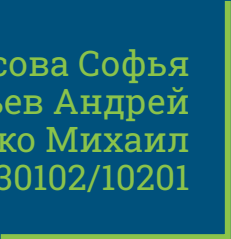




# Lua

## 4. Операторы



Корпусова Софья  
Афанасьев Андрей  
Лутченко Михаил  
5030102/10201

# Введение

---

- Стандартный набор операторов
  - Присваивание
  - управляющие структуры
  - Вызовы процедур
- Менее распространённые операторы

# Присваивание

- Множественное присваивание
- Сначала вычисляет значения, затем присваивает

pres\_4.lua

```
1  local x = 3
2  local a, b = 10, 2 * x
3  print(a, b)
4  a, b = b, a
5  print(a, b)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

sofakorpusova@Lenovo:~/lua\_scripts\$ lua pres\_4.lua

```
10      6
6       10
```

# Присваивание

- Количество значений приводится к количеству переменных
- Не быстрее одиночных присваиваний

```
7  local a, b, c = 0, 1
8  print(a, b, c)
9  a, b = a + 1, b + 1, b + 2 --значение b+2 игнорируется
10 print(a, b)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● sofakorpusova@Lenovo:~/lua\_scripts\$ lua pres\_4.lua

0 1 nil

1 2

# Управляющие конструкции: `if then else`

- Проверяет условие и выполняет одну из двух частей
- Часть с `else` необязательная

```
12  local a, b = -1, 5
13
14  if a < 0 then a = 0 end
15  print(a)
16
17  if a > b then print(a) else print(b) end
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

• `sofakorpusova@Lenovo:~/lua_scripts$ lua pres_4.lua`

0

5

# Управляющие конструкции: `if then else`

- `elseif`

```
19  local a, b = 3, 5
20  local op = io.read()
21  if op == "+" then
22      r = a + b
23  elseif op == "-" then
24      r = a - b
25  elseif op == "*" then
26      r = a * b
27  elseif op == "/" then
28      r = a / b
29  else
30      error("invalid operation")
31  end
32  print(r)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● sofakorpusova@Lenovo:~/lua\_scripts\$ lua pres\_4.lua

\*

15

# Управляющие конструкции: **while**

- Предусловие
- Можно использовать **break**

```
34  a = {3, 5, 8, -6, 5}
35  i = #a
36  while i > 0 do -- ищем в массиве отрицательное значение
37      |   if a[i] < 0 then break end -- если найдено, прерываем цикл
38      |   i = i - 1 -- иначе переходим к следующему элементу
39  end
40  if i > 0 then
41      |   print("Индекс отрицательного значения: "..i)
42  else
43      |   print("Массив не содержит отрицательных значений")
44  end
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

• **sofakorpusova@Lenovo:~/lua\_scripts\$** lua pres\_4.lua

Индекс отрицательного значения: 4

# Управляющие конструкции: **repeat**

- Постусловие - тело выполнится хотя бы 1 раз
- Можно использовать **break**

```
46  -- суммируем значения массива a, пока сумма не превысит 10
47  a = {3, 2, 5, 7, 9}
48  i = 0; sum = 0
49  repeat
50      i = i + 1
51      sum = sum + a[i]
52  until sum > 10
53  print("Сложено "..i.." элемента. Сумма равна "..sum)
--
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

- **sofakorpusova@Lenovo:~/lua\_scripts\$** lua pres\_4.lua  
Сложено 4 элемента. Сумма равна 17



# Управляющие конструкции: числовой **for**

- **for** **var** = **exp1**, **exp2**, **exp3** **do**

... -- тело цикла

**end**

- **exp1**, **exp2** и **exp3** вычисляются всего один раз
- Управляющая переменная - локальная
- Нельзя изменять значение управляющей переменной
- Можно использовать **break**

# Управляющие конструкции: числовой **for**

```
55  a = {3, 5, 8, -6, 5}
56  for i = 1,#a do -- ищем в массиве отрицательное значение
57      if a[i] < 0 then -- если найдено...
58          index = i -- сохраняем индекс найденного значения...
59          break -- и прерываем цикл
60      end
61  end
62  print("Индекс отрицательного значения: "..index)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● **sofakorpusova@Lenovo:~/lua\_scripts\$** lua pres\_4.lua

Индекс отрицательного значения: 4

# Управляющие конструкции: общий **for**

- **for** **var1, var2, ..., varN** **in** **<explist>** **do**  
    ... -- тело цикла  
**end**
- **var1, var2, ..., varN** локальны и их нельзя менять

# Управляющие конструкции: общий **for**

```
64 Days = {"Sunday" = 1, "Monday" = 2, "Tuesday" = 3, "Wednesday" = 4,  
65 "Thursday" = 5, "Friday" = 6, "Saturday" = 7}  
66  
67 for key, val in pairs(Days) do  
68     print("key == "..key.."; val == "..val)  
69 end
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

● **sofakorpusova@Lenovo:~/lua\_scripts\$** lua pres\_4.lua

key == Friday; val == 6

key == Thursday; val == 5

key == Sunday; val == 1

key == Wednesday; val == 4

key == Monday; val == 2

key == Tuesday; val == 3

key == Saturday; val == 7

# Управляющие конструкции: общий **for**

- **for** *key, val* **in** *next, t* **do**  
    *print("key == "..key.."; val == "..val)*  
**end**
- Несколько итераторов:
  - **io.lines** - строки файла
  - **ipars** - элементы последовательности
  - **string.gmatch** - слова внутри строки

# Операторы `break` и `return`

- `break` прерывает цикл, передавая управление конструкции сразу за ним
- `return` возвращает результаты из функции, всегда стоит в конце блока
- `do return end`

# Оператор `goto`

- `goto name`

...

`::name::`

- Нельзя прыгнуть внутрь блока
- Нельзя выпрыгнуть из функции
- Нельзя переместиться внутрь области видимости локальной переменной
- Моделирование конструкции из другого языка

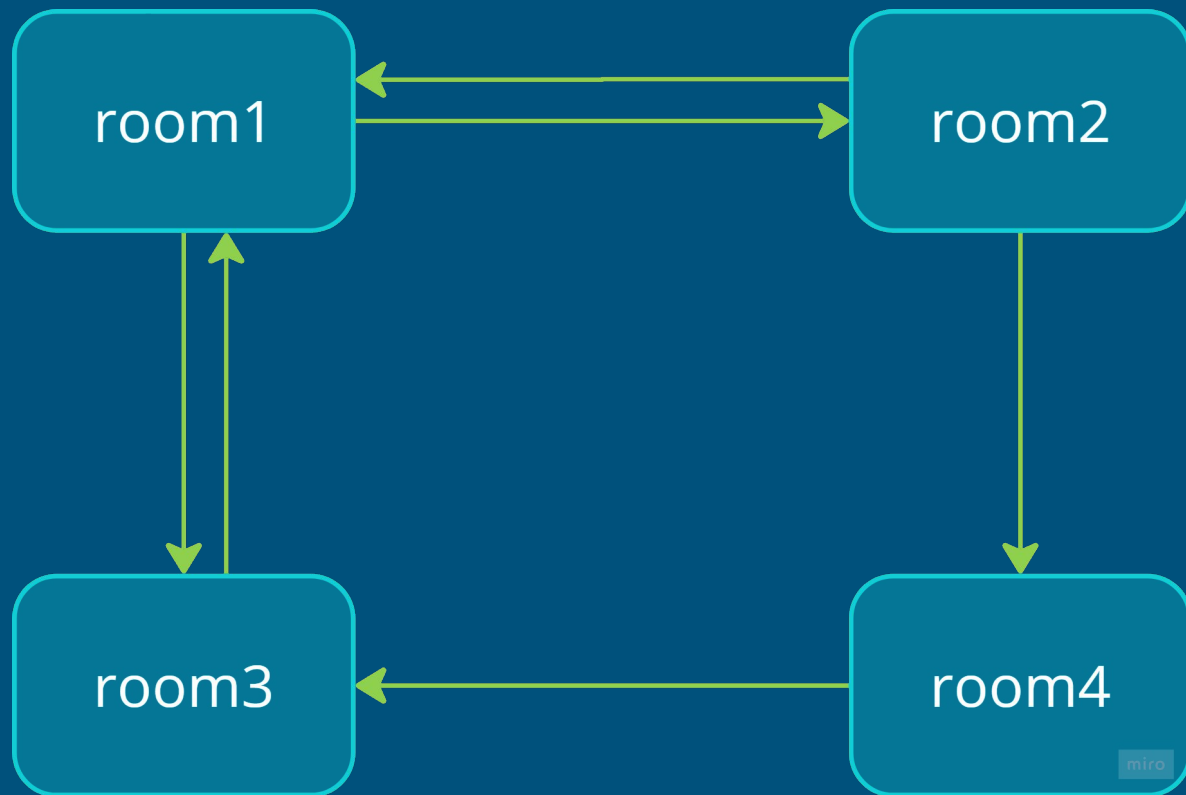
# Оператор `goto`

`continue` — это просто безусловный переход к метке в конце блока цикла. `redo` - переход к началу блока.

```
for while some_condition do
  ::redo::
  if some_other_condition then goto continue
  else if yet_another_condition then goto redo
end
<какой-нибудь код>
::continue::
end
```



# Оператор `goto` - лабиринт



# Оператор `goto` - лабиринт

```
72 goto room1 -- начальная комната
73 ::room1:: do
74     local move = io.read()
75     if move == "south" then goto room3
76     elseif move == "east" then goto room2
77     else
78         print("invalid move")
79         goto room1 -- остаемся в этой же комнате
80     end
81 end
82
83 ::room2:: do
84     local move = io.read()
85     if move == "south" then goto room4
86     elseif move == "west" then goto room1
87     else
88         print("invalid move")
89         goto room2
90     end
91 end
```

```
93 ::room3:: do
94     local move = io.read()
95     if move == "north" then goto room1
96     elseif move == "east" then goto room4
97     else
98         print("invalid move")
99         goto room3
100    end
101 end
102
103 ::room4:: do
104     print("Congratulations, you won!")
105 end
```

```
● sofakorpusova@Lenovo:~/lua_scripts$ lua pres_4.lua
east
south
Congratulations,      you      won!
```

# В следующей презентации

---

- **Функции**
  - Множественные результаты
  - Вариадические функции
  - Именованные аргументы
- **Замыкания**
- **Неглобальные функции**
- **Корректные хвостовые вызовы**

# Литература

---

- Операторы в Lua
- Программирование на языке  
Lua - Роберту Иерузалимски

