

Desafío - Objetos

Consideraciones

- Para realizar este desafío debes haber revisado todo el material disponibilizado en la plataforma.
- Crea una carpeta y guarda cada archivo `.rb` con el número de la pregunta, siguiendo las instrucciones de manera local con **Sublime** o **Atom**.
- Luego guarda los cambios y súbelos a tu repositorio de GitHub.
- Luego de pusheados los últimos cambios, sube el link de GitHub en el desafío de la sección correspondiente en la plataforma.

1) Sintaxis

- Copia el siguiente código y ejecútalo. Luego, corrige los errores para poder imprimir ejecutar ambos métodos y finalmente obtener el valor de la variable de `a`.

```
class Anything
  def foo
    a = 5
  end

  def bar
    @a += 1
  end
end

any = Anything.new
any.foo
any.bar
any.a
```

2) Sintaxis

Corrige el siguiente código para instanciar el objeto y generar la salida de manera correcta.

```
class Car(model, year)
  @model = model
  @year = year
end

car = Car.new('Camaro', 2016)
puts "Mi auto favorito es un #{car.model} del año #{car.year}!"
```

3) Sintaxis

Copia el siguiente código y ejecútalo. Luego corrige los errores para poder imprimir el nombre de la tienda.

```
class Store
  def initialize(name)
    name = @name
  end
end

store = store.new('Tienda 1')
puts store.name
```

4) Constructor con argumentos

Crea una clase llamada *Dog* cuyo constructor reciba como argumento un *hash* con la siguiente estructura:

```
propiedades = {nombre: 'Beethoven', raza: 'San Bernardo', color: 'Café'}
```

Instanciar un nuevo perro a partir de las propiedades contenidas en el *hash*. Luego generar un método llamado **ladrar** que, mediante interpolación, imprima *"Beethoven está ladrando!"*

5) Traductor entero a código Morse

Se tiene la clase *Morseable* que contiene un método de instancia *generate_hash* los datos de traducción de número entero a código morse.

```
class Morseable
  def initialize(number)
    @number = number
  end

  def generate_hash(number)
    # Esto es una aberración y debe ser refactorizado!
    h = '-----' if number == 0
    h = '-.----' if number == 1
    h = '..---' if number == 2
    h = '...--' if number == 3
    h = '....-' if number == 4
    h = '.....' if number == 5
    h = '-.....' if number == 6
    h = '--....' if number == 7
    h = '---...' if number == 8
    h = '----.' if number == 9
  end

  def to_morse
    self.generate_hash(@number)
  end
end

m = Morseable.new(3)
print m.to_morse
```

- Refactorizar el código del método de instancia *generate_hash* para que los datos estén contenidos en un *hash* donde **los números serán las claves y la traducción los valores**. El método *generate_hash* además debe retornar la traducción del número recibido como argumento.

6) Objetos y archivos

Se tiene un archivo llamado *catalogo.txt* que contiene los nombres de los productos que ofrece una tienda junto con los precios de las tallas L, M, S y XS.

```
Polera, 10990, 7990, 4990, 2990
Jeans, 14990, 10990, 7990, 5990
Poleron, 12990, 8990, 5990, 3990
Chaleco, 11990, 8990, 6990, 4990
Parka, 19990, 14990, 11990, 9990
```

El siguiente código define la clase *Product* y además realiza la lectura del archivo.

```

class Product
  def initialize(name, large, medium, small, xsmall)
    @name = name
    @large = large
    @medium = medium
    @small = small
    @xsmall = xsmall
  end
end

products_list = []
data = []
File.open('catalogo.txt', 'r') { |file| data = file.readlines}
data.each do |prod|
  ls = prod.split(' ', ' ')
  products_list << Product.new(ls[0], ls[1], ls[2], ls[3], ls[4])
end

puts products_list

```

Se pide:

- Optimizar el código implementando el operador *splat* al momento de instanciar los productos
- Generar un método que permita calcular el promedio de precio por producto.

7) Objetos y archivos

Utilizando el mismo archivo del ejercicio anterior:

La tienda desea generar un nuevo catálogo que **no incluya** el último precio correspondiente a cada producto debido a que ya no comercializa productos de talla XS.

Se pide **generar un método** que reciba como argumento los datos del archivo *catalogo.txt* y luego imprima el nuevo catálogo solicitado en un archivo llamado *nuevo_catalogo.txt*