

Лабораторная работа 1

Julia. Установка и настройка. Основные принципы.

Ланцова Я. И.

Российский университет дружбы народов, Москва, Россия

Информация

- Ланцова Яна Игоревна
- студентка
- Российский университет дружбы народов

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

Установка Julia и знакомство с синтаксисом

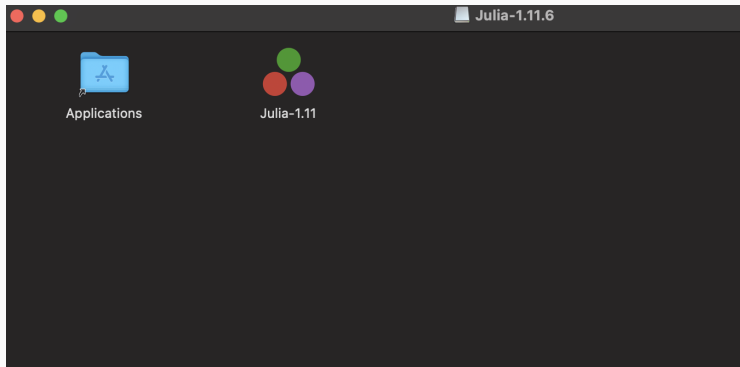


Рис. 1: Установка Julia

Выполнение лабораторной работы

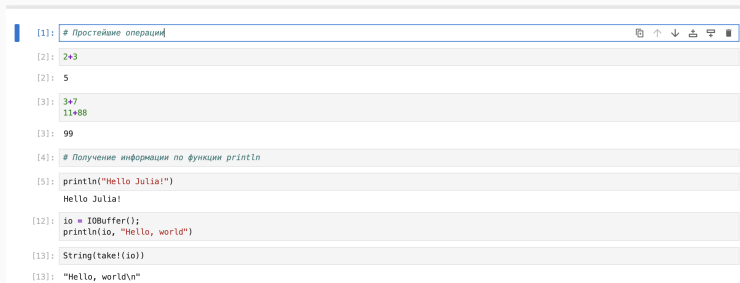
```
[julia> using Pkg

[julia> Pkg.add("IJulia")
    Updating registry at `~/\.julia/registries/General.toml`
    Resolving package versions...
    No Changes to `~/\.julia/environments/v1.11/Project.toml`
    No Changes to `~/\.julia/environments/v1.11/Manifest.toml`

[julia> using IJulia

[julia> notebook()
[ Info: running setenv(`/Users/yana/.julia/conda/3/aarch64/bin/jupyter notebook`,["XPC_FLAGS=0x0", "_CE_M=", "PATH=/Users/yana/.julia/conda/3/aarch64/bin:/Users/yana/anaconda3/condabin:/opt/homebrew/bin:/opt/homebrew/sbin:/opt/local/bin:/opt/local/sbin:/Library/Frameworks/Python.framework/Versions/3.12/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/Library/Frameworks/Python.framework/Versions/3.9/bin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Users/yana/Downloads/harbour/bin:/Applications/VMware Fusion.app/Contents/Public:/Library/TeX/texbin:/usr/local/go/bin:/opt/homebrew/bin:/usr/local/share/dotnet:/usr/local/share/dotnet/tools:/Library/Apple/usr/bin:/Applications/quarto/bin:/Applications/Postgres.app/Contents/Versions/latest/bin", "PWD=/Users/yana", "XPC_SERVICE_NAME=0", "TERM_PROGRAM=Apple_Terminal", "HOMEBREW_PREFIX=/opt/homebrew", "SHELL=/bin/zsh", "_CF_USER_TEXT_ENCODING=0x1F5:0x7:0x31", "LC_CTYPE=UTF-8", "_CFBundleIdentifier=com.apple.Terminal", "TMPDIR=/var/folders/mj/y0wsc8sn0d5cpn858fdg38dw0000gn/T/", "CONDARC=/Users/yana/.julia/conda/3/aarch64/condarc-julia.yml", "HOMEBREW_CELLAR=/opt/homebrew/Cellar", "LOGNAME=yana", "SHLVL=1", "LaunchInstanceID=8E3AD041-B64A-480C-B4B6-5730D1DFD1CD", "CONDA_PREFIX=/Users/yana/.julia/conda/3/aarch64", "SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.Uh5791XTQ2/Listeners", "TERM_SESSION_ID=61B3C30C-3032-4AAF-B3D8-7F48DF2E925B", "OLDPWD=/Users/yana", "INFOPATH=/opt/homebrew/share/info:/opt/homebrew/share/info:", "=/Applications/Julia-1.11.app/Contents/Resources/julia/bin/julia", "HOMEBREW_REPOSITORY=/opt/homebrew", "OPENBLAS_DEFAULT_NUM_THREADS=1", "_CE_CONDA=", "SECURITYSESSIONID=186b2", "USER=yana", "TERM=xterm-256color", "HOME=/Users/yana", "TERM_PROGRAM_VERSION=447", "OPENBLAS_MAIN_FREE=1", "PYTHONIOENCODING=UTF-8"]])
```

Рис. 2: Установка пакетов



The screenshot displays a Jupyter Notebook with the following content:

- Cell [1]: `# Простейшие операции`
- Cell [2]: `2+3` (Output: 5)
- Cell [2]: `5` (Output: 5)
- Cell [3]: `3+7` (Output: 11)
- Cell [3]: `11*88` (Output: 99)
- Cell [3]: `99` (Output: 99)
- Cell [4]: `# Получение информации по функции println`
- Cell [5]: `println("Hello Julia!")` (Output: Hello Julia!)
- Cell [12]: `io = IOBuffer(); println(io, "Hello, world")`
- Cell [13]: `String{take!(io)}` (Output: "Hello, world\n")

Рис. 3: Выполнение примеров из лабораторной

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

[18]: f(9)

[18]: 81

[17]: g(x)=x^2;

[19]: g(11)

[19]: 121

[25]: # вектор-строка
      a = [4 7 6]

[25]: 1×3 Matrix{Int64}:
      4  7  6

[27]: # вектор-столбец
      b = [1, 2, 3]

[27]: 3-element Vector{Int64}:
      1
      2
      3

[30]: a[1], b[1]

[30]: (4, 1)
```

Рис. 4: Выполнение примеров из лабораторной

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

[33]: 2x2 Matrix{Int64}:
      1 2
      3 4

[34]: Am[1,1]

[34]: 1

[35]: Am'

[35]: 2x2 adjoint{::Matrix{Int64}} with eltype Int64:
      1 3
      2 4

[ ]:
```

Рис. 5: Выполнение примеров из лабораторной

Задание 1

```
[41]: read("lab01_julia.txt", String)
[41]: "Hello everyone!\nWelcome to Julia!"

[40]: print("Enter your name: ")
      your_name = readline()
      Enter your name:
      stdin> Yana
[40]: "Yana"

[42]: readline("lab01_julia.txt")
[42]: "Hello everyone!"

[43]: readlines("lab01_julia.txt", keep=true)
[43]: 2-element Vector{String}:
      "Hello everyone!\n"
      "Welcome to Julia!"
```

Рис. 6: Чтение файла

Выполнение лабораторной работы

```
[45]: using DelimitedFiles

[58]: readddlm("lab01_julia.txt", Int)

[58]: 3x2 Matrix{Int64}:
      7  2
      4  5
      6  6

[62]: io = IOBuffer();
      print(io, "Yana the best")
      String(take!(io))

[62]: "Yana the best"

[66]: for i in 1:1:5
      print(i)
      end
      12345

[67]: for i in 1:1:5
      println(i)
      end
      1
      2
      3
      4
      5

[65]: show("I am 4 yo")
      "I am 4 yo"
```

Рис. 7: Вывод на печать

```
[71]: io = IOBuffer();  
      write(io, "I am 4 yo")  
[71]: 9  
[72]: String(take!(io))  
[72]: "I am 4 yo"  
[73]: write(io, "I love") + write(io, " eating food")  
[73]: 18  
[74]: String(take!(io))  
[74]: "I love eating food"  
[ ]:
```

Рис. 8: Команда записи

Задание 2

```
[75]: parse(Int, "1388")  
[75]: 1388  
  
[76]: parse(Float64, "1.388")  
[76]: 1.388  
  
[77]: parse(Int, "a", base = 16)  
[77]: 10  
  
[ ]: |
```

Рис. 9: Примеры использования функции `parse()`

Задание 3

```
[87]: ex1 = 2 + 4;
      ex2 = 2.2 + 4.4;
      println(ex1, '\n', ex2)

      6
      6.6000000000000005

[88]: ex3 = 2 - 4;
      ex4 = 2.2 - 4.4;
      println(ex3, '\n', ex4)

      -2
      -2.2

[90]: ex5 = 10 * 5;
      ex6 = 1.21 * 10;
      println(ex5, '\n', ex6)

      50
      12.1

[91]: ex7 = 10 / 5;
      ex8 = 1.21 / 10;
      println(ex7, '\n', ex8)

      2.0
      0.121

[92]: ex9 = 10 ^ 5;
      ex10 = 1.1 ^ 2;
      println(ex9, '\n', ex10)

      100000
      1.2100000000000002
```

[]: |

Рис. 10: Примеры базовых математических операций

Выполнение лабораторной работы

```
[93]: ex11 = sqrt(121)
[93]: 11.0

[96]: first = 6;
      second = 9;
      first == second
[96]: false

[99]: first != second
[99]: true

[100]: first > second
[100]: false

[101]: first < second
[101]: true

[102]: true && false
[102]: false

[103]: true && true
[103]: true

[104]: true || false
[104]: true

[105]: false || false
[105]: false
```

Рис. 11: Примеры базовых математических операций

Задание 4

```
[108]: vec1 = [2 5];  
      vec2 = [8 1];  
      matrix1 = [2 5; 9 1];  
      matrix2 = [8 1; 0 3];
```

```
[109]: vec1 + vec2
```

```
[109]: 1x2 Matrix{Int64}:  
      10  6
```

```
[110]: matrix1 + matrix2
```

```
[110]: 2x2 Matrix{Int64}:  
      10  6  
       9  4
```

```
[111]: vec1 - vec2
```

```
[111]: 1x2 Matrix{Int64}:  
      -6  4
```

```
[112]: matrix1 - matrix2
```

```
[112]: 2x2 Matrix{Int64}:  
      -6  4  
       9 -2
```

```
[113]: matrix1 * matrix2
```

```
[113]: 2x2 Matrix{Int64}:  
      16  17  
      72  12
```

```
[116]: vec1 * matrix1
```

```
[116]: 1x2 Matrix{Int64}:  
      49  15
```

Рис. 12: Примеры операций над матрицами

```
[117]: matrix1''  
[117]: 2×2 Matrix{Int64}:  
      2  5  
      9  1  
[118]: vec1''  
[118]: 1×2 Matrix{Int64}:  
      2  5  
[119]: matrix1 * 2  
[119]: 2×2 Matrix{Int64}:  
      4  10  
     18   2  
[121]: vec1 * 2  
[121]: 1×2 Matrix{Int64}:  
      4  10  
[123]: using LinearAlgebra  
[124]: dot(vec1, vec2)  
[124]: 21  
[128]: cross([0;1;0], [0;0;1])  
[128]: 3-element Vector{Int64}:  
      1  
      0  
      0
```

Рис. 13: Примеры операций над матрицами

Выводы

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.