

Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра моделювання систем і технологій

ЕКЗАМЕНАЦІЙНА РОБОТА

з дисципліни: «Мови прикладного програмування»

Виконав: студент групи КС33
Кучеренко Олег Анатолійович

Перевірив: старший викладач
кафедри МСіТ
Паршенцев Богдан Володимирович

Харків
2023

1. Паттерн Singleton - це один з патернів проектування, який гарантує, що клас має тільки один екземпляр і забезпечує глобальний доступ до нього.

Тип Singleton в Ruby:

У Ruby singleton реалізується досить просто за допомогою модулю Singleton. Цей модуль визначає метод `instance` для отримання єдиного екземпляра класу.

Переваги Singleton в Ruby:

Гарантується, що в межах одного процесу буде лише один екземпляр класу. Забезпечує глобальний доступ до цього єдиного екземпляра з будь-якої точки програми.

Недоліки Singleton в Ruby:

Іноді глобальний доступ може призвести до поганого організованості коду і зростання залежностей.

Тестування може бути важчим, оскільки клас має глобальний стан, який може впливати на інші тести.

Приклад:

```
class Singleton
  @instance = new
  private_class_method :new
  def self.instance
    @instance
  end
  def some_business_logic
  end
end
```

2. Метаклас (також відомий як "синглтон клас" або "eigenclass") - це спеціальний клас, який пов'язаний з конкретним об'єктом. Кожен об'єкт має свій власний метаклас, який використовується для зберігання методів, які визначаються тільки для цього конкретного об'єкта.

Метакласи використовуються для динамічного додавання методів до окремих об'єктів або класів. Це дозволяє вам розширювати функціонал об'єктів лише у визначеному контексті, не змінюючи сам клас або всі екземпляри цього класу.

3. Символи є спеціальним типом об'єкта, який представляє неперевірений, нещодавно змінюваний літерал. Символи визначаються за допомогою двокрапок (:), і вони є імутабельними на відміну від рядків.

4. У Ruby існує кілька типів змінних, і їхні різновиди впливають на їхню видимість та область дії. Основні типи змінних в Ruby включають локальні, глобальні, екземплярні та класові змінні.

Локальні змінні:

Видимість: Локальні змінні видимі лише у межах того блоку чи методу, в якому вони були визначені. Зовнішні блоки чи методи не можуть звертатися до локальних змінних.

Область дії: Локальні змінні існують тільки в межах блоку чи методу, де вони були створені.

Глобальні змінні:

Видимість: Глобальні змінні видимі в усьому програмному коді, включаючи всі блоки та методи.

Область дії: Глобальні змінні існують протягом усього життєвого циклу програми.

Екземплярні та класові змінні:

Видимість: Екземплярні змінні доступні тільки для конкретного об'єкта класу, а класові змінні є спільними для всіх екземплярів класу.

Область дії: Екземплярні змінні існують протягом життєвого циклу конкретного об'єкта, а класові змінні - протягом життєвого циклу всього класу.