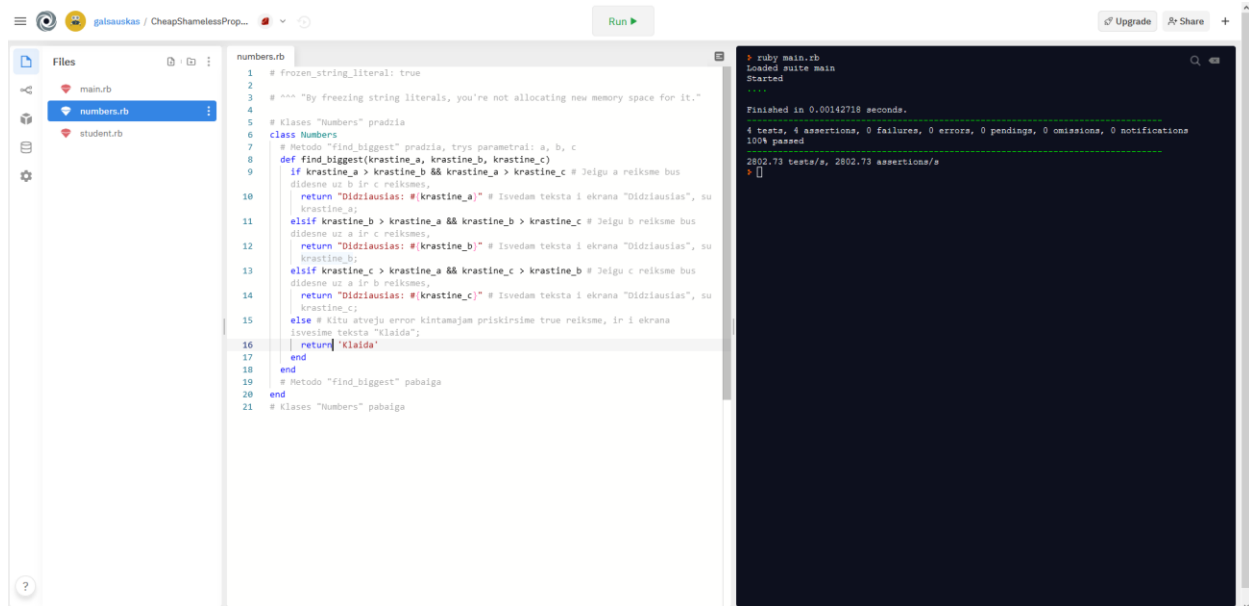


PIRMA PROGRAMA

Numbers.rb klasė;

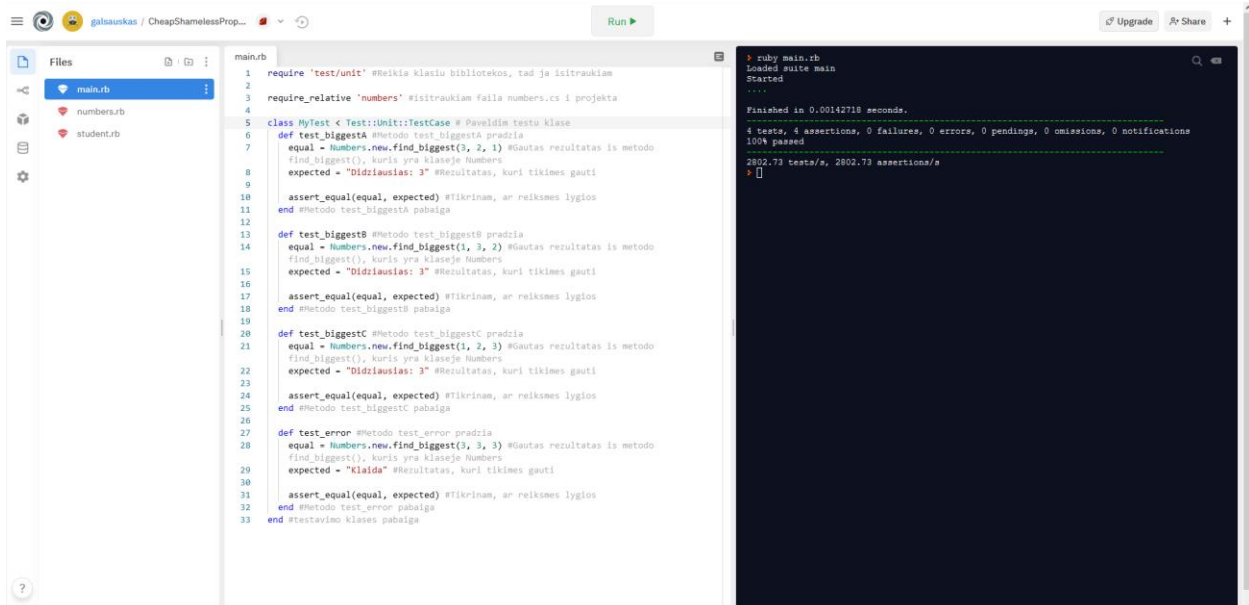


The screenshot shows a Ruby IDE with a file explorer on the left containing 'main.rb', 'numbers.rb', and 'student.rb'. The 'numbers.rb' file is open in the center, showing a Ruby class definition for 'Numbers' with a 'find_biggest' method. The method takes three arguments (a, b, c) and returns the largest value. It includes comments in Lithuanian. On the right, the test results for 'main.rb' are displayed, showing that 4 tests passed, 4 assertions were successful, and the suite finished in 0.00142718 seconds.

```
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, tryk parametrų: 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       return "Didziausias: #{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       return "Didziausias: #{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       return "Didziausias: #{krastine_c}" # Ivedam teksta i ekrana "Didziausias", su
20       krastine_c;
21     else # Kita atveju error kintamajam priskirsime true reiksme, ir i ekrana
22       isvedime teksta "Klaida";
23     end
24   end
25 end
26 # Metodo "find_biggest" pabaiga
27 end
28 # Klases "Numbers" pabaiga
```

```
> ruby main.rb
Loaded suite main
Started
.....
Finished in 0.00142718 seconds.
4 tests, 4 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
2802.73 tests/s, 2802.73 assertions/s
> []
```

Numbers.rb kalsės metodo testai



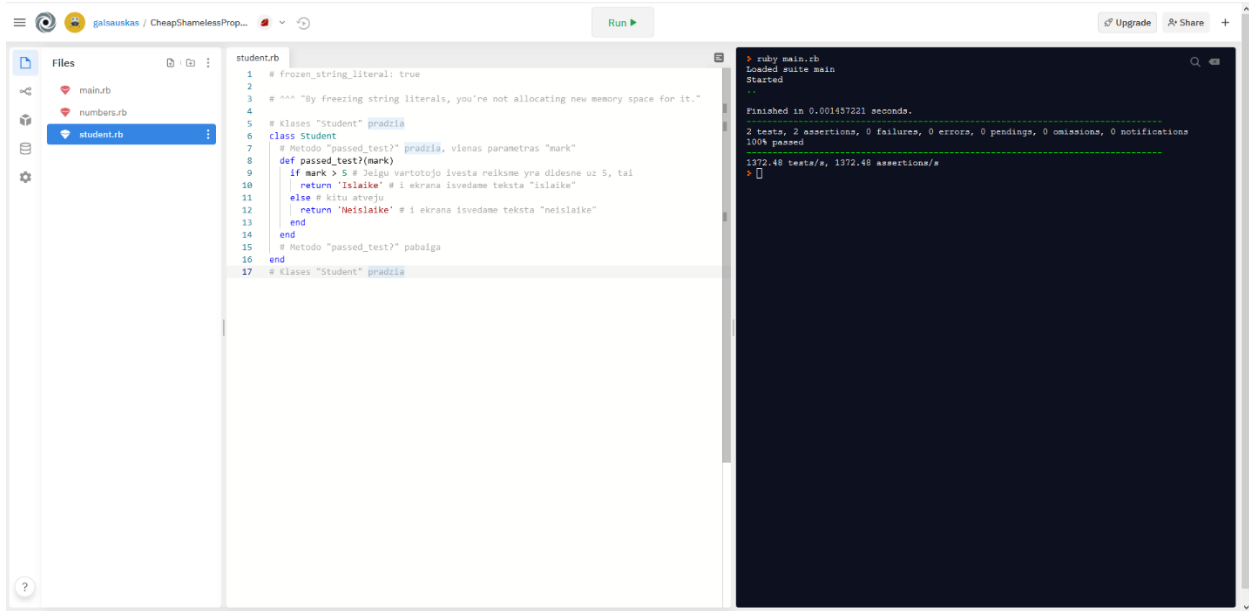
The screenshot shows the same Ruby IDE with the 'main.rb' file open. It contains unit tests for the 'Numbers' class using the 'Test::Unit' framework. The tests include 'test_biggestA', 'test_biggestB', 'test_biggestC', and 'test_error', each verifying the 'find_biggest' method's output for specific inputs. The test results on the right are identical to the previous screenshot, showing 4 tests passed.

```
1 require 'test/unit' #Reikia klasiu bibliotekos, tad je isitraukiam
2
3 require_relative "numbers" #isitraukiam faila numbers.cs i projekta
4
5 class MyTest < Test::Unit::TestCase # Paveldim testu klase
6   def test_biggestA #Metodo test_biggestA pradžia
7     equal = Numbers.new.find_biggest(1, 2, 1) #Gautas rezultatas is metodo
8     find_biggest(), kuris yra klaseje Numbers
9     expected = "Didziausias: 1" #Rezultatas, kuri tikimes gauti
10    assert_equal(equal, expected) #tikrinam, ar reiksmes lygios
11    end #Metodo test_biggestA pabaiga
12
13   def test_biggestB #Metodo test_biggestB pradžia
14     equal = Numbers.new.find_biggest(1, 3, 2) #Gautas rezultatas is metodo
15     find_biggest(), kuris yra klaseje Numbers
16     expected = "Didziausias: 3" #Rezultatas, kuri tikimes gauti
17     assert_equal(equal, expected) #tikrinam, ar reiksmes lygios
18     end #Metodo test_biggestB pabaiga
19
20   def test_biggestC #Metodo test_biggestC pradžia
21     equal = Numbers.new.find_biggest(1, 2, 3) #Gautas rezultatas is metodo
22     find_biggest(), kuris yra klaseje Numbers
23     expected = "Didziausias: 3" #Rezultatas, kuri tikimes gauti
24     assert_equal(equal, expected) #tikrinam, ar reiksmes lygios
25     end #Metodo test_biggestC pabaiga
26
27   def test_error #Metodo test_error pradžia
28     equal = Numbers.new.find_biggest(3, 3, 3) #Gautas rezultatas is metodo
29     find_biggest(), kuris yra klaseje Numbers
30     expected = "Klaida" #Rezultatas, kuri tikimes gauti
31     assert_equal(equal, expected) #tikrinam, ar reiksmes lygios
32     end #Metodo test_error pabaiga
33 end #testavimo klases pabaiga
```

```
> ruby main.rb
Loaded suite main
Started
.....
Finished in 0.00142718 seconds.
4 tests, 4 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
2802.73 tests/s, 2802.73 assertions/s
> []
```

ANTRA PROGRAMA

Student.rb klasė;



The screenshot shows a Ruby IDE with a file explorer on the left containing 'main.rb', 'numbers.rb', and 'student.rb'. The main editor displays the 'student.rb' file with the following code:

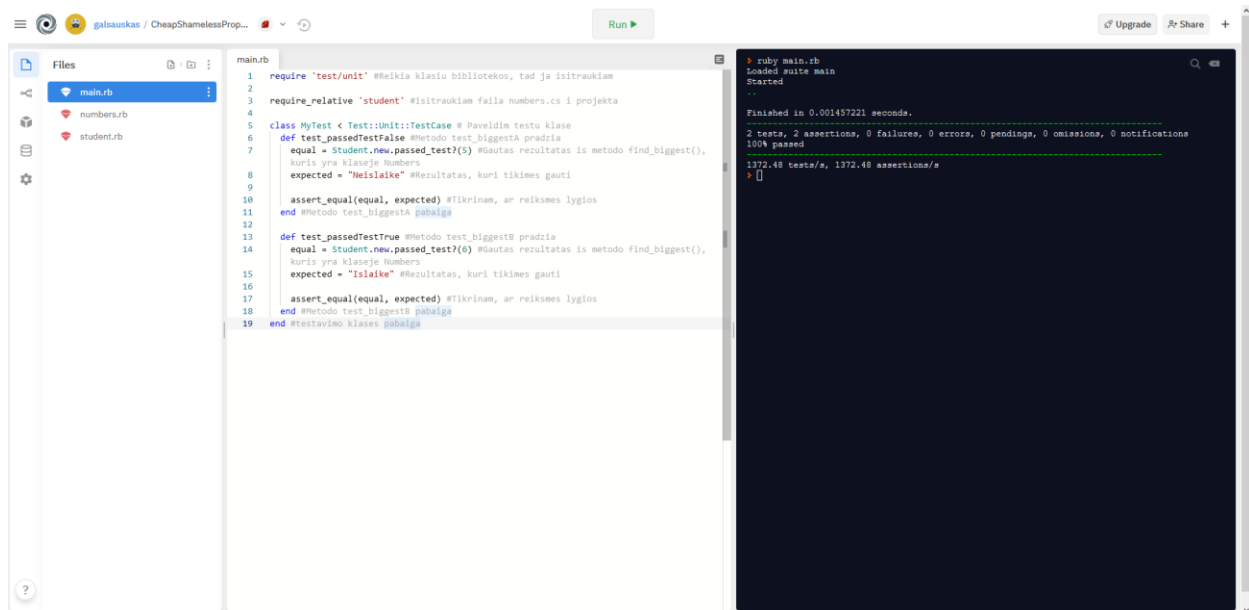
```
1 # frozen_string_literal: true
2
3 # *** "By freezing string literals, you're not allocating new memory space for it."
4
5 # Klases "Student" pradžia
6 class Student
7   # Metodo "passed_test?" pradžia, vienas parametras "mark"
8   def passed_test?(mark)
9     if mark > 5 # Jeigu vartotojo ivesta reiksme yra didesne uz 5, tai
10       return 'Islaike' # i ekrana isvedame teksta "islaike"
11     else # kitu atveju
12       return 'Neislaike' # i ekrana isvedame teksta "neislaike"
13     end
14   end
15   # Metodo "passed_test?" pabaiga
16 end
17 # Klases "Student" pradžia
```

The right sidebar shows the test results for 'ruby main.rb':

```
> ruby main.rb
Loaded suite main
Started

Finished in 0.001457221 seconds.
2 tests, 2 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
1372.48 tests/s, 1372.48 assertions/s
> []
```

Student.rb klasės metodo testai



The screenshot shows a Ruby IDE with a file explorer on the left containing 'main.rb', 'numbers.rb', and 'student.rb'. The main editor displays the 'main.rb' file with the following code:

```
1 require 'test/unit' #Reikia klasiu bibliotekos, tad ja isitraukiam
2
3 require_relative 'student' #isitraukiam faila numbers.cs i projekta
4
5 class MyTest < Test::Unit::TestCase # Paveldim testu klase
6   def test_passedTestFalse #Metodo test_biggestA pradžia
7     equal = Student.new.passed_test?(5) #Gautas rezultatas is metodo find_biggest(),
8     kuriu yra klaseje Numbers
9     expected = "Islaike" #Rezultatas, kuri tikimes gauti
10    assert_equal(equal, expected) #Tikrinam, ar reiksmes lygios
11  end #Metodo test_biggestA pabaiga
12
13   def test_passedTestTrue #Metodo test_biggestB pradžia
14     equal = Student.new.passed_test?(6) #Gautas rezultatas is metodo find_biggest(),
15     kuriu yra klaseje Numbers
16     expected = "Islaike" #Rezultatas, kuri tikimes gauti
17     assert_equal(equal, expected) #Tikrinam, ar reiksmes lygios
18   end #Metodo test_biggestB pabaiga
19 end #testavimo klases pabaiga
```

The right sidebar shows the test results for 'ruby main.rb':

```
> ruby main.rb
Loaded suite main
Started

Finished in 0.001457221 seconds.
2 tests, 2 assertions, 0 failures, 0 errors, 0 pendings, 0 omissions, 0 notifications
100% passed
1372.48 tests/s, 1372.48 assertions/s
> []
```