

Цель работы

Освоить применение:

- циклов
- условных операторов
- функций
- сторонних пакетов Julia

для решения задач линейной алгебры и работы с матрицами.

Задание

- 1 Повторить примеры из раздела 3.2
- 2 Выполнить задания для самостоятельной работы (раздел 3.4)

Julia — высокоуровневый язык программирования с динамической типизацией, предназначенный для математических вычислений.

Использовалась официальная документация Julia.

Примеры: циклы

Циклы while и for

```
[1] 0 OK
n = 0
while n < 10
  n += 1
  println(n)
end
```

```
1
2
3
4
5
6
7
8
9
10
```

```
[3] 0 OK
i = 1
myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
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] 0 OK
for n in 1:2:10
  println(n)
end
myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
for friend in myfriends
  println("Hi $friend, it's great to see you!")
end
```

```
1
3
5
7
9
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!
```

```
[5] 0 OK
n, m = 5, 5
A = fill{0, (n, m)}
for i in 1:n
  for j in 1:m
```

Примеры: условные выражения

Условные выражения

```
[9]
0 сек.
# используем `&&` для реализации операции "AND"
# операция % вычисляет остаток от деления
N = 99
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

```
[10]
0 сек.
if a
b
else
c
end
```

```
*** UnhandledExceptionError: `a` not defined in `Main`
Suggestion: check for spelling errors or missing imports.

Stacktrace:
 [1] top-level scope
      @ In[10]:1
```

Далее: [Объяснить ошибку](#)

Примеры: функции

Функции

```
[11]
✓ 0
ошибка
function sayhi(name)
println("Hi $name, it's great to see you!")
end
# функция возведения в квадрат:
function f(x)
x^2
end
```

*** f (generic function with 1 method)

```
[12]
✓ 0
ошибка
A = [1 + 3*j for j in 0:2, i in 1:3]
```

3×3 Matrix{Int64}:

1	2	3
4	5	6
7	8	9

Примеры: сторонние библиотеки

Сторонние библиотеки (пакеты) в Julia

```
[13] import Pkg
✓ 43 obs.
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'
[7876af07] + Example v0.5.5
Updating '~/.julia/environments/v1.11/Manifest.toml'
[7876af07] + Example v0.5.5
Precompiling project...
5409.9 ms ✓ Example
1 dependency successfully precompiled in 18 seconds. 468 already precompiled.
```

```
[14] Pkg.add("Colors")
✓ 11 obs.
using Colors

***
Resolving package versions...
Updating '~/.julia/environments/v1.11/Project.toml'
[5ae59895] + Colors v0.13.1
No Changes to '~/.julia/environments/v1.11/Manifest.toml'
```

```
[15] palette = distinguishable_colors(100)
✓ 2 obs.
```



```
[19] rand(palette, 3, 3)
✓ 0 obs.
```



Задание №1

Использование циклов while и for.

✓ 1

[25]

0
✓ ОК.

```
print([i for i in 1:100])  
print("\n", "квадрат")  
  
print("\n", [i^2 for i in 1:100])
```

✓

```
... [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,  
квадрат  
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225,
```

[27]

1
✓ ОК.

```
squares = Dict()  
  
for i in 1:10  
    squares[i] = i^2  
end  
print(squares)
```

✓

```
... Dict{Any, Any}{5 => 25, 4 => 16, 6 => 36, 7 => 49, 2 => 4, 10 =
```


Задание №2

Условный оператор:

- если число чётное — вывести число
- если нечётное — вывести строку “нечётное”

Переписано с использованием тернарного оператора.

✓ 2

[30]

1

сек.



```
a = readline()

a = parse(Int64, a)
if a % 2 == 0
    println(a)
else
    print("Нечетное")
end
```

Задание №3

Функция `add_one`, увеличивающая входное значение на 1.

✓ 3

[32]

✓ 0 сек.

```
function add_one(num)
  num += 1
end
```

✓

`add_one` (generic function with 1 method)

[33]

✓ 0 сек.



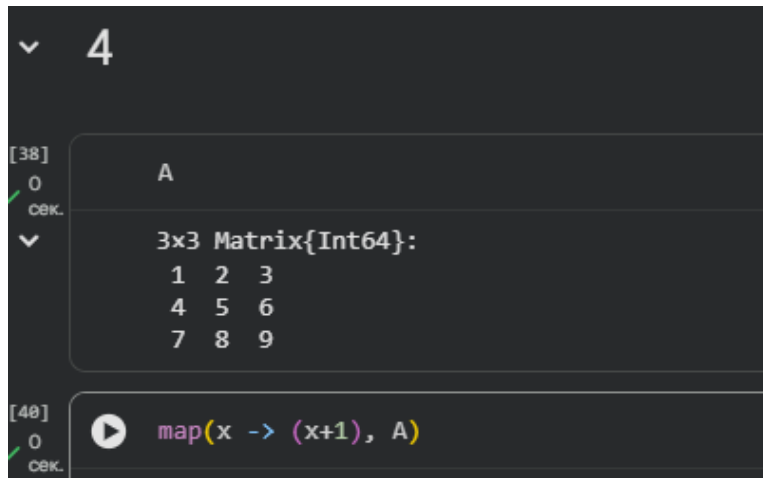
```
add_one(5)
```

✓

... 6

Задание №4

Использование `map()` или `broadcast()` для создания матрицы `A` с последовательным увеличением элементов.



The screenshot shows a Jupyter Notebook interface with two code cells. The first cell, labeled [38], contains the variable `A` and its output, a 3x3 matrix of type `Matrix{Int64}` with values 1 through 9. The second cell, labeled [40], contains the code `map(x -> (x+1), A)` and a play button icon, indicating it is ready to be executed.

```
[38] A
0 сек.
3x3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

[40] map(x -> (x+1), A)
0 сек.
```

Задание №5

- Задана матрица A
- Найдено (A^3)
- Третий столбец заменён суммой второго и третьего

5

✓ Новый раздел

[41] 0 сек.
▶ $A = \begin{bmatrix} 1 & 1 & 3; 5 & 2 & 6; -2 & -1 & -3 \end{bmatrix}$

▼
... 3x3 Matrix{Int64}:
1 1 3
5 2 6
-2 -1 -3

[42] 0 сек.
▶ $g(x) = x^3$
 $B = g.(A)$

▼
... 3x3 Matrix{Int64}:
1 1 27
125 8 216
-8 -1 -27

[43]

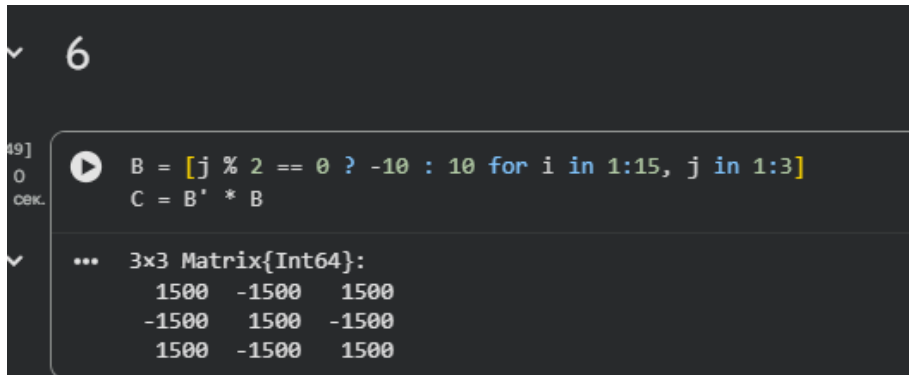
Задание №6

Матрица B:

$$[B_{i1}=10, \quad B_{i2}=-10, \quad B_{i3}=10]$$

Вычислена матрица:

$$[C = B^T B]$$



The screenshot shows a Jupyter Notebook interface with a dark theme. At the top left, there is a checkmark and the number '6'. Below it, the code cell is visible, showing the definition of matrix B and the calculation of matrix C. The code is as follows:

```
B = [j % 2 == 0 ? -10 : 10 for i in 1:15, j in 1:3]
C = B' * B
```

Below the code cell, the output is displayed, showing the resulting 3x3 matrix C. The output is a 3x3 Matrix{Int64} with the following values:

1500	-1500	1500
-1500	1500	-1500
1500	-1500	1500

Задание №7

Созданы матрицы:

- (Z) — нулевая 6×6
- (E) — единичная 6×6

Построены требуемые матрицы с использованием циклов.

```
Z = fill(0, 6, 6)
E = fill(1, 6, 6)

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

```
6×6 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
```

Задание №8

Реализация функции `outer()` и проверка её работы.

```
▶ Z3 = copy(Z)
  for i in 1:6, j in 1:6
    if (6 - i + 1) == j || abs(6 - i + 1 - j) == 2
      Z3[i, j] = 1
    end
  end
  Z3
```

```
... 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
```

```
▶ Z4 = copy(Z)
  for i in 1:6, j in 1:6
    if (i+j) % 2 == 0
      Z4[i, j] = 1
    end
  end
  Z4
```

Задание №10–11

Анализ количества элементов матрицы,
удовлетворяющих заданным условиям.

Вычисление выражений.

11

[83]

✓ 0
сек.



```
sum1 = sum(i^4 / (3 + j) for i in 1:20, j in 1:5)
sum2 = sum(i^4 / (3 + i*j) for i in 1:20, j in 1:5)
print("\n", sum1)
print("\n", sum2)
```



```
639215.2833333338
89912.02146097131
```


Выводы

В ходе лабораторной работы:

- Освоено применение циклов и функций в Julia
- Изучена работа с матрицами
- Получены навыки использования сторонних библиотек
- Закреплены методы решения задач линейной алгебры