

Prueba - Introducción a la programación con Ruby

En esta prueba validaremos nuestros conocimientos de Programación Orientada a Objetos. Para lograrlo, necesitarás aplicar los conocimientos adquiridos hasta este punto.

Lee todo el documento antes de comenzar el desarrollo **individual**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Tiempo asociado: 2 horas cronológicas

Descripción

Aplicando los conceptos y herramientas aprendidas hasta ahora, nos enfrentamos a una solicitud de un cliente en el cual nos solicita que hagamos refactorización del siguiente código.

```
class Person
  def initialize(first, last, age, type)
    @first_name = first
    @last_name = last
    @age = age
    @type = type
  end

  def birthday
    @age += 1
  end

  def talk
    if @type == "Student"
      puts "Aquí es la clase de programación con Ruby?"
    elsif @type == "Teacher"
      puts "Bienvenidos a la clase de programación con Ruby!"
    elsif @type == "Parent"
      puts "Aquí es la reunión de apoderados?"
    end
  end
end
```

```
def introduce
  if @type == "Student"
    puts "Hola profesor. Mi nombre es #{@first_name} #{@last_name}."
  elsif @type == "Teacher"
    puts "Hola alumnos. Mi nombre es #{@first_name} #{@last_name}."
  elsif @type == "Parent"
    puts "Hola. Soy uno de los apoderados. Mi nombre es #{@first_name}
#{@last_name}."
  end
end
end
```

Utilizando los principios de Herencia y Polimorfismo, refactoriza el siguiente código para eliminar la innecesaria cantidad de sentencias if en los métodos talk e introduce.

Requerimientos

1. Identificar las posibles subclases que aparecen en el código a refactorizar. Cada una de ella contendrá los métodos correspondientes, `talk` e `introduce`.
(3 Puntos)

Hint: Teacher sería una clase.

2. A partir de la refactorización del código se debe generar herencia de ancestros aplicando la sintaxis de Ruby, esto para la subclases.
(3 Puntos)
3. El método `introduce` al imprimir el mensaje muestra el nombre y apellido haciendo uso de interpolación de variables de instancia.
(1 Puntos)
4. Al generar las instancias de cada objeto, se muestra un mensaje en función del método que sea invocado.
(3 Puntos)



¡Mucho éxito!