

Лабораторная работа №3

Управляющие структуры

Клюкин М. А.

Российский университет дружбы народов, Москва, Россия

- Ключин Михаил Александрович
- студент
- Российский университет дружбы народов
- 1132226431@pruf.ru
- <https://MaKYaro.github.io/ru/>



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).

Выполнение лабораторной работы

Цикл while

```
while <условие>  
    <тело цикла>  
end
```


Пример использования цикла while

```
# 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
```

Рис. 1: Пример использования цикла while для формирования элементов массива

Пример использования цикла while

```
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!
```

Рис. 2: Пример использования цикла while для работы со строковыми элементами массива

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

Пример использования цикла for

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

1
3
5
7
9

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

1
2
3
4
5
6
7
8
9
10

Рис. 3: Пример использования цикла for для формирования элементов массива

Пример использования цикла for

```
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!
```

Рис. 4: Пример использования цикла for для работы со строковыми элементами массива

Пример использования цикла for

```
# 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
```

```
5x5 Matrix{Int64}:  
 2  3  4  5  6  
 3  4  5  6  7  
 4  5  6  7  8  
 5  6  7  8  9  
 6  7  8  9 10
```

Рис. 5: Пример использования цикла for для создания двумерного массива

Пример использования цикла for

```
# инициализация массива m x n из нулей:  
B = fill(0, (m, n))  
for i in 1:m, j in 1:n  
    B[i, j] = i + j  
end  
B
```

5x5 Matrix{Int64}:

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

Рис. 6: Пример использования цикла for для создания двумерного массива

Пример использования цикла for

```
C = [i + j for i in 1:m, j in 1:n]  
C
```

5×5 Matrix{Int64}:

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

Рис. 7: Пример использования цикла for для создания двумерного массива


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

Пример использования условного выражения

```
# используем `&&` для реализации операции "AND"  
# операция % вычисляет остаток от деления  
N = 15  
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
```

FizzBuzz

Рис. 8: Пример использования условного выражения

Тернарный оператор

`a ? b : c`

```
if a  
  b  
else  
  c  
end
```

Пример использования тернарного оператора

```
x = 5  
y = 10  
(x > y) ? x : y  
  
10
```

Рис. 9: Пример использования тернарного оператора

Функции

Пример написания функций с помощью ключевых слов

```
function sayhi(name)
    println("Hi $name, it's great to see you!")
end
# функция возведения в квадрат:
function f(x)
    x^2
end
```

f (generic function with 1 method)

```
sayhi("C-3P0")
f(42)
```

```
Hi C-3P0, it's great to see you!
1764
```

Рис. 10: Пример написания функций с помощью ключевых слов

Пример написания функций в одной строке

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

f2 (generic function with 1 method)

```
sayhi2("C-3P0")  
f2(42)
```

Hi C-3P0, it's great to see you!
1764

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

#5 (generic function with 1 method)

```
sayhi3("C-3P0")  
f3(42)
```

Hi C-3P0, it's great to see you!
1764

Рис. 11: Пример написания функций в одной строке

Пример изменения аргумента функции

```
# задаём массив v:  
v = [3, 5, 2]  
sort(v)  
v
```

```
3-element Vector{Int64}:  
 3  
 5  
 2
```

```
sort!(v)  
v
```

```
3-element Vector{Int64}:  
 2  
 3  
 5
```

```
v = [3, 5, 2]  
result = sort(v)  
result
```

```
3-element Vector{Int64}:  
 2  
 3  
 5
```

Рис. 12: Пример изменения аргумента функции

Пример вызова функции map

```
map(f, [1, 2, 3])
```

```
3-element Vector{Int64}:
```

```
1
```

```
4
```

```
9
```

Рис. 13: Пример вызова функции map

Пример вызова функции broadcast

```
f(x) = x^2  
broadcast(f, [1, 2, 3])
```

```
3-element Vector{Int64}:  
 1  
 4  
 9
```

Рис. 14: Пример вызова функции broadcast

Пример возведения матрицы в квадрат

```
# Задаём матрицу A:  
A = [i + 3*j for j in 0:2, i in 1:3]  
A
```

3×3 Matrix{Int64}:

1	2	3
4	5	6
7	8	9

f(A)

3×3 Matrix{Int64}:

30	36	42
66	81	96
102	126	150

Рис. 15: Пример возведения матрицы в квадрат

Пример вызова функции broadcast

```
broadcast(f, A)
```

3×3 Matrix{Int64}:

1	4	9
16	25	36
49	64	81

Рис. 16: Пример вызова функции broadcast

Пример записи математического выражения через функцию `bbroadcast`

```
A .+ 2 .* f.(A) ./ A
```

```
3×3 Matrix{Float64}:
```

```
 3.0  6.0  9.0  
12.0 15.0 18.0  
21.0 24.0 27.0
```

```
broadcast(x -> x + 2 * f(x) / x, A)
```

```
3×3 Matrix{Float64}:
```

```
 3.0  6.0  9.0  
12.0 15.0 18.0  
21.0 24.0 27.0
```

Рис. 17: Пример записи математического выражения через функцию `bbroadcast`

Сторонние библиотеки в Julia

Пример использования пакетов

```
import Pkg
Pkg.add("Example")

Updating registry at ~/julia/registries/General.toml
Resolving package versions...
Installing Example - v0.5.5
Updating ~/julia/environments/v1.11/Project.toml
(7678a467) = Example v0.5.5
Updating ~/julia/environments/v1.11/Manifest.toml
(7678a467) = Example v0.5.5
Precompiling project...
1217.3 ms / ConcreteStructs
1214.5 ms / IteratorInterfaceExtensions
2861.4 ms / SimpleBase
1214.2 ms / ExprTools
1221.8 ms / PosteriorFactorizations
1228.7 ms / MuladdMacro
1222.8 ms / LatexStrings
1226.8 ms / AbstractFFTs
1226.1 ms / FunctionWrappers
465.7 ms -- Example

Pkg.add("Colors")
using Colors

Resolving package versions...
Installing Colors - v0.13.1
Updating ~/julia/environments/v1.11/Project.toml
(3bc99051) = Colors v0.13.1
Updating ~/julia/environments/v1.11/Manifest.toml
(3bc99051) = Colors v0.13.0 - v0.13.1
Precompiling project...
4089.9 ms / Colors
865.1 ms / SparseMatrixColorings - SparseMatrixColoringsColorSet
3223.6 ms / ColorSchemes
38248.0 ms / Plots.jl
2383.1 ms / PlotThemes
4263.5 ms / RecipesPipeline
83123.0 ms / Plots
4187.8 ms / Plots - DefaultLook
5634.2 ms / Plots - IJuliaLook
9 dependencies successfully precompiled in 95 seconds, 454 already precompiled.

palette = distinguishable_colors(100)

randipalette, 3, 3)
```

Рис. 18: Пример использования пакетов

Задания для самостоятельного выполнения

Вывод чисел и их квадратов с помощью цикла while

```
n = 1
while n <= 100
  println(n, " ", n^2)
  n += 1
end
```

```
1, 1
2, 4
3, 9
4, 16
5, 25
6, 36
7, 49
8, 64
9, 81
10, 100
11, 121
12, 144
13, 169
14, 196
15, 225
16, 256
17, 289
18, 324
19, 361
20, 400
21, 441
22, 484
```

Рис. 19: Вывод чисел и их квадратов с помощью цикла while

Вывод чисел и их квадратов с помощью цикла for

```
for n in 1:100  
    println(n, " ", n^2)  
end
```

```
1, 1  
2, 4  
3, 9  
4, 16  
5, 25  
6, 36  
7, 49  
8, 64  
9, 81  
10, 100  
11, 121  
12, 144  
13, 169  
14, 196  
15, 225  
16, 256  
17, 289  
18, 324  
19, 361  
20, 400  
21, 441  
22, 484  
23, 529  
24, 576
```

Рис. 20: Вывод чисел и их квадратов с помощью цикла for

```
squares = Dict()
for n in 1:100
    squares[n] = n^2
end
println(squares)
```

```
Dict{Any, Any}(5 => 25, 56 => 3136, 35 => 1225, 55 => 3025, 60 => 3600, 30 => 900, 32 => 1024, 6 => 36, 67 => 4489, 45 => 2025, 73 => 5329, 64 => 4096, 90 => 8100, 4 => 16, 13 => 169, 54 => 2916, 63 => 3969, 86 => 7396, 91 => 8281, 62 => 3844, 58 => 3364, 52 => 2704, 12 => 144, 28 => 784, 75 => 5625, 23 => 529, 92 => 8464, 41 => 1681, 43 => 1849, 11 => 121, 36 => 1296, 68 => 4624, 69 => 4761, 98 => 9604, 82 => 6724, 85 => 7225, 39 => 1521, 84 => 7056, 77 => 5929, 7 => 49, 25 => 625, 95 => 9025, 71 => 5041, 66 => 4356, 76 => 5776, 34 => 1156, 50 => 2500, 59 => 3481, 93 => 8649, 2 => 4, 10 => 100, 18 => 324, 26 => 676, 27 => 729, 42 => 1764, 87 => 7569, 100 => 10000, 79 => 6241, 16 => 256, 20 => 400, 81 => 6561, 19 => 361, 49 => 2401, 44 => 1936, 9 => 81, 31 => 961, 74 => 5476, 61 => 3721, 29 => 841, 94 => 8836, 46 => 2116, 57 => 3249, 70 => 4900, 21 => 441, 38 => 1444, 88 => 7744, 78 => 6084, 72 => 5184, 24 => 576, 8 => 64, 17 => 289, 37 => 1369, 1 => 1, 53 => 2809, 22 => 484, 47 => 2209, 83 => 6889, 99 => 9801, 89 => 7921, 14 => 196, 3 => 9, 80 => 6400, 96 => 9216, 51 => 2601, 33 => 1089, 40 => 1600, 48 => 2304, 15 => 225, 65 => 4225, 97 => 9409)
```

Рис. 21: Создание словаря

```
squares_arr = [n^2 for n in 1:100]  
println(squares_arr)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576,  
625, 676, 729, 784, 841, 900, 961, 1024, 1089, 1156, 1225, 1296, 1369, 1444, 1521, 1600, 1681, 1764, 1849, 1  
936, 2025, 2116, 2209, 2304, 2401, 2500, 2601, 2704, 2809, 2916, 3025, 3136, 3249, 3364, 3481, 3600, 3721, 3  
844, 3969, 4096, 4225, 4356, 4489, 4624, 4761, 4900, 5041, 5184, 5329, 5476, 5625, 5776, 5929, 6084, 6241, 6  
400, 6561, 6724, 6889, 7056, 7225, 7396, 7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9  
604, 9801, 10000]
```

Рис. 22: Создание массива

Пример условного оператора

```
n = 3
if n % 2 == 0
    println(n)
else
    println("Нечетное")
end
```

Нечетное

Рис. 23: Пример условного оператора

Пример тернарного оператора

```
n = 4  
println(n % 2 == 0 ? n : "Нечетное")  
4
```

Рис. 24: Пример тернарного оператора

```
function add_one(x)
    return x + 1
end
println(add_one(1))
```

2

Рис. 25: Пример функции

Пример использования map

```
A = [1 2 3; 1 2 3; 1 2 3]  
B = map(x -> x + 1, A)  
B
```

3×3 Matrix{Int64}:

2 3 4

2 3 4

2 3 4

Рис. 26: Пример использования map

Пример возведения матрицы в степень

```
A = [1 1 3; 5 2 6; -2 -1 -3]
println(map(x -> x ^ 3, A))

[1 1 27; 125 8 216; -8 -1 -27]
```

Рис. 27: Пример возведения матрицы в степень

Пример замены столбца в матрице

```
A[:, 3] = A[:, 2] + A[:, 3]
```

A

3x3 Matrix{Int64}:

1	1	4
5	2	8
-2	-1	-4

Рис. 28: Пример замены столбца в матрице

Создание матрицы B

```
B = repeat([10 -10 10], 15, 1)
```

15x3 Matrix{Int64}:

10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10
10	-10	10

Рис. 29: Создание матрицы B

Вычисление матрицы $B^T B$

```
C = B' * B
```

```
C
```

```
3x3 Matrix{Int64}:
```

```
 1500  -1500  1500  
-1500   1500 -1500  
 1500  -1500  1500
```

Рис. 30: Вычисление матрицы $B^T B$

Создание матриц Z, E, Z1

```
Z = zeros(Int, 6, 6)
E = ones(Int, 6, 6)

n = 6
Z1 = copy(Z)
for i in 1:n
    for j in 1:n
        if abs(i - j) == 1
            Z1[i, j] = 1
        end
    end
end
Z1
```

```
6x6 Matrix{Int64}:
 0  1  0  0  0  0
 1  0  1  0  0  0
 0  1  0  1  0  0
 0  0  1  0  1  0
 0  0  0  1  0  1
 0  0  0  0  1  0
```

Рис. 31: Создание матриц Z, E, Z1

Создание матриц Z2

```
Z2 = copy(Z)
for i in 1:n
    if i+2 < n
        Z2[i, i] = 1
        Z2[i, i+2] = 1
    elseif (i == 3) || (i == 4)
        Z2[i, i] = 1
        Z2[i, i+2] = 1
        Z2[i, i-2] = 1
    else
        Z2[i, i] = 1
        Z2[i, i-2] = 1
    end
end
Z2
```

```
6×6 Matrix{Int64}:
 1  0  1  0  0  0
 0  1  0  1  0  0
 0  0  1  0  1  0
 0  1  0  1  0  1
 0  0  1  0  1  0
 0  0  0  1  0  1
```

Рис. 32: Создание матриц Z2

Создание матриц Z3

```
Z3 = copy(Z)
for i in 1:n
    for j in 1:n
        if (i+j) % 2 != 0
            Z3[i, j] = 1
        end
    end
end
Z3
```

6×6 Matrix{Int64}:

0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0

Рис. 33: Создание матриц Z3

Создание матриц Z3

```
Z4 = copy(Z)
for i in 1:n
    for j in 1:n
        if (i+j) % 2 == 0
            Z4[i, j] = 1
        end
    end
end
Z4
```

6×6 Matrix{Int64}:

1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1

Рис. 34: Создание матриц Z4

```
function outer(x, y, operation)
    return [operation(i, j) for i in x, j in y]
end
```

outer (generic function with 1 method)

Рис. 35: Создание функции outer

Создание матрицы A_1

```
A1 = outer(0:4, 0:4, +)
```

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. 36: Создание матрицы A_1

```
A2 = outer(0:4, 1:5, ^)
```

```
A2
```

```
5x5 Matrix{Int64}:
```

0	0	0	0	0
1	1	1	1	1
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024

Рис. 37: Создание матрицы A_2

Создание матрицы A_3

```
A3 = outer(0:4, 0:4, (x, y) -> mod(x+y, 5))  
A3
```

5x5 Matrix{Int64}:

0	1	2	3	4
1	2	3	4	0
2	3	4	0	1
3	4	0	1	2
4	0	1	2	3

Рис. 38: Создание матрицы A_3

Создание матрицы A_4

```
A4 = outer(0:9, 0:9, (x, y) -> mod(x+y, 10))  
A4
```

10×10 Matrix{Int64}:

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

Рис. 39: Создание матрицы A_4

Создание матрицы A_5

```
A5 = outer(0:8, 0:8, (x, y) -> mod(x-y, 9))  
A5
```

9×9 Matrix{Int64}:

0	8	7	6	5	4	3	2	1
1	0	8	7	6	5	4	3	2
2	1	0	8	7	6	5	4	3
3	2	1	0	8	7	6	5	4
4	3	2	1	0	8	7	6	5
5	4	3	2	1	0	8	7	6
6	5	4	3	2	1	0	8	7
7	6	5	4	3	2	1	0	8
8	7	6	5	4	3	2	1	0

Рис. 40: Создание матрицы A_5

Решение системы линейных уравнений

```
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;  
]  
b = [7; -1; -3; 5; 17]  
  
x = A \ b  
x
```

```
5-element Vector{Float64}:  
-2.00000000000000036  
 3.00000000000000058  
 4.9999999999999998  
 1.9999999999999991  
-3.9999999999999999
```

Рис. 41: Решение системы линейных уравнений

Создание матрицы M

```
M = rand(1:10, 6, 10)
```

```
M
```

```
6×10 Matrix{Int64}:
```

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

Рис. 42: Создание матрицы M

Поиск числа элементов в каждой строке матрицы M

```
N = 4  
greater_than_N = sum(M .> N)  
greater_than_N
```

41

Рис. 43: Поиск числа элементов в каждой строке матрицы M

Поиск строк с двумя числами M

```
twice = [i for i=1:6 if sum(M[i, :].==7)==2]  
twice
```

```
Int64[]
```

Рис. 44: Поиск строк с двумя числами M

Поиск столбцов матрицы M

```
K = 75
pairs = []
for i in 1:size(M, 2)-1
    for j in i + 1:size(M, 2)
        if sum(M[:, i] .+ M[:, j]) > K
            push!(pairs, (i, j))
        end
    end
end
pairs
```

18-element Vector{Any}:

```
(1, 2)
(1, 3)
(1, 7)
(1, 9)
(2, 3)
(2, 5)
(2, 6)
(2, 7)
(2, 8)
(2, 9)
(2, 10)
(3, 5)
(3, 6)
(3, 7)
(3, 8)
(3, 9)
(3, 10)
(7, 9)
```

Рис. 45: Поиск столбцов матрицы M

Вычисление выражений

```
sum1 = sum(i^4 / (3 + j) for i in 1:20 for j in 1:5)  
sum1
```

639215.2833333334

```
sum2 = sum(i^4 / (3 + i * j) for i in 1:20 for j in 1:5)
```

89912.02146097136

Рис. 46: Вычисление выражений

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