

Prueba de ruby

El objetivo de esta prueba es demostrar que se pueden resolver problemas básicos utilizando objetos, módulos y procs en ruby.

Los problemas a evaluar en esta prueba son del tipo abrir un archivo, procesar información, guardarla dentro de objetos, y luego utilizar los métodos de estos objetos para procesar la información.

Los objetos que se piden suelen ser de distintos tipos y estar agrupados en módulos.

Buena suerte !!

Comienza la prueba

1

Se tiene el siguiente código que no funciona, para arreglarlo se pide que método1 funcione como método de instancia.

```
class T
  def method1()
  end
end

T.method1
```

2

Se tiene el siguiente código que no funciona, para arreglarlo se pide que método1 funcione como método de clase.

```
class T
  def method1()
  end
end

T.method1
```

3

Se tiene el siguiente código, se pide convertir la clase en un módulo.

```
class Prueba
  @@A = 5
  def self.A
    @@A
  end
end
```

4

Se tiene la siguiente clase.

```
class Complejo
  def initialize(x, y)
    @x = x
    @y = y
  end

  def doble(complejo)
    Complejo.new(@x + complejo.x, @y + complejo.y)
  end
end
```

Pero tiene un problema menor, arreglarlo.

5

Se tienen las siguientes clases:

```
class T
  def method1
  end
end

class Q
end
```

Se pide que:

- Q herede de T
- Q al hacer inicializado llame a method1
- method1 debe devolver un número al azar
- El método devuelto debe ser guardado en una variable de instancia de Q

6

Dado el siguiente código:

```
module Priceable
  attr_accessor :price
  def discount_price(discount)
    @price - @price * discount
  end
end

class Product
  include Priceable
  def initialize(price)
    @price = price
  end
end
```

Crear el módulo Stockable, este módulo introduce la variable stock y debe tener un método llamado has_stock?, que devuelva true si el stock es mayor que cero.

7

Dado el siguiente código

```
class Car
  @@t_instances = 0
  @@q_instances = 0
end

class T
end

class Q
end
```

Modificar T, Q y Car para que Car pueda contar las instancias respectivas de T y Q, además crear métodos dentro de car para obtener el número de instancias de cada uno.

Para probar Crear 20 instancias de T y 25 de Q dentro de Main.