

# Лабораторная работа № 3

Управляющие структуры  
Герра Гарсия Паола Валентина

## Содержание

Цель работы .....	1
Задание .....	1
Теоретическое введение .....	2
Выполнение лабораторной работы .....	2
Выводы .....	7
Список литературы.....	7

## Список иллюстраций

Выполнение примеров с циклами .....	2
Выполнение примеров с условными выражениями .....	2
Выполнение примеров со сторонними библиотеками .....	3
Задание №1 .....	3
Задание №2 .....	4
Задание №3 .....	4
Задание №4 .....	4
Задание №5 .....	5
Реализация функции outer() .....	5
Проверка работы функции outer().....	6
Решение систему линейных уравнений .....	6
Задание №10.....	6
Задание №11.....	7

## Цель работы

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

## Задание

1. Используя Jupyter Lab, повторите примеры из раздела 3.2.

## 2. Выполните задания для самостоятельной работы (раздел 3.4)

### Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений [@julialang]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia [@juliadoc].

### Выполнение лабораторной работы

Для начала выполним примеры из лабораторной работы, чтобы познакомиться с циклами, условными операторами, функциями и работой со сторонними библиотеками (рис. [-@fig:001]-[-@fig:003]).

```
jupyter Lab3_статистический_анализ Last Checkpoint: ayer a las 3:04 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Logout
Not Trusted | Julia 1.12 O
In [5]: n = 0
while n < 10
    n += 1
    println(n)
end
1
2
3
4
5
6
7
8
9
10

In [5]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
i = 1
while i <= length(myfriends)
    friend = myfriends[i]
    println("Hi $friend, it's great to see you!")
    i += 1
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 [6]: for n in 1:2:10
    println(n)
end
1
3
5
```

#### Выполнение примеров с циклами

##### Условные выражения

```
In [12]: # используем '&&' для реализации операции "AND"
# операция % вычисляет остаток от деления
N = 5
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
```

Buzz

#### Выполнение примеров с условными выражениями

### Сторонние библиотеки (пакеты) в Julia

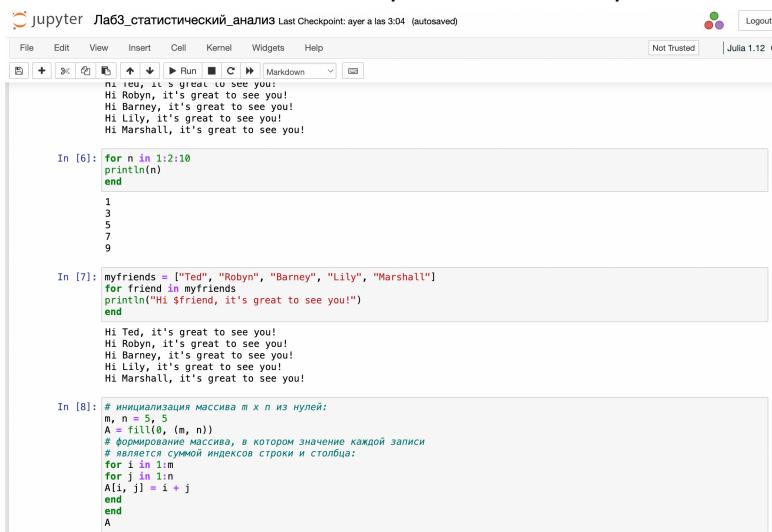
```
In [17]: import Pkg  
Pkg.add("Example")  
  
Resolving package versions...  
Installed Example - v0.5.5  
Updating `~/julia/environments/v1.12/Project.toml'  
[7876af07] + Example v0.5.5  
Updating `~/julia/environments/v1.12/Manifest.toml'  
[7876af07] + Example v0.5.5  
Precompiling packages...  
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`
```

### Выполнение примеров со сторонними библиотеками

Теперь перейдем к выполнению заданий для самостоятельной работы.

Используя циклы while и for (рис. [-@fig:004]):

- выведем на экран целые числа от 1 до 100 и напечатаем их квадраты;
- создадим словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;
- создадим массив squares\_arr, содержащий квадраты всех чисел от 1 до 100.



The screenshot shows a Jupyter Notebook interface with three code cells:

- In [6]:**

```
for n in 1:2:10  
    println(n)
```

 Output: 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!")
```

 Output: 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  
# равно сумме индексов строки и столбца:  
for i in 1:m  
    for j in 1:n  
        A[i, j] = i + j  
    end  
end  
A
```

### Задание №1

Напишем условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишем код, используя тернарный оператор (рис. [-@fig:005]).

```

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

```

### Задание №2

Напишем функцию `add_one`, которая добавляет 1 к своему входу (рис. [-@fig:006]).

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

6

### Задание №3

Используем `map()` или `broadcast()` для задания матрицы  $A$ , каждый элемент которой увеличивается на единицу по сравнению с предыдущим. (рис. [-@fig:007])

```

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

```

### Задание №4

Зададим матрицу  $A$ . Найдем  $A^3$ . Заменим третий столбец матрицы  $A$  на сумму второго и третьего столбцов (рис. [-@fig:008]).

The screenshot shows a Jupyter Notebook interface with the title "Jupyter Лаб6\_статистический\_анализ Last Checkpoint: ayer a las 3:04 (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a "Logout" button. A status bar at the top right indicates "Not Trusted" and "Julia 1.12". Below the menu is a toolbar with various icons for file operations. The main area contains several code cells:

```
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

In [38]: A = [1 1 3; 5 2 6; -2 -1 -3]
#^3
A_cubo = A * A * A
println("A^3 = ",A_cubo)
A^3 = [0 0 0; 0 0 0; 0 0 0]

In [39]: #reemplazar 3ra columna por suma de col2 + col3
A_mod = copy(A)
A_mod[:, 3] = A[:, 2] + A[:, 3]
Out[39]: 3-element Vector{Int64}:
 4
 8
-4

In [40]: B = repeat([10 -10 10]', 15) #15 filas, 3 columnas
C = B * B
println("C = B B = ",C)
C = B B = [45000;]

In [41]: n = 5
```

## Задание №5

Напишем свою функцию, аналогичную функции outer() языка R. Функция должна иметь следующий интерфейс: outer(x,y,operation) (рис. [-@fig:009],[-@fig:010]).

The screenshot shows a Jupyter Notebook interface with the title "Jupyter Лаб6\_статистический\_анализ Last Checkpoint: ayer a las 3:04 (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a "Logout" button. A status bar at the top right indicates "Not Trusted" and "Julia 1.12". Below the menu is a toolbar with various icons for file operations. The main area contains several code cells:

```
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 mas simple para multiplicacion
function outer_simple(A, B)
    return A * B
end

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

## Реализация функции outer()

```

Jupyter Lab3_статистический_анализ Last Checkpoint: ayer a las 3:04 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Logout
Not Trusted Julia 1.12
A4 = ((i+j-2) % n for i in 1:n4, j in 1:n4]
println("A4(10x10):")
for i in 1:n4
    println(join([lpad(A4[i,j], 3) for j in 1:n4], " "))
end

A4(10x10):
 0 1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9 0
 2 3 4 5 6 7 8 9 0 1
 3 4 5 6 7 8 9 0 1 2
 4 5 6 7 8 9 0 1 2 3
 5 6 7 8 9 0 1 2 3 4
 6 7 8 9 0 1 2 3 4 5
 7 8 9 0 1 2 3 4 5 6
 8 9 0 1 2 3 4 5 6 7
 9 0 1 2 3 4 5 6 7 8

In [50]: #A5: 9x9 con patron descendente
n5 = 9
A5 = [(j-i+5) % n5 for i in 1:n5, j in 1:n5]
println("A5(9x9):")
for i in 1:n5
    println(join([lpad(A5[i,j], 3) for j in 1:n5], " "))
end

A5 (9x9):
 0 1 2 3 4 5 6 7 8
 8 0 1 2 3 4 5 6 7
 7 8 0 1 2 3 4 5 6
 6 7 8 0 1 2 3 4 5
 5 6 7 8 0 1 2 3 4
 4 5 6 7 8 0 1 2 3
 3 4 5 6 7 8 0 1 2
 2 3 4 5 6 7 8 0 1
 1 2 3 4 5 6 7 8 0

```

## Проверка работы функции outer()

Решим систему линейных уравнений с 5 неизвестными (рис. [-@fig:011]).

```

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")

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

Solucion: x= [-2.000000000000013, 3.000000000000027, 4.999999999999964, 2.000000000000027, -4.000000000000001]
A*x deber ser y: *

```

## Решение системы линейных уравнений

В 10 задании произведем анализ количества элементов матрицы, удовлетворяющих необходимым условиям (рис. [-@fig:012]).

```

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([(1, 8), (1, 10), (2, 8), (2, 10), (4, 8), (4, 10), (8, 9), (8, 10), (9, 10)])

```

## Задание №10

Вычислим выражения (рис. [-@fig:013]).

```
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
```

Задание №11

## Выводы

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.

## Список литературы