

Компьютерный практикум по статистическому анализу данных лабораторная работа №3

Управляющие структуры

Ким Илья Владиславович НФИбд-01-21

Содержание

Цель работы	3
Задание	4
Выполнение работы. Повтор примеров из раздела 3.2	5
3.2.1. Циклы While и for	5
3.2.2. Условные выражения	8
3.2.3. Функции	9
3.2.4. Сторонние библиотеки (пакеты) в Julia	13
3.4 Задания для самостоятельного выполнения	15
Задание 1	15
Задание 2	17
Задание 3	17
Задание 4	17
Задание 5	18
Задание 6	18
Задание 7	19
Задание 10	21
Задание 11	23
Выводы	24

Цель работы

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

Задание

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

Выполнение работы. Повтор примеров из раздела 3.2

3.2.1. Циклы While и for

1. Цикл, который пока $n < 10$ прибавляет к n единицу и печатает значение

3.2.1 Циклы While и for

```
[1]: n = 0
while n<10
  n+=1
  println(n)
end
```

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

```
[10]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
```

```
i = 1
while i <= length(myfriends)
  friend = myfriends[i]
  println("Hi $friend, it's gear to see you!")
  i+=1
end
```

```
Hi Ted, it's gear to see you!
Hi Robyn, it's gear to see you!
Hi Barney, it's gear to see you!
Hi Lily, it's gear to see you!
Hi Marshall, it's gear to see you!
```

2. Задали массив со строковыми элементами. Далее в цикле подставляется имя из массива в заданную строку и выводит на экран.

```

6]: for n in 1:2:10
    println(n)
end
myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]

for friend in myfriends
    println("Hi $friend, it's gear to see you!")
end

1
3
5
7
9
Hi Ted, it's gear to see you!
Hi Robyn, it's gear to see you!
Hi Barney, it's gear to see you!
Hi Lily, it's gear to see you!
Hi Marshall, it's gear to see you!

0]: m,n = 5, 5
A=fill(0,(m,n))

for i in 1:m
    for j in 1:n
        A[i,j] = i+j
    end
end
A

0]: 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

```

3. Использование цикла for для создание двумерного массива, в котором значение каждой записи является суммой индексов строки и столбца

```
4]: B = fill(0,(m,n))
```

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

```
4]: 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
```

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

```
8]: 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
```

3.2.2. Условные выражения

4. Пример, в котором используется условие: если N делится на 3 выводится “Fizz”, если на 5 “Buzz”, если на 3 и на 5 “FizzBuzz”

3.2.2 Условные выражения

```
[48]: 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
```

3.2.3. Функции

5. Написали функцию, при вызове которой подставляет принятое имя в строку.
Вторая функция - функция возведения в квадрат

3.2.3. Функции

```
[61]: function sayhi(name)
      println("Hi $name, it's great to see you!")
      end

[61]: sayhi (generic function with 1 method)

[63]: sayhi("Ilya")
      hi Ilya, it's great to see you!

[65]: function f(x)
      x^2
      end

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

[67]: f(5)

[67]: 25

[71]: sayhi2(name) = println("Hi $name, it's great to see you!")

[71]: sayhi2 (generic function with 1 method)

[75]: sayhi2("Ilya")
      Hi Ilya, it's great to see you!

[77]: f2(x) = x^2

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

[79]: f2(2)

[79]: 4

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

[81]: #9 (generic function with 1 method)

[83]: sayhi3("Ilya")
      Hi Ilya, it's great to see you!

[89]: f3(6)

[89]: 36
```

6. Задал массив `v` и отсортировал его функцией `sort(v)` и `sort!(v)`. Разница в них была в том, что `sort!(v)` меняли значения в нашем массиве, а `sort(v)` - нет.

Функция `map()` применила функцию `f` ко всем элементам массива

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

[105]: 3-element Vector{Int64}:
      3
      5
      2

[101]: sort!(v)
      v

[101]: 3-element Vector{Int64}:
      2
      3
      5

[107]: map(f, [1,2,3])

[107]: 3-element Vector{Int64}:
      1
      4
      9

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

[111]: 3-element Vector{Int64}:
      1
      8
      27
```

7. Применил функцию `broadcast`, которая привела все объекты к общему измерению, в отличие от функции `map`, которая напрямую применяет функцию поэлементно.

Задал матрицу `A` и применили к ней функцию `f(A)` возведения в квадрат, получил `A*A`.

Далее применил функцию `f.(A)`, получил матрицу, где каждый элемент возведен в квадрат.

```

[119]: broadcast(f,[1,2,3])

[119]: 3-element Vector{Int64}:
       1
       4
       9

[121]: f.([1,2,3])

[121]: 3-element Vector{Int64}:
       1
       4
       9

[125]: A = [i+3*j for j in 0:2, i in 1:3]

[125]: 3x3 Matrix{Int64}:
       1  2  3
       4  5  6
       7  8  9

[127]: f(A)

[127]: 3x3 Matrix{Int64}:
      30  36  42
      66  81  96
     102 126 150

[129]: B = f.(A)

[129]: 3x3 Matrix{Int64}:
       1  4  9
      16 25 36
      49 64 81

```

8. Попробовал пример записей с синтаксисом broadcast.

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

```
[135]: 3x3 Matrix{Float64}:  
  3.0  6.0  9.0  
 12.0 15.0 18.0  
 21.0 24.0 27.0
```

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

```
[137]: 3x3 Matrix{Float64}:  
  3.0  6.0  9.0  
 12.0 15.0 18.0  
 21.0 24.0 27.0
```

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

```
[139]: 3x3 Matrix{Float64}:  
  3.0  6.0  9.0  
 12.0 15.0 18.0  
 21.0 24.0 27.0
```

3.2.4 Сторонние библиотеки (пакеты) в Julia

9. Загрузил пакеты и попробовал их использовать

3.2.4. Сторонние библиотеки (пакеты) в Julia

```
[142]: import Pkg
```

```
[144]: Pkg.add("Example")
```

```
Updating registry at `C:\Users\ksudz\.julia\registries\General.toml`  
Resolving package versions...  
Installed Example - v0.5.5  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Project.toml`  
[707fa107] + Example v0.5.5  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Manifest.toml`  
[707fa107] + Example v0.5.5  
Precompiling project...  
✓ Example  
1 dependency successfully precompiled in 1 seconds. 17 already precompiled.
```

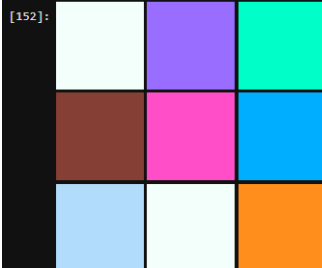
```
[146]: Pkg.add("Colors")  
using Colors
```

```
Resolving package versions...  
Installed FixedPointNumbers - v0.8.5  
Installed ColorTypes - v0.12.0  
Installed Colors - v0.13.0  
Installed Reexport - v1.2.2  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Project.toml`  
[56e09095] + Colors v0.13.0  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Manifest.toml`  
[56e09095] + ColorTypes v0.12.0
```

```
[148]: palette = distinguishable_colors(100)
```



```
[152]: rand(palette, 3, 3)
```



3.4 Задания для самостоятельного выполнения

Задание 1

1.1 Использовал цикл for для печати чисел от 1 до 100 и их квадратов.

```
1. Используя циклы while и for:

1.1. Выведите на экран целые числа от 1 до 100 и напечатайте их квадраты

[5]: for i in range(1, 101):
      print(i, i**2)

1 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 1936 2025 2116 2209 2304 2401 2500 2601 2704 2809 2916 3025 3136 3249 3364 3481 3600 3721 3844 3969 4096 4225 4356 4489 4624 4761 4900 5041 5184 5329 5476 5625 5776 5929 6084 6241 6400 6561 6724 6891 7056 7225 7396 7569 7744 7921 8100 8281 8464 8649 8836 9025 9216 9409 9604 9801 10000

[5]: i = 1
      while i <= 100:
          print(i, i**2)
          i += 1

1 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 1936 2025 2116 2209 2304 2401 2500 2601 2704 2809 2916 3025 3136 3249 3364 3481 3600 3721 3844 3969 4096 4225 4356 4489 4624 4761 4900 5041 5184 5329 5476 5625 5776 5929 6084 6241 6400 6561 6724 6891 7056 7225 7396 7569 7744 7921 8100 8281 8464 8649 8836 9025 9216 9409 9604 9801 10000
```

1.2 Объявил, что squares - это словарь. Использовал цикл for от 1 до 100, где присвоил целые числа к ключам, а их квадраты в качестве пар-значений.

```
1.2. Создайте словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений

[5]: squares = {}
      for i in range(1, 101):
          squares[i] = i**2
      squares

[5]: Dict({'key': 'key'}) with 100 entries:
    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
    25 -> 625
    26 -> 676
    27 -> 729
    28 -> 784
    29 -> 841
    30 -> 900
    31 -> 961
    32 -> 1024
    33 -> 1089
    34 -> 1156
    35 -> 1225
    36 -> 1296
    37 -> 1369
    38 -> 1444
    39 -> 1521
    40 -> 1600
    41 -> 1681
    42 -> 1764
    43 -> 1849
    44 -> 1936
    45 -> 2025
    46 -> 2116
    47 -> 2209
    48 -> 2304
    49 -> 2401
    50 -> 2500
    51 -> 2601
    52 -> 2704
    53 -> 2809
    54 -> 2916
    55 -> 3025
    56 -> 3136
    57 -> 3249
    58 -> 3364
    59 -> 3481
    60 -> 3600
    61 -> 3721
    62 -> 3844
    63 -> 3969
    64 -> 4096
    65 -> 4225
    66 -> 4356
    67 -> 4489
    68 -> 4624
    69 -> 4761
    70 -> 4900
    71 -> 5041
    72 -> 5184
    73 -> 5329
    74 -> 5476
    75 -> 5625
    76 -> 5776
    77 -> 5929
    78 -> 6084
    79 -> 6241
    80 -> 6400
    81 -> 6561
    82 -> 6724
    83 -> 6891
    84 -> 7056
    85 -> 7225
    86 -> 7396
    87 -> 7569
    88 -> 7744
    89 -> 7921
    90 -> 8100
    91 -> 8281
    92 -> 8464
    93 -> 8649
    94 -> 8836
    95 -> 9025
    96 -> 9216
    97 -> 9409
    98 -> 9604
    99 -> 9801
    100 -> 10000
```

1.2.1 Сделал тоже самое, используя цикл while

```
[23]: squares1 = Dict()
      i=1
      while i!=101
          squares1[i]=i^2
          i+=1
      end
      squares1
```

```
[23]: 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
⋮  => ⋮
```

1.3 Задал массив нулей `squares_arr` размерностью 100, использовал цикл `for` для возведения `i` в квадрат, записи в массив и вывода на экран. Сделал тоже самое, используя цикл `while`.


```
13. Создайте массив squares_arr, содержащий квадраты всех чисел от 1 до 100

[7]: squares_arr = zeros(1004,100)
for i in 1:100
    squares_arr[i, :] = i^2
    println(squares_arr[i, :])
end

[8]: squares_arr = zeros(1004,100)
for i in 1:100
    squares_arr[i, :] = i^2
    println(squares_arr[i, :])
    i+=1
end

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 1936 2025 2116 2209 2304 2401 2500 2601 2704 2809 2916 3025 3136 3249
9 100 400 900 1600 2500 3600 4900 6400 8100 10000 12100 14400 16900 19600 22500 25600 28900 32400 36100 40000 44100 48400 52900 57600 62500 67600 72900 78400 84100 90000 96100 102400 108900 115600 122500 129600 136900 144400 152100 160000 168100 176400 184900 193600 202500 211600 220900 230400 240100 250000 260100 270400 280900 291600 302500 313600 324900
81 324 648 972 1296 1620 1944 2268 2592 2916 3240 3564 3888 4212 4536 4860 5184 5508 5832 6156 6480 6804 7128 7452 7776 8100 8424 8748 9072 9396 9720 10044 10368 10692 11016 11340 11664 11988 12312 12636 12960 13284 13608 13932 14256 14580 14904 15228 15552 15876 16200 16524 16848 17172 17496 17820 18144 18468 18792 19116 19440 19764 20088 20412 20736 21060 21384 21708 22032 22356 22680 23004 23328 23652 23976 24300 24624 24948 25272 25596 25920 26244 26568 26892 27216 27540 27864 28188 28512 28836 29160 29484 29808 30132 30456 30780 31104 31428 31752 32076 32400 32724 33048 33372 33696 34020 34344 34668 34992 35316 35640 35964 36288 36612 36936 37260 37584 37908 38232 38556 38880 39204 39528 39852 40176 40500 40824 41148 41472 41796 42120 42444 42768 43092 43416 43740 44064 44388 44712 45036 45360 45684 46008 46332 46656 46980 47304 47628 47952 48276 48600 48924 49248 49572 49896 50220 50544 50868 51192 51516 51840 52164 52488 52812 53136 53460 53784 54108 54432 54756 55080 55404 55728 56052 56376 56700 57024 57348 57672 57996 58320 58644 58968 59292 59616 59940 60264 60588 60912 61236 61560 61884 62208 62532 62856 63180 63504 63828 64152 64476 64800 65124 65448 65772 66096 66420 66744 67068 67392 67716 68040 68364 68688 69012 69336 69660 69984 70308 70632 70956 71280 71604 71928 72252 72576 72900 73224 73548 73872 74196 74520 74844 75168 75492 75816 76140 76464 76788 77112 77436 77760 78084 78408 78732 79056 79380 79704 80028 80352 80676 81000 81324 81648 81972 82296 82620 82944 83268 83592 83916 84240 84564 84888 85212 85536 85860 86184 86508 86832 87156 87480 87804 88128 88452 88776 89100 89424 89748 90072 90396 90720 91044 91368 91692 92016 92340 92664 92988 93312 93636 93960 94284 94608 94932 95256 95580 95904 96228 96552 96876 97200 97524 97848 98172 98496 98820 99144 99468 99792 100116 100440 100764 101088 101412 101736 102060 102384 102708 103032 103356 103680 104004 104328 104652 104976 105300 105624 105948 106272 106596 106920 107244 107568 107892 108216 108540 108864 109188 109512 109836 110160 110484 110808 111132 111456 111780 112104 112428 112752 113076 113400 113724 114048 114372 114696 115020 115344 115668 115992 116316 116640 116964 117288 117612 117936 118260 118584 118908 119232 119556 119880 120204 120528 120852 121176 121500 121824 122148 122472 122796 123120 123444 123768 124092 124416 124740 125064 125388 125712 126036 126360 126684 127008 127332 127656 127980 128304 128628 128952 129276 129600 129924 130248 130572 130896 131220 131544 131868 132192 132516 132840 133164 133488 133812 134136 134460 134784 135108 135432 135756 136080 136404 136728 137052 137376 137700 138024 138348 138672 138996 139320 139644 139968 140292 140616 140940 141264 141588 141912 142236 142560 142884 143208 143532 143856 144180 144504 144828 145152 145476 145800 146124 146448 146772 147096 147420 147744 148068 148392 148716 149040 149364 149688 150012 150336 150660 150984 151308 151632 151956 152280 152604 152928 153252 153576 153900 154224 154548 154872 155196 155520 155844 156168 156492 156816 157140 157464 157788 158112 158436 158760 159084 159408 159732 160056 160380 160704 161028 161352 161676 162000 162324 162648 162972 163296 163620 163944 164268 164592 164916 165240 165564 165888 166212 166536 166860 167184 167508 167832 168156 168480 168804 169128 169452 169776 170100 170424 170748 171072 171396 171720 172044 172368 172692 173016 173340 173664 173988 174312 174636 174960 175284 175608 175932 176256 176580 176904 177228 177552 177876 178200 178524 178848 179172 179496 179820 180144 180468 180792 181116 181440 181764 182088 182412 182736 183060 183384 183708 184032 184356 184680 185004 185328 185652 185976 186300 186624 186948 187272 187596 187920 188244 188568 188892 189216 189540 189864 190188 190512 190836 191160 191484 191808 192132 192456 192780 193104 193428 193752 194076 194400 194724 195048 195372 195696 196020 196344 196668 196992 197316 197640 197964 198288 198612 198936 199260 199584 199908 200232 200556 200880 201204 201528 201852 202176 202500 202824 203148 203472 203796 204120 204444 204768 205092 205416 205740 206064 206388 206712 207036 207360 207684 208008 208332 208656 208980 209304 209628 209952 210276 210600 210924 211248 211572 211896 212220 212544 212868 213192 213516 213840 214164 214488 214812 215136 215460 215784 216108 216432 216756 217080 217404 217728 218052 218376 218700 219024 219348 219672 220000
```

Задание 2

2. Использовал цикл for от 1 до 10 с условием: если остаток от деления i на 2 равен нулю, то выводит на экран, что число i чётное, в обратном случае - нечётное

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

for i in 1:10
    if i % 2 == 0
        println("Число ", i, " - чётное")
    else
        println("Число ", i, " - нечётное")
    end
end

MacOS 1 - нечётное
MacOS 2 - чётное
MacOS 3 - нечётное
MacOS 4 - чётное
MacOS 5 - нечётное
MacOS 6 - чётное
MacOS 7 - нечётное
MacOS 8 - чётное
MacOS 9 - нечётное
MacOS 10 - чётное
```

Задание 3

3. Написал функцию add_one, которая прибавляет единицу к своему входному значению.

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

[47]: function add_one(x)
        x+=1
    end
    add_one(5)

[47]: 6
```

Задание 4

4. Задал матрицу A размерностью 3x3, использовал функцию broadcast, в которой прибавил ко всем элементам матрицы единицу. Сделал тоже самое, используя функцию map() и add_one().

```

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

[10]: A = [[1]] for j in range(3): A = A + 1
[10]: 3x3 Matrix{Int64}:
 1 2 3
 4 5 6
 7 8 9

[10]: broadcast(ones, A)
[10]: 3x3 Matrix{Int64}:
 2 3 4
 5 6 7
 8 9 10

[10]: mapadd_ones(A)
[10]: 3x3 Matrix{Int64}:
 2 3 4
 5 6 7
 8 9 10

```

Задание 5

5. Задал матрицу A и функцию возведения в куб f, применил функцию f(A) и получил матрицу нулей. Во втором пункте использовал цикл for от 3 до 9 с шагом 3, в котором присваивал к 3, 6 и 9 элементам сумму первого и второго.

5. Задайте матрицу A следующего вида:

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

- Найдите A^3

```

106]: A = [[1, 5, -2] [1, 2, -1] [3, 6, -3]]
      f(x)=x^3
      f(A)

106]: 3x3 Matrix{Int64}:
 0 0 0
 0 0 0
 0 0 0

```

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

```

123]: for i in 3:2:9
      A[i] = A[i-2] + A[i-1]
      end
      A

123]: 3x3 Matrix{Int64}:
 1 1 3
 5 2 6
 6 3 9

```

Задание 6

- 6.. Создал матрицу B используя функцию repeat, и нашел произведение транспонированной B на B

```

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

[179]: 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

[181]: B

[181]: 3x15 adjoint(::Matrix{Int64}) with eltype 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

[183]: C=B*B

[183]: 3x3 Matrix{Int64}:
1500 1500 1500
1500 1500 1500
1500 1500 1500

```

Задание 7

7.1 Создал матрицу Z размерности 6х6, где все элементы равны нулю и матрицу E, где все элементы равны 1

```

7. Создайте матрицу Z размерности 6 × 6, все элементы которой равны нулю, и матрицу E, все элементы которой равны 1.
[194]: Z = zeros{(),6}
display(Z)
Z = zeros{(),6}

6x6 Matrix{Float64}:
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
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

[194]: 6x6 Matrix{Float64}:
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0

```

7.2 Задал 4 матрицы заполненные нулями размерностью 6х6, далее нашел закономерности, по которым их нужно привести к нужным матрицам, и используя цикл for и условие if выполнил задание.

```

Используя цикл while или for и закономерности расположения элементов, создайте матрицы размерности 6 × 6:


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



$$Z_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad Z_4 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$


[255]: Z1 = zeros{Int64,6,6}
Z2 = zeros{Int64,6,6}
Z3 = zeros{Int64,6,6}
Z4 = zeros{Int64,6,6}
for i in 1:6, j in 1:6
    if i == j+1 || i == j-1
        Z1[i,j] = 1
    end
end
Z1

[255]: 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

```

```
[261]: for i in 1:6, j in 1:6
        if i==j || i==j+2 || i==j-2
            Z2[i,j] = 1
        end
    end
end
Z2
```

```
[261]: 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
```

```
[259]: for i in 1:6, j in 1:6
        if i==7-j || i==5-j || i== 9-j
            Z3[i,j] = 1
        end
    end
end
Z3
```

```
[259]: 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
```

```
[263]: for i in 1:6, j in 1:6
        if i%2!=0 && j%2!=0
            Z4[i,j] = 1
        elseif i%2==0 && j%2==0
            Z4[i,j] = 1
        end
    end
end
Z4
```

```
[263]: 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
```

Задание 10

10.1 Используя функцию `rand`, создал матрицу `M` размерности 6×10 , элементами которой являются целые числа, выбранные случайным образом из диапазона $1, 2, \dots, 10$.

Используя цикл `for` перебрал всю матрицу и нашел количество элементов, которые больше 4

```
10. Создайте матрицу M размерности 6 x 10, элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности 1, 2, ..., 10.
Найдите число элементов в каждой строке матрицы M, которые больше числа N (например, N = 4).

[366]: M = randi([1,2,3,4,5,6,7,8,9,10],6,10)

[366]:
ans =
     3     8     1     7    10     6     3     6     7
     1     2     6     6     9     4     7    10     4
     5     3     1     2     8    10     7
     9     4     3     9     6     5     4     7     5
     1     7     8     6     9     6     7     9     8
     4     2     3     5     2     9     2     5     9

[370]: k=0;
N=4;
for i=1:size(M)[1], j=1:size(M)[2]
    if M[i,j] > N
        k=k+1;
    end
end
k

[370]: 34
```

10.2 Задал $P=7$, использовал цикл `for` от 1 до размерности матрицы по строкам (6), ввел счетчик `z`, который будет обнуляться при движении `i`, ввел второй цикл `j` от 1 до размерности матрицы по столбцам (10), ввел условие, что если элемент матрицы равен P , то срабатывает счетчик. После того, как мы прошли один круг цикла `j` проверяется условие, что если $z==2$, то напечатать строку. И так проверяется каждая строка.

```
Определите, в каких строках матрицы M число P (например, P = 7) встречается ровно 2 раза?

[374]: P=7;
for i=1:size(M)[1]
    z=0;
    for j=1:size(M)[2]
        if M[i,j] == P
            z=z+1;
        end
    end
    if z==2
        println("В строке №",i, " число ", P, " встречается ровно 2 раза")
    end
end

В строке №1 число 7 встречается ровно 2 раза
В строке №5 число 7 встречается ровно 2 раза
```

10.3 Использовал цикл `for` для нахождения суммы элементов каждого столбца, далее присваиваю это значение к новому массиву `A` и веду счетчик для порядка элементов в массиве `A`. После использую еще один цикл, в котором перебираю элементы массива `A` и ищу пары столбцов, сумма элементов которых будет больше 75, если она больше, то печатает на экран.

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

```
[475]: K = 75
z=1
A = zeros(Int64,size(M)[2])
display(M)
for i in 1:size(M)[2]
    G=0
    for j in 1:size(M)[1]
        G = G+M[j,i]
    end
    A[z]=G
    z+=1
end
for i in 1:size(A)[1]
    for j in 2:size(A)[1]
        if A[i]+A[j]>75
            println("Сумма столбцов №",i," и №",j," больше 75")
        end
    end
end
end
```

```
6x10 Matrix{Int64}:
 3  8  3  3  7 10  6  3  6  7
 1  3  2  6  6  9  4  7 10  4
 6  3  4  1  1  2  8  8 10  7
 9  4  8  3  9  6  5  4  7  5
 1  7  4 10  6  8  8  7  9  8
 4  2  3  5  2  9  2  3  9  2
Сумма столбцов №2 и №9 больше 75
Сумма столбцов №4 и №9 больше 75
Сумма столбцов №5 и №9 больше 75
Сумма столбцов №6 и №6 больше 75
Сумма столбцов №6 и №7 больше 75
Сумма столбцов №6 и №8 больше 75
Сумма столбцов №6 и №9 больше 75
Сумма столбцов №6 и №10 больше 75
Сумма столбцов №7 и №6 больше 75
Сумма столбцов №7 и №9 больше 75
Сумма столбцов №8 и №6 больше 75
Сумма столбцов №8 и №9 больше 75
Сумма столбцов №9 и №2 больше 75
Сумма столбцов №9 и №4 больше 75
Сумма столбцов №9 и №5 больше 75
Сумма столбцов №9 и №6 больше 75
Сумма столбцов №9 и №7 больше 75
Сумма столбцов №9 и №8 больше 75
Сумма столбцов №9 и №9 больше 75
Сумма столбцов №9 и №10 больше 75
Сумма столбцов №10 и №6 больше 75
Сумма столбцов №10 и №9 больше 75
```

Задание 11

11. Вычислите:

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

```
[471]: sum([sum([i^4/(3+j)] for j in 1:5) for i in 1:20])
```

```
[471]: 1-element Vector{Float64}:  
 639215.2833333334
```

```
[473]: sum([sum([i^4/(3+i*j)] for j in 1:5) for i in 1:20])
```

```
[473]: 1-element Vector{Float64}:  
 89912.02146097136
```

Выводы

Используя Jupyter lab повторил примеры из раздела 3.2 и выполнил задания для самостоятельной работы.