Implement Openstruct by Metaprogramming way

Wowwww I am experiencing the magic of Metaprogramming what beauty you sweety!!!! .please don’t get embarrassed that I am going to write a post about my girl friend yes absolutely right Metaprogramming ruby is my my girl friend ,what a beauty it has if I start explain about its beauty you will also become a great fan of it.lets implement OpenStruc t using Ruby Metaprogramming.

OpenStruct kya hei ??

OpenStruct is a data structure close to hash ,it allows to define key and values and access like a hash. If you call the key with object as a receiver corresponding value will be returned.

require ‘ostruct’

str = OpenStruct.new

str.name=”Vinod”

str.age=”23”

puts str.age #=> 23

puts str.name #=> “Vinod”

Here we will try to implement the functionality of Openstruct using Ruby Metaprogramming

class MyStruct

def initialize

@my\_hash={} #empty hash

end

def method\_missing(name,\*args)

new\_key = name.to\_s #convert method name to string

if new\_key =~ /=$/ #check whether ‘=’ pattern is available in method name

@my\_hash[new\_key.chop] = args[0] #assign value to the corresponding key in a hash

else

@my\_hash[new\_key] #assign empty value to hash

end

end

end

vcet=MyStruct.new

vcet.nick = "vinod"

puts vcet.nick

vcet.age = 23

puts vcet.age

Explanation:

\*create an empty hash named @my\_hash = {}

\*key name is gonna be dynamic so we need to implement method\_missing() to handle the unknown method and the arguments(name will contain method name and args[0] will contain the value)

\*if the method name has a pattern of ‘=’ then method name will be passed as a key in to the hash and value assigned to method will be value of the key in hash.

\*if the method is not in such pattern then method name will be passed as a key in to hash and value will be null.

\*here initialize ‘vcet’ object for MyStruct class and call vcet.nick=”Vinod”

\*here nick= will be the method called and method\_missing() will be called because it is an unknown method and (nick= and “Vinod”) will be passed as an argument.

\*hash will be defined as my\_hash[nick]=”Vinod”.