Linguagem Lua

Nome: Luciano Januario Barbosa Filho

Origem

- Lua é uma linguagem de programação dinâmica desenvolvida por um time de desenvolvedores do instituto Tecgraf da PUC-Rio em 1993, no início para ser usada em um projeto da Petrobras.

- Devido a sua facilidade de aprendizado, sintaxe intuitiva e eficiência passou a ser utilizada em várias aplicações industriais e jogos.

Influências

- Linguagens que influenciaram na sua criação foram Icon e Pascal, em sua concepção, e Python pela sua facilidade de utilização e aprendizado.
- Em um artigo publicado no Dr. Dobb's Journal, os criadores da linguagem Lua afirmaram que Lisp e Scheme tiveram grande influência no desenvolvimento da tabela, essa sendo sua principal estrutura de dados de Lua.

Classificação

Dinâmica e Interpretada:

- Lua é uma linguagem de tipagem dinâmica e compilada que utiliza um ambiente que os seus recursos são suportados e utilizados em tempo de execução.
- A Linguagem oferece um ambiente de interpretação REPL (Read-eval-print loop) que permite o usuário usufruir das características dinâmicas da linguagem.

Metatabelas e Metamétodos

- Lua é uma linguagem que não é orientada a objetos, no entanto, esta é orientada a tabelas, metatabelas e metamétodos, estes que emulam o funcionamento de classes e objetos dentro da linguagem.
- Metatabelas controlam o comportamento de estruturas de dados na linguagem, esta utiliza de chaves, que são derivadas a partir de nomes dos eventos; o os valores correspondentes a estes são chamados de metamétodos.
- Um exemplo é o evento "add", o seu metamétodo é a função que realiza a adição.

Exemplo do uso de metatabelas:

Operações com Vetores A(X, Y) e B(X, Y):

```lua
local Vetor2_meta = {
    __add = function(a, b)
        return Vetor2 (a.x + b.x, a.y + b.y)
    end,

    __sub = function(a, b)
        return Vetor2 (a.x - b.x, a.y - b.y)
    end,

    __mul = function (a, b)
        return Vetor2 (a.x * b.x, a.y * b.y)
    end,
    __call = function (self, ...)
        print("{" .. self.x .. "," .. self.y .. "}")
    end,
    __tostring = function(self)
        return "{" .. self.x .. "," .. self.y .. "}"
    end, -- Caso não houvesse a sobrescrição do metametodo iria ser retornado o endereço da table.
}

function Vetor2(x, y)
    local v = {x = x or 0 , y = y or 0}
    setmetatable(v, Vetor2_meta)
    return v
end
```

```lua
function Vetor2(x, y)
    local v = {x = x or 0 , y = y or 0}
    setmetatable(v, Vetor2_meta)
    return v
end

local a = Vetor2(5, 7)
local b = Vetor2(10, 3)
local soma = Vetor2(0, 0)
local subtracao = Vetor2(0, 0)

soma = a + b -- soma de vetores
subtracao = a - b -- subtração de vetores

local produto = Vetor2(0,0)

produto = a * b -- Produto de vetores

print("Vetor soma dos vetores A e B: ", soma)
print("Vetor subtração dos vetores A e B: ", subtracao)
print("Vetor produto escalar dos vetores A e B: ", produto)
```

Output:

Vetor soma dos vetores A e B: {15,10}

Vetor subtração dos vetores A e B: {-5,4}

Vetor produto escalar dos vetores A e B: {50,21}