Лабораторная работа № 3

Покрас Илья Михайлович

2023, Москва

Цель работы

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

Ход работы

```
println("for cycle:")
for i in 1:100
    print(i, ":)", i+2, " ")
end
println()
println("\nwhile cycle:")
i = 1
while i <= 100
    print(i, ":)", i+2, " ")
    i += 1
end
println()
squares = Dict(i => i^2 for i = 1:20)
println("squares: $squares")
println()
while i <= 20
    squares[i] = i^2
    i += 1
end
println("\nsquares dict:")
println(squares)

squares_arr = [i^2 for i in 1:200]
squares_arr = []
i = 1
while i <= 100
    push!(squares_arr, i^2)
    i += 1
end
println("\nsquares array:")
println(squares_arr)
```

```
for cycle:
1 :) 3 2 :) 4 3 :) 5 4 :) 6 5 :) 7 6 :) 8 7 :) 9 8 :) 10 9 :) 11 10 :) 12 11 :) 13 12 :) 14 13 :) 15 14 :) 16 15 :) 17 16 :) 18 17 :) 19 18 :) 20 19 :) 21 20 :) 22 21 :) 23 22 :) 24 23 :) 25 24 :) 26 25 :) 27 26 :) 28 27 :) 29 28 :) 30 29 :) 31 30 :) 32 31 :) 33 32 :) 34 33 :) 35 34 :) 36 35 :) 37 36 :) 38 37 :) 39 38 :) 40 39 :) 41 40 :) 42 41 :) 43 42 :) 44 43 :) 45 44 :) 46 45 :) 47 46 :) 48 47 :) 49 48 :) 50 49 :) 51 50 :) 52 51 :) 53 52 :) 54 53 :) 55 54 :) 56 55 :) 57 56 :) 58 57 :) 59 58 :) 60 59 :) 61 60 :) 62 61 :) 63 62 :) 64 63 :) 65 64 :) 66 65 :) 67 66 :) 68 67 :) 69 68 :) 70 69 :) 71 70 :) 72 71 :) 73 72 :) 74 73 :) 75 74 :) 76 75 :) 77 76 :) 78 77 :) 79 78 :) 80 79 :) 81 80 :) 82 81 :) 83 82 :) 84 83 :) 85 84 :) 86 85 :) 87 86 :) 88 87 :) 89 88 :) 90 89 :) 91 90 :) 92 91 :) 93 92 :) 94 93 :) 95 94 :) 96 95 :) 97 96 :) 98 97 :) 99 98 :) 100 99 :) 101 100 :) 102

while cycle:
1 :) 3 2 :) 4 3 :) 5 4 :) 6 5 :) 7 6 :) 8 7 :) 9 8 :) 10 9 :) 11 10 :) 12 11 :) 13 12 :) 14 13 :) 15 14 :) 16 15 :) 17 16 :) 18 17 :) 19 18 :) 20 19 :) 21 20 :) 22 21 :) 23 22 :) 24 23 :) 25 24 :) 26 25 :) 27 26 :) 28 27 :) 29 28 :) 30 29 :) 31 30 :) 32 31 :) 33 32 :) 34 33 :) 35 34 :) 36 35 :) 37 36 :) 38 37 :) 39 38 :) 40 39 :) 41 40 :) 42 41 :) 43 42 :) 44 43 :) 45 44 :) 46 45 :) 47 46 :) 48 47 :) 49 48 :) 50 49 :) 51 50 :) 52 51 :) 53 52 :) 54 53 :) 55 54 :) 56 55 :) 57 56 :) 58 57 :) 59 58 :) 60 59 :) 61 60 :) 62 61 :) 63 62 :) 64 63 :) 65 64 :) 66 65 :) 67 66 :) 68 67 :) 69 68 :) 70 69 :) 71 70 :) 72 71 :) 73 72 :) 74 73 :) 75 74 :) 76 75 :) 77 76 :) 78 77 :) 79 78 :) 80 79 :) 81 80 :) 82 81 :) 83 82 :) 84 83 :) 85 84 :) 86 85 :) 87 86 :) 88 87 :) 89 88 :) 90 89 :) 91 90 :) 92 91 :) 93 92 :) 94 93 :) 95 94 :) 96 95 :) 97 96 :) 98 97 :) 99 98 :) 100 99 :) 101 100 :) 102

squares dict:
Dict(5 => 25, 15 => 256, 20 => 400, 12 => 144, 8 => 64, 17 => 289, 1 => 1, 13 => 169, 6 => 36, 11 => 121, 3 => 9, 14 => 196, 9 => 81, 16 => 256, 2 => 4, 18 => 100, 10 => 324)

squares array:
[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. 1: Код выполнения пункта 1

```
a = parse(Int, readline())
if isodd(a)
    println("Нечётное")
else
    println(a)
end
```

```
stdin>  3
Нечётное
```

Рис. 2: Код выполнения пункта 2

```
function add_one(numb)
    println(numb + 1)
end

for i in 1:3
    a = parse(Int, readline())
    add_one(a)
end
```

```
stdin>  2
3
stdin>  1
2
stdin>  4
5
```


Рис. 3: Код выполнения пункта 3

Ход работы

```
println("Matrix1:")
matrix1 = map(x -> x, reshape(Array(1:20), 4, 5))
display(matrix1)

println("\nMatrix2:")
matrix2 = broadcast(x -> x, reshape(Array(1:20), 4, 5))
display(matrix2)
```

```
Matrix1:
4×5 Matrix{Int64}:
 1  5   9  13  17
 2  6  10  14  18
 3  7  11  15  19
 4  8  12  16  20

Matrix2:
4×5 Matrix{Int64}:
 1  5   9  13  17
 2  6  10  14  18
 3  7  11  15  19
 4  8  12  16  20
```

Рис. 4: Код выполнения пункта 4

Ход работы

```
A = [1 1 3; 5 2 6; -2 -1 -3]
println("Matrix:")
display(A)

println("\nMatrix^3:")
display(A.^3)

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

println("\nMatrix with sum:")
display(A)
```

```
Matrix:
3×3 Matrix{Int64}:
  1   1   3
  5   2   6
 -2  -1  -3

Matrix^3:
3×3 Matrix{Int64}:
   1   1   27
 125   8  216
  -8  -1  -27

Matrix with sum:
3×3 Matrix{Int64}:
  1   1   2
  5   2   7
 -2  -1  -3
```

Рис. 5: Код выполнения пункта 5

```
B = zeros(15, 3)

for i in 1:15
    B[i,1] = 10
    B[i,2] = -10
    B[i,3] = 10
end

Bt = transpose(B)
C = Bt * B

println("\nMuliplication matrix:")
display(C)
```

```
Muliplication matrix:
3×3 Matrix{Float64}:
  1500.0  -1500.0   1500.0
 -1500.0   1500.0  -1500.0
  1500.0  -1500.0   1500.0
```

Рис. 6: Код выполнения пункта 6

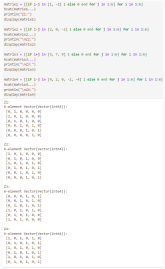

Рис. 7: Код выполнения пункта 7

Ход работы

```
outer(x, y, operation) = transpose(hcat([[sum(operation(x[i, k], y[k, j]) for k in 1:size(x)[2]) for j in 1:size(y)[2]] for i in 1:size(x)[1]]...))

B = reshape(rand(1:5, 12), 2, 6)
A = reshape(rand(1:5, 12), 6, 2)
println("Source matrices:")
display(A)
display(B)
println("\nSum matrix:")
display(outer(A, B, +))

println("\nSub matrix:")
display(outer(A, B, -))

println("\nMul matrix:")
display(outer(A, B, *))

println("\nDiv matrix:")
display(outer(A, B, /))

println("\nPow matrix:")
display(outer(A, B, ^))

println("\nA1:")
display(outer(reshape(0:4, 5, 1), reshape(0:4, 1, 5), +))

println("\nA2:")
println(outer(reshape(0:4, 5, 1), reshape(1:5, 1, 5), ^))

println("\nA3:")
display(outer(hcat([[if i==j 1 else 0 end for j in 0:4] for i in 0:4]...), hcat([Vector(1:i+4).%5 for i in 0:4]...), *))

println("\nA4:")
display(outer(hcat([[if i==j 1 else 0 end for j in 0:9] for i in 0:9]...), hcat([Vector(1:i+9).%10 for i in 0:9]...), *))

println("\nA5:")
display(outer(hcat([[if i==j 1 else 0 end for j in 0:8] for i in 0:8]...), hcat([Vector(i+9:-1:i+1).%9 for i in 0:8]...), *))
```

Рис. 8: Код выполнения пункта 8

```
M = [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]
N = [7, -1, -3, 5, 17]
println("Solution Vector: ")
display(M \ N)
```

```
Solution Vector:
5-element Vector{Float64}:
 -2.0000000000000036
  3.0000000000000058
  4.999999999999998
  1.9999999999999991
 -3.999999999999999
```

Рис. 9: Код выполнения пункта 9

Ход работы

```
matrix = rand(1:10, 6, 10)
println("Matrix: ")
display(matrix)
num = 5

ans_vec1 = []
for i in 1:size(matrix)[1]
    counter = 0
    for j in 1:size(matrix)[2]
            if matrix[i, j] > num
                counter = counter +1
            end
    end
    push!(ans_vec1, counter)
end
println("The amount of numbers, greater than N: ")
println(ans_vec1)


ans_vec2 = []
M = 7
for i in 1:size(matrix)[1]
    counter = 0
    for j in 1:size(matrix)[2]
            if matrix[i, j] == M
                counter = counter + 1
        end
    end
    push!(ans_vec2, counter)
end
println("\nthe amount of numbers, that equal M: ")
println(ans_vec2)

K = 75
ans_vec3 = [(i, j) for i in 1:size(matrix)[2]-1 for j in i+1:size(matrix)[2] if sum(matrix[:, i] + matrix[:, j]) > K]
println("\nthe amount of column pairs, sum of which is greater than K: ")
println(ans_vec3)
```

```
Matria:
6×10 Matrix{Int64}:
 9   2  5  2  10   6  6  5   8  6
 6   5  3  4   7   9  9  5   3  4
 6  10  4  4  10   6  5  4   1  8
 4   5  3  7   6  10  9  9  10  3
 7   3  2  3   2  10  5  1   4  9
 5   5  6  2   2   2  2  2   1  6
The amount of numbers, greater than N:
Any[6, 6, 5, 6, 5, 2]

The amount of numbers, that equal M:
Any[0, 1, 0, 1, 1, 0]

The amount of column pairs, sum of which is greater than K:
[(1, 6), (5, 6), (6, 7), (6, 10)]
```

Рис. 10: Код выполнения пункта 10

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

```
639215.2833333333
89912.02146097136
```

Рис. 11: Код выполнения пункта 11

Выводы

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