[TOC] #

0

emmylua emmylua https://emmylua.github.io/zh_CN/annotation.html emmylua : https://emmylua.github.io/annotation.html

1

C++ struct struct struct

C++

```
//    struct
struct People{
  string name;
  string address;
  int age;
};

//   People
PeoPle one; //   one    name address age
```

Lua lua

Lua

```
---@class People
---@field name string
---@field address string
---@field age int

---@type People
local one;     --   one    name address age
```

avatar

* —(???) * —(???) * —(???)

lua —(???)

2

- 1

  —(???) TYPE{|OTHER_TYPE} (???)

- 2 C++ struct

  —(???) TYPE[: PARENT_TYPE {, PARENT_TYPE}] (???)

- 3 class

  —(??? public|protected|private) field_name TYPE{|OTHER_TYPE} (???)

- 4

  —(???) param_name TYPE{| OTHER_TYPE} (???)

- 5

  —(???) TYPE{| OTHER_TYPE} (???)

- 6

  —(???) new_type TYPE{ | OTHER_TYPE}

- 7

  —(???) G1 [: PARENT_TYPE] [, G2 [: PARENT_TYPE]]

3

3.1 type

`type`

-

  —(???) TYPE{|OTHER_TYPE} (???)

* TYPE OTHER_TYPE lua * (???) comment

* {} 0 * [] 0 1

Lua : * nil * boolean bool true false * number * string: * function * userdata * thread Lua * any * table table * void

1.

```
---@class People
---@field name string
---@field address string
---@field age int

---@type People @instance of People
local one = {}
```

- Lua |

```
---@type People | number @ People      number
local one
```

2.

```
---@type People @global variable type
global_people = {}
```

3. table

```
local team = {}
---@type People @    People
team.one = getPeople()
```

* 1

```
---@type string, number @one string  @two number
local one, two
```

- 2

```
---@type string @one string
---@type number @two number
local one, two
```

3.2

TYPE[]

-

—(???) **TYPE**[] (???)

-

```
---@class People
---@field name string
---@field address string
---@field age int

---@type People[]
local people_list = {}

-- one   People
local one = people_list[1]

for i, two in pairs(people_list) do
    -- i    number
    -- two    People
end
```

-

avatar

avatar

3.3

table

-

—(???) table<VALUE_TYPE, VALUE_TYPE> (???)

-

```lua
---@type table<string, People> @ People   string key, value People
local dict = {}

-- one  People
local one = dict['key']

for key, two in pairs(dict) do
    -- key  string
    -- two  People
end
```

-

avatar

-

  table key value **—(???) table (???)**

3.4 fun

`@fun`

- :

  —(???) fun(param1: PARAM_TYPE1, param2: PARAM_TYPE2) {:RETURN_TYPE1, RETURN_TYPE2} (???)

-

  1) `lua ---@type fun() @ local fun1`

  2 `lua ---@type fun(param1 : string) @ local fun2`

  3 `lua ---@type fun(param1 : string | number, param2 : number) @ strin`

  4) `lua ---@type fun(param1 : string, param2 : number) : number @ stri`

  5) `lua ---@type fun(param1 : string, param2 : number) : number, string | numb`

3.5 class

`@class C++ struct` .

-

  —(???) TYPE[: PARENT_TYPE {, PARENT_TYPE}] (???)

- ,

1 `lua ---@class People @ People ---@field name string ---@field address st`

2 `lua ---@class Man : People @ Man ---@field phone number` avatar

3 `lua ---@class Man : People, Team @ Man ---@field attr string @`

-

  1 class class @type `lua ---@class People @ People ---@field name str`

  2 class class field “‘lua —(???) People (???) —(???) name string —(???) address string —(???) age int

  —(???) People local one – one People

  one.attr = 1 – People attr function one.fun1() – People fun1 end

  —(???) People local two “‘ avatar

3.6 field class

`@field` `class`

- —(??? **public|protected|private) field_name TYPE{|OTHER_TYPE} (???)**
- field @class :

  ```lua
  ---@class People @ People
  ---@filed public name string @    class  name
  local one = {} --     class     People  one name
  ```

3.7 param

`@param`

- —(???) **param_name TYPE{| OTHER_TYPE} (???)**
-

  1

  ```lua
  ---@param people People @  People
  ---@param age number @
  ---@param info string | number @   string   number
  local function setInfo(people, age, info)
  end
  ```

  avatar

  2

  ```lua
  ---@param info string | number @   string   number
  setCallback(function(info)
  ```

```lua
end)
```

3

```lua
---@param k number @
---@param info People @  People
for k, info in pairs(info_table) do
end
```

3.8 return

`@return`

- :

  1) —(???) TYPE{| OTHER_TYPE} (???)

  2 —(???) TYPE{| OTHER_TYPE}, TYPE{| OTHER_TYPE} (???) (???)

-

  1 “‘lua —(???) People (???) local function create()

  end “‘ avatar

  2 “‘lua —(???) number (???) —(???) People (???) local function getInfo()

  end “‘

  3 “‘lua —(???) number, string (???) (???) —(???) People (???) local function getInfo()

  end “‘

3.9 alias

`@alias` `C++` `typedef`

-

  —(???) new_type TYPE{ | OTHER_TYPE}

-

  1 “‘lua —(???) NewOne string | number (???)

  —(???) NewOne (???) local one “‘

  2 “‘lua —(???) Handler fun(one: string, two : number) : void (???)

  —(???) handler Handler function addHandler(handler) end “‘

3.10 generic

~~~
 @generic        C++        C++
```C++
// T
template <typename T>
T const& Max (T const& a, T const& b)
{
    return a < b ? b:a;
}
```
~~~

- **—(???) G1 [: PARENT_TYPE] [, G2 [: PARENT_TYPE]]**

- 1)   “‘lua —(???) T —(???) param1 T —(???) T local function test_one(param1) end

  —(???) People local one

  local value = test_one(one) – one People test_one People value People

  `![avatar](https://raw.githubusercontent.com/yinfei8/LuaHelper/master/images/annotate/g`

  ~~~
  2 )
  ```lua
  ---@generic T : Transport, K
  ---@param param1 T
  ---@param param2 K
  ---@return T
  local function test_two(param1, param2)
      -- todo
  end
  ~~~

4

~~~
```lua
-- 1  type
---@class People @ People
---@field name string @
---@field address string @
---@field age int @

---@type People
local one           --      People       one.    People
print(one.address)

-- 2
--      name         People
~~~

```
---@param name string @
---@return People @     People
function create_people(name)
end

--       People   two    People
local two = create_people("yinfei")

--     two.    People
print(two.age)

-- 3
--         two   People   three  three   People
local three = two

--     three.    People
print(three.age)

-- 4    lua
---@class Car @ Car
---@field name string
---@field color string

---@type Car
local car   --   car  People

car.age = 1    --    car attr     car  Car            Car  age
function car.run()  --    car fun1     car  Car            Car  run
end

--  Car  4  name color attr run
---@type Car
local car1
--     car1.    Car   name color attr run
print(car1.age)

-- 5
---@class Man : People @ Man    People
---@field height number @man    height
local four      --  four   class       four   four   Man

--   four.    People      height
print(four.height)

-- 6
-- test_one     param1
---@generic T
---@param param1 T
```

~~~
---@return T
local function test_one(param1)
end

---@type People
local one

local value = test_one(one)  -- one  People  test_one    People    value  People
--    value.     People
print(value.age)


```
~~~