

Julia. Установка и настройка. Основные принципы

Лабораторная работа № 1

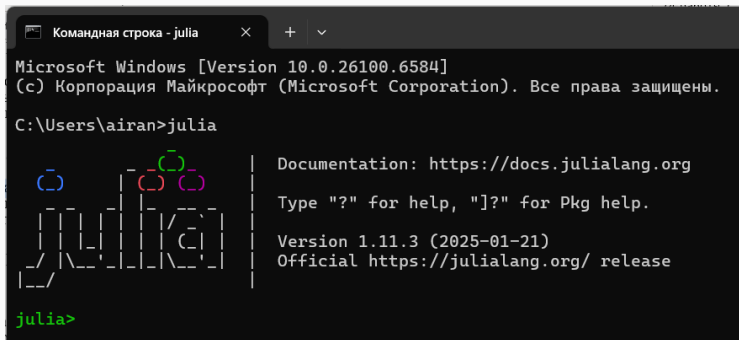
Шулуужук А. В.

22 март 2025

Российский университет дружбы народов, Москва, Россия

Основная цель работы — подготовить рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомиться с основами синтаксиса Julia.

Выполнение лабораторной работы



```
Командная строка - julia
Microsoft Windows [Version 10.0.26100.6584]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\airan>julia

      _ _ _ _ _
     ( ) ( ) ( )
    _ _ _ _ _ _ _
   | | | | | | | |
  _/ | \ _ _ _ _ _
 |__/_/

Documentation: https://docs.julialang.org
Type "?" for help, "]"?" for Pkg help.

Version 1.11.3 (2025-01-21)
Official https://julialang.org/ release

julia>
```

Рис. 1: Julia

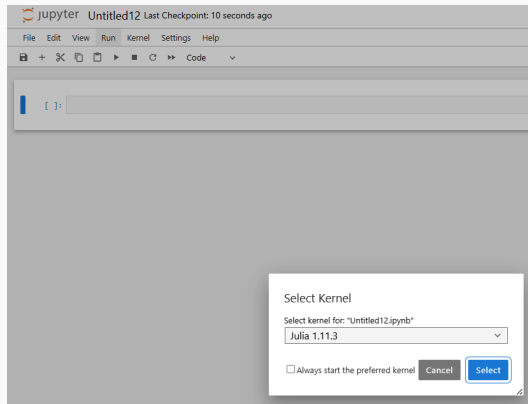


Рис. 2: Jupyter Lab

Примеры определения типа числовых данных

```
[3]: typeof(3), typeof(3.5), typeof(3/3.55), typeof(sqrt(3+4im)), typeof(pi)

[3]: (Int64, Float64, Float64, ComplexF64, Irrational{::π})

[10]: 1.0/0.0, 1.0/(-0.0), 0.0/0.0

[10]: (Inf, -Inf, NaN)

[12]: typeof(1.0/0.0), typeof(1.0/(-0.0)), typeof(0.0/0.0)

[12]: (Float64, Float64, Float64)

[14]: for T in [Int8, Int16, Int32, Int64, Int128, UInt8, UInt16, UInt32, UInt64, UInt128]
      println("${lpad(T,7)}: [$(typemin(T)),$(typemax(T))]" )
    end

      Int8: [-128,127]
      Int16: [-32768,32767]
      Int32: [-2147483648,2147483647]
      Int64: [-9223372036854775808,9223372036854775807]
      Int128: [-170141183460469231731687303715884105728,170141183460469231731687303715884105727]
      UInt8: [0,255]
      UInt16: [0,65535]
      UInt32: [0,4294967295]
      UInt64: [0,18446744073709551615]
      UInt128: [0,340282366920938463463374607431768211455]
```

Рис. 3: Примеры определения типа числовых величин

Примеры приведения аргументов к одному типу

```
[21]: Int64(2.0), Char(2), typeof(Char(2))
```

```
[21]: (2, '\x02', Char)
```

```
[23]: convert{Int64, 2.0}, convert{Char, 2}
```

```
[23]: (2, '\x02')
```

```
[25]: typeof(promote{Int8(1), Float16(4.5), Float32(4.1)})
```

```
[25]: Tuple{Float32, Float32, Float32}
```

```
[29]: Bool(1), Bool(0)
```

```
[29]: (true, false)
```

Рис. 4: Примеры приведения аргументов к одному типу

▼ Примеры определения функции

```
[35]: function f(x)
      x^2
      end
```

```
[35]: f (generic function with 1 method)
```

```
[37]: f(4)
```

```
[37]: 16
```

```
[39]: g(x)=x^2
```

```
[39]: g (generic function with 1 method)
```

```
[41]: g(8)
```

```
[41]: 64
```

Рис. 5: Примеры определения функций

Примеры работы с массивами

```
[44]: a = [4 7 6]  
      b = [1, 2, 3]  
      a[2], b[2]
```

```
[44]: (7, 2)
```

```
[46]: a = 1; b = 2; c = 3; d = 4  
      Am = [a b; c d]
```

```
[46]: 2x2 Matrix{Int64}:  
      1  2  
      3  4
```

```
[48]: Am[1,1], Am[1,2], Am[2,1], Am[2,2]
```

```
[48]: (1, 2, 3, 4)
```

```
[60]: aa = [1 2]  
      AA = [1 2; 3 4]  
      aa*AA*aa'
```

```
[60]: 1x1 Matrix{Int64}:  
      27
```

```
[62]: aa, AA, aa'
```

```
[62]: ([1 2], [1 2; 3 4], [1; 2; ;])
```

Рис. 6: Примеры работы с массивами

```
[5]: # print() u println()
      print("Hello, ")
      println("Julia!")
      println("Julia!")

      Hello, Julia!
      Julia!
```

Рис. 7: Функции print() и println()

```
] : # show()  
x = [5,21,3]  
show(x)  
  
[5, 21, 3]
```

Рис. 8: Функция show()

```
# read()
data_bytes = open("test.txt") do f
  read(f)
end
```

53-element Vector{UInt8}:

0xd0
0xad
0xd1
0x82
0xd0
0xbe
0x20
0xd1
0x81
0xd1
0x82
0xd1
0x80
:
0x20
0xd0
0xb2
0x20
0xd1
0x84
0xd0
0xb0
0xd0
0xb9
0xd0
0xbb

Рис. 9: Функция read()

```
# readline() - чтение одной строки  
line = open("test.txt") do f  
  readline(f)  
end
```

"Это строка, записанная в файл"

```
# readlines() - чтение всех строк  
lines = open("test.txt") do f  
  readlines(f)  
end
```

1-element Vector{String}:
"Это строка, записанная в файл"

Рис. 10: Функции readline(), readlines()

```
: parse(Float64, "3.14")  
  
: 3.14  
  
: parse(Int, "0987654432")  
  
: 987654432
```

Рис. 11: Функция parse()

Выполнение лабораторной работы

[137]:	<code>#сложение</code> <code>5+4</code>
[137]:	9
[139]:	<code>#вычитание</code> <code>235-12</code>
[139]:	223
[141]:	<code>#умножение</code> <code>4*2</code>
[141]:	8
[152]:	<code>#деление</code> <code>9/4</code>
[152]:	2.25
[156]:	<code>#деление с остатком</code> <code>9%4</code>
[156]:	1
[162]:	<code>#возведение в степень</code> <code>12^2</code>
[162]:	144
[164]:	<code>#извлечение корня</code> <code>sqrt(225)</code>
[164]:	15.0

Рис. 12: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня

```
[174]: #комплексные числа  
      z = 6 + 8im  
      abs(z)  
  
[174]: 10.0  
  
[176]: angle(z)  
  
[176]: 0.9272952180016122  
  
[180]: #сравнения и логические операции  
      10 > 8  
  
[180]: true  
  
[182]: 10 == 8  
  
[182]: false  
  
[184]: 10 != 8  
  
[184]: true  
  
[186]: !true  
  
[186]: false  
  
[188]: true&&false  
  
[188]: false
```


Выполнение лабораторной работы

```
[200]: v1 = [1, 2, 3] #вектор столбец  
       v2 = [4 5 6] #вектор строка  
  
[200]: 1x3 Matrix{Int64}:  
       4 5 6  
  
[204]: v1 + v1  
  
[204]: 3-element Vector{Int64}:  
       2  
       4  
       6  
  
[206]: v1 - v1  
  
[206]: 3-element Vector{Int64}:  
       0  
       0  
       0  
  
[208]: v1 * 2  
  
[208]: 3-element Vector{Int64}:  
       2  
       4  
       6  
  
[210]: 2 * v1  
  
[210]: 3-element Vector{Int64}:  
       2  
       4  
       6  
  
[220]: v1'  
  
[220]: 1x3 adjoint{::Vector{Int64}} with eltype Int64:  
       1 2 3
```

Рис. 14: операции над матрицами

```
[224]: A = [1 2; 3 4]  
      B = [5 6; 7 8]
```

```
A + B
```

```
[224]: 2x2 Matrix{Int64}:  
      6  8  
     10 12
```

```
[226]: A - B
```

```
[226]: 2x2 Matrix{Int64}:  
     -4 -4  
     -4 -4
```

```
[228]: A * B
```

```
[228]: 2x2 Matrix{Int64}:  
     19 22  
     43 50
```

Рис. 15: операции над матрицами

Выводы

В результате выполнения лабораторной работы подготовили рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомились с основами синтаксиса Julia.