodo "passed_test?" pabaiga
16 end
17 # Klases "Student" pradzia
18
19 puts 'Iveskite studento pazini:' # Isvaden teksta i ekrana "Iveskite studento pazini:"
20 mark = gets.to_i # Apsirasom a kintamaji, i ji prasmom vartotojo ivest reiksme, kuria
21                 # konvertuojam i int.
22
23 student = Student.new # Klases "Student" kvietimas/inicializavimas
24 student.passed_test?(mark) # Metodo "passed_test?", esancio klaseje "Student"
25                           # kvietimas, su parametru "mark"
```

```
> ruby main.rb
Iveskite studento pazini:
5
Islaikė
```

# ANTRA UŽDUOTIS

The screenshot shows a web-based Ruby REPL interface. The left sidebar displays a file explorer with 'main.rb' selected. The main editor area contains the following Ruby code:

```
1 # frozen_string_literal: true
2
3 # "" "By freezing string literals, you're not allocating new memory space for it."
4
5 # Klases "Numbers" pradžia
6 class Numbers
7   # Metodo "find_biggest" pradžia, trys parametrai: a, b, c
8   def find_biggest(krastine_a, krastine_b, krastine_c)
9     if krastine_a > krastine_b && krastine_a > krastine_c # Jeigu a reiksme bus
10       puts "Didžiausias: #{krastine_a}" # Ivedam teksta i ekrana "Didziausias", su
11       krastine_a
12     elsif krastine_b > krastine_a && krastine_b > krastine_c # Jeigu b reiksme bus
13       puts "Didžiausias: #{krastine_b}" # Ivedam teksta i ekrana "Didziausias", su
14       krastine_b
15     elsif krastine_c > krastine_a && krastine_c > krastine_b # Jeigu c reiksme bus
16       puts "Didžiausias: #{krastine_c}" # Ivedam teksta i ekrana "Didziausias", su
17       krastine_c
18     else # Kitu atveju error kintamajam priskirsime true reiksme, ir i ekrana isvesime
19       teksta "Klaida"
20     end
21   end
22   # Metodo "find_biggest" pabaiga
23 end
24 # Klases "Numbers" pabaiga
25
26 puts "Iveskite 3 skaicius:" # i ekrana isvedame teksta "Iveskite 3 skaicius:"
27 a = gets.to_i # Prasome vartotojo ivesti reiksme, kuria veliau konvertuojame i int
28   tipa
29 b = gets.to_i # Prasome vartotojo ivesti reiksme, kuria veliau konvertuojame i int
30   tipa
31 c = gets.to_i # Prasome vartotojo ivesti reiksme, kuria veliau konvertuojame i int
32   tipa
33
34 numbers = Numbers.new # Klases "Numbers" kvietimas/inicializavimas
35 numbers.find_biggest(a, b, c) # Metodo "find_biggest", esancio klaseje "Numbers"
36   kvietimas, su parametrais a, b ir c
```

The right sidebar shows the output of the script:

```
> ruby main.rb
Iveskite 3 skaicius:
1
2
Didziausias: 3
```

The screenshot shows a web-based Ruby REPL interface. The left sidebar displays a file explorer with 'main.rb' selected. The main editor area contains the following Ruby code:

```
1 # frozen_string_literal: true
2
3 # "" "By freezing string literals, you're not allocating new memory space for it."
4
5 # Klases "Numbers" pradžia
6 class Numbers
7   # Metodo "find_biggest" pradžia, trys parametrai: a, b, c
8   def find_biggest(krastine_a, krastine_b, krastine_c)
9     if krastine_a > krastine_b && krastine_a > krastine_c # Jeigu a reiksme bus
10       didesne uz b ir c reiksme,
11       puts "Didžiausias: #{krastine_a}" # Ivedam teksta i ekrana "Didziausias", su
12       krastine_a
13     elsif krastine_b > krastine_a && krastine_b > krastine_c # Jeigu b reiksme bus
14       didesne uz a ir c reiksme,
15       puts "Didžiausias: #{krastine_b}" # Ivedam teksta i ekrana "Didziausias", su
16       krastine_b
17     elsif krastine_c > krastine_a && krastine_c > krastine_b # Jeigu c reiksme bus
18       didesne uz a ir b reiksme,
19       puts "Didžiausias: #{krastine_c}" # Ivedam teksta i ekrana "Didziausias", su
20       krastine_c
21     else # Kitu atveju error kintamajam priskirsime true reiksme, ir i ekrana isvesime
22       teksta "Klaida"
23     end
24   end
25   # Metodo "find_biggest" pabaiga
26 end
27 # Klases "Numbers" pabaiga
28
29 puts "Iveskite 3 skaicius:" # i ekrana isvedame teksta "Iveskite 3 skaicius:"
30 a = gets.to_i # Prasome vartotojo ivesti reiksme, kuria veliau konvertuojame i int
31   tipa
32 b = gets.to_i # Prasome vartotojo ivesti reiksme, kuria veliau konvertuojame i int
33   tipa
34 c = gets.to_i # Prasome vartotojo ivesti reiksme, kuria veliau konvertuojame i int
35   tipa
36
37 numbers = Numbers.new # Klases "Numbers" kvietimas/inicializavimas
38 numbers.find_biggest(a, b, c) # Metodo "find_biggest", esancio klaseje "Numbers"
39   kvietimas, su parametrais a, b ir c
```

The right sidebar shows the output of the script:

```
> ruby main.rb
Iveskite 3 skaicius:
1
2
Didziausias: 3
```

The screenshot shows the Replit IDE interface. On the left, a file explorer shows 'main.rb' selected. The main editor displays a Ruby script with the following content:

```
1 # frozen_string_literal: true
2
3 # *** "By freezing string literals, you're not allocating new memory space for it."
4
5 # Klases "Numbers" pradzia
6 class Numbers
7   # Metodo "find_biggest" pradzia, tryps parametrui: a, b, c
8   def find_biggest(krastine_a, krastine_b, krastine_c)
9     if krastine_a > krastine_b && krastine_a > krastine_c # Jeigu a reiksme bus
10       didziausias: #krastine_a) # Ispedam teksta i ekrana "Didziausias", su
11       krastine_a
12     elsif krastine_b > krastine_a && krastine_b > krastine_c # Jeigu b reiksme bus
13       didziausias: #krastine_b) # Ispedam teksta i ekrana "Didziausias", su
14       krastine_b
15     elsif krastine_c > krastine_a && krastine_c > krastine_b # Jeigu c reiksme bus
16       didziausias: #krastine_c) # Ispedam teksta i ekrana "Didziausias", su
17       krastine_c
18     else # Kitu atveju error kintamajam priskirsime true reiksme, ir i ekrana isvesime
19       teksta "klaida";
20       puts 'klaida'
21     end
22   end
23   # Metodo "find_biggest" pabaiga
24 end
25
26 # Klases "Numbers" pabaiga
27
28 puts 'Iveskite 3 skaicius:' # i ekrana isvedame teksta "Iveskite 3 skaicius:"
29 a = gets.to_i # Prasome vartotojo ivesti reiksme, kurla vellau konvertuojame i int
30 tipa.
31 b = gets.to_i # Prasome vartotojo ivesti reiksme, kurla vellau konvertuojame i int
32 tipa.
33 c = gets.to_i # Prasome vartotojo ivesti reiksme, kurla vellau konvertuojame i int
34 tipa.
35
36 numbers = Numbers.new # Klases "Numbers" kvietimas/inicializavimas
37 numbers.find_biggest(a, b, c) # Metodo "find_biggest", esancio klaseje "Numbers"
38 kvietimas, su parametrais a, b ir c
39
40
```

On the right, the output console shows the following text:

```
> ruby main.rb
Iveskite 3 skaicius:
1
2
3
Didziausias: 3
>
```

This screenshot is identical to the one above, showing the same Ruby script in the Replit IDE. The script defines a 'Numbers' class with a 'find\_biggest' method and a main section that prompts the user for three numbers and prints the largest one. The output console shows the user has entered three numbers (1, 2, 3) and the program has correctly identified 3 as the largest number.

## **MANO PASTEBĖJIMAI ATLIEKANT ŠIĄ UŽDUOTĮ:**

1. Perkėliau komentarus, kadangi negali jie būt toj pačioj eilutėj, kur paminėti žodžiai `def`, `class` ir dar kažkokie yra;
2. Negalima naudot `ascii` simbolių komentaruose, t.y teko ištrint lietuviškas raides;
3. Pataisė lygiavimą automatiškai pats `rubocop`;
4. Parametrų pavadinimai daugiau nei iš 2-jų raidžių turi būt. Antroj programoj teko pakeist kintamųjų `a`, `b` ir `c` pavadinimus;
5. Kaikur komentarai buvo per ilgi, yra tam tikras simbolių kiekis numatytas;
6. Antroje programoje `complexity` metodo buvo per didelis, metė klaidą, kad per daug eilučių naudoju net ir dar keletą klaidų apie tą metodą, tad supaprastinau jį ir klaida dingo;