РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина: Компьютерный практикум по статистическому анализу данных

Студент: Евдокимов Иван Андреевич

Группа: НФИбд-01-20

МОСКВА

2023 г.

Цель работы:

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

In [1]:
```
;date
```

```
Fri Nov 10 16:54:21 RTZ 2023
```

In [2]:
```
;whoami
```

```
win-orl1fa978ms\admin
```

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, поясня особенности их применения.

Подготовка:

In [3]:
```
using DelimitedFiles
x = [1; 2; 3; 4];
y = [5; 6; 7; 8];
z = [5; 6; 7; 8];
open("delim_file.txt", "w") do io
           writedlm(io, [x y z])
       end
```

Вывод с помощью read():

In [4]:
```
read("delim_file.txt", String)
```

Out[4]:
```
"1\t5\t5\n2\t6\t6\n3\t7\t7\n4\t8\t8\n"
```

Вывод с помощью readline():

In [5]:
```
readline("delim_file.txt")
```

Out[5]:
```
"1\t5\t5"
```

Вывод с помощью readlines():

In [6]:
```
readlines("delim_file.txt")
```

Out[6]:
```
4-element Vector{String}:
 "1\t5\t5"
 "2\t6\t6"
 "3\t7\t7"
 "4\t8\t8"
```

Вывод с помощью readdlm():

In [7]:
```
readdlm("delim_file.txt", '\t', Int, '\n')
```

Out[7]:
```
4×3 Matrix{Int64}:
 1  5  5
 2  6  6
 3  7  7
 4  8  8
```

Запись с помощью write():

In [8]:
```
# Открываю файл для записи
a = [1; 2; 3; 4]
file = open("output.txt", "w")
# Строка для вывода
output_string = "Привет, мир!"
# Используем функцию write() для записи строки в файл
write(file, output_string)
# Закрываем файл
close(file)
```

Вывод с помощью print():

In [9]:
```
print(a, " ", output_string)
print(a, " ", output_string)
```

```
[1, 2, 3, 4] Привет, мир![1, 2, 3, 4] Привет, мир!
```

Вывод с помощью println():

In [10]:
```
println(a, " ", output_string)
println(a, " ", output_string)
```

```
[1, 2, 3, 4] Привет, мир!
[1, 2, 3, 4] Привет, мир!
```

Вывод с помощью show():

In [11]:
```
show(a)
show(output_string)
```

```
[1, 2, 3, 4]"Привет, мир!"
```

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

In [12]:
```
parse(Int, "1234")
```

Out[12]: 1234

In [13]:
```
parse(Int, "1234", base = 5)
```

Out[13]: 194

In [14]:
```
parse(Int, "afc", base = 16)
```

Out[14]: 2812

In [15]:
```
parse(Float64, "1.2e-3")
```

Out[15]: 0.0012

In [16]:
```
parse(Complex{Float64}, "3.2e-1 + 4.5im")
```

Out[16]: 0.32 + 4.5im

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 применени.

In [17]:
```
a,b,c,d = 5,3,true,false
```

Out[17]: (5, 3, true, false)

Операция сложение:

In [18]:
```
a + b
```

Out[18]: 8

Операция вычитание:

In [19]:
```
a - b
```

Out[19]: 2

Операция умножение:

In [20]:
```
a * b
```

Out[20]: 15

Операция деление:

In [21]:
```
a / b
```

Out[21]: 1.6666666666666667

Операция возведение в степень:

In [22]:
```
a ^ b
```

Out[22]: 125

Операция извлечение корня:

In [23]:
```
sqrt(a)
```

Out[23]: 2.23606797749979

Операция сравнение:

In [24]:
```
a == b
```

Out[24]:
```
false
```

Логические операции:

In [25]:
```
~a # Побитовое НЕ
```

Out[25]:
```
-6
```

In [26]:
```
a & b # Побитовое И
```

Out[26]:
```
1
```

In [27]:
```
a | b # Побитовое ИЛИ
```

Out[27]:
```
7
```

In [28]:
```
a ? b # Побитовое исключающее ИЛИ
```

```
syntax: colon expected in "?" expression

Stacktrace:
 [1] top-level scope
   @ In[28]:1
```

In [29]:
```
a >>> b # Логический сдвиг вправо
```

Out[29]:
```
0
```

In [30]:
```
a >> b # Побитовый/логический сдвиг вправо
```

Out[30]:
```
0
```

In [31]:
```
a << b # Побитовый/логический сдвиг влево
```

Out[31]:
```
40
```

In [32]:
```
c && d # Логическое И
```

Out[32]:
```
false
```

In [33]:
```
c || d # Логическое ИЛИ
```

Out[33]:
```
true
```

In [34]:
```
!c # Логическое НЕ
```

Out[34]:
```
false
```

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

Подготовка:

In [35]:
```
A = [1 2; 2 3]
B = [3 2; 2 3]
C = [21; 45]
```

Out[35]:
```
2-element Vector{Int64}:
 21
 45
```

Сумма матриц:

In [36]:
```
A + B
```

Out[36]:
```
2×2 Matrix{Int64}:
 4  4
 4  6
```

Разность матриц:

In [37]:
```
A - B
```

Out[37]:
```
2×2 Matrix{Int64}:
 -2  0
  0  0
```

Транспонирую матрицу A:

In [38]:
```
transpose(A)
```

Out[38]:
```
2×2 transpose(::Matrix{Int64}) with eltype Int64:
 1  2
 2  3
```

Умножение матрицы на число:

In [39]:
```
A * 2
```

Out[39]:
```
2×2 Matrix{Int64}:
 2  4
 4  6
```

Умножение матрицы на матрицу:

In [40]:
```
A * B
```

Out[40]:
```
2×2 Matrix{Int64}:
  7   8
 12  13
```

Выводы:

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

Список литературы:

1. Julia 1.5 Documentation. — 2020. — URL: https://docs.julialang.org/en/v1/.
2. Klok H.,Nazarathy Y. Statistics with Julia: Fundamentals for Data Science,Machine Learning and Artificial Intelligence. — 2020. — URL: https://statisticswithjulia.org/.
3. Ökten G. First Semester in Numerical Analysis with Julia. — Florida State University, 2019. — DOI: 10.33009/jul.
4. Антонюк В. А. Язык Julia как инструмент исследователя. — М. : Физический факультет МГУ им. М. В. Ломоносова, 2019.
5. Шиндин А. В. Язык программирования математических вычислений Julia. Базовое руководство. — Нижний Новгород : Нижегородский госуниверситет, 2016.