

Лабораторная работа № 1. Julia. Установка и настройка. Основные принципы.

Абакумова О. М.

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Абакумова Олеся Максимовна
- Студентка
- Российский университет дружбы народов
- 1132220832@pfur.ru
- <https://github.com/omabakumova>



Цель работы

Цель работы

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

Задания

Задания

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

Выполнение лабораторной работы

Подготовка инструментария к работе

```
lesya@lesya-Aspire-A115-32: ~ julia
The latest version of Julia in the 'release' channel is 1.11.6+0.x64.linux.gnu. You currently have '1.11.3+0.x64.linux.gnu' installed. Run:
juliaup update
in your terminal shell to install Julia 1.11.6+0.x64.linux.gnu and update the 'release' channel to that version.

  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: Запуск Julia

Подготовка инструментария к работе

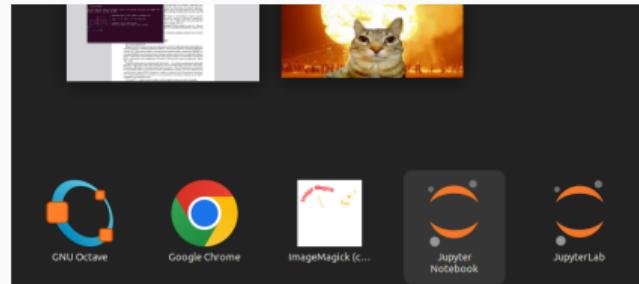
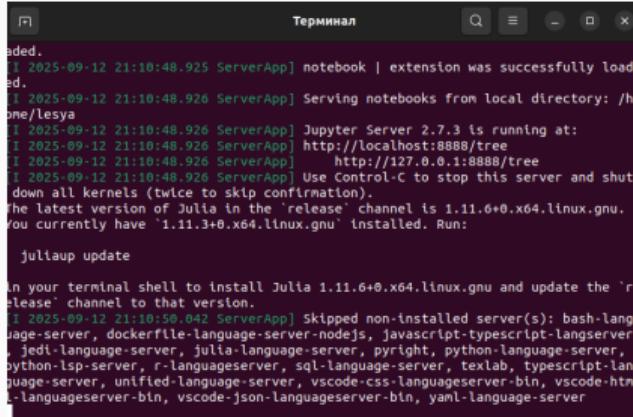


Рис. 2: Запуск Jupyter

Подготовка инструментария к работе



```
[I 2025-09-12 21:10:48.925 ServerApp] notebook | extension was successfully loaded.
[I 2025-09-12 21:10:48.926 ServerApp] Serving notebooks from local directory: /home/lesya
[I 2025-09-12 21:10:48.926 ServerApp] Jupyter Server 2.7.3 is running at:
[I 2025-09-12 21:10:48.926 ServerApp] http://localhost:8888/tree
[I 2025-09-12 21:10:48.928 ServerApp] http://127.0.0.1:8888/tree
[I 2025-09-12 21:10:48.928 ServerApp] Use Control-C to stop this server and shut
down all kernels (twice to skip confirmation).
The latest version of Julia in the 'release' channel is 1.11.6+0.x64.linux.gnu.
You currently have '1.11.3+0.x64.linux.gnu' installed. Run:
    juliaup update
In your terminal shell to install Julia 1.11.6+0.x64.linux.gnu and update the 'r'
channel to that version.
[I 2025-09-12 21:10:50.042 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langs
erver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-languag
e-server, unified-language-server, vscode-css-languageserver-bin, vscode-htm
l-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
```

Рис. 3: Запуск Jupyter

Подготовка инструментария к работе

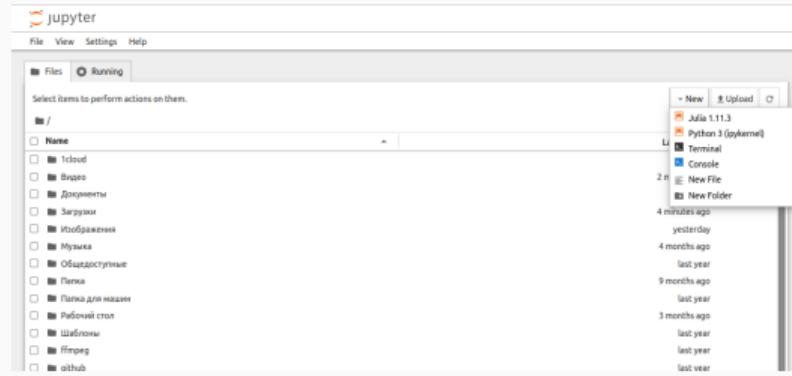


Рис. 4: Создание нового блокнота для Julia

Подготовка инструментария к работе

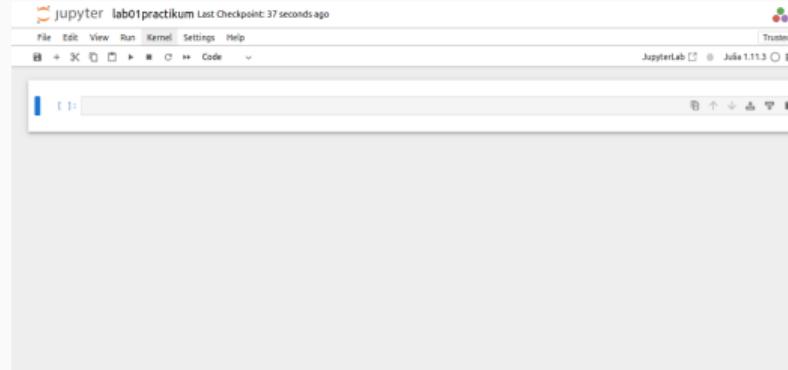


Рис. 5: Блокнот имеет ядро Julia

Подготовка инструментария к работе

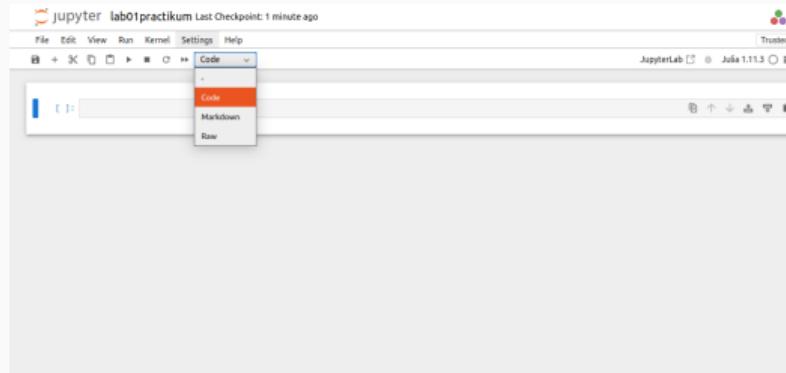
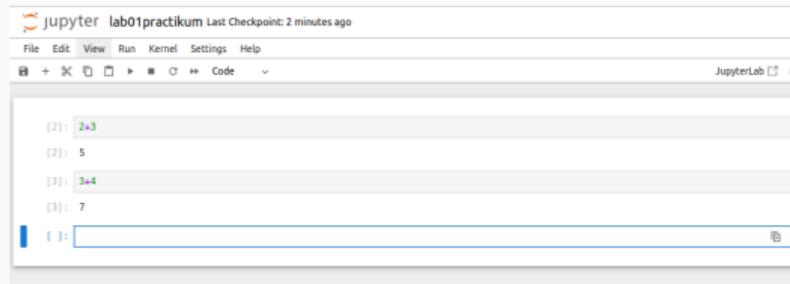


Рис. 6: Режимы вставки ячейки

Основы работы в блокноте Jupyter



The screenshot shows a Jupyter Lab interface with a toolbar at the top. The title bar reads "jupyter lab01praktikum Last Checkpoint: 2 minutes ago". The menu bar includes File, Edit, View, Run, Kernel, Settings, and Help. A "Code" dropdown is open. The main area displays a code editor with four cells containing Julia code and their results:

```
[2]: 2+3
[2]: 5
[3]: 3+4
[3]: 7
[1]: 
```

Рис. 7: Простейшие операции на языке Julia в Jupyter Lab

Основы работы в блокноте Jupyter

```
[4]: ?println
search: println print sprint pointer printstyled
[4]: println([io::IO], xs...)
Print (using print) xs to io followed by a newline. If io is not supplied, prints to the default output stream stdout.
See also printstyled to add colors etc.

Examples
julia> println("Hello, world")
Hello, world

julia> io = IOBuffer();
julia> println(io, "Hello", '.', " world.")
julia> String(take!(io))
"Hello, world.\n"
```

Рис. 8: Пример получения информации по функции println на языке Julia в Jupyter Lab

Основы работы в блокноте Jupyter

```
[5]: :date
Пт 12 сен 2025 21:17:28 MSK

[6]: :whoami
lesya

[ ]:
```

Рис. 9: Пример получения информации о дате и пользователе ОС Linux в Jupyter Lab

Основы работы в блокноте Jupyter

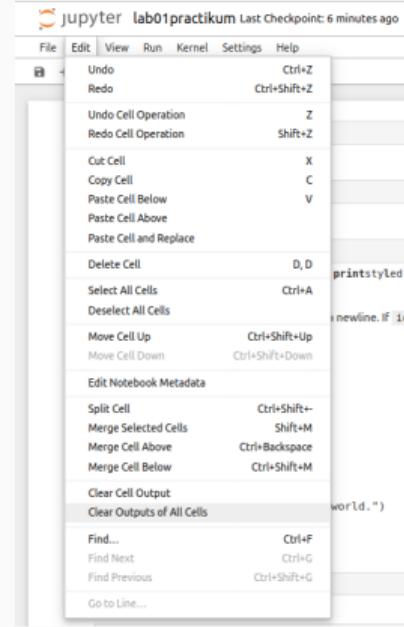


Рис. 10: Очистка результатов выполнения ячеек

Основы синтаксиса Julia на примерах



The screenshot shows a Julia REPL window. The user has run the command `typeof(Number)` which returns `DataType`. Then, they have run a loop that prints the minimum and maximum values for various integer types. The output is as follows:

```
[7]: typeof(Number)
[7]: DataType
[8]: 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: [0,340282366928938463463374607431768211455]
```

Рис. 11: Определение крайних значений диапазонов целочисленных числовых величин и определение типа числовой величины

Основы синтаксиса Julia на примерах

```
[9]: Int64(2.0), Char(2)
[9]: (2, '\x02')

[10]: convert(Int64, 2.0), convert(Char, 2)
[10]: (2, '\x02')

[11]: Bool(1), Bool(0)
[11]: (true, false)

[12]: promote(Int8(1), Float16(4.5), Float32(4.1))
[12]: (1.0f0, 4.5f0, 4.1f0)
```

Рис. 12: Примеры приведения аргументов к одному типу

Основы синтаксиса Julia на примерах

```
[13]: function f(x)
           x^2
       end
f(4)

[13]: 16

[14]: a = [4 7 6] # вектор-строка
       b = [1, 2, 3] # вектор-столбец
       a[2], b[2] # вторые элементы векторов a и b

[14]: (7, 2)

[1]: |
```

Рис. 13: Пример определения одномерных массивов и пример определения функций

Основы синтаксиса Julia на примерах

```
[15]: g(x) = x^2
[15]: g (generic function with 1 method)
[16]: g(8)
[16]: 64
```

Рис. 14: Пример определения функций

Основы синтаксиса Julia на примерах

```
[18]: a = 1; b = 2; c = 3; d = 4 # присвоение значений  
Am = [a b; c d] # матрица 2 x 2
```

```
[18]: 2x2 Matrix{Int64}:  
      1 2  
      3 4
```

```
[19]: Am[1,1], Am[1,2], Am[2,1], Am[2,2] # элементы матрицы
```

```
[19]: (1, 2, 3, 4)
```

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

```
[25]: 1x1 Matrix{Int64}:  
      27
```

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

```
[24]: ([1 2], [1 2; 3 4], [1; 2;:])
```

```
[ ]:
```

Рис. 15: Примеры работы с массивами

Задания для самостоятельной работы

1. Изучите документацию по основным функциям Julia для чтения / записи / вывода информации на экран: `read()`, `readline()`, `readlines()`, `readdlm()`, `print()`, `println()`, `show()`, `write()`. Приведите свои примеры их использования, поясняя особенности их применения.

Задания для самостоятельной работы

```
new
[1] using DelimitedFiles
[2] data = readtable("Datasets/csv/caterpillar_faces.csv", ",")  

[3] 382x4 Matrix{key: "Vertex 1", "Vertex 2", "Vertex 3", "Vertex 4"}  

   0   1   2   3   4  

   1   2   22  21  

   2   3   23  22  

   3   4   24  23  

   4   5   25  24  

   5   6   26  25  

   6   7   27  26  

   7   8   28  27  

   8   9   29  28  

   9   10  30  29  

  10  11  31  30  

  11  12  32  31  

  
 367  368  388  387  

  369  369  389  388  

  370  370  390  389  

  370  371  391  390  

  371  372  392  391  

  372  373  393  392  

  373  374  394  393  

  374  375  395  394  

  375  376  396  395  

  376  377  397  396  

  377  378  398  397  

  378  379  399  398  

  
[4] println("Hello")
Hello
[5] println("World")
World
[6] show(stdout, 3.14159)
3.14159
[7] write("output.htm", data)
```

Рис. 17: Функции readlm(), print(), println(), show(), write()

Задания для самостоятельной работы

2. Изучите документацию по функции `parse()`. Приведите свои примеры её использования, поясняя особенности её применения.

Задания для самостоятельной работы

```
*[10]: x = parse(Int, "42")# 42 (целое число)
[10]: 42
*[11]: y = parse(Float64, "3.14")# 3.14 (число с плавающей точкой)
[11]: 3.14
*[12]: z = parse(Bool, "true")# true (логическое значение)
[12]: true
```

Рис. 18: Функция parse()

Задания для самостоятельной работы

3. Изучите синтаксис Julia для базовых математических операций с разным типом переменных: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, сравнение, логические операции. Приведите свои примеры с пояснениями по особенностям их применения.

Задания для самостоятельной работы

```
[14]: a = 2
[14]: 2
[16]: b = 4
[16]: 4
*[18]: a + b
# сложение
[18]: 6
*[19]: a - b
# вычитание
[19]: -2
*[20]: a * b
# умножение
[20]: 8
*[21]: a / b
# деление (вещественное)
[21]: 0.5
*[22]: a % b
# целочисленное деление (div(a, b))
[22]: 0
*[23]: a % b
# остаток от деления (mod(a, b))
[23]: 2
*[24]: a ^ b
# возведение в степень
[24]: 16
*[25]: sqrt(a)
# квадратный корень
[25]: 1.4142135623730951
```

Рис. 19: Математические операции с разным типом переменных

Задания для самостоятельной работы

```
[26]: a == b
# равно
[26]: false

[27]: a != b
# не равно
[27]: true

[28]: a < b
# меньше
[28]: true

[29]: a > b
# больше
[29]: false

[30]: a == b
# меньше или равно
[30]: true

[31]: a != b
# больше или равно
[31]: false

[39]: a == true
[39]: true

[40]: b == false
[40]: false

[41]: "a"
# строка
[41]: false

[42]: a && b
# логическое И
[42]: false

[43]: a || b
# логическое ИЛИ
[43]: true
```

Рис. 20: Математические операции с разным типом переменных(логические операции)

Задания для самостоятельной работы

4. Приведите несколько своих примеров с пояснениями с операциями над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр.

Задания для самостоятельной работы

```
[46]: using LinearAlgebra
A = [1 2; 3 4] # матрица 2x2
B = [5 6; 7 8]
v = [1, 2, 3] # вектор-столбец
w = [4, 5, 6]

[46]: 2x2 Matrix{Int64}:
 5 12
21 32

[47]: A + B
# поллементное сложение

[47]: 2x2 Matrix{Int64}:
 6 0
10 12

[48]: A - B
# поллементное вычитание

[48]: 2x2 Matrix{Int64}:
 -4 -4
 -4 -4

[49]: A * B
# матричное умножение

[49]: 2x2 Matrix{Int64}:
 19 22
 43 50

[50]: A'
```

транспонирование

```
[51]: 2x2 offdiag() Matrix{Int64} with eltype Int64:
 1 3
 2 4

[52]: dot(v, w)
# скалярное произведение векторов

[52]: 32

[53]: A .* B
# поллементное умножение

[53]: 2x2 Matrix{Int64}:
 5 12
21 32
```

Рис. 21: Операции над матрицами и векторами

Выводы

Выводы

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