Лабораторная работа №3.

Управляющие структуры

Тазаева А. А.

Российский университет дружбы народов, Москва, Россия

Цели работы

Цели работы

Освоить применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

Задание

Задание

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

Циклы while и for

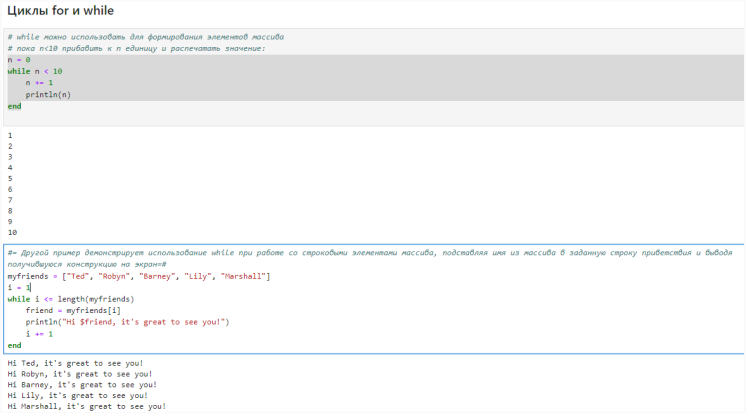
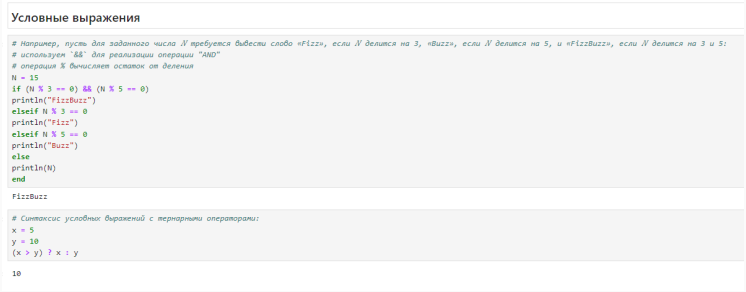


Рис. 1: Циклы while и for. Примеры. Часть 1

Условные выражения



Условные выражения

```
# Например, пусть для заданного числа N требуется вывести слово «Fizz», если N делится на 3, «Buzz», если N делится на 5, и «FizzBuzz», если N делится на 3 и 5:
# 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
```

```
# Синтаксис условных выражений с тернарными операторами:
x = 5
y = 10
(x > y) ? x : y
```

```
10
```

Рис. 2: Условные выражения. Примеры

Функции

```
#Julia дает нам несколько разных способов написать функцию. Первый требует ключевых слов function и end:
function sayhi(name)
    println("Hi $name, it's great to see you!")
end
# функция возведения в квадрат:
function f(x)
    x^2
end
sayhi("Nastya")
f(9)
```

```
Hi Nastya, it's great to see you!
81
```

```
# В качестве альтернативы, можно объявить любую из выше определённых функций в одной строке:
sayhi2(name) = println("Hi $name, it's great to see you!")
f2(x) = x^2
sayhi("Nastya")
f(9)
```

```
Hi Nastya, it's great to see you!
81
```

```
# Наконец, можно объявить выше определённые функции как «анонимные»:
sayhi3 = name -> println("Hi $name, it's great to see you!")
f3 = x -> x^2
sayhi("Nastya")
f(9)
```

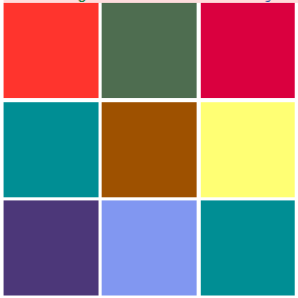
```
Hi Nastya, it's great to see you!
81
```


Рис. 3: Функции. Примеры. Часть 1

Сторонние библиотеки (пакеты) в Julia

```
import Pkg
Pkg.add("Example")
Pkg.add("Colors")
using Colors
palette = distinguishable_colors(100)
rand(palette, 3, 3)
```

```
  Resolving package versions...
 No Changes to `C:\Users\noname\.julia\environments\v1.10\Project.toml`
 No Changes to `C:\Users\noname\.julia\environments\v1.10\Manifest.toml`
  Resolving package versions...
 No Changes to `C:\Users\noname\.julia\environments\v1.10\Project.toml`
 No Changes to `C:\Users\noname\.julia\environments\v1.10\Manifest.toml`
```



Самостоятельная работа

Самостоятельная работа

Задание 1. Используя циклы while и for выведите на экран целые числа от 1 до 100 и напечатайте их квадраты

```
n = 0
while n < 100
    n += 1
    print(n, "^2 = ", n*n, " | ")
end
```

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

```
for n in 1:1:100
    print(n, "^2 = ", n*n, " | ")
end
```

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

Рис. 5: Самостоятельная работа. Задание 1.1

Самостоятельная работа

Задание 1. Используя циклы while и for создайте словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений

```
squares = Dict()
n = 0
while n<10
    n += 1
    squares[n] = n^n
end
pairs(squares)
```

```
Dict{Any, Any} with 10 entries:
  5  => 25
  4  => 16
  6  => 36
  7  => 49
  2  => 4
  10 => 100
  9  => 81
  8  => 64
  3  => 9
  1  => 1
```

```
for n in 1:10
    squares[n] = n^n
end
pairs(squares)
```

```
Dict{Any, Any} with 10 entries:
  5  => 25
  4  => 16
  6  => 36
  7  => 49
  2  => 4
  10 => 100
  9  => 81
  8  => 64
  3  => 9
  1  => 1
```

Рис. 6: Самостоятельная работа. Задание 1.2

Самостоятельная работа

Задание 1. Используя циклы while и for создайте массив squares_arr, содержащий квадраты всех чисел от 1 до 100 ¶

```
squares_arr=[]
n = 0
while n<100
    n += 1
    push!(squares_arr, [n^n])
end
print(squares_arr)
```

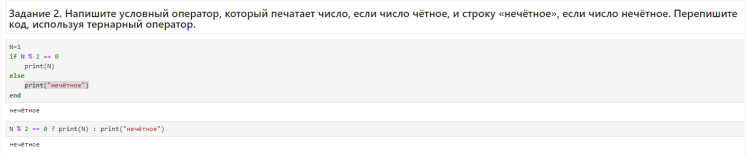
```
Any[[1], [4], [9], [16], [25], [36], [49], [64], [81], [100], [121], [144], [169], [196], [225], [256], [289], [324], [361], [400], [441], [484], [529], [576], [625], [676], [729], [784], [841], [900], [961], [1024], [1089], [1156], [1225], [1296], [1369], [1444], [1521], [1600], [1681], [1764], [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], [6889], [7056], [7225], [7396], [7569], [7744], [7921], [8100], [8281], [8464], [8649], [8836], [9025], [9216], [9409], [9604], [9801], [10000]]
```

```
squares_arr_2 = [i^i for i in 1:100]
print(squares_arr_2)
```

```
[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, 6889, 7056, 7225, 7396, 7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9604, 9801, 10000]
```

Рис. 7: Самостоятельная работа. Задание 1.3

Самостоятельная работа



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

```
N=1
if N % 2 == 0
    print(N)
else
    print("нечётное")
end
```

```
нечётное
```

```
N % 2 == 0 ? print(N) : print("нечётное")
```

```
нечётное
```

Рис. 8: Самостоятельная работа. Задание 2

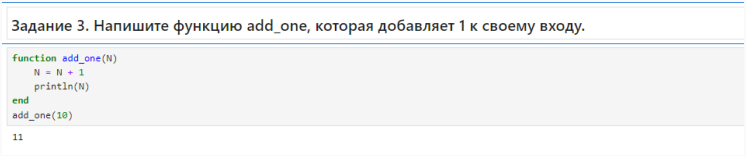


Рис. 9: Самостоятельная работа. Задание 3

Самостоятельная работа

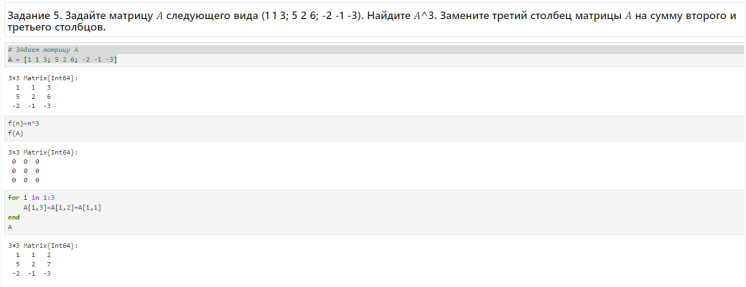


Рис. 10: Самостоятельная работа. Задание 5

Самостоятельная работа

Задание 6. Создайте матрицу B с элементами $Bi1 = 10$, $Bi2 = -10$, $Bi3 = 10$, $i = 1, 2, \ldots, 15$. Вычислите матрицу $C = B^{T}*B$.

```
# Задает матрицу B
B = [j%2==1 ? x=10 : x=-10 for i in 1:15,j in 1:3]
B
```

```
15×3 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
```

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

```
3×3 Matrix{Int64}:
  1500  -1500   1500
 -1500   1500  -1500
  1500  -1500   1500
```

Рис. 11: Самостоятельная работа. Задание 6

Самостоятельная работа

Задание 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 = zeros(6,6)
E = ones(6,6)

Z1 = copy(Z)
for i in 1:6, j in 1:6
    if abs(i-j) == 1
        Z1[i,j]=Z[i,j]+E[i,j]
    end
    Z1[i,j] |> Int64
end
Z1
```

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

```
Z2 = copy(Z)
for i in 1:6, j in 1:6
    if abs(i-j) == 2 || i == j
        Z2[i,j]=Z[i,j]+E[i,j]
    end
    Z2[i,j] |> Int64
end
Z2
```

```
6×6 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. 12: Самостоятельная работа. Задание 7.1

Самостоятельная работа

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

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

```
6×10 Matrix{Int64}:
 10   6  3  8  7   4  5   7  3  4
 10   8  8  6  4  10  1  10  4  5
  9   1  9  7  3   3  3   4  1  5
 10   1  2  3  8  10  4   7  5  5
  7  10  3  7  9   6  5   8  5  3
 10   1  1  8  7   1  1   9  2  7
```

- Найдите число элементов в каждой строке матрицы M, которые больше числа N (например, N = 4).

```
N = 4
res = sum(M .> N, dims=2)
println(res)
```

```
[6; 7; 4; 6; 8; 5;;]
```

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

```
M_value = 7
rows_res = findall(x -> count(==(M_value), x) == 2, eachrow(M))
println(rows_res)
```

```
[1, 5, 6]
```

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

```
K = 75
pairs_res = []
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_res, (i, j))
        end
    end
end
println(pairs_res)
```

```
Any[(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 8), (1, 9), (1, 10), (4, 5), (4, 8), (5, 8), (6, 8)]
```

Рис. 13: Самостоятельная работа. Задание 10

Самостоятельная работа

Задание 11. Вычислите

```
sum_1 = sum(1^4 * (3 + j) for i in 1:20 for j in 1:5)
println(sum_1)
```

```
21679980
```

```
sum_2 = sum(1^4 * (3 + i * j) for i in 1:20 for j in 1:5)
println(sum_2)
```

```
195839490
```

Рис. 14: Самостоятельная работа. Задание 11

Выводы по проделанной работе

Выводы по проделанной работе

В ходе лабораторной работы мною было освоено применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.