

# Лабораторная работа №1

Julia. Установка и настройка. Основные принципы.

---

Ким Реачна<sup>1</sup>

10 ноября, 2023, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и задачи

---

## Цель лабораторной 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.
3. Выполните задания для самостоятельной работы.

# Процесс выполнения лабораторной работы

---

- `read()`: функция `read()` используется для чтения данных из стандартного ввода (клавиатуры) или из файла.
- `readline()`: функция читает строку из стандартного ввода или файла.
- `readlines()`: считывание всех строк из файла или потока ввода-вывода.
- `readdlm()`: используется для чтения данных из текстового файла, разделенных определенным разделителем (по умолчанию - запятой) и возвращения их в виде двумерного массива. Это полезно для обработки структурированных данных, таких как таблицы.

# Основные функции Julia

Пример:

```
[27]: message = IOBuffer("hello, world!");  
      read(message, String)  
  
[27]: "hello, world!"  
  
[28]: open("myfile.txt", "w") do io  
      write(io, "Hello this is lab1!\n");  
      end  
  
[28]: 28  
  
[29]: readLine("myfile.txt")  
  
[29]: "Hello this is lab1!"  
  
[30]: readLine("myfile.txt", keep = true)  
  
[30]: "Hello this is lab1!\n"  
  
[31]: write("my_file.txt", "Using function write()!\n Lorem Ipsum is simply dummy  
      text of the printing and typesetting industry.\n Lorem Ipsum has been  
      the industry's standard dummy text ever since the 1500s.")  
  
[31]: 108  
  
[32]: readLines("my_file.txt")  
  
[32]: 3-element Vector{String}:  
      "Using function write()!\n"  
      "Lorem Ipsum is simply dummy text of the printing and typesetting industry.\n"  
      "Lorem Ipsum has been the industry's standard dummy text ever since the 1500s."  
  
[33]: readLines("my_file.txt", keep = true)  
  
[33]: 3-element Vector{String}:  
      "Using function write()!\n"  
      "Lorem Ipsum is simply dummy text of the printing and typesetting industry.\n"  
      "Lorem Ipsum has been the industry's standard dummy text ever since the 1500s."  
  
[34]: using DelimitedFiles  
      x = [1; 2; 3; 4];  
      y = [5; 6; 7; 8];  
      open("delimited_file.txt", "w") do io  
          write(io, {x y})  
      end  
      readLine("delimited_file.txt", Int12)  
  
[34]: 4x2 Matrix{Int32}:  
      1 5  
      2 6  
      3 7  
      4 8
```

Рис. 1: Пример основных функций Julia

- `print()`: функция выводит аргументы на стандартный вывод без добавления символа новой строки в конце.
- `println()`; функция аналогична `print()`, но добавляет символ новой строки в конце.
- `show()`: функция предназначена для вывода объектов на экран. Она вызывается автоматически при использовании функций `println()` и `print()` для пользовательских типов данных.
- `write()`: записывает общепринятое двоичное представление значения в данный поток ввода-вывода или файл. Возвращает количество байтов, записанных в поток.



# Основные функции Julia

```
[34]: print("Hello,")  
      print("World!")  
  
      Hello,World!  
  
[35]: println("Hello,")  
      println("World!")  
  
      Hello,  
      World!  
  
[36]: show("This is lab1")  
  
      "This is lab1"  
  
[37]: data_to_write = "Using function write()"  
      write("myfile.txt", data_to_write)  
  
[37]: 22
```

Рис. 2: Пример основных функций Julia

Функция `parse()` в Julia используется для преобразования строк в значения определенного типа данных.

```
[39]: str_num = "18"  
      parse_int = parse{Int, str_num}  
      println("Parsed integer: ", parse_int)  
  
      Parsed integer: 18  
  
[40]: parse{Complex{Float64}, "1.2e-1 + 3.4im"}  
  
[40]: 0.12 + 3.4im
```

Рис. 3: Функция `parse()`

# Основные математические операции

```
[41]: a = 3
      b = 4
      println("Сложение: ", a + b)
      println("Вычитание: ", a - b)
      println("Умножение: ", a * b)
      println("Деление: ", a / b)
      println("Возведение в степень: ", a ^ b)
      println("Извлечение корня: ", sqrt(a))

      Сложение: 7
      Вычитание: -1
      Умножение: 12
      Деление: 0.75
      Возведение в степень: 81
      Извлечение корня: 1.7320508075688772

[42]: # Сравнение
      x = 5
      y = 6
      println("Is x equal to y? ", x == y)
      println("Is x not equal to y? ", x != y)
      println("Is x smaller than y? ", x < y)
      println("Is x greater than y? ", x > y)

      Is x equal to y? false
      Is x not equal to y? true
      Is x smaller than y? true
      Is x greater than y? false

[43]: # Логические операции
      false && false

[43]: false

[44]: false && true

[44]: false

[45]: true || false

[45]: true

[46]: println("Logical AND: ", x < 10 && y < 10)
      println("Logical OR: ", x < 10 || x > 10)

      Logical AND: true
      Logical OR: true
```

Рис. 4: Основные математические операции

# Основные математические операции

```
[47]: A = [1 2 3; 4 5 6; 7 8 9]

[47]: 3x3 Matrix{Int64}:
      1 2 3
      4 5 6
      7 8 9

[48]: B = [1 2 2; 2 3 3; 5 4 4]

[48]: 3x3 Matrix{Int64}:
      1 2 2
      2 3 3
      5 4 4

[49]: # сложение
      A + B

[49]: 3x3 Matrix{Int64}:
      2 4 5
      6 8 9
     12 12 13

[50]: # вычитание
      A - B

[50]: 3x3 Matrix{Int64}:
      0 0 1
      2 2 3
      2 4 5

[51]: # скалярное произведение векторов
      v1 = [1, 2, 3]
      v2 = [4, 5, 6]
      print("Скалярное произведение векторов: ", sum(v1 .* v2))

Скалярное произведение векторов: 32

[52]: # транспонирование матрицы A
      A'

[52]: 3x3 adjoint{::Matrix{Int64}} with eltype Int64:
      1 4 7
      2 5 8
      3 6 9

[53]: # умножение на скаляр
      s = 3
      print("Матрица A, умножение на скаляр: ", A * s)

Матрица A, умножение на скаляр: [ 3 6 9; 12 15 18; 21 24 27]
```

Рис. 5: Основные математические операции

## Выводы по проделанной работе

---

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