Лабораторная работа №1

Julia. Установка и настройка. Основные принципы.

Шияпова Дарина Илдаровна

Содержание

Список иллюстраций

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 лабораторной работы.
3. Выполните задания для самостоятельной работы.

3 Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений [**julialang?**]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia [**juliadoc?**].

4 Выполнение лабораторной работы

У меня уже были установлены Julia и Jupyter.

Теперь повторим простейшие примеры для знакомства с синтаксисом Julia (рис. 4.1-4.3).

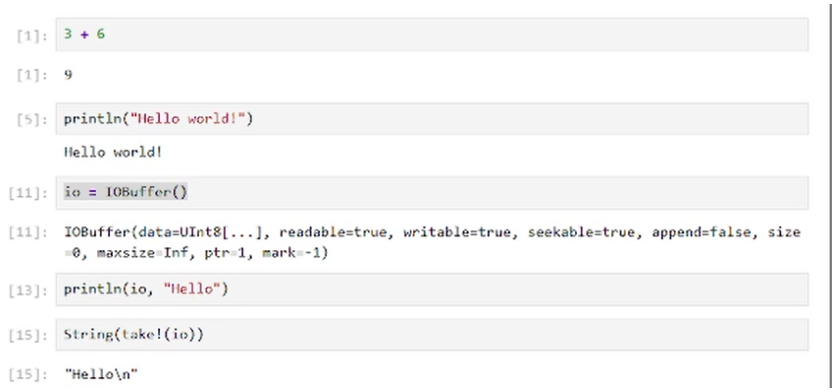

Рис. 4.1: Выполнение примеров из лабораторной

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

[21]: f (generic function with 1 method)

[23]: f(5)

[23]: 25

[25]: g(x) = x^3

[25]: g (generic function with 1 method)

[27]: g(2)

[27]: 8
```

Рис. 4.2: Выполнение примеров из лабораторной

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

[37]: Am = [1 2; 3 4]

[37]: 2×2 Matrix{Int64}:
       1  2
       3  4

[43]: Am[1, 1], Am[1, 2], Am[2, 1], Am[2, 2]

[43]: (1, 2, 3, 4)

[45]: aa = [1 2]
      AA = [1 2; 3 4]

[45]: 2×2 Matrix{Int64}:
       1  2
       3  4
```

Рис. 4.3: Выполнение примеров из лабораторной

Теперь перейдем к выполнению заданий.

Задание №1

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

их применения.

Для того, чтобы ознакомиться с документацией достаточно поставить знак ? перед интересующей функцией. Например, `?print`.

Создадим текстовый файл с любым содержанием в папке, где мы работаем. Откроем его на чтение и прочитаем с помощью команды `read()`. Текст вывелся в одну строку с разделителями `\r\n`. Также прочитаем текст используя функцию `readline()` - выведется только первая строка. Чтобы прочитать все строки в файле используем команду `readlines()` (рис. 4.4).

Рис. 4.4: Чтение файла

Далее посмотрим, как работает команда `println()`, `print()` и `show()` (рис. 4.5-4.6).

```
[79]: using DelimitedFiles
      x = [1; 2; 3; 4];
      y = ["a"; "s"; "d"; "f"];

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

      readdlm("1.txt")
```

```
[79]: 4×2 Matrix{Any}:
       1  "a"
       2  "s"
       3  "d"
       4  "f"
```

Рис. 4.5: Вывод на печать

```
[89]: for i in 1:1:3
          print(i, "     ")
      end
```

```
1     2     3
```

```
[91]: for i in 1:2:10
          println(i)
      end
```

```
1
3
5
7
9
```

```
[93]: show("Hello!!!")
```

```
"Hello!!!"
```

```
[95]: print("Hello!!!")
```

```
Hello!!!
```

Рис. 4.6: Вывод на печать

Посмотрим на работу функции `write()` (рис. 4.7).

```
[97]: io = IOBuffer()
      write(io, "la-la-la", "laa-laa")

[97]: 15

[99]: String(take!(io))

[99]: "la-la-lalaa-laa"

[101]: String(take!(io))

[101]: ""
```

Рис. 4.7: Команда записи

Задание №2

Изучим документацию по функции parse(). Приведем свои примеры её использования, поясняя особенности её применения (рис. 4.8).

```
parse(Int, "9994567765423")
```

```
[116]:
```

```
9994567765423
```

```
[126]:
```

```
parse(Int, "11100",  base = 2)
```

```
[126]:
```

```
28
```

```
[128]:
```

```
parse(Int, "a",  base = 16)
```

```
[128]:
```

```
10
```

Рис. 4.8: Примеры использования функции parse()

Задание №3

Изучим синтаксис Julia для базовых математических операций с разным типом переменных: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, сравнение, логические операции. Приведем примеры с пояснениями по особенностям их применения (рис. 4.9-4.10).

```
y = 22 / 11
y1 = 24.1234 / 10.124
println(y, '\n', y1)
```

```
2.0
2.3827933623073885
```

[156]:

```
y = 2 ^ 3
y1 = 2.1 ^ 3
println(y, '\n', y1)
```

```
8
9.261000000000001
```

[158]:

```
x = sqrt(25)
```

[158]:

```
5.0
```

Рис. 4.9: Примеры базовых математических операций

Рис. 4.10: Примеры базовых математических операций

Задание №4

Приведем несколько примеров с пояснениями с операциями над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр (рис. 4.11-4.12).

```
[210]: A + B

[210]: 2×2 Matrix{Int64}:
         4  9
        17  5

[212]: A - B

[212]: 2×2 Matrix{Int64}:
        -2  -5
         1  -3

[214]: A*B

[214]: 2×2 Matrix{Int64}:
        19  15
        35  67
```

Рис. 4.11: примеры операций над матрицами

```
[227]: A'

[227]: 2×2 adjoint(::Matrix{Int64}) with eltype Int64:
        1  9
        2  1

[237]: using LinearAlgebra
       q = [1, 2, 3]
       w = [2, 3, 4]
       p = dot(q, w)

[237]: 20

       r = cross(q, w)
```

Рис. 4.12: примеры операций над векторами

5 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.

Список литературы