

# Лабораторная работа № 3

Управляющие структуры

---

Герра Гарсия Паола Валентина

Российский университет дружбы народов, Москва, Россия

## Информация

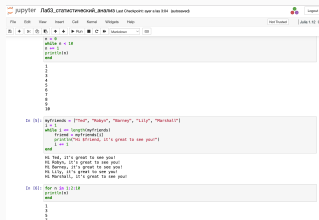
---

- Герра Гарсия Паола Валентина
- студентка
- Российский университет дружбы народов
- 1032225472@pfur.ru

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

1. Используя Jupyter Lab, повторите примеры из раздела 3.2.
2. Выполните задания для самостоятельной работы (раздел 3.4)

# Выполнение лабораторной работы



```

In [0]:
while 4 < 10:
    print(4)
end

1
2
3
4
5
6
7
8
9
10

In [3]: myFriends = ["Tom", "Moby", "Barney", "Lily", "Marshall"]
while 1 < len(myFriends):
    friend = myFriends[0]
    print("The friend, it's great to see you!")
    1 = 1
end
No Tom, it's great to see you!
No Moby, it's great to see you!
No Barney, it's great to see you!
No Lily, it's great to see you!
No Marshall, it's great to see you!

In [3]: for n in 1:2:10
    print(n)
end

1
3
5
7
9

```

Рис. 1: Выполнение примеров с циклами

### Условные выражения

```
In [121]: # использовать "do" для реализации операции "AND"
# операция % вычисляет остаток от деления
N = 5
ST = (M % 3 == 0) AND (M % 5 == 0)
print("Fizzbuzz")
elseif M % 3 == 0
print("Fizz")
elseif M % 5 == 0
print("buzz")
else
print(IN)
end
buzz
```

Рис. 2: Выполнение примеров с условными выражениями

## Сторонние библиотеки (пакеты) в Julia

```
In [17]: import Pkg
          Pkg.add("Example")

          Resolving package versions...
          Installed Example ~ v0.5.5
          Updating `~/julia/environments/v1.12/Project.toml`
          [7876af87] + Example v0.5.5
          Updating `~/julia/environments/v1.12/Manifest.toml`
          [7876af87] + Example v0.5.5
          Precompiling packages...
          948.9 ms / Example
          1 dependency successfully precompiled in 1 seconds. 43 already precompiled.

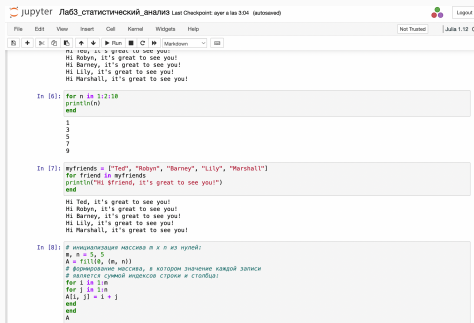
In [19]: Pkg.add("Colors")
          using Colors

          Resolving package versions...
          Project No packages added to or removed from `~/julia/environments/v1.12/Project.toml`
          Manifest No packages added to or removed from `~/julia/environments/v1.12/Manifest.toml`
```

Рис. 3: Выполнение примеров со сторонними библиотеками



# Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook window titled "Лаб3\_статистический\_анализ Last Checkpoint: ayer a las 3:04 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Windows, Help) and a toolbar with icons for file operations, running, and saving. The notebook contains three code cells:

```
ma, row, 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!
Hi Marshall, it's great to see you!
```

```
In [6]: for n in 1:2:10
        println(n)
      end

1
3
5
7
9
```

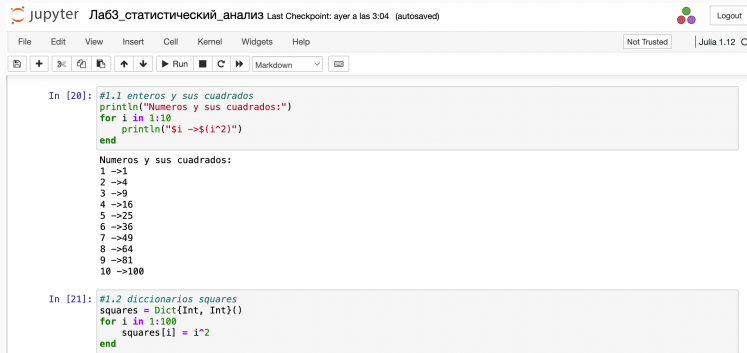
```
In [7]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
        for friend in myfriends
          println("Hi $friend, it's great to see you!")
        end

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!
Hi Marshall, it's great to see you!
```

```
In [8]: # инициализация массива m x n из нулей:
m, n = 5, 5
A = fill{0, (m, n)}
# формирование массива, в котором значение каждой записи
# является суммой индексов строки и столбца:
for i in 1:m
  for j in 1:n
    A[i, j] = i + j
  end
end
A
```

Рис. 4: Задание №1

# Выполнение лабораторной работы



Jupyter Лаб3\_статистический\_анализ Last Checkpoint: ayer a las 3:04 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.12

In [20]:

```
#1.1 enteros y sus cuadrados
println("Numeros y sus cuadrados:")
for i in 1:10
    println("$i ->$(i^2)")
end
```

Numeros y sus cuadrados:

```
1 ->1
2 ->4
3 ->9
4 ->16
5 ->25
6 ->36
7 ->49
8 ->64
9 ->81
10 ->100
```

In [21]:

```
#1.2 diccionarios squares
squares = Dict{Int, Int}{}
for i in 1:100
    squares[i] = i^2
end
```

Рис. 5: Задание №2

```
In [35]: add_one(x) = x+1  
         println(add_one(5)) #6  
  
6
```

Рис. 6: Задание №3

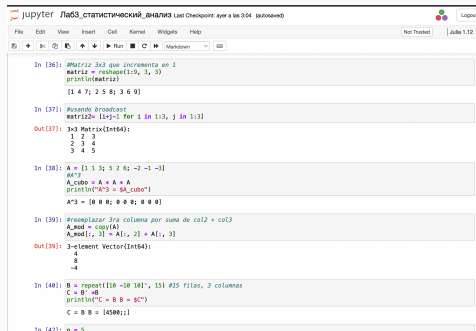
```
In [36]: #Matriz 3x3 que incrementa en 1  
matriz = reshape(1:9, 3, 3)  
println(matriz)
```

```
[1 4 7; 2 5 8; 3 6 9]
```

```
In [37]: #usando broadcast  
matriz2= [i+j-1 for i in 1:3, j in 1:3]
```

```
Out[37]: 3x3 Matrix{Int64}:  
 1  2  3  
 2  3  4  
 3  4  5
```

Рис. 7: Задание №4



The screenshot shows a Jupyter Notebook window titled "jupyter Лаб3\_статистический\_анализ Last Checkpoint: ayer a las 3:04 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar indicating "Not trusted" and "Julia 1.12.0". The notebook contains several code cells:

- In [36]:** A Julia code cell that creates a 3x3 matrix by incrementing values and then reshapes it into a 1x9 vector. The output is a 1x9 matrix: `[1 4 7; 2 5 8; 3 6 9]`.
- In [37]:** A Julia code cell that uses broadcasting to create a 3x3 matrix. The output is a 3x3 matrix: `1 2 3; 2 3 4; 3 4 5`.
- In [38]:** A Julia code cell that creates a 3x3 matrix A, calculates its cube (A^3), and prints the result. The output is a 3x3 matrix: `0 0 0; 0 0 0; 0 0 0`.
- In [39]:** A Julia code cell that creates a 3x3 matrix A, copies it to A\_mod, and then modifies the third column. The output is a 3x1 vector: `4; 8; -4`.
- In [40]:** A Julia code cell that creates a 15x3 matrix B by repeating a 1x3 vector, calculates B\*B, and prints the result. The output is a 15x3 matrix: `4500; 1; 1`.
- In [42]:** A Julia code cell that prints the value of n, which is 5.

Рис. 8: Задание №5

```
In [43]: function outer(x, y, op)
        result = zeros(size(x, 1), size(y, 2))
        for i in 1:size(x, 1)
            for j in 1:size(y, 2)
                result[i,j] = op(x[i,:], y[:,j])
            end
        end
        return result
    end

Out[43]: outer (generic function with 1 method)

In [44]: #version was simple para multiplication
        function outer_simple(A, B)
            return A * B
        end

        #create matrices A1, A2, A3, A4, A5
        m=5
        A1 = [i+j-2 for i in 1:m, j in 1:n]

Out[44]: 5x5 Matrix{Int64}:
 0  1  2  3  4
 1  2  3  4  5
 2  3  4  5  6
 3  4  5  6  7
 4  5  6  7  8
```

Рис. 9: Реализация функции outer()

The screenshot shows a Jupyter Notebook with a single code cell. The code defines a function to generate a Latin square of size 10. It uses a list 'a' to track used numbers in each row and column. The output is a 10x10 grid of numbers from 1 to 10, where each row and column contains all numbers exactly once.

```

In [1]: # Python script to generate a Latin square of size 10
# A Latin square is an n x n grid of numbers 1 to n, such that each row and column contains each number exactly once.
# We will use a list 'a' to keep track of used numbers in each row and column.

def generateLatinSquare(n):
    # Create a list 'a' of size n, initialized with 0s. This list will be used to keep track of used numbers in each row and column.
    a = [0] * n

    # Create a 2D list 's' of size n x n, initialized with 0s. This list will be used to store the Latin square.
    s = [[0] * n for i in range(n)]

    # Generate the Latin square using nested loops.
    for i in range(n):
        for j in range(n):
            # Find a number that is not used in the current row and column.
            num = 1
            while num in a[i] or num in a[j]:
                num += 1

            # Place the number in the current cell.
            s[i][j] = num

            # Mark the number as used in the current row and column.
            a[i] = num
            a[j] = num

    # Print the Latin square.
    for i in range(n):
        for j in range(n):
            print(s[i][j], end=" ")
        print()

# Generate a Latin square of size 10.
generateLatinSquare(10)

```

The output of the script is a 10x10 grid of numbers:

```

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

```

Рис. 10: Проверка работы функции `outer()`

```
In [53]: #Matriz A del sistema
A = [1 2 3 4 5;
      2 1 2 3 4;
      3 2 1 2 3;
      4 3 2 1 2;
      5 4 3 2 1]
y = [7, -1, -3, 5, 17]

#Resolver Ax = y
x = A\y
println("Solucion: x= $x")

#verify
println("A*x deber ser y: $(A*x = y)")

Solucion: x= [-2.00000000000000013, 3.00000000000000027, 4.99999999999999964, 2.00000000000000027, -4.0000000000000001]
A*x deber ser y: *
```

Рис. 11: Решение систему линейных уравнений



# Выполнение лабораторной работы

```
In [54]: using Random
Random.seed!(123)

M = rand(1:10, 6, 10)
N = 4
M_num = 7
K = 75

mayores_N = [count(x -> x > N, M[i,:]) for i in 1:6]

filas_M2 = [i for i in 1:6 if count(x -> x == M_num, M[i,:]) == 2]

pares_columnas = []
for j1 in 1:9
    for j2 in (j1+1):10
        if sum(M[:,j1]) + sum(M[:,j2]) > K
            push!(pares_columnas, (j1, j2))
        end
    end
end

println("Mayores que $N por fila : $mayores_N")
println("Filas con $M_num exactamente 2 veces: $filas_M2")
println("Pares de columnas con suma > $K: $pares_columnas")

Mayores que 4 por fila : [3, 3, 3, 3, 3, 3]
Filas con 7 exactamente 2 veces: [1]
Pares de columnas con suma > 75: Any{Tuple{Int64, Int64}} = [(1, 8), (1, 10), (2, 8), (2, 10), (4, 8), (4, 10), (8, 9), (8, 10), (9, 10)]
```

Рис. 12: Задание №10

## Выполнение лабораторной работы

```
In [55]: sum1 = 0.0
        for i in 1:20
            for j in 1:5
                sum1 += i^4 / (3+j)^4
            end
        end
```

```
In [56]: sum2 = 0.0
        for i in 1:20
            for j in 1:5
                sum2 += i^4 / (3+i)^4
            end
        end
```

```
In [57]: println("Sumatoria 1: $sum1")
        println("Sumatoria 2: $sum2")

Sumatoria 1: 5014.209868467763
Sumatoria 2: 32.98305525107463
```

```
In [58]: sum1_vec = sum([i^4/(3+j)^4 for i in 1:20, j in 1:5])
        sum2_vec = sum([i^4/(3+j)^4 for i in 1:20, j in 1:5])
```

```
Out[58]: 5014.209868467762
```

Рис. 13: Задание №11

В результате выполнения данной лабораторной работы я освоила применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

1. JuliaLang [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://julialang.org/> (дата обращения: 11.10.2024).
2. Julia 1.11 Documentation [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://docs.julialang.org/en/v1/> (дата обращения: 11.10.2024).