

Лабораторная работа №3

Компьютерный практикум по статистическому анализу данных

Николаев Дмитрий Иванович

Содержание

1	Цель работы	6
2	Выполнение лабораторной работы	7
2.1	Повторение примеров	7
2.2	Самостоятельная работа	16
3	Выводы	35
	Список литературы	36

Список иллюстраций

2.1	Циклы while и for (1)	7
2.2	Циклы while и for (2)	8
2.3	Циклы while и for (3)	9
2.4	Циклы while и for (4)	10
2.5	Условные выражения	10
2.6	Функции (1)	11
2.7	Функции (2)	12
2.8	Функции (3)	13
2.9	Функции (4)	14
2.10	Функции (5)	15
2.11	Сторонние библиотеки и пакеты (1)	15
2.12	Сторонние библиотеки и пакеты (2)	16
2.13	Задание 1. Пункт 1	16
2.14	Задание 1. Пункт 2 (1)	17
2.15	Задание 1. Пункт 2 (2)	18
2.16	Задание 1. Пункт 3 (1)	19
2.17	Задание 1. Пункт 3 (2)	20
2.18	Задание 2. Тернарный условный оператор	21
2.19	Задание 3.	21
2.20	Задание 4.	21
2.21	Задание 5. Пункт 1	22
2.22	Задание 5. Пункт 2	22
2.23	Задание 5. Пункт 3	23
2.24	Задание 6 (1).	23
2.25	Задание 6 (2).	23
2.26	Задание 7.	24
2.27	Задание 7. Матрица Z_1	25
2.28	Задание 7. Матрица Z_2	26
2.29	Задание 7. Матрица Z_3	26
2.30	Задание 7. Матрица Z_4	27
2.31	Задание 8. Пункт 1. Написание функции outer	27
2.32	Задание 8. Пункт 2. Матрица A_1	27
2.33	Задание 8. Пункт 2. Матрицы A_2 и A_3	28
2.34	Задание 8. Пункт 2. Матрицы A_4 и A_5	29
2.35	Задание 9. Матрица коэффициентов	30
2.36	Задание 9. Метод Гаусса	30
2.37	Задание 9. Решение системы уравнений	31

2.38	Задание 10. Создание матрицы со случайными целыми числами .	32
2.39	Задание 10. Пункт 1.	32
2.40	Задание 10. Пункт 2.	32
2.41	Задание 10. Пункт 3.	33
2.42	Задание 11. Пункт 1	33
2.43	Задание 11. Пункт 2	34

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

Выполняем задания, следуя указаниям [1].

2.1 Повторение примеров

Повторим примеры, представленные в лабораторной работе. Циклы `while` и `for` ([2.1-2.4]), Условные выражения ([2.5]), функции ([2.6-2.10]) и сторонние библиотеки/пакеты ([2.11,2.12]).

▾ Лабораторная работа № 3. Управляющие структуры

Повторение примеров

Циклы `while` и `for`

```
[4]: # 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. 2.1: Циклы `while` и `for` (1)

```
[6]: 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!
```

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

```
1  
3  
5  
7  
9
```

```
[8]: 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. 2.2: Циклы while и for (2)


```
[9]: # 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
```

```
[9]: 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
```

Другая реализация этого же примера:

```
[10]: # 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
```

```
[10]: 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. 2.3: Циклы while и for (3)

Ещё одна реализация этого же примера:

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

```
[11]: 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. 2.4: Циклы while и for (4)

Условные выражения

```
[13]: # 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

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

```
[14]: 10
```

Рис. 2.5: Условные выражения

Функции

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

[15]: f (generic function with 1 method)

Вызов функции осуществляется по её имени с указанием аргументов, например:

```
[16]: sayhi("C-3P0")
      f(42)

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

[16]: 1764

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

[17]: f2 (generic function with 1 method)

Наконец, можно объявить выше определённые функции как «анонимные»:

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

[18]: #5 (generic function with 1 method)

Рис. 2.6: Функции (1)

```
[20]: # задаём массив v:  
v = [3, 5, 2]  
sort(v)  
  
[20]: 3-element Vector{Int64}:  
      2  
      3  
      5  
  
[21]: v  
  
[21]: 3-element Vector{Int64}:  
      3  
      5  
      2  
  
[22]: sort!(v)  
  
[22]: 3-element Vector{Int64}:  
      2  
      3  
      5  
  
[23]: v  
  
[23]: 3-element Vector{Int64}:  
      2  
      3  
      5
```

Рис. 2.7: Функции (2)

```
[24]: f(x) = x^2  
      map(f, [1, 2, 3])
```

```
[24]: 3-element Vector{Int64}:  
      1  
      4  
      9
```

```
[25]: x -> x^3  
      map(x -> x^3, [1, 2, 3])
```

```
[25]: 3-element Vector{Int64}:  
      1  
      8  
     27
```

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

```
[26]: 3-element Vector{Int64}:  
      1  
      4  
      9
```

```
[27]: f.([1, 2, 3])
```

```
[27]: 3-element Vector{Int64}:  
      1  
      4  
      9
```

Рис. 2.8: Функции (3)

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

[28]: 3x3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

[29]: # Вызываем функцию f возведения в квадрат
f(A)

[29]: 3x3 Matrix{Int64}:
30  36  42
66  81  96
102 126 150

[30]: B = f.(A)

[30]: 3x3 Matrix{Int64}:
 1  4  9
16 25 36
49 64 81
```

Рис. 2.9: Функции (4)

```
[31]: A .+ 2 .* f.(A) ./ A
```

```
[31]: 3x3 Matrix{Float64}:  
  3.0  6.0  9.0  
 12.0 15.0 18.0  
 21.0 24.0 27.0
```

```
[32]: @. A + 2 * f(A) / A
```

```
[32]: 3x3 Matrix{Float64}:  
  3.0  6.0  9.0  
 12.0 15.0 18.0  
 21.0 24.0 27.0
```

```
[33]: broadcast(x -> x + 2 * f(x) / x, A)
```

```
[33]: 3x3 Matrix{Float64}:  
  3.0  6.0  9.0  
 12.0 15.0 18.0  
 21.0 24.0 27.0
```

Рис. 2.10: Функции (5)

Сторонние библиотеки (пакеты) в Julia

```
[34]: import Pkg  
      Pkg.add("Example")  
  
      Updating registry at `C:\Users\User\.julia\registries\General.toml`  
      Resolving package versions...  
      Installed Example - v0.5.3  
      Updating `C:\Users\User\.julia\environments\v1.8\Project.toml`  
      [7876af07] + Example v0.5.3  
      Updating `C:\Users\User\.julia\environments\v1.8\Manifest.toml`  
      [7876af07] + Example v0.5.3  
      Precompiling project...  
      ✓ Example  
      1 dependency successfully precompiled in 7 seconds. 287 already precompiled.  
  
[35]: Pkg.add("Colors")  
      using Colors  
  
      Resolving package versions...  
      Updating `C:\Users\User\.julia\environments\v1.8\Project.toml`  
      [5ae59095] + Colors v0.12.10  
      No Changes to `C:\Users\User\.julia\environments\v1.8\Manifest.toml`
```

Рис. 2.11: Сторонние библиотеки и пакеты (1)

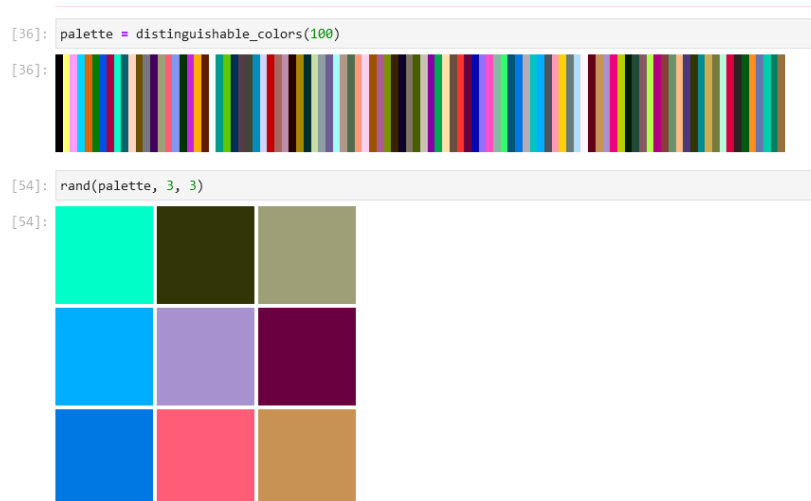


Рис. 2.12: Сторонние библиотеки и пакеты (2)

2.2 Самостоятельная работа

1. Используем циклы while и for (задания написаны на скриншотах):

1) Пункт 1 ([2.13])

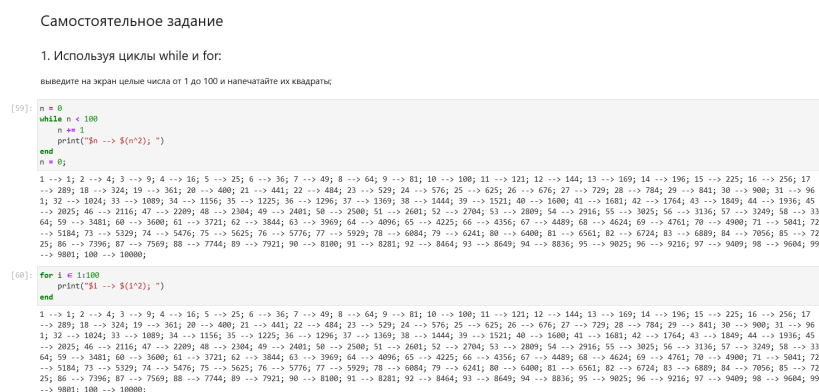


Рис. 2.13: Задание 1. Пункт 1

2) Пункт 2 ([2.14,2.15])

– создайте словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;

```
[71]: squares_1 = Dict()
      while n < 100
        n += 1
        squares_1[n] = n^2
      end
      n = 0;
      pairs(squares_1)
```

[71]: Dict{Any, Any} with 100 entries:

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
:	=>	:

Рис. 2.14: Задание 1. Пункт 2 (1)

```
[72]: squares_2 = Dict()
      for i ∈ 1:100
          squares_2[i] = i^2
      end
      squares_2
```

[72]: Dict{Any, Any} with 100 entries:

```
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
⋮  => ⋮
```

Рис. 2.15: Задание 1. Пункт 2 (2)

3) Пункт 3 ([2.16,2.17])

– создайте массив squares_arr, содержащий квадраты всех чисел от 1 до 100.

```
[73]: squares_arr_1 = (Int64)[]  
      while n < 100  
        n += 1  
        push!(squares_arr_1, n^2)  
      end  
      n = 0;  
      squares_arr_1
```

```
[73]: 100-element Vector{Int64}:  
      1  
      4  
      9  
     16  
     25  
     36  
     49  
     64  
     81  
    100  
    121  
    144  
    169  
     ⋮  
   7921  
  8100  
  8281  
  8464  
  8649  
  8836  
  9025  
  9216  
  9409  
  9604  
  9801  
 10000
```

Рис. 2.16: Задание 1. Пункт 3 (1)

```
[75]: squares_arr_2 = (Int64[])  
      for i ∈ 1:100  
          push!(squares_arr_2, i^2)  
      end  
      squares_arr_2
```

```
[75]: 100-element Vector{Int64}:  
      1  
      4  
      9  
     16  
     25  
     36  
     49  
     64  
     81  
    100  
    121  
    144  
    169  
     ⋮  
   7921  
   8100  
   8281  
   8464  
   8649  
   8836  
   9025  
   9216  
   9409  
   9604  
   9801  
  10000
```

Рис. 2.17: Задание 1. Пункт 3 (2)

2. Напишем условное выражение, используя тернарный оператор ([2.18])

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

```
[80]: x = 5; y = 6;
[81]: iseven(x) ? println(x) : println("Нечётное")
      iseven(y) ? println(y) : println("Нечётное")
      Нечётное
      6
```

Рис. 2.18: Задание 2. Тернарный условный оператор

3. Напишем функцию добавления единицы к аргументу ([2.19])

3. Напишите функцию add_one, которая добавляет 1 к своему входу.

```
[85]: add_one(x) = x + 1
[85]: add_one (generic function with 1 method)
[86]: add_one(1)
[86]: 2
```

Рис. 2.19: Задание 3.

4. Создадим матрицу, элементы которой различаются на единицу ([2.20])

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

```
[87]: A = zeros(3, 3)
[87]: 3x3 Matrix{Float64}:
      0.0  0.0  0.0
      0.0  0.0  0.0
      0.0  0.0  0.0
[92]: add_one.(A)
[92]: 3x3 Matrix{Float64}:
      1.0  1.0  1.0
      1.0  1.0  1.0
      1.0  1.0  1.0
[93]: map(add_one, A)
[93]: 3x3 Matrix{Float64}:
      1.0  1.0  1.0
      1.0  1.0  1.0
      1.0  1.0  1.0
[98]: x = 0
      for i ∈ 1:3, j ∈ 1:3
          A[i, j] = add_one(x)
          x += 1
      end
      A
[98]: 3x3 Matrix{Float64}:
      1.0  2.0  3.0
      4.0  5.0  6.0
      7.0  8.0  9.0
```

Рис. 2.20: Задание 4.

5. Работа с матрицей определенного вида (задания написаны на скриншотах):

1) Пункт 1 ([2.21])

5. Задайте матрицу A следующего вида:

$$A = \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix}$$

```
[99]: A = [[1 1 3]; [5 2 6]; [-2 -1 -3]]
```

```
[99]: 3x3 Matrix{Int64}:
```

```
 1  1  3
 5  2  6
-2 -1 -3
```

Рис. 2.21: Задание 5. Пункт 1

2) Пункт 2 ([2.22])

– Найдите A^3 .

```
[100]: g(x) = x^3
```

```
[100]: g (generic function with 1 method)
```

```
[105]: g(A)
```

```
[105]: 3x3 Matrix{Int64}:
```

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

Рис. 2.22: Задание 5. Пункт 2

3) Пункт 3 ([2.23])

– Замените третий столбец матрицы A на сумму второго и третьего столбцов.

```
[106]: for i in 1:3
        A[i, 3] = A[i, 1] + A[i, 2]
    end
    A
```

```
[106]: 3x3 Matrix{Int64}:
      1  1  2
      5  2  7
     -2 -1 -3
```

Рис. 2.23: Задание 5. Пункт 3

6. Создадим матрицу ([2.24]) и вычислим ее произведение с транспонированной версией ([2.25]).

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

```
[118]: N = 15
        B = zeros{Float64}(N, 3)
        for i = 1:N
            B[i, 1] = 10
            B[i, 2] = -10
            B[i, 3] = 10
        end
        B

[118]: 15x3 Matrix{Float64}:
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
 10.0 -10.0 10.0
```

Рис. 2.24: Задание 6 (1).

```
[120]: C = B'*B
```

```
[120]: 15x15 Matrix{Float64}:
 1500.0 -1500.0 1500.0 -1500.0 ... -1500.0 1500.0 -1500.0 1500.0
-1500.0 1500.0 -1500.0 1500.0 1500.0 -1500.0 1500.0 -1500.0
 1500.0 -1500.0 1500.0 -1500.0 -1500.0 1500.0 -1500.0 1500.0
-1500.0 1500.0 -1500.0 1500.0 1500.0 -1500.0 1500.0 -1500.0
 1500.0 -1500.0 1500.0 -1500.0 -1500.0 1500.0 -1500.0 1500.0
-1500.0 1500.0 -1500.0 1500.0 ... 1500.0 -1500.0 1500.0 -1500.0
 1500.0 -1500.0 1500.0 -1500.0 -1500.0 1500.0 -1500.0 1500.0
-1500.0 1500.0 -1500.0 1500.0 1500.0 -1500.0 1500.0 -1500.0
 1500.0 -1500.0 1500.0 -1500.0 -1500.0 1500.0 -1500.0 1500.0
-1500.0 1500.0 -1500.0 1500.0 ... 1500.0 -1500.0 1500.0 -1500.0
 1500.0 -1500.0 1500.0 -1500.0 -1500.0 1500.0 -1500.0 1500.0
-1500.0 1500.0 -1500.0 1500.0 1500.0 -1500.0 1500.0 -1500.0
 1500.0 -1500.0 1500.0 -1500.0 -1500.0 1500.0 -1500.0 1500.0
```

Рис. 2.25: Задание 6 (2).

7. Создадим 4 матрицы с некоторыми закономерностями ([2.26-2.30]).

7. Создайте матрицу Z размерности 6×6 , 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 6×6 :

$$Z_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad Z_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad Z_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad Z_4 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

```
[129]: E = [1 for i in 1:6, j in 1:6]
```

```
[129]: 6x6 Matrix{Int64}:
```

```
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
```

```
[133]: Z = [0 for i in 1:6, j in 1:6]
```

```
[133]: 6x6 Matrix{Int64}:
```

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

Рис. 2.26: Задание 7.


```
[135]: Z_1 = Int.(zeros(6, 6))
      Z_2 = Int.(zeros(6, 6))
      Z_3 = Int.(zeros(6, 6))
      Z_4 = Int.(zeros(6, 6))
```

```
[135]: 6x6 Matrix{Int64}:
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
```

```
[141]: for i ∈ 1:6, j ∈ 1:6
      if (j == i + 1) || (j == i - 1)
          Z_1[i, j] = E[i, j]
      else
          Z_1[i, j] = Z[i, j]
      end
  end
Z_1
```

```
[141]: 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. 2.27: Задание 7. Матрица Z_1

```
[142]: for i ∈ 1:6, j ∈ 1:6
        if (j == i + 2) || (j == i - 2) || (j == i)
            Z_2[i, j] = E[i, j]
        else
            Z_2[i, j] = Z[i, j]
        end
    end
    Z_2
```

```
[142]: 6x6 Matrix{Int64}:
 1  0  1  0  0  0
 0  1  0  1  0  0
 1  0  1  0  1  0
 0  1  0  1  0  1
 0  0  1  0  1  0
 0  0  0  1  0  1
```

Рис. 2.28: Задание 7. Матрица Z_2

```
[144]: for i ∈ 1:6, j ∈ 1:6
        if (j == 7 - i) || (j == 5 - i) || (j == 9 - i)
            Z_3[i, j] = E[i, j]
        else
            Z_3[i, j] = Z[i, j]
        end
    end
    Z_3
```

```
[144]: 6x6 Matrix{Int64}:
 0  0  0  1  0  1
 0  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  0
 1  0  1  0  0  0
```

Рис. 2.29: Задание 7. Матрица Z_3

```
[146]: for i ∈ 1:6, j ∈ 1:6
        if abs(1 - j) == 0 || abs(1 - j) == 2 || abs(1 - j) == 4 # (j == i + 2) || (j == i - 2) || (j == i) || (j == i + 4) || (j == i - 4)
            Z_4[i, j] = E[i, j]
        else
            Z_4[i, j] = Z[i, j]
        end
    end
end
Z_4

[146]: 6x6 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. 2.30: Задание 7. Матрица Z_4

8. Написание функции `outer` (внешнее произведение, с возможностью замены операции) и построение матриц с ее помощью (задания написаны на скриншотах):

1) Пункт 1 ([2.31])

8. В языке R есть функция `outer()`. Фактически, это матричное умножение с возможностью изменить применяемую операцию (например, заменить произведение на сложение или возведение в степень).

– Напишите свою функцию, аналогичную функции `outer()` языка R. Функция должна иметь следующий интерфейс: `outer(x,y,operation)`. Таким образом, функция вида `outer(A,B,*)` должна быть эквивалентна произведению матриц A и B размерностями $L \times M$ и $M \times N$ соответственно, где элементы результирующей матрицы C имеют вид $C_{ij} = \sum_{k=1}^M A_{ik} B_{kj}$ (или в тензорном виде $C' = \sum_{k=1}^M A'_k B'_k$).

```
[168]: function outer(x, y, operation)
        Res = zeros(length(x), length(y))
        for i ∈ 1:length(x), j ∈ 1:length(y)
            Res[i, j] = operation(x[i], y[j])
        end
        return Res
    end

[168]: outer (generic function with 1 method)

[178]: x = [1, 2, 3];
        y = [4, 5, 6];
        outer(x, y, *)

[178]: 3x3 Matrix{Float64}:
 5.0  6.0  7.0
 6.0  7.0  8.0
 7.0  8.0  9.0
```

Рис. 2.31: Задание 8. Пункт 1. Написание функции `outer`

2) Пункт 2 — создаем матрицы разной структуры ([2.32-2.34])

– Используя написанную вами функцию `outer()`, создайте матрицы следующей структуры: $A_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 8 & 16 & 32 \\ 3 & 9 & 27 & 81 & 243 \\ 4 & 16 & 64 & 256 & 1024 \end{pmatrix}$, $A_3 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{pmatrix}$.

$A_4 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \\ : & : & : & : & : & : & : & : & : & : \\ 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$, $A_5 = \begin{pmatrix} 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 \\ 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 \\ 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{pmatrix}$

```
[176]: a_1 = [0, 1, 2, 3, 4]
        b_1 = [0, 1, 2, 3, 4]
        A_1 = outer(a_1, b_1, *)

[176]: 5x5 Matrix{Float64}:
 0.0  1.0  2.0  3.0  4.0
 1.0  2.0  3.0  4.0  5.0
 2.0  3.0  4.0  5.0  6.0
 3.0  4.0  5.0  6.0  7.0
 4.0  5.0  6.0  7.0  8.0
```

Рис. 2.32: Задание 8. Пункт 2. Матрица A_1

```
[179]: a_2 = a_1
b_2 = [1, 2, 3, 4, 5]
A_2 = outer(a_2, b_2, ^)
```

```
[179]: 5x5 Matrix{Float64}:
 0.0  0.0  0.0  0.0  0.0
 1.0  1.0  1.0  1.0  1.0
 2.0  4.0  8.0 16.0 32.0
 3.0  9.0 27.0 81.0 243.0
 4.0 16.0 64.0 256.0 1024.0
```

```
[190]: a_3 = [0, 1, 2, 3, 4]
b_3 = [0, 1, 2, 3, 4]
A_3 = outer(a_3, b_3, +) .%5
```

```
[190]: 5x5 Matrix{Float64}:
 0.0  1.0  2.0  3.0  4.0
 1.0  2.0  3.0  4.0  0.0
 2.0  3.0  4.0  0.0  1.0
 3.0  4.0  0.0  1.0  2.0
 4.0  0.0  1.0  2.0  3.0
```

Рис. 2.33: Задание 8. Пункт 2. Матрицы A_2 и A_3

```
[192]: a_4 = [i for i in range(0,9)]
b_4 = [i for i in range(0,9)]
A_4 = outer(a_4, b_4, '+').%10
```

```
[192]: 10x10 Matrix{Float64}:
 0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0
 1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0
 2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0
 3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0
 4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0
 5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0
 6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0
 7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0
 8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0
 9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0
```

```
[199]: a_5 = [i for i in range(0,8)]
b_5 = [i for i in range(9,-1,1)]
A_5 = (outer(a_5, b_5, '+')).%9
```

```
[199]: 9x9 Matrix{Float64}:
 0.0  8.0  7.0  6.0  5.0  4.0  3.0  2.0  1.0
 1.0  0.0  8.0  7.0  6.0  5.0  4.0  3.0  2.0
 2.0  1.0  0.0  8.0  7.0  6.0  5.0  4.0  3.0
 3.0  2.0  1.0  0.0  8.0  7.0  6.0  5.0  4.0
 4.0  3.0  2.0  1.0  0.0  8.0  7.0  6.0  5.0
 5.0  4.0  3.0  2.0  1.0  0.0  8.0  7.0  6.0
 6.0  5.0  4.0  3.0  2.0  1.0  0.0  8.0  7.0
 7.0  6.0  5.0  4.0  3.0  2.0  1.0  0.0  8.0
 8.0  7.0  6.0  5.0  4.0  3.0  2.0  1.0  0.0
```

Рис. 2.34: Задание 8. Пункт 2. Матрицы A_4 и A_5

9. Решим систему уравнений определенного вида. Составим матрицу коэффициентов ([2.35]), реализуем метод Гаусса ([2.36]) и найдем решение системы уравнений ([2.37]).

9. Решите следующую систему линейных уравнений с 5 неизвестными:

$$\begin{aligned}x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 &= 7, \\2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 &= -1, \\3x_1 + 2x_2 + x_3 + 2x_4 + 3x_5 &= -3, \\4x_1 + 3x_2 + 2x_3 + x_4 + 2x_5 &= 5, \\5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 &= 17,\end{aligned}$$

рассмотрев соответствующее матричное уравнение $Ax = y$. Обратите внимание на особый вид матрицы A . Метод, используемый для решения данной системы уравнений, должен быть легко обобщаем на случай большого числа уравнений, где матрица A будет иметь такую же структуру.

```
[349]: M = 5
A = zeros(M, M)
for k in 1:M
    for i in k:M
        A[k, i] = i - k + 1
        A[i, k] = i - k + 1
    end
end
A

[349]: 5x5 Matrix{Float64}:
 1.0  2.0  3.0  4.0  5.0
 2.0  1.0  2.0  3.0  4.0
 3.0  2.0  1.0  2.0  3.0
 4.0  3.0  2.0  1.0  2.0
 5.0  4.0  3.0  2.0  1.0

[350]: b = [7, -1, -3, 5, 17]

[350]: 5-element Vector{Int64}:
 7
 -1
 -3
 5
 17
```

Рис. 2.35: Задание 9. Матрица коэффициентов

```
[351]: function Gauss_Straight(A)
    for i in 1:(size(A)[1])
        for k in (i+1):(size(A)[2]-1)
            A[k, :] = A[k, :] - A[i, :]*(A[k, i] / A[i, i])
        end
    end
    return A
end

function Gauss_Reverse(_A)
    x = zeros(size(_A)[1])
    for i in (size(_A)[1]):-1:1
        b = _A[i, size(_A)[2]]
        for j in (size(_A)[1]):-1:i
            if j != i
                b -= x[j]*_A[i, j]
            end
        end
        x[i] = b / _A[i, i]
    end
    return x
end

function Gauss(A, b)
    A_ = hcat(A, b)
    _A = Gauss_Straight(A_)
    X = Gauss_Reverse(_A)
    return X
end
```

[351]: Gauss (generic function with 2 methods)

Рис. 2.36: Задание 9. Метод Гаусса

```
[352]: _A_ = hcat(A, b)
        #print(size(_A_))
        A__ = Gauss_Straight(_A_)

[352]: 5x6 Matrix{Float64}:
        1.0  2.0  3.0  4.0  5.0  7.0
        0.0 -3.0 -4.0 -5.0 -6.0 -15.0
        0.0  0.0 -2.66667 -3.33333 -4.0 -4.0
        0.0  0.0  0.0 -2.5 -3.0  7.0
        0.0  0.0  0.0  0.0 -2.4  9.6

[353]: X = Gauss_Reverse(A__)

[353]: 5-element Vector{Float64}:
        -2.0000000000000044
         3.0000000000000002
         4.9999999999999964
         2.0000000000000044
        -4.0000000000000002

[354]: x = round.(X)

[354]: 5-element Vector{Float64}:
        -2.0
         3.0
         5.0
         2.0
        -4.0

[355]: x2 = round.(Gauss(A, b))

[355]: 5-element Vector{Float64}:
        -2.0
         3.0
         5.0
         2.0
        -4.0
```

Рис. 2.37: Задание 9. Решение системы уравнений

10. Создадим матрицу со случайными целыми числами ([2.38]) и проведем с ней несколько операций (задания написаны на скриншотах):

10. Создайте матрицу M размерности 6×10 , элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности $1, 2, \dots, 10$.

```
[260]: M = rand(1:10, (6, 10))

[260]: 6x10 Matrix{Int64}:
 7 10 1 1 7 4 1 6 2 1
 7 4 6 6 6 8 5 5 3 3
 6 8 9 8 7 4 2 8 4 2
10 10 7 7 10 7 4 3 5 5
 6 1 7 3 3 2 5 3 5 3
 8 6 8 3 10 6 3 5 6 1
```

Рис. 2.38: Задание 10. Создание матрицы со случайными целыми числами

1) Пункт 1 ([2.39])

– Найдите число элементов в каждой строке матрицы M , которые больше числа N_1 (например, $N_1 = 4$).

```
[261]: N1 = 4
Numbers_1 = Int.(zeros(6))
for i ∈ 1:6
    for j ∈ 1:10
        if M[i, j] > N1
            Numbers_1[i] += 1
        end
    end
end
Numbers_1

[261]: 6-element Vector{Int64}:
 4
 7
 6
 8
 4
 7
```

Рис. 2.39: Задание 10. Пункт 1.

2) Пункт 2 ([2.40])

– Определите, в каких строках матрицы M число N_2 (например, $N_2 = 7$) встречается ровно 2 раза?

```
[264]: N2 = 7
Numbers_2 = Int.(zeros(6))
for i ∈ 1:6
    for j ∈ 1:10
        if M[i, j] == N2
            Numbers_2[i] += 1
        end
    end
end
print(Numbers_2)
findall(x -> x == 2, Numbers_2)

[2, 1, 1, 3, 1, 0]

[264]: 1-element Vector{Int64}:
 1
```

Рис. 2.40: Задание 10. Пункт 2.

3) Пункт 3 ([2.41])

– Определите все пары столбцов матрицы M , сумма элементов которых больше K (например, $K = 75$).

```
[263]: K = 75
Numbers_3 = []
for i ∈ 1:9
    for j ∈ (i+1):10
        if (sum(M[i, i]) + sum(M[i, j]) > K)
            push!(Numbers_3, (i, j))
        end
    end
end
Numbers_3
```

```
[263]: 6-element Vector{Any}:
 (1, 2)
 (1, 3)
 (1, 5)
 (2, 3)
 (2, 5)
 (3, 5)
```

Рис. 2.41: Задание 10. Пункт 3.

11. Найдем две суммы ([2.42,2.43]).

11. Вычислите:

$$\sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+j)}$$

```
[265]: S_1 = 0
for i ∈ 1:20
    for j ∈ 1:5
        S_1 += i^4 / (3 + j)
    end
end
S_1
```

```
[265]: 639215.2833333334
```

Рис. 2.42: Задание 11. Пункт 1

$$\sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+ij)}$$

```
[266]: S_2 = 0
        for i ∈ 1:20
            for j ∈ 1:5
                S_2 += i^4 / (3 + i*j)
            end
        end
        S_2
```

```
[266]: 89912.02146097136
```

Рис. 2.43: Задание 11. Пункт 2

3 Выводы

В ходе выполнения лабораторной работы я освоил работу с циклами и сторонние библиотеки Julia для решения задач линейной алгебры и работы с матрицами.

Список литературы

1. Королькова А. В., Кулябов Д. С. Лабораторная работа № 3. Управляющие структуры [Электронный ресурс]. RUDN, 2023. URL: https://esystem.rudn.ru/pluginfile.php/2231345/mod_resource/content/2/003-lab_control-structures.pdf.