

Отчет по лабораторной работе №1

Julia.Установка и настройка.Основные принципы.

Легиньких Галина Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Вывод	16

Список иллюстраций

3.1	Загрузка Far Manager	7
3.2	Загрузка Notepad++	8
3.3	Загрузка Julia	8
3.4	Установка пакетов для работы с Jupyter	8
3.5	Тип величины	9
3.6	Диапазон величин	9
3.7	Преобразование типов	10
3.8	Функции	10
3.9	Одномерные массивы	10
3.10	Двумерные массивы	11
3.11	Функция записи	11
3.12	Разные функции чтения	11
3.13	Разные функции вывода	12
3.14	Функция parse()	12
3.15	Математические операции 1	13
3.16	Математические операции 2	14
3.17	Операции над матрицами 1	14
3.18	Операции над матрицами 2	15

Список таблиц

1 Цель работы

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

2 Задание

1. Установите под свою операционную систему Julia, Jupyter.
2. Используя Jupyter Lab, повторите примеры из раздела 1.3.3.
3. Выполните задания для самостоятельной работы:
 - Изучите документацию по основным функциям Julia для чтения / записи / вывода информации на экран: `read()`, `readline()`, `readlines()`, `readline()`, `print()`, `println()`, `show()`, `write()`. Приведите свои примеры их использования, поясняя особенности их применения.
 - Изучите документацию по функции `parse()`. Приведите свои примеры её использования, поясняя особенности её применения.
 - 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.
 - Приведите несколько своих примеров с пояснениями с операциями над матрицами и векторами: сложение

3 Выполнение лабораторной работы

1. Для начала нужно было скачать менеджер пакетов Chocolatey. Как оказалось у меня это уже скачено.

2. Через Administrative Shell я скачала Far Manager (рис. 3.1), Notepad++ (рис. 3.2), Julia (рис. 3.3). Anaconda пришлось скачать с официального сайта.

```
PS D:\> choco install far -y
Chocolatey v2.2.2
Installing the following packages:
far
By installing, you accept licenses for the packages.
Progress: Downloading Far 3.0.6364... 100%
far v3.0.6364 [Approved]
far package files install completed. Performing other installation steps.
Downloading Far 64 bit
from 'https://www.farmanager.com/files/Far30b6364.x64.20240824.msi'
Progress: 100% - Completed download of C:\Users\galin\AppData\Local\Temp\chocolatey\Far\3.0.6364\Far30b6364.x64.20240824.msi (15.48 MB).
Download of Far30b6364.x64.20240824.msi (15.48 MB) completed.
Hashes match.
Installing Far...
Far has been installed.
far may be able to be automatically uninstalled.
The install of far was successful.
Software installed to 'C:\Program Files\Far Manager\'
Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

Рис. 3.1: Загрузка Far Manager

```

PS D:\> choco install notepadplusplus -y
Chocolatey v2.2.2
Installing the following packages:
notepadplusplus
By installing, you accept licenses for the packages.
Progress: Downloading notepadplusplus.install 8.7.1.20241108... 100%

notepadplusplus.install v8.7.1.20241108 [Approved]
notepadplusplus.install package files install completed. Performing other installation steps.
Installing 64-bit notepadplusplus.install...
notepadplusplus.install has been installed.
WARNING: No registry key found based on 'Notepad\+.*'
WARNING: Can't find notepadplusplus.install install location
notepadplusplus.install can be automatically uninstalled.
The install of notepadplusplus.install was successful.
Software installed as 'exe', install location is likely default.
Progress: Downloading notepadplusplus 8.7.1.20241108... 100%

notepadplusplus v8.7.1.20241108 [Approved]
notepadplusplus package files install completed. Performing other installation steps.
The install of notepadplusplus was successful.
Software install location not explicitly set, it could be in package or
default install location of installer.

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

```

Рис. 3.2: Загрузка Notepad++

```

PS D:\> choco install julia -y
Chocolatey v2.2.2
Installing the following packages:
julia
By installing, you accept licenses for the packages.
julia v1.10.5 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- julia - julia v1.10.5 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
PS D:\>

```

Рис. 3.3: Загрузка Julia

3. После установки запустила Julia в режиме REPL. Перешла в пакетный режим Julia и ввела add IJulia для установки пакета для работы с Jupyter. (рис. 3.4)

```

(@v1.10) pkg> add IJulia
Installing known registries into 'C:\Users\galin\.julia'
Added 'General' registry to C:\Users\galin\.julia\registries
Updating registry at 'C:\Users\galin\.julia\registries\General.toml'
Resolving package versions...
Installed Parsers _____ v2.8.1
Installed ZMQ _____ v1.3.0
Installed JSON _____ v0.21.4
Installed Conda _____ v1.10.2
Installed Preferences _____ v1.4.3
Installed IJulia _____ v1.26.0
Installed JLLWrappers _____ v1.6.1
Installed PrecompileTools _____ v1.2.1
Installed VersionParsing _____ v1.3.0
Installed SoftGlobalScope _____ v1.1.0
Installed libsodium_jll _____ v1.0.20+1
Installed ZeroMQ_jll _____ v4.3.5+1
Installed MbedTLS _____ v1.1.9

```

Рис. 3.4: Установка пакетов для работы с Jupyter

4. Я буду использовать Jupyter Lab. Перешла через Anaconda в Jupyter Lab и создала новый блокнот.

5. Перешла к следующему пункту задания. Нужно было повторить примеры синтаксиса Julia. (Некоторые комментарии есть на скриншотах в блокноте)

- Определение типа числовой величины. (рис. 3.5)

```
[5]: typeof(3)
[5]: Int64

[7]: typeof(3.5)
[7]: Float64

[9]: typeof(3/3.5)
[9]: Float64

[11]: typeof(5/0)
[11]: Float64
```

Рис. 3.5: Тип величины

- Определение крайних значений диапазонов целочисленных числовых величин. (рис. 3.6)

```
[15]: 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. 3.6: Диапазон величин

- Преобразование типов. (рис. 3.7)

```

[17]: Int64(2.0)
[17]: 2
[19]: Char(2)
[19]: '\x02': ASCII/Unicode U+0002 (category Cc: Other, control)
[21]: convert(Int64, 2.0)
[21]: 2
[23]: convert(Char, 2)
[23]: '\x02': ASCII/Unicode U+0002 (category Cc: Other, control)
[25]: Bool(1)
[25]: true
[27]: promote(Int8(1), Float16(4.5), Float32(4.1))
[27]: (1.0f0, 4.5f0, 4.1f0)
[29]: typeof(1.0f0)
[29]: Float32

```

Рис. 3.7: Преобразование типов

- Синтаксис определения функций. (рис. 3.8)

```

[35]: function f(x)
      x^2
      end
      f(4)
[35]: 16
[37]: g(x)=x^2
      g(5)
[37]: 25

```

Рис. 3.8: Функции

- Одномерные массивы. (рис. 3.9)

```

[43]: a = [4 7 6] # вектор-строка
      b = [1, 2, 3] # вектор-столбец
      a[2], b[2] # вторые элементы векторов a и b
[43]: (7, 2)

```

Рис. 3.9: Одномерные массивы

- Двумерные массивы. (рис. 3.10)

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

[45]: (1, 2, 3, 4)

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

[47]: 1x1 Matrix{Int64}:
      27
```

Рис. 3.10: Двумерные массивы

6. Перешла к заданиям для самостоятельной работы. Нужно было изучить документацию по некоторым функциям. Ниже будет функция и пример к ней.

7. Функции для чтения/записи/вывода информации:

- write() (рис. 3.11)

```
[83]: io = IOBuffer()
      write("io.txt", "Мы изучаем функцию write.\n", "Мы изучили функцию write.")

[83]: 83
```

Рис. 3.11: Функция записи

- read()
- readline()
- readlines() (рис. 3.12)

```
[85]: read("io.txt", String)

[85]: "Мы изучаем функцию write.\nМы изучили функцию write."

[91]: readline("io.txt")

[91]: "Мы изучаем функцию write."

[93]: readlines("io.txt")

[93]: 2-element Vector{String}:
      "Мы изучаем функцию write."
      "Мы изучили функцию write."
```

Рис. 3.12: Разные функции чтения

- `print()`
- `println()`
- `show()` (рис. 3.13)

```
[107]: print("Компьютерный практикум")
      print("Компьютерный практикум")
      Компьютерный практикумКомпьютерный практикум

[115]: println("Компьютерный практикум")
      println("Компьютерный практикум")
      Компьютерный практикум
      Компьютерный практикум

[119]: show("Компьютерный практикум")
      "Компьютерный практикум"
```

Рис. 3.13: Разные функции вывода

8. Изучила и придумала пример к функции `parse()`. Эта функция преобразует строковое представление числа в число. (рис. 3.14)

```
[125]: parse(Int, "234")
[125]: 234

[129]: parse(Int, "234", base = 5)
[129]: 69

[131]: parse(Float64, "1.0e-3")
[131]: 0.001

[135]: parse(Complex{Float64}, "3.0e-1 + 3.1im")
[135]: 0.3 + 3.1im

[137]: parse(Complex{Int}, "3 + 3im")
[137]: 3 + 3im
```

Рис. 3.14: Функция `parse()`

9. Попробовала базовые математические операций с разным типом переменных: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, сравнение, логические операции. Подробно нет смысла описывать, некоторые комментарии будут на скриншоте. Привела свои примеры. (рис. 3.15) (рис. 3.16)

```

[159]: 3 + 3.4
[159]: 6.4

[143]: 4 + 1.3e+4
[143]: 13004.0

[145]: 5.78 - 4
[145]: 1.7800000000000002

[147]: 7 * 4.7
[147]: 32.9

[149]: 5/0
[149]: Inf

[151]: 3.4/6
[151]: 0.5666666666666667

[153]: 5.9^2
[153]: 34.81

[155]: 4^(1/2)
[155]: 2.0

```

Рис. 3.15: Математические операции 1

```

[157]: 4.1^(1/2)
[157]: 2.0248456731316584

[161]: 2 == 3
[161]: false

[163]: 2 >= 1
[163]: true

[165]: 1.0 == 1
[165]: true

[169]: !true
[169]: false

[171]: true && true
[171]: true

[175]: true && false
[175]: false

[177]: true || false
[177]: true

```

Рис. 3.16: Математические операции 2

10. Привела примеры операций над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр. (рис. 3.17) (рис. 3.18)

```

[179]: a = [1 2 3]
[179]: 1x3 Matrix{Int64}:
 1  2  3

[181]: b = [1, 3, 4]
[181]: 3-element Vector{Int64}:
 1
 3
 4

[189]: c = [1 2; 3 4]
[189]: 2x2 Matrix{Int64}:
 1  2
 3  4

[191]: d = [2 3; 5 6]
[191]: 2x2 Matrix{Int64}:
 2  3
 5  6

```

Рис. 3.17: Операции над матрицами 1

```

[195]: a * c
[195]: 1x3 Matrix{Int64}:
      3  6  9

[200]: b
[200]: 1x3 adjoint{::Vector{Int64}} with eltype Int64:
      1  3  4

[209]: d'
[209]: 2x2 adjoint{::Matrix{Int64}} with eltype Int64:
      2  5
      3  6

[211]: a * 5
[211]: 1x3 Matrix{Int64}:
      5 10 15

```

Рис. 3.18: Операции над матрицами 2

4 Вывод

Подготовила рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомилась с основами синтаксиса Julia.