Sintaxis del algebra de traits

* Definir un trait:

CreadorTrait.definirTrait(nombreTrait, bloqueMetodosTrait)

Ejemplo:

CreadorTrait.definirTrait('MiTrait') do

agregarMethod :saludar do

'Hola! ' + self.nombre

end

agregarMethod :chau do |num|

5+ num

end

end

* Para poder usar un trait:

Ejemplo:

class Persona

uses MiTrait, EstrategiaPorCorte.new() {|resultado| resultado>15}

attr\_accessor :nombre

def initialize(nombre)

self.nombre = Juan

end

end

* Suma de traits

Ejemplo:

CreadorTrait.definirTrait('OtroTrait') do

agregarMethod :saludar do

'Hola otro trait!'

end

agregarMethod :chau do |num|

7+ num

end

end

class Persona

uses MiTrait + OtroTrait, EstrategiaPorCorte.new() {|resultado| resultado>15}

attr\_accessor :nombre

def initialize(nombre)

self.nombre = Juan

end

end

* Resta de selectores

Ejemplo:

class PersonaRestaMetodo

uses MiTrait + (MiOtroTrait - :chau) , EstrategiaTodosLosMensajes.new()

attr\_accessor :nombre

def initialize(nombre)

self.nombre = nombre

end

end

* Renombrar selectores

Ejemplo:

class PersonaConAlias

uses MiTrait.<<(:saludar, :unSaludo) , EstrategiaTodosLosMensajes.new()

attr\_accessor :nombre

def initialize(nombre)

self.nombre = nombre

end

end

* Resolución de conflictos

Las estrategias se pasan como parámetro en el uses

class Persona

uses MiTrait.<<(:saludar, :unSaludo) , EstrategiaTodosLosMensajes.new()

end

class Persona

uses MiTrait.<<(:saludar, :unSaludo) , EstrategiaExcepcion.new()

end

class Persona

uses MiTrait.<<(:saludar, :unSaludo) , EstrategiaPorFuncion.new() { |n1,n2| n1+n2 }

end

class Persona

uses MiTrait.<<(:saludar, :unSaludo) , EstrategiaPorCorte.new() { |resultado| resultado>0 }

end