Контрольна робота №2 з дисципліни
“Мови прикладного програмування”
Виконав студент групи КС-33
Мішуров Михайло Сергійович

Теоретична частина
1. Що таке метапрограмування та в чому його переваги?
2. Що є лямбда-вираз? У чому відмінність від блоку?
Практична частина
Створіть модуль Printable, який додає метод print_info до будь-якого класу. Метод має виводити інформацію про поточний об'єкт

Результати виконання завдання

1. Метапрограмування - це парадигма побудови коду інформаційної системи з динамічною зміною поведінки або структури в залежності від даних, дій користувача або взаємодії з іншими системами. Завдання метапрограмування: підвищення абстракції коду та його гнучкості, повторне використання, прискорення розробки, спрощення міжсистемної інтеграції. Прикладом мета програмування є метамовна абстракція - це процес вирішення складних завдань шляхом створення нової мови або створення словника з метою кращого розуміння предметної галузі. Такі мови називають мовами предметної області (DSL - Domain Specific Languages).

2. Блок це будь-який код, визначений усередині фігурних дужок { }, або всередині контейнера do-end. Код усередині блоку це всього лише логіка, яка ще не обернута в об'єкт. Коли цей блок передається методу і в цьому методі використовується вираз yield, тоді Ruby оберне блок в об'єкт типу Proc. Як тільки блок стає об'єктом Proc, Ruby може використовувати його і код, що міститься в блоці оживає. Lambda - це різновид об'єкта Proc, лямбда майже повністю ідентична Proc. Якщо лямбда викликається з більшою чи меншою кількістю аргументів, ніж необхідно, тоді отримуємо помилку ArgumentError. Однак, коли Proc викликається з більшою кількістю аргументів, ніж необхідно, жодної помилки не повертається і зайві аргументи просто відкидаються. Коли процедура викликається з меншою кількістю аргументів, то параметри, які не отримали необхідних значень, набувають значення nil. Коли лямбда-функція повертає значення і ця лямбда викликається всередині методу, то метод просто продовжуватиме своє виконання після закінчення роботи лямбди. Однак, коли Proc повертає значення і ця процедура викликається всередині методу, виконання цього методу буде зупинено. Через це Proc слід використовувати обережно.

3.
Вивід програми: #<Sample:0x00000088be8ea930 @var1=10, @var2=20>
Код програми:

```
 module Printable
  def print_info
    puts self.inspect
  end
end

class Sample
  include Printable
  def initialize(v1, v2)
    @var1 = v1
    @var2 = v2
  end
end

if __FILE__ == $0
  samp = Sample.new(10, 20)
  samp.print_info
end
```