

Лабораторная работа № 3

Управляющие структуры

Герра Гарсия Паола Валентина

Российский университет дружбы народов, Москва, Россия

Информация

- Герра Гарсия Паола Валентина
- студентка
- Российский университет дружбы народов
- 1032225472@pfur.ru

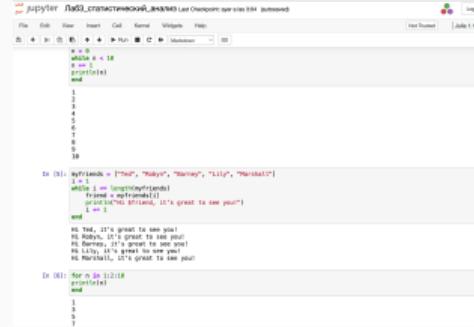
Цель работы

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

Задание

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

Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook interface with the title "ipython - Lab3_спланированный_блокнот.ipynb Last Opened: 2023-04-19 11:14 [modified]". The code cell contains the following Python script:

```
x = 0
while x < 10:
    print(x)
    x += 1

fr = ["Bob", "Mikey", "Baron", "Lily", "Marshall"]
for i in range(len(fr)):
    friend = fr[i]
    print(f"Hi {friend}, it's great to see you!")

for n in range(10):
    print(n)
```

The output pane displays the results of the code execution:

```
0
1
2
3
4
5
6
7
8
9
10
Hi Bob, it's great to see you!
Hi Mikey, it's great to see you!
Hi Baron, it's great to see you!
Hi Lily, it's great to see you!
Hi Marshall, it's great to see you!
0
1
2
3
```

Рис. 1: Выполнение примеров с циклами

Выполнение лабораторной работы



Условные выражения

```
3n 1121: # используем "if" для выполнения операции "НЕ"
# используем % вычисляет остаток от деления
if
    if M \% 3 == 0 then (M \% 5 == 0)
        print("FizzBuzz")
    elseif M \% 3 == 1
        print("Fizz")
    elseif M \% 3 == 2
        print("Buzz")
    else
        print(M)
    end
else
```

Рис. 2: Выполнение примеров с условными выражениями

Выполнение лабораторной работы

Сторонние библиотеки (пакеты) в Julia

```
In [17]: import Pkg  
Pkg.add!("Example")  
Resolving package versions...  
 Installed Example = v0.5.5  
 Updating `~/.julia/environments/v1.12/Project.toml'  
 [7876af07] + Example v0.5.5  
 └─ Colors v0.7.0  
     └─ Project.toml v1.12  
     └─ Manifest.toml  
 [7876af07] + Example v0.5.5  
Precompiling packages...  
 948.9 ms ✓ Example  
 1 dependency successfully precompiled in 1 seconds. 43 already precompiled.  
  
In [19]: Pkg.add!("Colors")  
using Colors  
Resolving package versions...  
 Project No packages added to or removed from `~/.julia/environments/v1.12/Project.toml'  
 Manifest No packages added to or removed from `~/.julia/environments/v1.12/Manifest.toml`
```

Рис. 3: Выполнение примеров со сторонними библиотеками

Выполнение лабораторной работы

The screenshot shows a Jupyter Notebook window with three code cells and their corresponding outputs:

- In [6]:**

```
for n in 1:2:10
print(n)
end
```

Output:
1
3
5
7
9
- In [7]:**

```
friends = ["Ted", "Robin", "Barney", "Lily", "Marshall"]
for friend in friends
print("Hi $friend, it's great to see you!")
end
```

Output:
Hi Ted, it's great to see you!
Hi Robin, 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!
- In [8]:**

```
# инициализация массива A из нулей:
m, n = 5, 5
A = fill(0, (m, n))
# Выводим массив A, в котором значение каждой записи
# равно сумме индексов строки и столбца:
for i in 1:m
for j in 1:n
A[i, j] = i + j
end
A
```

Рис. 4: Задание №1

Выполнение лабораторной работы

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter Лаб3_статистический_анализ Last Checkpoint: ayer a las 3:04 (autosaved)
- User Area:** Logout, Not Trusted, Julia 1.12
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help; icons for New, Open, Save, Run, Stop, Kernel, Help, and Cell Type.
- Cell 20 Output:**

```
In [20]: #1.1 enteros y sus cuadrados
println("Numeros y sus cuadrados:")
for i in 1:10
    println("$i ->$(i^2)")
end
```

Numeros y sus cuadrados:
1 ->1
2 ->4
3 ->9
4 ->16
5 ->25
6 ->36
7 ->49
8 ->64
9 ->81
10 ->100
- Cell 21 Output:**

```
In [21]: #1.2 diccionarios squares
squares = Dict{Int, Int}()
for i in 1:100
    squares[i] = i^2
end
```

Рис. 5: Задание №2

Выполнение лабораторной работы

```
In [35]: add_one(x) = x+1
          println(add_one(5)) #6
```

6

Рис. 6: Задание №3

Выполнение лабораторной работы

```
In [36]: #Matriz 3x3 que incrementa en 1
matriz = reshape(1:9, 3, 3)
println(matriz)

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

In [37]: #usando broadcast
matriz2= [i+j-1 for i in 1:3, j in 1:3]

Out[37]: 3x3 Matrix{Int64}:
 1  2  3
 2  3  4
 3  4  5
```

Рис. 7: Задание №4

Выполнение лабораторной работы

The screenshot shows a Jupyter Notebook interface with a Julia kernel. The notebook has several cells containing code and their corresponding outputs.

- In [36]:** #Matrix 3x3 que incrementa en 1
matrix = reshape(1:9, 3, 3)
print(matrix)
[1 4 7; 2 5 8; 3 6 9]
- In [37]:** #usando broadcast
matrix2= [i+j-1 for i in 1:3, j in 1:3]
Out[37]: 3x3 Matrix{Int64}:
1 2 3
2 3 4
3 4 5
- In [38]:** A = [1 1 3; 5 2 6; -2 -1 -3]
#A^3
A_cubo = A * A * A
println("A^3 = \$A_cubo")
A^3 = [0 0 0; 0 0 0; 0 0 0]
- In [39]:** #reemplazar 3ra columna por suma de col2 + col3
A_mod = copy(A)
A_mod[:, 3] = A[:, 2] + A[:, 3]
Out[39]: 3-element Vector{Int64}:
4
8
-4
- In [40]:** B = repeat([10 -10 10]', 15) #15 filas, 3 columnas
C = B' * B
println("C = B * B = \$C")
C = B * B = [45000;]
- In [41]:** n = 4

Рис. 8: Задание №5

Выполнение лабораторной работы

```
In [43]: function outer(x, y, op)
    result = zeros(size(x, 1), size(y, 2))
    for i in 1:size(x, 1)
        for j in 1:size(y, 2)
            result[i,j] = op(x[i,:], y[:,j])
        end
    end
    return result
end

Out[43]: outer (generic function with 1 method)

In [44]: #vectorizar single para multiplicacion
function outer_simple(A, B)
    return A * B
end

#crear matrices A1, A2, A3, A4, A5
n5
A1 = [1:i:j^2 for i in 1:n, j in 1:n]

Out[44]: 5x5 Matrix{Int64}:
 1  2  3  4
 2  4  6  8
 3  6  9  12
 4  8  12 16
 5  10 15 20
```

Рис. 9: Реализация функции outer()

Выполнение лабораторной работы

```
In [1]: def outer(A1):
    for i in range(3):
        for j in range(3):
            print(A1[i][j], end=' ')
    print()

A1 = [[1,2,3], [4,5,6], [7,8,9]]
```

```
In [2]: outer(A1)
```

```
A1([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

```
In [3]: def outer(A2):
    for i in range(3):
        for j in range(3):
            print(A2[i][j], end=' ')
    print()

A2 = [[1,2,3], [4,5,6], [7,8,9]]
```

```
In [4]: outer(A2)
```

```
A2([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

Рис. 10: Проверка работы функции `outer()`

Выполнение лабораторной работы

```
In [53]: #Matriz A del sistema
A = [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]
y = [7, -1, -3, 5, 17]

#Resolver Ax = y
x = A\y
println("Solucion: x= $x")

#verify
println("A*x deber ser y: $(A*x = y)")

Solucion: x= [-2.000000000000013, 3.000000000000027, 4.999999999999964, 2.000000000000027, -4.000000000000001]
A*x deber ser y: *
```

Рис. 11: Решение системы линейных уравнений

Выполнение лабораторной работы

```
In [54]: using Random
Random.seed!(123)

M = rand(1:10, 6, 10)
N = 4
M_num = 7
K = 75

mayores_N = [count(x .-> x >= N, M[i,:]) for i in 1:6]

filas_M2 = [i for i in 1:6 if count(x .-> x == M_num, M[i,:]) == 2]

pares_columnas = []
for j1 in 1:9
    for j2 in (j1+1):10
        if sum(M[:,j1]) + sum(M[:,j2]) > K
            push!(pares_columnas, (j1, j2))
        end
    end
end

println("Mayores que $N por fila : $mayores_N")
println("Filas con $M_num exactamente 2 veces: $filas_M2")
println("Pares de columnas con suma > $K: $pares_columnas")
```

Mayores que 4 por fila : [3, 3, 3, 3, 3, 3]
Filas con 7 exactamente 2 veces: [1]
Pares de columnas con suma > 75: Any([(1, 8), (1, 10), (2, 8), (2, 10), (4, 8), (4, 10), (8, 9), (8, 10), (9, 10)])

Рис. 12: Задание №10

Выполнение лабораторной работы

```
In [55]: sum1 = 0.0
for i in 1:20
    for j in 1:5
        sum1 += i^4 / (3+j)^4
    end
end
```

```
In [56]: sum2 = 0.0
for i in 1:20
    for j in 1:5
        sum2 += i^4 / (3+i)^4
    end
end
```

```
In [57]: println("Sumatoria 1: $sum1")
println("Sumatoria 2: $sum2")
```

```
Sumatoria 1: 5014.209868467763
Sumatoria 2: 32.98305525107463
```

```
In [58]: sum1_vec = sum([i^4/(3+j)^4 for i in 1:20, j in 1:5])
sum2_vec = sum([i^4/(3+j)^4 for i in 1:20, j in 1:5])
```

```
Out[58]: 5014.209868467762
```

Рис. 13: Задание №11

Выводы

В результате выполнения данной лабораторной работы я освоила применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

Список литературы

1. JuliaLang [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://julialang.org/> (дата обращения: 11.10.2024).
2. Julia 1.11 Documentation [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://docs.julialang.org/en/v1/> (дата обращения: 11.10.2024).