

Лабораторная работа 3

Управляющие структуры

Ланцова Я. И.

Российский университет дружбы народов, Москва, Россия

Информация

- Ланцова Яна Игоревна
- студентка
- Российский университет дружбы народов

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.

Выполнение лабораторной работы

Выполнение лабораторной работы

```
[3]: # пример
n = 3
while n < 10
  n = n + 1
  println(n)
end

4
5
6
7
8
9
10

[4]: myfriends = ["Ted", "Barney", "Collins", "Lily", "Marshall", "Lesya"];
i = 1
while i <= length(myfriends)
  friend = myfriends[i]
  println("Hi $friend, it's nice to see you!")
  i = i + 1
end

Hi Ted, it's nice to see you!
Hi Barney, it's nice to see you!
Hi Collins, it's nice to see you!
Hi Lily, it's nice to see you!
Hi Marshall, it's nice to see you!
Hi Lesya, it's nice to see you!

[5]: for n in 1:3:10
  println(n)
end

1
4
7
10
```

Рис. 1: Выполнение примеров с циклами

Выполнение лабораторной работы

```
[9]: m, n = 3, 3
    A = fill(0, (m, n))

[9]: 3x3 Matrix{Int64}:
    0  0  0
    0  0  0
    0  0  0

[11]: for i in 1:m
        for j in 1:n
            A[i, j] = i + j
        end
    end
    println(A)
    [2 3 4; 3 4 5; 4 5 6]

[12]: B = fill(0, (m, n))
    for i in 1:m, j in 1:n
        B[i, j] = i + j
    end
    println(B)
    [2 3 4; 3 4 5; 4 5 6]

[13]: C = [i + j for i in 1:m, j in 1:n]
    C
    [2 3 4
     3 4 5
     4 5 6]
```

Рис. 2: Выполнение примеров с циклами

Выполнение лабораторной работы

```
[14]: N = 52
      if (N % 2 == 0) && (N % 9 == 0)
        println("Cool number!")
      elseif N % 2 == 0
        println("Oh thats a nice num!")
      elseif N % 9 == 0
        println("Mehfeh!")
      else
        println(N)
      end
      Oh thats a nice num!

[15]: x = 5
      y = 2
      (x > y) ? x : y

[15]: 5

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

[17]: f2(x) = x^2
      f2(9)

[17]: 81

[18]: f_anonymous = x -> x^2
      f_anonymous(9)

[18]: 81
```

Рис. 3: Выполнение примеров с условными выражениями

Выполнение лабораторной работы

```
[19]: 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'
      [7876af87] + Example v0.5.5
      Updating '~/.julia/environments/v1.11/Manifest.toml'
      [7876af87] + Example v0.5.5
      Precompiling project...
      930.2 ms / Example
      1 dependency successfully precompiled in 2 seconds. 41 already precompiled.

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

      Resolving package versions...
      Installed Reexport - v1.2.2
      Installed Statistics - v1.11.1
      Installed ColorTypes - v0.12.1
      Installed FixedPointNumbers - v0.8.5
      Installed Colors - v0.13.1
      Updating '~/.julia/environments/v1.11/Project.toml'
      [5ae59095] + Colors v0.13.1
      Updating '~/.julia/environments/v1.11/Manifest.toml'
      [32ab0277] + ColorTypes v0.12.1
      [5ae59095] + Colors v0.13.1
      [53c48c17] + FixedPointNumbers v0.8.5
      [189e38b7] + Reexport v1.2.2
      [18745b36] + Statistics v1.11.1
      [37e2e46d] + LinearAlgebra v1.11.0
```

Рис. 4: Импорт сторонних библиотек

Выполнение лабораторной работы

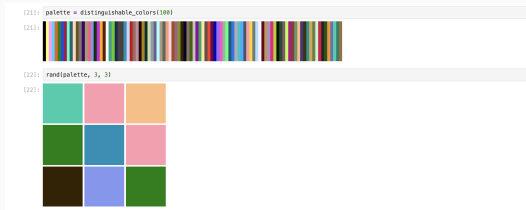


Рис. 5: Выполнение примеров со сторонними библиотеками

Задание 1

Выполнение лабораторной работы

Задание 1

- выведите на экран целые числа от 1 до 100 и напечатайте их квадраты

```
[28]: i = 1
while i <= 100
  print("Number  $i$ , square:  $i^2$ . ")
  i = i + 1
end

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

[28]: for i in 1:100
  print("Number  $i$ , square:  $i^2$ . ")
end

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

Рис. 6: Задание №1

Выполнение лабораторной работы

```
• создайте словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений
```

```
[36]: squares = Dict{Int, Int}()
      for i in 1:100
        squares[i] = i^2
      end
```

```
[37]: squares{
```

```
[37]: Dict{Int64, Int64} with 100 entries:
      5 => 25
      56 => 3136
      35 => 1225
      55 => 3025
      68 => 4624
      38 => 1444
      32 => 1024
      6 => 36
      67 => 4489
      45 => 2025
      73 => 5329
      64 => 4096
      98 => 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
```

Рис. 7: Задание №1

Выполнение лабораторной работы

```
• создайте массив squares_arr, содержащий квадраты всех чисел от 1 до 100

[10]: squares_arr = [i**2 for i in range(1,101)]
[10]: 100-element Vector[Int64]:
      1
       4
       9
      16
      25
      36
      49
      64
      81
     100
     121
     144
     169
    1921
    4000
    8100
   12101
   16004
   20449
   25636
   31625
   38416
   46009
   54404
   63601
   73600
  100000
```

Рис. 8: Задание №1

Задание 2

Выполнение лабораторной работы

Задание 2

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

```
[41]: numbers = [1, 2, 3, 4, 5, 6]
      for num in numbers
      if num%2 == 0
        print(num)
      else
        print("Нечётное")
      end
    end
    Нечётное
    2
    Нечётное
    4
    Нечётное
    6

[43]: for num in numbers
      print(num%2==0 ? num : "Нечётное")
    end
    Нечётное
    2
    Нечётное
    4
    Нечётное
    6
```

Рис. 9: Задание №2

Задание 3 и 4

Выполнение лабораторной работы

Задание 3

Напишите функцию `add_one`, которая добавляет 1 к своему вводу

```
[46]: function add_one(x)
      x = x + 1
      end

[46]: add_one (generic function with 1 method)

[47]: add_one(9)

[47]: 10
```

Задание 4

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

```
[48]: matrix = [1 2 3; 4 5 6; 7 8 9]

[48]: 3x3 Matrix{Int64}:
      1  2  3
      4  5  6
      7  8  9

[49]: matrix_plus_one = broadcast(x -> x + 1, matrix)

[49]: 3x3 Matrix{Int64}:
      2  3  4
      5  6  7
      8  9 10

[51]: matrix_plus_plus_one = map(x -> x + 1, matrix)

[51]: 3x3 Matrix{Int64}:
```

Рис. 10: Задание №3 и 4

Задание 5

Выполнение лабораторной работы

Задание 5

Задайте матрицу A и найдите

- A^3
- Замените третий столбец матрицы A на сумму второго и третьего столбцов.

```
[52]: A = [1 1 3; 5 2 0; -2 -1 -3];  
      A_cubed = A^3  
  
[52]: 3x3 Matrix{Int64}:  
      0  0  0  
      0  0  0  
      0  0  0  
  
[54]: A_modified = copy(A);  
      A_modified[i, 3] = A[i, 2] + A[i, 3];  
      A_modified  
  
[54]: 3x3 Matrix{Int64}:  
      1  1  4  
      5  2  0  
     -2 -1 -4
```

Рис. 11: Задание №5

Задание 6 и 7

Выполнение лабораторной работы

```
Задание 6

[58]: B = fill(0, (15, 3));

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

[59]: C = transpose(B) * B

[59]: 3x3 Matrix{Int64}:
 0  0  0
 0  0  0
 0  0  0

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 матрицы размерности 6 × 6

[78]: Z = fill(0, {6, 6});
      E = fill(1, {6, 6});
```

Рис. 12: Задание №6 и 7

Выполнение лабораторной работы

```
[84]: Z1 = fill(0, (6, 6))
      for i in 1:6
        for j in 1:6
          if abs(i - j) == 1
            Z1[i, j] = 1
          end
        end
      end
      Z1

[84]: 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

[85]: Z2 = fill(0, (6, 6))
      for i in 1:6
        for j in 1:6
          if i == j
            Z2[i, j] = 1
          elseif abs(i - j) == 2
            Z2[i, j] = 1
          end
        end
      end
      Z2

[85]: 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
```

Рис. 13: Задание №7

Выполнение лабораторной работы

```
[91]: Z3 = fill(0, (6, 6));  
for i in 1:6  
    for j in 1:6  
        if i == 1 && (j == 4 || j == 6)  
            Z3[i, j] = 1  
        elseif i == 2 && (j == 3 || j == 5)  
            Z3[i, j] = 1  
        elseif i == 3 && (j == 2 || j == 4 || j == 6)  
            Z3[i, j] = 1  
        elseif i == 4 && (j == 1 || j == 3 || j == 5)  
            Z3[i, j] = 1  
        elseif i == 5 && (j == 2 || j == 4)  
            Z3[i, j] = 1  
        elseif i == 6 && (j == 1 || j == 3)  
            Z3[i, j] = 1  
        end  
    end  
end  
Z3  
[91]: 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
```

Рис. 14: Задание №7

Выполнение лабораторной работы

```
[90]: Z4 = fill(0, (6, 6));  
      for i in 1:6  
        for j in 1:6  
          if (i + j) % 2 == 0  
            Z4[i, j] = 1  
          end  
        end  
      end  
      Z4  
[90]: 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
```

Рис. 15: Задание №7

Задание 8

Выполнение лабораторной работы

Задание 8

```
[98]: function outer(x, y, operation)
      %%%
      % Аналог функции outer() из R
      % x, y - векторы или матрицы
      % operation - функция, применяемая к каждой паре элементов
      %%%
      x_vec = vec(x)
      y_vec = vec(y)

      m = length(x_vec)
      n = length(y_vec)

      result = Matrix(Any)(undef, m, n)

      for i in 1:m
          for j in 1:n
              result[i, j] = operation(x_vec[i], y_vec[j])
          end
      end
      return result
  end

[98]: outer (generic function with 1 method)

[95]: A = [1, 2, 3];
      B = [4, 5, 6];

[96]: result1 = outer(A, B, *)

[96]: 3x3 Matrix{Any}:
      4  5  6
      8 10 12
     12 15 18
```

Рис. 16: Задание №8

Выполнение лабораторной работы

```
[97]: result2 = outer(A, B, +)

[97]: 3x3 Matrix[Any]:
  5 6 7
  6 7 8
  7 8 9

[100]: A1 = outer(0:4, 0:4, +)

[100]: 5x5 Matrix[Any]:
  0 1 2 3 4
  1 2 3 4 5
  2 3 4 5 6
  3 4 5 6 7
  4 5 6 7 8

[102]: A2 = outer(0:4, 1:5, *)

[102]: 5x5 Matrix[Any]:
  0 0 0 0 0
  1 1 1 1 1
  2 4 8 16 32
  3 9 27 81 243
  4 16 64 256 1024

[104]: A3 = outer(0:4, 0:4, (i,j) -> (i + j) % 5)

[104]: 5x5 Matrix[Any]:
  0 2 3 4 0
  1 2 3 4 0
  2 3 4 0 1
  3 4 0 1 2
  4 0 1 2 3
```

Рис. 17: Задание №8

Выполнение лабораторной работы

```
[105]: A4 = outer(0:9, 0:9, (i,j) -> (i + j) % 10)

[105]: 10x10 Matrix{Any}:
 0 1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9 0
 2 3 4 5 6 7 8 9 0 1
 3 4 5 6 7 8 9 0 1 2
 4 5 6 7 8 9 0 1 2 3
 5 6 7 8 9 0 1 2 3 4
 6 7 8 9 0 1 2 3 4 5
 7 8 9 0 1 2 3 4 5 6
 8 9 0 1 2 3 4 5 6 7
 9 0 1 2 3 4 5 6 7 8

[106]: A5 = outer(0:8, 0:8, (i,j) -> i - j < 0 ? i - j + 9 : i - j)

[106]: 9x9 Matrix{Any}:
 0 8 7 6 5 4 3 2 1
 1 0 8 7 6 5 4 3 2
 2 1 0 8 7 6 5 4 3
 3 2 1 0 8 7 6 5 4
 4 3 2 1 0 8 7 6 5
 5 4 3 2 1 0 8 7 6
 6 5 4 3 2 1 0 8 7
 7 6 5 4 3 2 1 0 8
 8 7 6 5 4 3 2 1 0
```

Рис. 18: Задание №8

Задание 9 и 10

Выполнение лабораторной работы

```
Задание 9

[122]: 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];
       x = A \ y;
       print(x)

[-2.0000000000000013, 3.0000000000000027, 4.9999999999999954, 2.0000000000000027, -4.000000000000001]

[121]: residual = A * x - y
       print(residual)

[0.0, -3.552713678800501e-15, 0.0, -5.329070518208751e-15, 0.0]

Задание 10

[115]: M = rand(1:10, 6, 10);
       N = 4;
       K = 75;
       count_M = sum(M.*M)

[115]: 39

[117]: count_M = [1 for i=1:6 if sum(M[i, :])>=2]

[117]: 2-element Vector{Int64}:
       1
       3

[120]: col = [i, j] for i=1:6, j=2:5 if i!=j && sum(M[i, :]) + M[i, j]>=K]

[120]: 3-element Vector{Tuple{Int64, Int64}}:
       (4, 5)
```

Рис. 19: Задание №9 и 10

Задание 11

Задание 11

```
[123]: sum1 = sum(i^4 / (3 + j) for i in 1:20, j in 1:5)
```

```
[123]: 639215.2833333338
```

```
[125]: sum2 = sum(i^4 / (3 + i*j) for i in 1:20, j in 1:5)
```

```
[125]: 89912.02146097131
```

Рис. 20: 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.