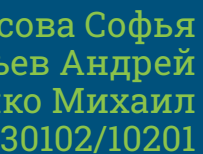




Lua

5. ФУНКЦИИ



Корпусова Софья
Афанасьев Андрей
Лутченко Михаил
5030102/10201

Введение

- Множественные результаты
- Вариативные функции
- Именованные аргументы
- Замыкания
- Неглобальные функции
- Корректные хвостовые вызовы

Множественные результаты

- Можно возвращать несколько значений \Rightarrow множественное присваивание
- Lua всегда приводит количество результатов функции к обстоятельствам ее вызова



```
pres_5.lua
1  s, e = string.find("hello Lua users", "Lua")
2  print(s, e)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

sofakorpusova@Lenovo:~/lua_scripts$ lua pres_5.lua
7      9
```

Множественные результаты

```
4  function    maximum (a)
5      local   mi  =  1
6      local   m   =  a[mi]
7      for i   =  1, #a do
8          if a[i] > m then
9              mi = i; m = a[i]
10         end
11     end
12     return  m,  mi
13 end
14 print(maximum({8,10,23,12,5}))
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● sofakorpusova@Lenovo:~/lua_scripts\$ lua pres_5.lua

23

3

Множественные результаты

```
16  function foo0() end
17  function foo1() return "a" end
18  function foo2() return "a", "b" end
19
20  x, y = foo2()
21  print(x, y)
22  x = foo2()
23  print(x)
24  x, y, z = 10, foo2()
25  print(x, y, z)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● sofakorpusova@Lenovo:~/lua_scripts\$ lua pres_5.lua

a b

a

10 a b

Множественные результаты

- Значений меньше, чем нужно \Rightarrow nil

```
27  x, y = foo0()
28  print(x, y)
29  x, y = foo1()
30  print(x, y)
31  x, y, z = foo2()
32  print(x, y, z)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_5.lua
nil      nil
a        nil
a        b      nil
```

Множественные результаты

- Вызов функции, который не является последним элементом в списке, всегда дает ровно один результат

```
34  x, y = foo2(), 20
35  print(x, y)
36  x, y = foo0(), 20, 30
37  print(x, y)
38  print(foo2())
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_5.lua
a      20
nil    20
a      b
```

Множественные результаты

- Получение только нужных значений

```
50  function foo3() return 1, 2, 3 end
51  local _, _, c = foo3()
52  print(_, c)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_5.lua
2      3
```


Вариадические функции

- Переменное число аргументов

```
40  function add (...)
41      local s = 0
42      for i, v in ipairs{...} do
43          s = s + v
44      end
45      return s
46  end
47  print(add(3, 4, 10, 25, 12))
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_5.lua
54
```

Вариадические функции

- Выражение с переменным числом аргументов

```
function foo (a, b, c)
```

```
function foo (...)
```

```
    local a, b, c = ...
```

Именованные аргументы

- Механизм передачи параметров - позиционный

```
56  -- недопустимый код
57  --rename(old="temp.lua", new="temp1.lua")
58
59  --rename{old="temp.lua", new="temp1.lua"} специальный синтаксис
60
61  function rename (arg)
62  |      return os.rename(arg.old, arg.new)
63  end
```

Замыкания

Замыкание—это функция вместе со всем, что ей нужно для правильного доступа к нелокальным переменным

```
66  function newCounter()  
67      local i = 0  
68      return function() --анонимная функция  
69          i = i + 1  
70          return i  
71      end  
72  end  
73  c1 = newCounter()  
74  print(c1(), c1())  
75  
76  c2 = newCounter()  
77  print(c2(), c1(), c2())
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● sofakorpusova@Lenovo:~/lua_scripts\$ lua pres_5.lua

1 2

1 3 2

Неглобальные функции

```
79  Lib = {}
80  Lib.foo = function (x, y) return x + y end
81  Lib.goo = function (x, y) return x - y end
82  --[[
83  ИЛИ конструктор
84  Lib = {
85      Lib.foo = function (x, y) return x + y end,
86      Lib.goo = function (x, y) return x - y end
87  }
88  ИЛИ специальный синтаксис
89  Lib = {}
90  function Lib.foo (x, y) return x + y end
91  function Lib.goo (x, y) return x - y end
92  ]]
93  print(Lib.foo(2, 3), Lib.goo(2, 3))
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● sofakorpusova@Lenovo:~/lua_scripts\$ lua pres_5.lua

5

-1

Неглобальные функции

- Функция в локальной переменной - локальная
- Удобно для пакетов:

```
local f = function (<параметры>)
    <тело>
end
local g = function (<параметры>)
    <какой-нибудь код>
    f()    -- 'f' здесь видима
    <какой-нибудь код>
end
```

Корректные хвостовые вызовы

- Хвостовой вызов - goto, выглядящий как вызов функции

```
function f (x) return g(x) end
```

- Хвостовым считается лишь вызов вида return func(args)

```
return x[i].foo(x[j] + a * b, i + j)
```

Дополнительные материалы

1. [Компиляция, выполнение и ошибки \(глава 8\)](#)
2. [Модули и пакеты \(глава 15\)](#)
3. [Библиотеки стандартных функций](#)
4. [Объектно-ориентированное программирование \(глава 16\)](#)
5. [Работа с файлами](#)
6. [Lua для игр и не только](#)
7. [Lua Tutorials](#)
8. [Официальный сайт Lua](#)

Литература

- Програмирование на языке Lua - Роберту Иерузалимски
- Возврат значений из функции