

Защита лабораторной работы № 3

Управляющие структуры.

Компьютерный практикум по статистическому анализу данных

Работу Выполнил:
Саинт-Амур Измаэль
Группа: НПИбд-01-20



Цель работы

Основная цель работы — освоить применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

Для различных операций, связанных с перебором индексируемых элементов структур данных, традиционно используются циклы while и for.

Синтаксис while

while <условие>

<тело цикла>

end

```
In [1]: # пока n<10 прибавить к n единицу и распечатать значение:
```

```
n = 0
while n < 10
n += 1
println(n)
end
```

```
1
2
3
4
5
6
7
8
9
10
```

```
In [2]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
```

```
Out[2]: 5-element Vector{String}:
```

```
"Ted"
"Robyn"
"Barney"
"Lily"
"Marshall"
```

Такие же результаты можно
получить при использовании цикла
for.

Синтаксис for
for <переменная> in <диапазон>
<тело цикла>
end

```
In [4]: for n in 1:2:10
        println(n)
        end
```

```
1
3
5
7
9
```

```
In [5]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
```

```
Out[5]: 5-element Vector{String}:
         "Ted"
         "Robyn"
         "Barney"
         "Lily"
         "Marshall"
```

```
In [6]: for friend in myfriends
        println("Hi $friend, it's great to see you!")
        end
```

```
Hi Ted, it's great to see you!
Hi Robyn, it's great to see you!
Hi Barney, it's great to see you!
Hi Lily, it's great to see you!
```

Довольно часто при решении задач требуется проверить выполнение тех или иных условий. Для этого используют условные выражения. Синтаксис условных выражений с ключевым словом:

```
if <условие 1>  
  <действие 1>  
elseif <условие 2>  
  <действие 2>  
else  
  <действие 3>  
end
```

```
[n [11]: # используем `&&` для реализации операции "AND"  
# операция % вычисляет остаток от деления  
N = 9  
if (N % 3 == 0) && (N % 5 == 0)  
println("FizzBuzz")  
elseif N % 3 == 0  
println("Fizz")  
elseif N % 5 == 0  
println("Buzz")  
else  
println(N)  
end
```

Fizz

```
[n [12]: x = 5  
y = 10  
(x > y) ? x : y
```

```
out[12]: 10
```

Функции

Julia дает нам несколько разных способов написать функцию. Первый требует ключевых слов `function` и `end`:
Вызов функции осуществляется по её имени с указанием аргументов

```
In [13]: function sayhi(name)
          println("Hi $name, it's great to see you!")
          end
```

```
Out[13]: sayhi (generic function with 1 method)
```

```
In [14]: # функция возведения в квадрат:
          function f(x)
              x^2
          end
```

```
Out[14]: f (generic function with 1 method)
```

```
In [15]: sayhi("C-3PO")
          f(42)
```

```
Hi C-3PO, it's great to see you!
```

```
Out[15]: 1764
```

```
In [16]: sayhi2(name) = println("Hi $name, it's great to see you!")
          f2(x) = x^2
```

```
Out[16]: f2 (generic function with 1 method)
```

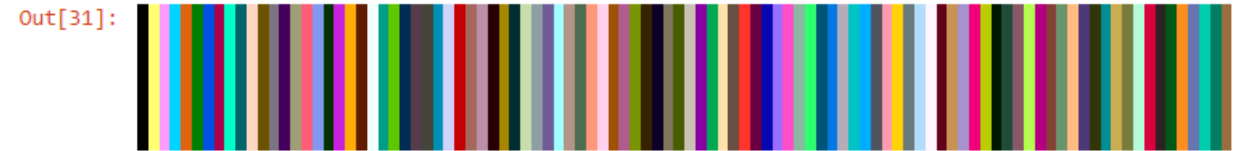
```
In [17]: sayhi3 = name -> println("Hi $name, it's great to see you!")
          f3 = x -> x^2
```


Использование сторонних пакетов, таких как Colors, дополняет функциональность языка Julia и предоставляет дополнительные возможности для работы с данными и визуализации результатов.
Команда `Pkg.add("Colors")` демонстрирует процесс добавления пакетов в Julia.

```
In [30]: Pkg.add("Colors")
```

```
Updating registry at `C:\Users\Scorpion 1.0\.julia\registries\General.toml`  
Resolving package versions...  
No Changes to `C:\Users\Scorpion 1.0\.julia\environments\v1.9\Project.toml`  
No Changes to `C:\Users\Scorpion 1.0\.julia\environments\v1.9\Manifest.toml`
```

```
In [31]: using Colors  
palette = distinguishable_colors(100)
```



```
In [32]: rand(palette, 3, 3)
```



В ходе работы по изучению , на освоение применения циклов, функций и сторонних пакетов для решения задач линейной алгебры и работы с матрицами в языке программирования Julia