

Харківський національний університет ім. В.Н. Каразіна
Факультет комп'ютерних наук
Кафедра електроніки та управляючих систем

ЗВІТ
З КОНТРОЛЬНОЇ РОБОТИ №1
дисципліни: «Мови прикладного програмування»

Виконав: студентка групи ЗКС31
Москалюк С.Ю.

Перевірив:
Паршенцев Б.В.

Харків
2023

Теоретична частина

1. Як можна використовувати методи `method_missing` та `define_method` для динамічного створення методів у Ruby?
2. Які переваги надають модулі у Ruby?

Практична частина

Напишіть функцію `word_count`, яка приймає рядок і повертає хеш, де ключами є слова, а значення — кількість разів, кожне кожне слово зустрічається в рядку.

Теоретична частина

1. У Ruby методи `method_missing` і `define_method` дозволяють створювати та обробляти динамічні методи. Коли викликаний метод не знайдений, викликається `method_missing`. За допомогою `define_method` ми можемо використовувати його для перехоплення викликів неіснуючих методів і динамічного створення відповідних реалізацій. Наприклад:

```
class DynamicMethodsExample
  def method_missing(name, *args, &block)
    # Перевіряємо, чи ім'я методу починається з "create_method_"
    if name.to_s =~ /^create_method_(.+)$/
      # Якщо так, отримуємо аргументи та блок, передані методу, і ви-
      значаємо новий метод
      define_dynamic_method($1, *args, &block)
    else
      # Якщо ім'я методу не відповідає, викликаємо метод_missing за
      замовчуванням
      super
    end
  end

  private

  def define_dynamic_method(name, *args, &block)
    # Створюємо новий метод з іменем name та використовуємо
    define_method
    self.class.send(:define_method, name) do |*method_args|
      puts "Dynamic method #{name} is called with arguments:
#{method_args.inspect}"
      block.call(*method_args) if block_given?
    end
  end

  # Приклад використання
  example = DynamicMethodsExample.new
  example.create_method_greet do |name|
    puts "Hello, #{name}!"
  end
end
```

```
example.greet("John") # Викличе створений метод greet з аргументом "John"
```

У цьому прикладі, коли викликається метод `create_method_greet`, він визиває `method_missing`, щоб перевірити, чи відповідає ім'я методу шаблону `create_method_`. Якщо відповідає, використовується функція `define_method` для створення нового методу з іменем, вказаним у шаблоні. Цей новий метод виводить аргументи та викликає блок. Далі можна викликати створений метод `greet`, передавши аргументи та блок у `create_method_greet`.

2. Ruby-модулі мають такі переваги:

Вкладеність та простори імен: допомагає створювати логічні блоки коду та запобігає конфліктам імен.

Множинне спадкування: дозволяє обходити обмеження множинного спадкування класів, включаючи один модуль у кілька класів.

Міксини(Mixins): дозволяють вставляти функціональність у класи, що робить код гнучким і дозволяє повторно використовувати код.

Розділення функціональності: полегшує розуміння та управління кодом, розділяючи функціональність на логічні групи.

Метод “include”: дозволяє використовувати функціональність модуля як методи екземпляра в класах.

Метод “extend”: дозволяє використовувати функціональність модуля як методи класу.

Callback-функції: Callback-функції можуть бути включені в модулі для виклику в певних точках виконання коду.

Ізоляція коду: Модулі полегшують роботу з великими проектами, оскільки вони дозволяють ізолювати код в окремих файлових одиницях.

Поліморфізм: Впровадження методів у різних модулях і включення їх у класи допомагає реалізувати поліморфізм.

Практична частина:

Для перевірки правильності роботи функції я взяла англійську версію «Сон», Т.Г. Шевченко.

Лістинг 1:

```
def word_count(sentence)
  words = sentence.split(/\s+/)
  word_count_hash = Hash.new(0)

  words.each do |word|
    word = word.gsub(/\W+/, '')
    word_count_hash[word.downcase] += 1 unless word.empty?
  end

  word_count_hash
end

sentence = "Each man on earth has his own fate,
Each one his highway wide:
This one builds up, that one lays waste,
And that casts greedy eyes
O'er all the globe, to find somewhere
A land not yet enslaved,
Which he could conquer and then bear
With him into the grave.

This fellow in his neighbour's home
His host cleans out at cards,
While that one in a corner hones
A blade for brother's heart.
Then there's the solid citizen,
The worthy, pious kind,
Who'll creep up like a cat and then
Bide patiently his time
Until hard luck hits your affairs,
Then pounce! – Don't plead your cause:
Your wife's appeals and children's tears
Won't save you from his claws.
And that one, generous and grand,
The fervent patriot,
So deeply loves his native land,
So worries o'er its lot,
As from his country's heart he sucks
The blood as though it was water!...
The brethren meanwhile sit and look,
Their eyes agape like saucers!
And bleat like lambs: Perhaps it was
Thus ordained from on high.
That's how it's meant to be! Because
There's no God in the sky!
You pull your yoke until your breath
Gives out and you are done,
Yet pray for heaven after death?
In vain! There's none! There's none!
Your labour's lost. Come to! Come to!
In this world every one –
The princes, and the beggars, too,
They all are Adam's sons.
Both he... and he... What's this I prate?
What is it all about?
I banquet every single day,
Carouse day in, day out,
```

```
While you with envy burn and hate!  
Don't scold: 'Twill do no good -  
I'm deaf to you! I drink my own,  
Not other people's blood!"  
result = word_count(sentence)  
puts result
```