

# Лабораторная работа №1

Julia. Установка и настройка. Основные принципы.

---

Беличева Д. М.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Беличева Дарья Михайловна
- студентка
- Российский университет дружбы народов
- 1032216453@pfur.ru
- <https://dmbelicheva.github.io/ru/>



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

1. Установите под свою операционную систему Julia, Jupyter.
2. Используя Jupyter Lab, повторите примеры из раздела лабораторной работы.
3. Выполните задания для самостоятельной работы.

## Выполнение лабораторной работы

---



# Простейшие примеры на Julia

```
println("Hello,world!")  
Hello,world!  
  
2+4  
6  
  
io = IOBuffer()  
IOBuffer(data=UInt8{...}, readable=true, writable=true, seekable=true, append=false, size=0, maxsize=Inf, ptr=1, mark=-1)  
  
println(io, "Hello,world!")  
  
String(take!(io))  
"Hello,world!\n"
```

Рис. 2: Выполнение примеров из лабораторной



## Простейшие примеры на Julia

```
[23] function f(x)
      x^2
      end;

[24] f(3)
... 9

[26] g(x) = x^2;

[27] g(4)
... 16

[28] a = [3 4 5]
... 1x3 Matrix{Int64}:
      3  4  5
```

Рис. 3: Выполнение примеров из лабораторной

## Простейшие примеры на Julia

```
[33] a[1], b[3]
... (3, 4)

[3] a = 1; b = 3; c = 4; d = 5;

[40] Am = [a b; c d]
... 2x2 Matrix{Int64}:
 1  3
 4  5

[41] Am[1,1]
... 1

[42] Am'
... 2x2 adjoint(::Matrix{Int64}) with eltype Int64:
 1  4
 3  5
```

Рис. 4: Выполнение примеров из лабораторной

```
▷ n = open("hello.txt", "r")  
[82]  
... IOStream(<file hello.txt>  
  
data = read(n, String)  
[83]  
... "Hello, world\r\nHello, mir\r\nManera krutit"  
  
m = open("hello.txt", "r")  
lines = readline(m)  
[92]  
... "Hello, world"  
  
m = open("hello.txt", "r")  
lines = readlines(m)  
[86]  
... 3-element Vector{String}:  
    "Hello, world"  
    "Hello, mir"  
    "Manera krutit"
```

Рис. 5: Чтение файла

```
using DelimitedFiles
x = [2; 3; 2; 1];

y = ["a"; "b"; "c"; "d"];

open("hello.txt", "w") do io
    for i in 1:length(x)
        println(io, string(x[i], " ", y[i]))
    end
end

readdlm("hello.txt")
```

4x2 Matrix{Any}:

```
2 "a"
3 "b"
2 "c"
1 "d"
```

```
for i in 1:1:3
    print(i, " ")
end
14]
· 1 2 3

for i in 1:1:3
    println(i)
end
15]
· 1
  2
  3

show("Hello, world")
16]
· "Hello, world"

print("Hello, world")
17]
· Hello, world
```

Рис. 7: Вывод на печать

```
io = IOBuffer()
write(io, "big boss", " is here")
[23]
... 16

String(take!(io))
[24]
... "big boss is here"

write(io, "big boss") + write(io, " is here")
[30]
... 16

String(take!(io))
[31]
... "big boss is here"
```

Рис. 8: Команда записи

```
?parse
[28]
... search: parse tryparse partialsortperm partialsortperm! pairs skipchars

... ParseError: KaTeX parse error: No such environment: verbatim at position 7: \begin{verbatim} parse(type, st...

Parse a string as a number. For  $\texttt{Integer}$  types, a base can be specified (the default is 10). For floating-point types, the string is parsed as a decimal floating-point number.  $\texttt{Complex}$  types accept strings of the form  $\texttt{R+lim}$  as a  $\texttt{Complex{R}}$  of the requested type;  $\texttt{I}$  or  $\texttt{J}$  can also be used instead of  $\texttt{im}$ , and  $\texttt{R}$  or  $\texttt{lim}$  are also permitted. If a valid number, an error is raised.

ParseError: KaTeX parse error: No such environment: quote at position 7: \begin{quote} \textbf{compat...

\section{Examples}\begin{verbatim} julia> parse{Int, "1234"} 1234

julia> parse{Int, "1234", base = 5} 194

julia> parse{Int, "abc", base = 16} 2812

julia> parse{Float64, "1.2e-3"} 0.0012

julia> parse{Complex{Float64}, "3.2e-1 + 4.5im"} 0.32 + 4.5im \end{verbatim} \rule{\textwidth}{1pt}

ParseError: KaTeX parse error: No such environment: verbatim at position 7: \begin{verbatim} parse{::Type{P...

Parses a string platform triplet back into a  $\texttt{Platform}$  object.
```

Рис. 9: Документация по функции `parse()`

```
[30] parse(Int, "134567")
... 134567

[32] parse(Float64, "3.14")
... 3.14

[48] parse(Int, "12"; base = 5)
... 7

[61] parse(Int, "hello", base = 30)
... 14167554
```

Функция `parse()` разбирает строки как числа. Аргумент `base` - это основание, показывающее систему счисления (по умолчанию десятичная)

Рис. 10: Примеры использования функции `parse()`



```
a = 2 + 4
b = 2.5 + 2.5
c = "Tom" * "Jerry"
println(a, '\n', b, '\n', c)

[39]
... 6
    5.0
    TomJerry

▷
a = 2 - 4
b = 2.5 - 2.5
println(a, '\n', b)

[40]
... -2
    0.0

a = 5/2
b = 2.5/10
println(a, '\n', b)

[41]
... 2.5
    0.25

a = 3^2
b = 4^3
println(a, '\n', b)

[42]
... 9
    64
```

Рис. 11: Примеры базовых математических операций

```
a = sqrt(64)

[40]
... 8.0

▷
a = 4 > 5
b = 5 == 5
c = 6 < 10
d = 1 != 4
println(a, '\n', b, '\n', c, '\n', d)

[52]
... false
... true
... true
... true

a = true && true
b = true && false
println(a, '\n', b)

[54]
... true
... false

a = true || true
b = true || false
println(a, '\n', b)

[56]
... true
... true

a = !true
b = !false
println(a, '\n', b)
```

Рис. 12: Примеры базовых математических операций

```
▶ A = [2 4; 5 6]
  B = [1 1; 1 1]
  C = A + B
  C

[64]

... 2x2 Matrix{Int64}:
      3  5
      6  7

      A = [1 4; 5 6]
      B = [1 1; 1 1]
      C = A - B
      C

[68]

... 2x2 Matrix{Int64}:
      0  3
      4  5

      A = [2 4; 5 6]
      B = [1 1; 1 1]
      C = A * B
      C

[69]

... 2x2 Matrix{Int64}:
      6  6
     11 11
```

Рис. 13: примеры операций над матрицами

```
A = [2 4; 5 6]
k = 2
C = k * A

[73]
*** 2x2 Matrix{Int64}:
    4  8
   10 12

A

[74]
*** 2x2 Matrix{Int64}:
    2  4
    5  6

A'

[75]
*** 2x2 adjoint{::Matrix{Int64}} with eltype Int64:
    2  5
    4  6

using LinearAlgebra
u = [1, 2, 3]
v = [1, 1, 2]
prod = dot(u,v)

[76]
*** 9

u = [1, 2, 3]
v = [1, 1, 2]
prod = cross(u,v)

[77]
*** 3-element Vector{Int64}:
    1
    1
   -1
```

Рис. 14: примеры операций над векторами

В результате выполнения данной лабораторной работы я подготовила рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомилась с основами синтаксиса Julia.

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).