

# **Отчет по лабораторной работе 3**

**Управляющие структуры**

Шалыгин Георгий Эдуардович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Задания для самостоятельного выполнения . . . . .	9
<b>3</b>	<b>Выводы</b>	<b>17</b>
	<b>Список литературы</b>	<b>18</b>

## Список иллюстраций

2.1	while . . . . .	6
2.2	for . . . . .	6
2.3	двойной for . . . . .	7
2.4	if-else . . . . .	7
2.5	функции . . . . .	8
2.6	способы задания функций . . . . .	8
2.7	операции inplace . . . . .	8
2.8	map, broadcast . . . . .	8
2.9	broadcast . . . . .	9
2.10	установка пакета . . . . .	9
2.11	1 . . . . .	10
2.12	2 . . . . .	10
2.13	3 . . . . .	11
2.14	4 . . . . .	11
2.15	5 . . . . .	11
2.16	6 . . . . .	12
2.17	7 . . . . .	12
2.18	8 . . . . .	13
2.19	9 . . . . .	13
2.20	10 . . . . .	13
2.21	21 . . . . .	14
2.22	22 . . . . .	14
2.23	23 . . . . .	15
2.24	24 . . . . .	15
2.25	25 . . . . .	16

## Список таблиц

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

1. Повторим примеры (fig. 2.7).

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

12345678910

```
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. 2.1: while

```
for n in 1:2:10
    print(n)
end
println()
myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
for friend in myfriends
    println("Hi $friend, it's great to see you!")
end
```

13579  
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. 2.2: for

```

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

```

```

B = fill(0, (m, n))
for i in 1:m, j in 1:n
    B[i, j] = i + j
end
@show B
#Ещё одна реализация этого же примера:
C = [i + j for i in 1:m, j in 1:n]
@show C

```

Рис. 2.3: двойной for

```

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

```

```

Buzz

```

```

10

```

Рис. 2.4: if-else

```
function sayhi(name)
    println("Hi $name, it's great to see you!")
end
# функция возведения в квадрат:
function f(x)
    x^2
end
sayhi("C-3P0")
f(42)
```

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

1764

Рис. 2.5: функции

```
sayhi2(name) = println("Hi $name, it's great to see you!")
f2(x) = x^2
#Наконец, можно объявить выше определённые функции как «анонимные»:
sayhi3 = name -> println("Hi $name, it's great to see you!")
f3 = x -> x^2
println(f3(10))
sayhi3("QWE")
```

100

Hi QWE, it's great to see you!

Рис. 2.6: способы задания функций

```
v = [3, 5, 2]
sort(v)
@show v
sort!(v)
@show v
```

v = [3, 5, 2]  
v = [2, 3, 5]

Рис. 2.7: операции inplace

```
f(x) = x^2
println(map(f, [1, 2, 3]))
println(map(x -> x^3, [1, 2, 3]))
f(x) = x^2
println(broadcast(f, [1, 2, 3]))
println(f.([1, 2, 3]))
```

[1, 4, 9]  
[1, 8, 27]  
[1, 4, 9]  
[1, 4, 9]

Рис. 2.8: map, broadcast



```
A = [i + 3*j for j in 0:2, i in 1:3]
f(A)
```

```
3×3 Matrix{Int64}:
 30  36  42
 66  81  96
102 126 150
```

```
f.(A)
```

```
3×3 Matrix{Int64}:
 1  4  9
16 25 36
49 64 81
```

Рис. 2.9: broadcast

```
using Pkg
Pkg.add("Colors")
using Colors
```

```
Updating registry at `D:\julia\depot\registries\General.toml`
Resolving package versions...
Updating `D:\julia\depot\environments\v1.8\Project.toml`
[5ae59095] + Colors v0.12.10
No Changes to `D:\julia\depot\environments\v1.8\Manifest.toml`
```

```
palette = distinguishable_colors(100)
rand(palette, 3, 3)
```



Рис. 2.10: установка пакета

## 2.1 Задания для самостоятельного выполнения

1. Используя циклы while и for:

– выведите на экран целые числа от 1 до 100 и напечатайте их квадраты;

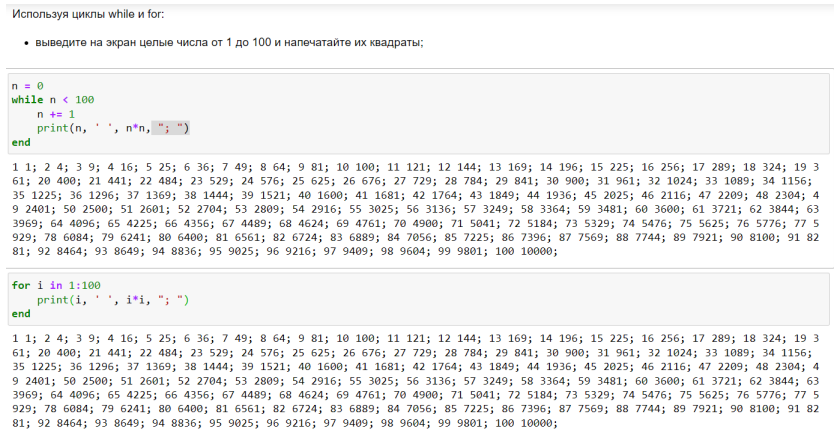


Рис. 2.11: 1

- создайте словарь `squares`, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;
- создайте массив `squares_arr`, содержащий квадраты всех чисел от 1 до 100

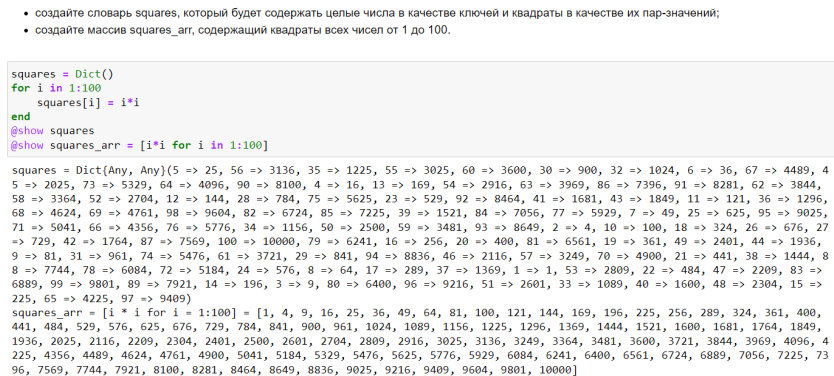


Рис. 2.12: 2

- Напишите условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишите код, используя тернарный оператор

Напишите условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишите код, используя тернарный оператор.

```
N = 10
if (N % 2 == 0)
    println("четное")
else
    println("нечетное")
end
println((N % 2 == 0) ? "четное" : "нечетное")

четное
четное
```

Рис. 2.13: 3

#### 4. Напишите функцию add\_one, которая добавляет 1 к своему входу

Напишите функцию add\_one, которая добавляет 1 к своему входу.

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

print(add_one(2))
```

3

Рис. 2.14: 4

#### 5. Используйте map() или broadcast() для задания матрицы $\boxtimes$ , каждый элемент которой увеличивается на единицу по сравнению с предыдущим.

Используйте map() или broadcast() для задания матрицы A, каждый элемент которой увеличивается на единицу по сравнению с предыдущим.

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

```
3x3 Matrix{Int64}:
 2  2  4
 6  3  7
-1  0 -2
```

Рис. 2.15: 5

#### 6. Задайте матрицу $\boxtimes$ . – Найдите $A^3$ . – Замените третий столбец матрицы $\boxtimes$ на сумму второго и третьего столбцов

Найдите  $A^3$

Замените третий столбец матрицы  $A$  на сумму второго и третьего столбцов

```
A^3
```

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

```
0  0  0  
0  0  0  
0  0  0
```

```
A[:, 3] = A[:, 1] + A[:, 2]
```

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

```
2  
7  
-3
```

Рис. 2.16: 6

7. Создайте матрицу  $\mathbf{X}$  с элементами  $\mathbf{X}_{11} = 10, \mathbf{X}_{12} = -10, \mathbf{X}_{13} = 10, \mathbf{X}_{i1} = 1, 2, \dots, 15$ . Вычислите матрицу  $\mathbf{C} = \mathbf{X}^T \mathbf{X}$ .

Создайте матрицу  $B$  с элементами

$B_{i1} = 10, B_{i2} = -10, B_{i3} = 10, i = 1, 2, \dots, 15$ . Вычислите матрицу  $C = B^T B$ .

```
B = hcat(fill(10, 15), fill(-10, 15), fill(10, 15))  
C = B' * B  
println(C)
```

```
[1500 -1500 1500; -1500 1500 -1500; 1500 -1500 1500]
```

Рис. 2.17: 7

8. Создайте матрицу  $\mathbf{X}$  размерности  $6 \times 6$ , все элементы которой равны нулю, и матрицу  $\mathbf{Y}$ , все элементы которой равны 1. Используя цикл `while` или `for` и закономерности расположения элементов, создайте следующие матрицы размерности  $6 \times 6$ :

. Создайте матрицу  $Z$  размерности  $6 \times 6$ , все элементы которой равны нулю, и матрицу  $E$ , все элементы которой равны 1. Используя цикл `while` или `for` и закономерности расположения элементов, создайте следующие матрицы размерности  $6 \times 6$ :

```
Z = zeros(6, 6)
E = ones(6, 6)
for i in 1:6, j in 1:6
    if abs(j-i) == 1
        Z[i, j] = 1
    end
end
Z
```

Рис. 2.18: 8

```
: Z = zeros(6, 6)
E = ones(6, 6)
for i in 1:6, j in 1:6
    if (abs(j-i) == 0) || abs(i-j) == 2
        Z[i, j] = 1
    end
end
Z

: 6x6 Matrix{Float64}:
 1.0  0.0  1.0  0.0  0.0  0.0
 0.0  1.0  0.0  1.0  0.0  0.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 0.0  0.0  1.0  0.0  1.0  0.0
 0.0  0.0  0.0  1.0  0.0  1.0
```

Рис. 2.19: 9

```
Z = zeros(6, 6)
E = ones(6, 6)
for i in 1:6, j in 1:6
    if j+i == 5 || i+j == 7 || i+j == 9
        Z[i, j] = 1
    end
end
Z

6x6 Matrix{Float64}:
 0.0  0.0  0.0  1.0  0.0  1.0
 0.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  0.0
 1.0  0.0  1.0  0.0  0.0  0.0
```

Рис. 2.20: 10

```

: Z = zeros(6, 6)
  E = ones(6, 6)
  for i in 1:6, j in 1:6
    if (j+i) % 2 == 0
      Z[i, j] = 1
    end
  end
end
Z

: 6×6 Matrix{Float64}:
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0
 1.0  0.0  1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0  0.0  1.0

```

Рис. 2.21: 21

9. Напишите свою функцию, аналогичную функции `outer()` языка R

```

function outer(A, B, f)
  if length(A) == 1 && length(B) == 1
    return f(A[1], B[1])
  end

  N = length(A)
  M = length(B)
  result = zeros(eltype(A), N, M)
  for i in 1:N
    for j in 1:M
      result[i, j] = f(A[i], B[j])
    end
  end
  return result
end

```

outer (generic function with 1 method)

Рис. 2.22: 22

10. Используя написанную вами функцию `outer()`, создайте матрицы следующей структуры:

```

n = 5
m = 6
A1 = outer(1:n, 0:m-1, +)
A2 = outer(0:n-1, 0:m-1, *)
A3 = outer(0:n-1, 1:n, (x, y) -> mod(x + y, n))
A4 = outer(0:m-1, 1:n, (x, y) -> mod(x + y, m))
A5 = outer(0:n-1, 1:n, (x, y) -> mod(x * y, n) + 1)
println("A1:\n", A1)
println("\nA2:\n", A2)
println("\nA3:\n", A3)
println("\nA4:\n", A4)
println("\nA5:\n", A5)

```

A1:  
[1 2 3 4 5 6; 2 3 4 5 6 7; 3 4 5 6 7 8; 4 5 6 7 8 9; 5 6 7 8 9 10]

A2:  
[0 0 0 0 0 0; 0 1 2 3 4 5; 0 2 4 6 8 10; 0 3 6 9 12 15; 0 4 8 12 16 20]

A3:  
[1 2 3 4 0; 2 3 4 0 1; 3 4 0 1 2; 4 0 1 2 3; 0 1 2 3 4]

A4:  
[1 2 3 4 5 0; 2 3 4 5 0 1; 3 4 5 0 1 2; 4 5 0 1 2 3; 5 0 1 2 3 4; 0 1 2 3 4 5]

A5:  
[1 1 1 1 1; 2 3 4 5 1; 3 5 2 4 1; 4 2 5 3 1; 5 4 3 2 1]

Рис. 2.23: 23

11. Решите следующую систему линейных уравнений с 5 неизвестными:

```

: function solve_matrix(A, b)
    return A \ b
end

n = 5
A = zeros(n, n)
for i in 1:n, j in 1:n
    A[i, j] = abs(i-j)+1
end
b = [7 -1 -3 5 17]
solve_matrix(A, b)

```

: 5x1 Matrix{Float64}:  
0.2707774798927614  
0.24128686327077742  
0.20643431635388731  
0.1554959785522788  
0.13136729222520108

Рис. 2.24: 24

12. Создайте матрицу  $\boxtimes$  размерности  $6 \times 10$ , элементами которой являются целые числа, выбранные случайным образом с повторениями из совокуп-

ности 1, 2, ..., 10. – Найдите число элементов в каждой строке матрицы  $\mathbb{X}$ , которые больше числа  $\mathbb{X}$  (например,  $\mathbb{X} = 4$ ). – Определите, в каких строках матрицы  $\mathbb{X}$  число  $\mathbb{X}$  (например,  $\mathbb{X} = 7$ ) встречается ровно 2 раза? – Определите все пары столбцов матрицы  $\mathbb{X}$ , сумма элементов которых больше  $\mathbb{X}$  (например,  $\mathbb{X} = 75$ ).

```
rows = 6
cols = 10
N = 4
M = 7
K = 75
M = rand(1:10, rows, cols)
println("Матрица M:")
println(M)

# 2. Находим число элементов в каждой строке, которые больше N
elements_greater_than_N = [count(x -> x > N, row) for row in eachrow(M)]
println("\nчисло элементов в каждой строке, больших чем $N:")
println(elements_greater_than_N)

# 3. Определяем, в каких строках число M встречается ровно 2 раза
rows_with_M_twice = sum(M .== M, dims=2) .== 2
println("\nСтроки, в которых число $M встречается ровно 2 раза:")
println(findall(rows_with_M_twice))

# 4. Определяем все пары столбцов, сумма элементов которых больше K
column_pairs_sum_greater_than_K = [(i, j) for i in 1:cols-1, j in i+1:cols if sum(M[:, i] + M[:, j]) > K]
println("\nПары столбцов, сумма элементов которых больше $K:")
println(column_pairs_sum_greater_than_K)
```

Рис. 2.25: 25



## 3 Выводы

В ходе работы были освоены применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

## **Список литературы**