

Лабораторная работа № 3

Управляющие структуры

Старовойтов Е. С.

30 ноября 2024

Информация

Докладчик

- Старовойтов Егор Сергеевич
- студент кафедры ТВиК
- Российский университет дружбы народов
- 1032212281@pfur.ru

Вводная часть

Цели и задачи

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

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

Результаты

Поставленные боевые задачи были выполнены, все цели достигнуты.

Выполнение лабораторной работы

```
lab3.ipynb x +
Notebook Julia 1.11.0-rc3

[1]: n = 0
while n < 10
    n += 1
    println(n)
end
1
2
3
4
5
6
7
8
9
10

[3]: 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!

[4]: for n in 1:2:10
    println(n)
end
1
3
5
7
9

[5]: 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!

[6]: # 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

[6]: 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

[7]: B = fill{0, (m, n)}
for i in 1:m, j in 1:n
    B[i, j] = i + j
end
B
```

```
Julia 1.11.0-rc3
B

[7]: 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

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

[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

[11]: N = 12
      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

      Fizz

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

[12]: 10

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

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

[14]: sayhi("C-3PO")
      f(42)

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

[14]: 1764

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

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

[16]: sayhi3 = name -> println("Hi $name, it's great to see you!")

[16]: #3 (generic function with 1 method)

[17]: f3 = x -> x^2

[17]: #5 (generic function with 1 method)

[18]: v = [3, 5, 2]
      sort(v)
      v
      sort!(v)
      v

[18]: 3-element Vector{Int64}:
```

```
lab3.ipynb X +
Notebook Julia 1.11.0-rc3

[18]: 3-element Vector{Int64}:
      2
      3
      5

[22]: # Задаём матрицу A:
      A = [i + 3*j for j in 0:2, i in 1:3]
      # Вызываем функцию f возведения в квадрат
      f(A)

[22]: 3×3 Matrix{Int64}:
      30  36  42
      66  81  96
      102 126 150

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

[23]: 3×3 Matrix{Float64}:
      3.0  6.0  9.0
      12.0 15.0 18.0
      21.0 24.0 27.0

[24]: import Pkg
      Pkg.add("Example")

      Updating registry at '~/.julia/registries/General.toml'
      Resolving package versions...
      Installed Example - v0.5.5
      Updating ~/.julia/environments/v1.11/Project.toml
      Updating ~/.julia/environments/v1.11/Manifest.toml
      Precompiling project...
      319.7 ms ✓ Example
      1 dependency successfully precompiled in 1 seconds. 45 already precompiled.

[25]: Pkg.add("Colors")
      using Colors

      Resolving package versions...
      Installed Statistics - v1.11.1
      Installed Reexport - v1.2.2
      Installed Colors - v0.13.0
      Installed ColorTypes - v0.12.0
      Installed FixedPointNumbers - v0.8.5
      Updating ~/.julia/environments/v1.11/Project.toml
      Updating ~/.julia/environments/v1.11/Manifest.toml
      Updating JuliaArrays v1.11.0
      Updating Colors v0.13.0
      Updating FixedPointNumbers v0.8.5
      Updating Reexport v1.2.2
      Updating Statistics v1.11.1
      Updating LinearAlgebra v1.11.0
      Updating CompilerSupportLibraries_jll v1.11.0
      Updating OpenBLAS_jll v0.3.27+1
      Updating libblastrampoline_jll v5.11.0-0
      Precompiling project...
      252.3 ms ✓ Statistics
      352.3 ms ✓ Reexport
      1003.9 ms ✓ FixedPointNumbers
      775.3 ms ✓ ColorTypes
      306.5 ms ✓ Colors
      1871.8 ms ✓ Colors
      6 dependencies successfully precompiled in 4 seconds. 47 already precompiled.

[26]: # Задание 1
      # Целые числа от 1 до 100 и их квадраты
      println("Целые числа и их квадраты:")
      for n in 1:100
          println("$n^2 = $(n^2)")
      end

      # Создаём squares
      squares = Dict{n => n^2 for n in 1:100}
```

```
lab3.ipynb
386.5 ms ✓
1871.8 ms ✓ Colors
6 dependencies successfully precompiled in 4 seconds. 47 already precompiled.

[26]: # Задание 1
# Целые числа от 1 до 100 и их квадраты
println("Целые числа и их квадраты:")
for n in 1:100
    println("$n^2 = $(n^2)")
end

# Словарь squares
squares = Dict{n => n^2 for n in 1:100}
println("Словарь squares:")
println(squares)

# Массив squares_arr
squares_arr = [n^2 for n in 1:100]
println("Массив squares_arr:")
println(squares_arr)

# Задание 2
function even_or_odd(n)
    if n % 2 == 0
        println(n)
    else
        println("нечётное")
    end
end

# Пример вызова
for n in 1:10
    even_or_odd(n)
end

# Тернарный оператор
for n in 1:10
    println((n % 2 == 0) ? n : "нечётное")
end

# Задание 3
add_one(x) = x + 1

# Пример вызова
println(add_one(10))

# Задание 4
A = [1 2; 3 4] # Примерная матрица для демонстрации
A_inc = map(x -> x + 1, A)
println("Матрица A с увеличенными элементами:")
println(A_inc)

# Задание 5
A = [1 1 3; 5 2 6; -2 -1 -3]
A3 = A * A * A
println("Матрица A^3:")
println(A3)

A[:, 3] = A[:, 2] + A[:, 3]
println("Матрица A с изменённым третьим столбцом:")
println(A)

# Задание 6: Создание матрицы B
B = [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]

println("Матрица B:")
println(B)

# Вычисляем C = B' * B
C = B' * B
println("Матрица C:")
```

```
lab3.ipynb
+
Code
Notebook
Julia 1.11.0-rc3

for i in 1:6
    for j in 1:6
        if abs(i - j) == 1
            Z1[i, j] = 1
        end
        if abs(i - j) % 2 == 0
            Z2[i, j] = 1
        end
        Z3[i, j] = (i + j) % 2
        Z4[i, j] = (i + j + 1) % 2
    end
end
println("Матрицы Z1, Z2, Z3, Z4:")
println(Z1)
println(Z2)
println(Z3)
println(Z4)

# Задание 8
function outer(x, y, operation)
    [operation(xi, yj) for xi in x, yj in y]
end

# A1
A1 = [i + j for i in 0:4, j in 0:4]
println("Матрица A1:")
println(A1)

# A2
A2 = [i^j for i in 0:4, j in 0:4]
println("Матрица A2:")
println(A2)

# A3
A3 = [(i + j) % 5 for i in 0:4, j in 0:4]
println("Матрица A3:")
println(A3)

# A4
A4 = [(i + j) % 10 for i in 0:9, j in 0:9]
println("Матрица A4:")
println(A4)

# A5
A5 = [(abs(i - j) % 9) for i in 0:8, j in 0:8]
println("Матрица A5:")
println(A5)

# Задание 9
# Решение системы линейных уравнений Ax = y
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] # Вектор правых частей
x = A \ y # Решение системы
println("Решение системы линейных уравнений:")
println(x)

# Задание 10
# Создание случайной матрицы M
M = [
    9 5 10 2 1 8 1 3 5 2;
    7 5 5 8 6 9 9 2 9 4;
    3 10 2 7 7 1 6 6 4 4;
    9 7 8 7 6 5 4 2 6 1;
    7 1 10 5 6 7 6 8 9 3;
    8 3 9 3 4 8 6 7 10 4
]
```

```
lab3.ipynb x +
+ 🔍 📄 🏠 🔄 ▶️ ⏏️ Code ▾
Notebook 📄 Julia 1.11.0-rc3 🌐 ☰
```

```
8 3 9 3 4 8 6 7 10 4
]
println("Матрица M:")
println(M)

# Подзадание 1: Число элементов в каждой строке больше N
N = 4
greater_than_N = sum(M .> N, dims=2) # Считаем элементы больше N по строкам
println("Количество элементов больше $N в каждой строке:")
println(greater_than_N)

# Подзадание 2: Строки, где число M_value встречается ровно два раза
M_value = 7
rows_with_M_twice = []
for i in 1:size(M, 1)
    if count(==(M_value), M[i, :]) == 2
        push!(rows_with_M_twice, i)
    end
end
println("Строки, где число $M_value встречается ровно два раза:")
println(rows_with_M_twice)

# Подзадание 3: Пары столбцов, сумма элементов которых больше K
K = 75
column_pairs = []
for i in 1:size(M, 2)
    for j in i+1:size(M, 2)
        if sum(M[:, i]) + sum(M[:, j]) > K
            push!(column_pairs, (i, j))
        end
    end
end
println("Пары столбцов с суммой элементов больше $K:")
println(column_pairs)

# Задание 11
# Подсчёт суммы 1
sum1 = sum(i^4 / (3 + j) for i in 1:20, j in 1:5)
println("Сумма 1:")
println(sum1)

# Подсчёт суммы 2
sum2 = sum(i^4 / (3 + i * j) for i in 1:20, j in 1:5)
println("Сумма 2:")
println(sum2)
```

Целые числа и их квадраты:

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

