Бублик Владислав КС31

КР з дисципліни МПП

Варіант 1

**Білет**

1.Як виконати обробку винятків (exception handling) в Ruby?

Обробка винятків в Ruby - це спосіб зміцнити програму проти несподіваних проблем, які можуть виникнути під час її роботи. Такі події, як ділення на нуль чи спроба отримати доступ до чогось, чого немає, можуть призвести до аварійного завершення програми. Однак обробка винятків дозволяє нам спланувати, яким чином програма повинна реагувати на такі проблеми. На практиці, це виглядає так: ми визначаємо певний блок коду, в якому можуть виникнути помилки, і вказуємо програмі, як реагувати на ці помилки, якщо вони все ж виникають. Ми також можемо вказати, які дії повинні відбутися завжди, незалежно від того, чи сталася помилка. В цілому, обробка винятків допомагає зробити нашу програму більш стабільною і передбачуваною, навіть у випадку помилок.

2. Що таке модуль в Ruby? Як ви створюєте та використовуєте модулі в Ruby?

Модуль (module) в Ruby - це спосіб організації та групування функцій та класів, які можна використовувати у програмі. Модулі служать для збереження методів та констант, які можна використовувати в інших частинах вашої програми. Приклад створення та використання модулей в Ruby:

module MyModule

def some\_method

# Main

end

def another\_method

# Main

end

end

Тут визначається модуль MyModule і включає в нього два методи some\_method та another\_method.

Використання модуля:

Для використання методів і констант з модуля в інших частинах програми, потрібно включити цей модуль у клас або інший модуль:

class MyClass

include MyModule # Включення модуля MyModule в клас MyClass

end

obj = MyClass.new

obj.some\_method # Виклик методу із модуля

Так модуль дає змогу використовувати один і той самий код в різних частинах програми, для полегшення роботи з великим проектом.

3. Напишіть програму, яка приймає список чисел і сортує їх у порядку зростання або

спадання за вибором користувача.

Вихідний код:

def sort\_numbers(numbers, ascending)

if ascending

numbers.sort!

else

numbers.sort! { |a, b| b <=> a }

end

end

puts "Enter a list of numbers separated by spaces:"

input = gets.chomp

numbers = input.split.map { |num| num.to\_i }

puts "Choose the sorting order:"

puts "1. Ascending"

puts "2. Descending"

choice = gets.chomp.to\_i

if choice == 1

sort\_numbers(numbers, true)

puts "Sorted in ascending order: #{numbers.join(', ')}"

elsif choice == 2

sort\_numbers(numbers, false)

puts "Sorted in descending order: #{numbers.join(', ')}"

else

puts "Invalid choice. Please select 1 for ascending or 2 for descending."

end

4. Напишіть програму, яка генерує випадковий пароль заданої довжини для

користувача.

def generate\_password(length)

characters = ('a'..'z').to\_a + ('A'..'Z').to\_a + ('0'..'9').to\_a

password = ''

length.times do

random\_index = rand(characters.length)

password << characters[random\_index]

end

password

end

puts "Enter the desired password length:"

password\_length = gets.chomp.to\_i

if password\_length > 0

password = generate\_password(password\_length)

puts "Generated password: #{password}"

else

puts "Invalid password length. Please enter a positive integer."

end