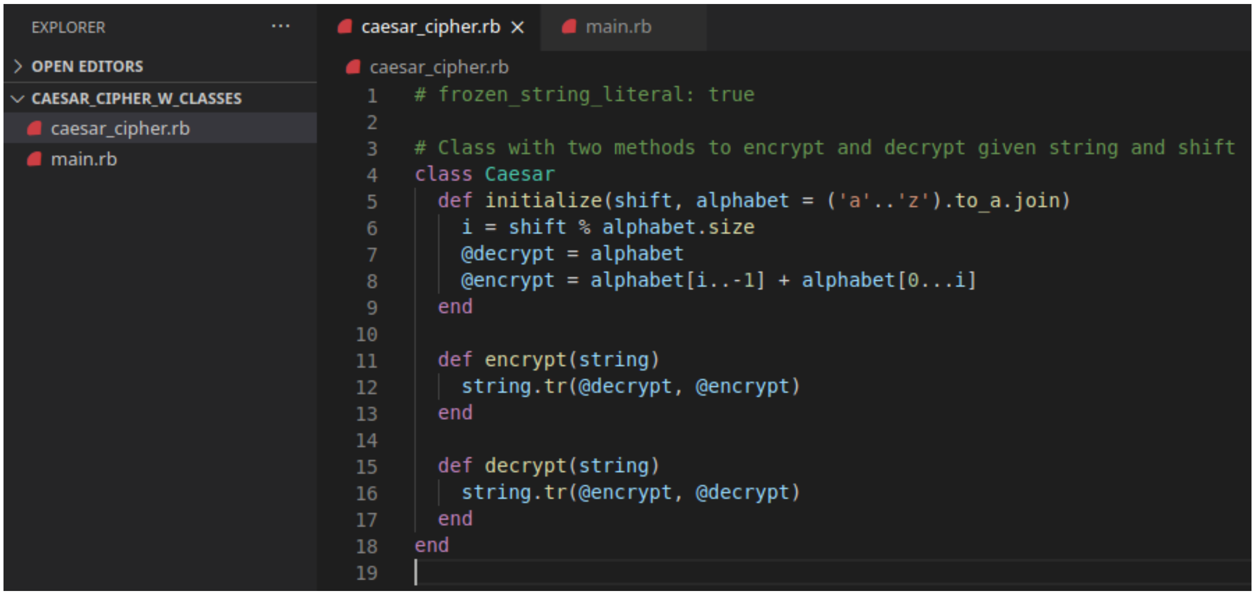
TASK 3

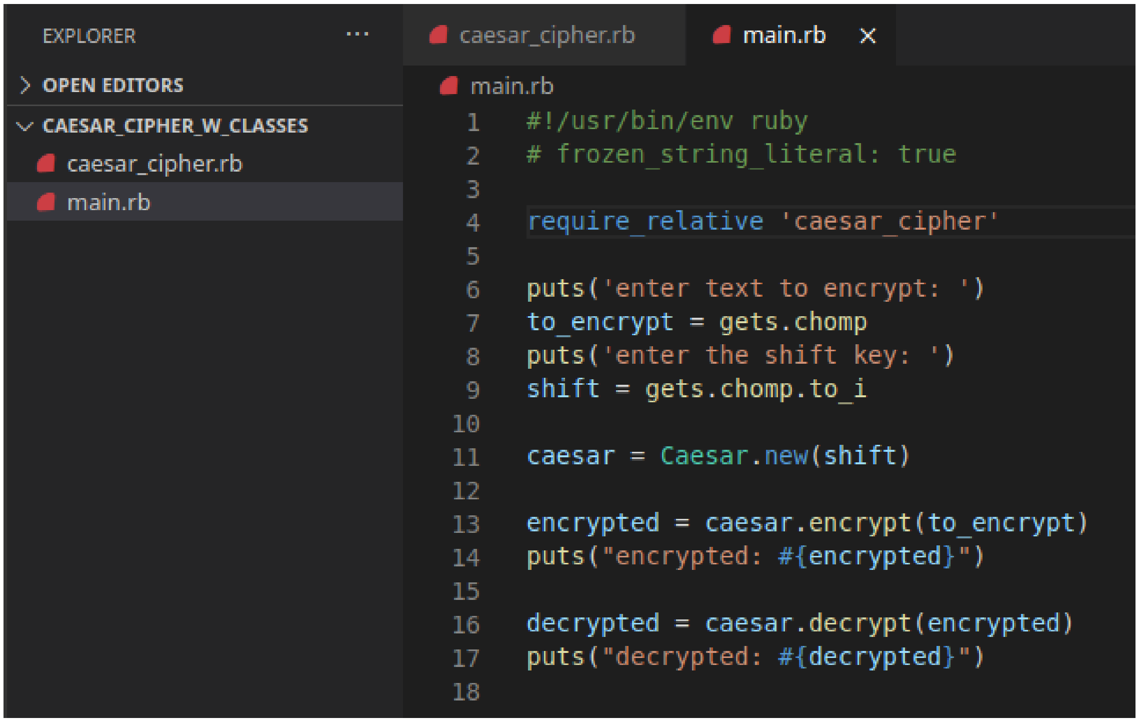
Caesar cipher

After rewriting Task2 Caeser cipher according to Object-Oriented programming paradigm:



```ruby
# frozen_string_literal: true

# Class with two methods to encrypt and decrypt given string and shift
class Caesar
  def initialize(shift, alphabet = ('a'..'z').to_a.join)
    i = shift % alphabet.size
    @decrypt = alphabet
    @encrypt = alphabet[i..-1] + alphabet[0...i]
  end

  def encrypt(string)
    string.tr(@decrypt, @encrypt)
  end

  def decrypt(string)
    string.tr(@encrypt, @decrypt)
  end
end
```

Figure 1 – Caesar cipher program using OOP paradigm

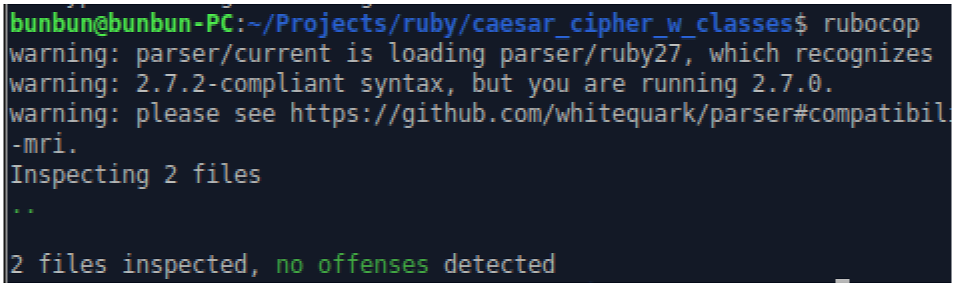


```ruby
#!/usr/bin/env ruby
# frozen_string_literal: true

require_relative 'caesar_cipher'

puts('enter text to encrypt: ')
to_encrypt = gets.chomp
puts('enter the shift key: ')
shift = gets.chomp.to_i

caesar = Caesar.new(shift)

encrypted = caesar.encrypt(to_encrypt)
puts("encrypted: #{encrypted}")

decrypted = caesar.decrypt(encrypted)
puts("decrypted: #{decrypted}")
```

Figure 2 - main file for output

The encryption, decryption and alphabet values are now initialized inside the class. Main.rb file is for output only with the import for the cipher class.

Fixed mistakes with rubocop, no offenses detected.



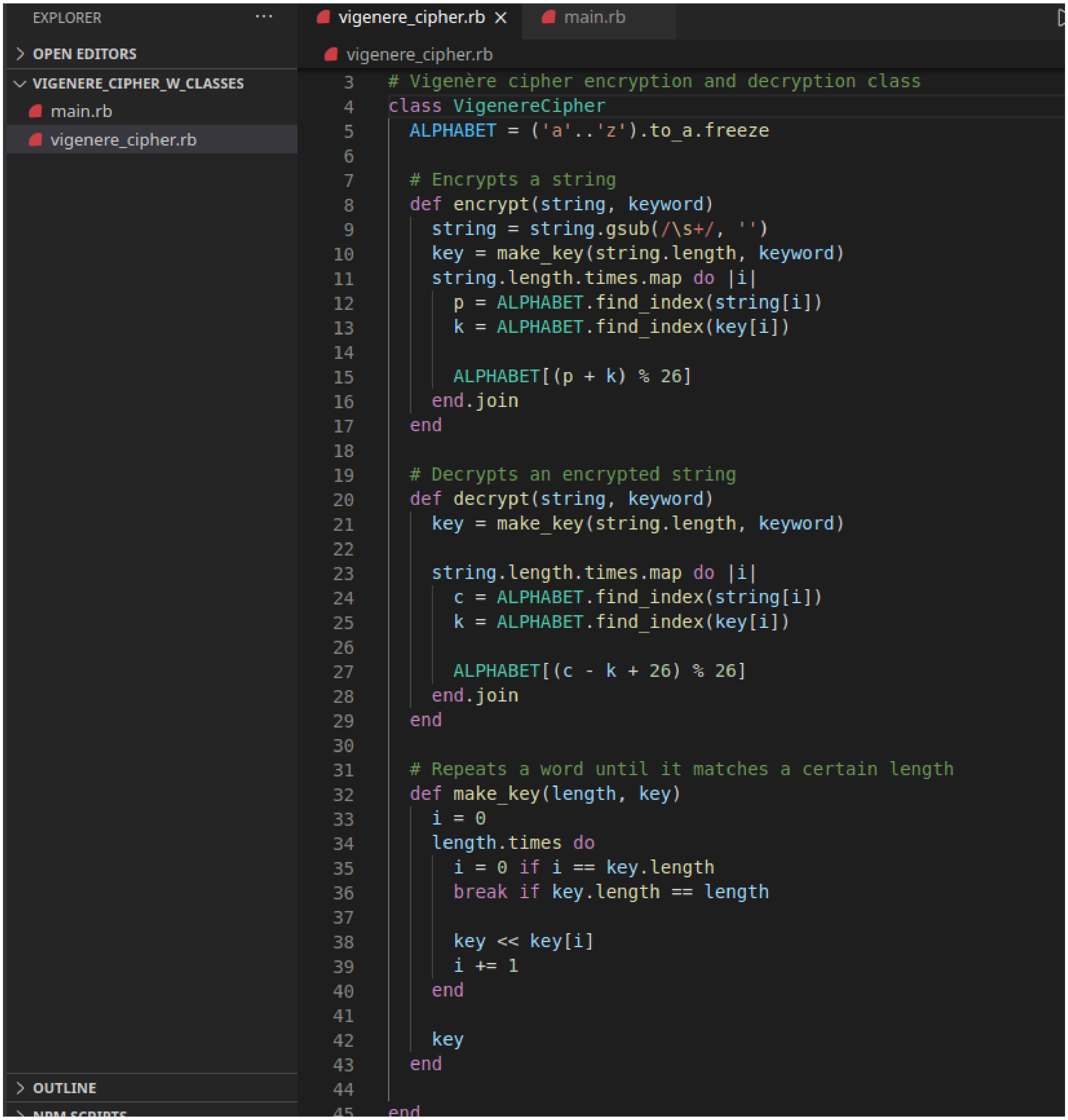
```
bunbun@bunbun-PC:~/Projects/ruby/caesar_cipher_w_classes$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.0.
warning: please see https://github.com/whitequark/parser#compatibil
-mri.
Inspecting 2 files
..

2 files inspected, no offenses detected
```

Figure 3 - rubocop result

The outputs are the same as in task 2.

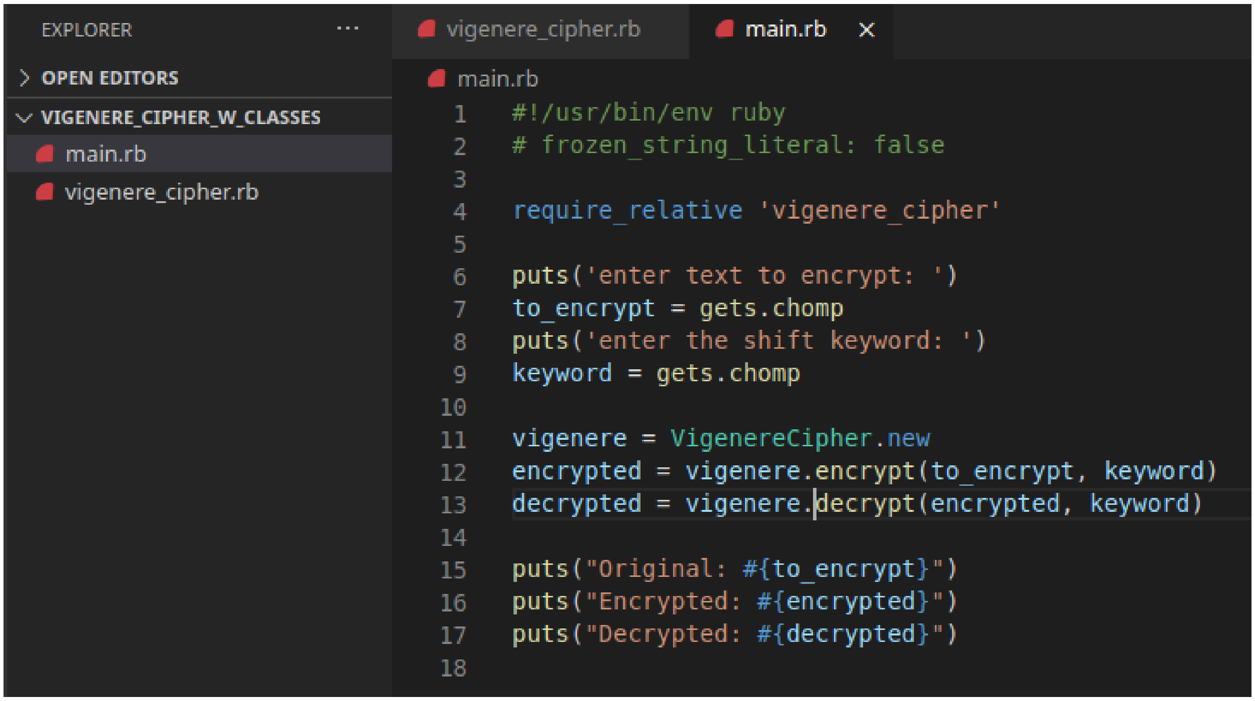
Vigenere cipher

After rewriting task 2 Vigenere cipher according to Object-Oriented programming paradigm:



```ruby
# Vigenère cipher encryption and decryption class
class VigenereCipher
  ALPHABET = ('a'..'z').to_a.freeze

  # Encrypts a string
  def encrypt(string, keyword)
    string = string.gsub(/\s+/, '')
    key = make_key(string.length, keyword)
    string.length.times.map do |i|
      p = ALPHABET.find_index(string[i])
      k = ALPHABET.find_index(key[i])

      ALPHABET[(p + k) % 26]
    end.join
  end

  # Decrypts an encrypted string
  def decrypt(string, keyword)
    key = make_key(string.length, keyword)

    string.length.times.map do |i|
      c = ALPHABET.find_index(string[i])
      k = ALPHABET.find_index(key[i])

      ALPHABET[(c - k + 26) % 26]
    end.join
  end

  # Repeats a word until it matches a certain length
  def make_key(length, key)
    i = 0
    length.times do
      i = 0 if i == key.length
      break if key.length == length

      key << key[i]
      i += 1
    end

    key
  end

end
```

Figure 4 – Vigenere cipher program using OOP paradigm

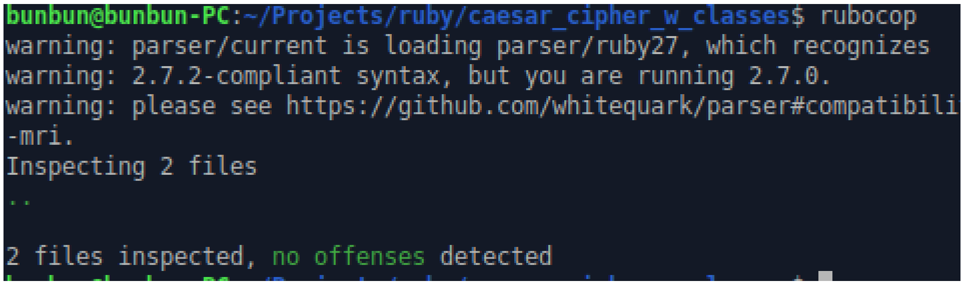


```ruby
#!/usr/bin/env ruby
# frozen_string_literal: false

require_relative 'vigenere_cipher'

puts('enter text to encrypt: ')
to_encrypt = gets.chomp
puts('enter the shift keyword: ')
keyword = gets.chomp

vigenere = VigenereCipher.new
encrypted = vigenere.encrypt(to_encrypt, keyword)
decrypted = vigenere.decrypt(encrypted, keyword)

puts("Original: #{to_encrypt}")
puts("Encrypted: #{encrypted}")
puts("Decrypted: #{decrypted}")

```

Figure 5 - main file for output

Everything works in the same way as in task 2, the only difference is that all of the methods are in a class now and it is called from the main file where the output is handled.

Fixed mistakes with rubocop, no offenses detected.



```
bunbun@bunbun-PC:~/Projects/ruby/caesar_cipher_w_classes$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.0.
warning: please see https://github.com/whitequark/parser#compatibili
-mri.
Inspecting 2 files
..

2 files inspected, no offenses detected
```

Figure 6 - rubocop result

The outputs are the same as in task 2.