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 математическому моделированию

Студент: Яссин Мохамад Аламин

Группа: НКНбд-01-20

МОСКВА

2023 г.

Постановка задачи

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 (разделы 1.3.1 и 1.3.2).
2. Используя Jupyter Lab, повторите примеры из раздела 1.3.3.
3. Выполните задания для самостоятельной работы (раздел 1.3.4).

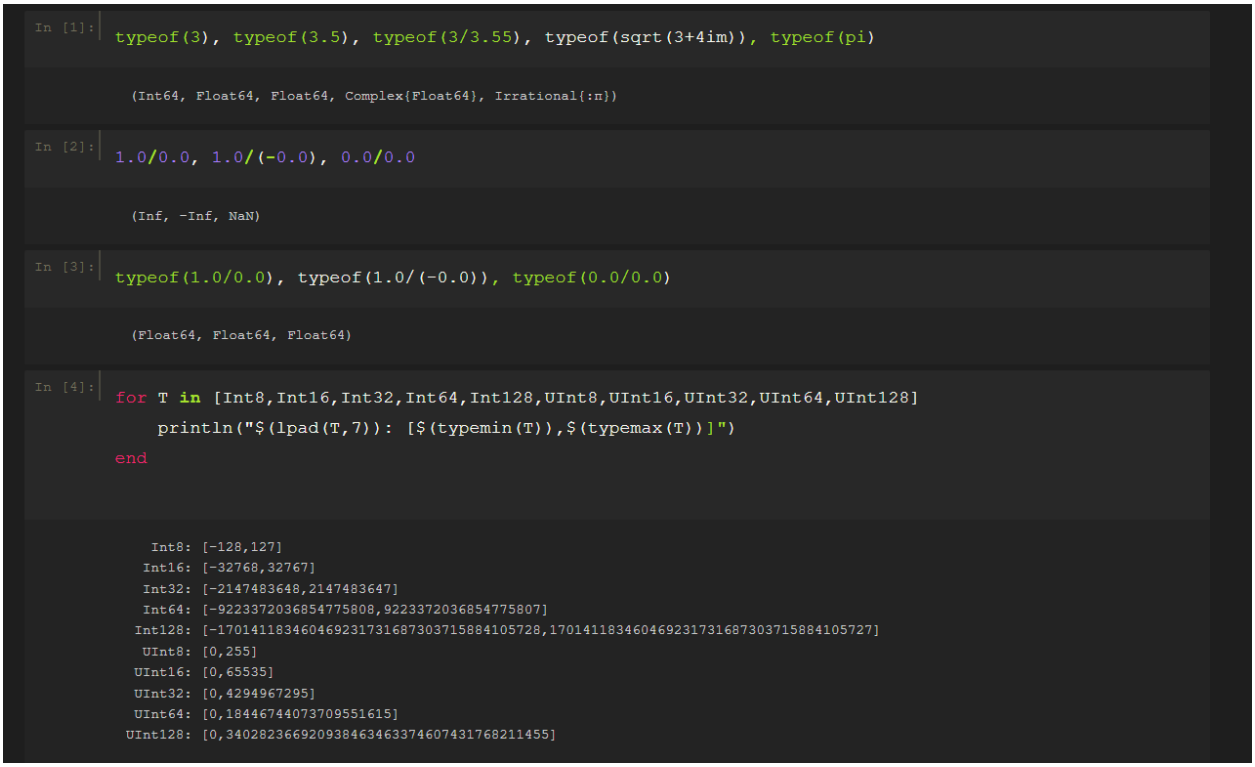
Выполнение работы

1. Установите под свою операционную систему Julia, Jupyter (разделы 1.3.1 и 1.3.2).

Посмотрели ознакомительное видео и выполнили установку требуемых приложений

2. Используя Jupyter Lab, повторите примеры из раздела 1.3.3.

Получим минимальные и максимальные значения целочисленных типов



Примеры приведения типов

```
In [5]: Int64(2.0), Char(2), typeof(Char(2))

        (2, '\x02', Char)

In [6]: convert(Int64, 2.0), convert(Char,2)

        (2, '\x02')

In [7]: typeof(promote(Int8(1), Float16(4.5), Float32(4.1)))

        Tuple{Float32,Float32,Float32}
```

Примеры работы с функциями

```
In [8]: function f(x)
            x^2
        end

        f (generic function with 1 method)

In [9]: f(4)

        16

In [10]: g(x)=x^2

         g (generic function with 1 method)

In [11]: g(8)

         64
```

Примеры работы с матрицами

```
In [12]: a = [4 7 6]
         b = [1, 2, 3]
         a[2], b[2]

         (7, 2)

In [13]: a=1; b=2; c=3; d=4
         Am = [a b; c d]

         2×2 Array{Int64,2}:
          1  2
          3  4

In [14]: Am[1,1], Am[1,2], Am[2,1], Am[2,2]

         (1, 2, 3, 4)

In [15]: aa = [1 2]
         AA = [1 2; 3 4]
         aa*AA*aa'
```

3. Выполните задания для самостоятельной работы (раздел 1.3.4).

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, поясняя особенности их применения

- read() – считывает значение указанного типа в бинарном представлении read(stream, type)

```
In [4]: buffer=IOBuffer("33 20 21")
        X=read(buffer)
        X

8-element Vector{UInt8}:
 0x33
 0x33
 0x20
 0x32
 0x30
 0x20
 0x32
 0x31
```

- readline() – считывает строку до символа завершения строки, включая этот символ

```
In [19]: readline("file.txt", keep=true)

          "100 51 57 26 82 100 50 72 44 \n"

In [20]: str=readline()
         str

          stdin> 595455652


          "595455652"
```

- readlines() – считывает все строки, поданные на вход в виде массива

```
In [21]: STR=readlines("file.txt", keep=true)

          5-element Array{String,1}:
           "100 51 57 26 82 100 50 72 44 \n"
           "92 100 51 85 62 13 69\n"
           " 44 75 67 26 66 19 16 20 36 43 14 \n"
           " 75 1 7 5 36 11 40 60 42 80 92 9 81 \n"
           " 23 15 36 68 72 72 46 89 11 44"
```

- print() – распечатывает на экран содержимое переменной
- println() – распечатывает на экран с новой строки содержимое переменной

```
In [6]: print(X)

         UInt8[0x33, 0x33, 0x20, 0x32, 0x30, 0x20, 0x32, 0x31]

In [8]: println(X)

         UInt8[0x33, 0x33, 0x20, 0x32, 0x30, 0x20, 0x32, 0x31]
```

- show() – распечатывает на экран содержимое переменной, для сложных выражений выполняет форматирование, можно указать поток вывода

```
In [9]: show(X)

        UInt8[0x33, 0x33, 0x20, 0x32, 0x30, 0x20, 0x32, 0x31]
```

- write() – выполняет запись содержимого переменной в указанный поток, возвращает число записанных байт

```
In [54]: io=open("output.txt", "w")
         write(io, X)

         20
```

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

Функция parse() преобразовывает строку в численный тип, можно указать основание системы счисления и поток ввода.

```
In [10]: a=parse(Int64,"4534")

         4534

In [11]: typeof(a)

         Int64

In [12]: a=parse(Int, "4747", base=16)

         18247

In [29]: a=parse(Float32, "32.15")

         32.15f0
```

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 применения.

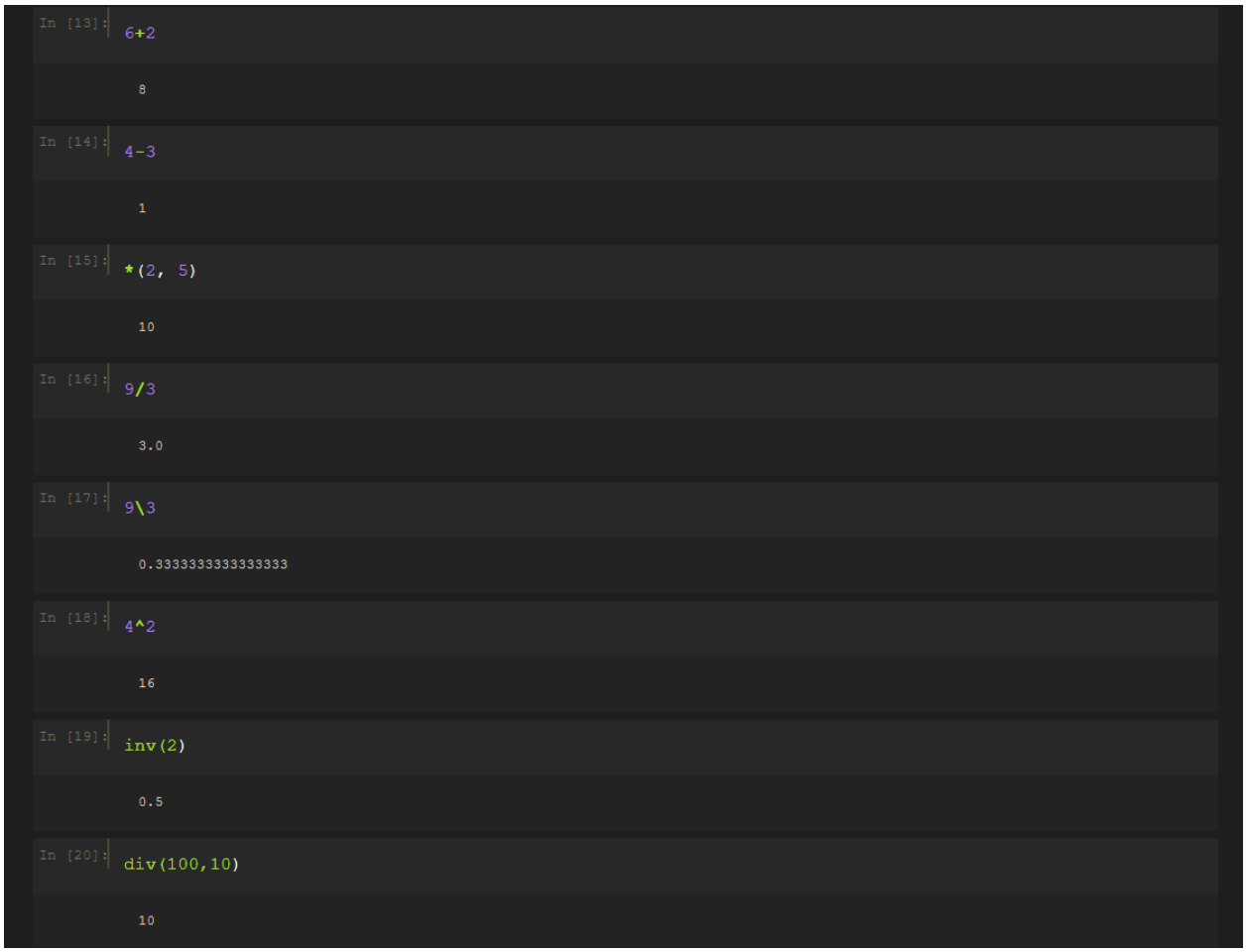
Синтаксис предполагает использование стандартный символов в качестве обозначения операций

+	Сложение
-	Вычитание, унарный минус
*	умножение
/	правостороннее деление
\	левостороннее деление
^	степень, работает также для матриц
fma()	вычисление без округления
inv()	обратное число
div()	целочисленное деление
mod()	остаток от деления
==	сравнение на равенство
!=	сравнение на неравество
<	сравнение на меньшее
>	сравнение на большее
<=	меньшее либо равное
>=	большее либо равное
~	побитовое НЕ
&	побитовое И
|	побитовое ИЛИ
!	логическое НЕ
xor()	побитовое ИСКЛ ИЛИ
&&	логическое И
||	логическое ИЛИ
sin(), cos(), tan()	тригонометрические в радианах
sind(), cosd(), tand()	тригонометрические в градусах
sinh(), cosh(), tanh()	гиперболические тригонометрические функции

Для всех тригонометрических функций реализованы обратные функции

log(x)	натуральный логарифм
log(b, x)	логарифм по основанию b
exp()	экспонента e^x

exp2()	2^x
exp10()	10^x
round()	математическое округление
abs()	модуль
abs2()	квадрат модуля
sqrt()	квадратный корень
cbrt()	кубический корень
factorial()	факториал



```
In [16]: 9/3
```

```
3.0
```

```
In [17]: 9\3
```

```
0.3333333333333333
```

```
In [18]: 4^2
```

```
16
```

```
In [19]: inv(2)
```

```
0.5
```

```
In [20]: div(100,10)
```

```
10
```

```
In [21]: mod(66,6)
```

```
0
```

```
In [22]: 111.0==11
```

```
false
```

```
In [23]: 2!=9
```

```
true
```

```
In [24]: 9>3
```

```
In [25]: 97<21
```

```
false
```

```
In [26]: ~(16)
```

```
-17
```

```
In [27]: (9<11) || (15<10)
```

```
true
```

```
In [28]: !(true)
```

```
false
```

```
In [29]: tan(30)
```

```
-6.405331196646276
```

```
In [30]: exp(2)
```

```
7.38905609893065
```

```
In [31]: exp2(9)
```

```
512.0
```

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

```
M1=[ 60 26 62 54; 14 37 52 16; 22 10 42 91; 70 85 64 84 ]
M2=[ 82 32 52 96; 21 17 85 76; 14 57 96 79; 69 14 84 38 ]
println(M1+M2)
println(M1-M2)
println(M1*M2)
println(M1')
```

```
[142 58 114 150; 35 54 137 92; 36 67 138 170; 139 99 148 122]
[-22 -6 10 -42; -7 20 -33 -60; 8 -47 -54 12; 1 71 -20 46]
[10060 6652 15818 14686; 3757 4265 10209 8872; 8881 4542 13670 9648; 14217 8509 24065 21428]
[60 14 22 70; 26 37 10 85; 62 52 42 64; 54 16 91 84]
```