

# 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: Ли Тимофей Александрович

Группа: НФИбд-01-18

МОСКВА

2021 г.

## Постановка задачи

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

## Выполнение работы

Сначала выполнил все примеры к лабораторной работе №1:

```
[1]: 2+3
[1]: 5

[2]: 3+4
     1+2
[2]: 3

[3]: 3+5
     4+5;
[4]: ?println
      search: println printstyled print sprint isprint

[4]: println([io::IO], xs...)
      Print (using print) xs followed by a newline. If io is not supplied, prints to stdout.
```

## Examples

```
julia> println("Hello, world")
Hello, world
```

```
[7]: for T in [Int8,Int16,Int32,Int64,Int128,UInt8, UInt16,UInt32,UInt64,UInt128]
      println("$(@pad(T,7)): [$(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]

[8]: typeof(3),typeof(3.5), typeof(3/3.55), typeof(sqrt(3+4im)),typeof(pi)

[8]: (Int64, Float64, Float64, ComplexF64, Irrational{::π})

[9]: 1.0/0.0, 1.0/(-0.0), 0.0/0.0

[9]: (Inf, -Inf, NaN)

[10]: Int64(2.0), Char(2), typeof(Char(2))

[10]: (2, '\x02', Char)
```

```
[11]: convert(Int64, 2.0), convert(Char, 2)
```

```
[11]: (2, '\x02')
```

```
[12]: typeof(promote{Int8(1),Float16(4.5),Float32(4.1)})
```

```
[12]: Tuple{Float32, Float32, Float32}
```

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

```
[13]: f (generic function with 1 method)
```

```
[14]: f(4)
```

```
[14]: 16
```

```
[15]: g(x)=x^2
```

```
[15]: g (generic function with 1 method)
```

```
[16]: g(8)
```

```
[16]: 64
```

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

```
[17]: (7, 2)
```

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

```
[18]: 2×2 Matrix{Int64}:
      1  2
      3  4
```

```
[19]: Am[1,1], Am[1,2],Am[2,1],Am[2,2]
```

```
[19]: (1, 2, 3, 4)
```

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

```
[20]: 1×1 Matrix{Int64}:
      27
```

```
[21]: aa,AA,aa'
```

```
[21]: ([1 2], [1 2; 3 4], [1; 2])
```

1. Изучил документацию к функциям read(), readline(), readlines(), readlm(), print(), println(), show(), write() и привел примеры их использования.

```

[23]: io = IOBuffer("i know how to use the command 'read'")

[23]: IOBuffer(data=UInt8[...], readable=true, writable=false, seekable=true, append=false, size=36, maxsize=Inf, ptr=1,
mark=-1)

[24]: read(io, Char)

[24]: 'i': ASCII/Unicode U+0069 (category Ll: Letter, lowercase)

[25]: read(io, String)

[25]: " know how to use the command 'read'"

[27]: io = IOBuffer("first line \nsecond line")

[27]: IOBuffer(data=UInt8[...], readable=true, writable=false, seekable=true, append=false, size=23, maxsize=Inf, ptr=1,
mark=-1)

[28]: readline(io)

[28]: "first line "

[29]: readline(io)

[29]: "Second line"

[31]: io = IOBuffer("first line \nsecond line")

[31]: IOBuffer(data=UInt8[...], readable=true, writable=false, seekable=true, append=false, size=23, maxsize=Inf, ptr=1,
mark=-1)

[32]: readlines(io)

[32]: 2-element Vector{String}:
      "first line "
      "second line"

[5]: a=["a"; "c"; "e"; "g"]

[5]: 4-element Vector{String}:
      "a"
      "c"
      "e"
      "g"

[6]: b=[1; 2; 3; 4]

[6]: 4-element Vector{Int64}:
      1
      2
      3
      4

[7]: open("testfile.txt", "w") do io
      writedlm(io, [a b])
    end

[10]: readaddlm("testfile.txt", Char)

[10]: 4x2 Matrix{Char}:
      'a' '1'
      'c' '2'
      'e' '3'
      'g' '4'

```

```

[12]: print("i printed this phrase")
      i printed this phrase
[60]: f=open("testfile.txt","w")
[60]: IOStream(<file testfile.txt>)
[61]: println(f, "first line")
[62]: print(f, "second line")
[63]: close(f)
[64]: readlines("testfile.txt")
[64]: 2-element Vector{String}:
      "first line"
      "second line"

[66]: show(1)
      1
[67]: show("line")
      "line"
[74]: io=IOBuffer()
[74]: IOBuffer(data=UInt8[...], readable=true, writable=true, seekable=true, append=false, size=0, maxsize=Inf, ptr=1, mark=-1)
[75]: write(io, "first line\n")
[75]: 11
[76]: write(io, "second line")
[76]: 11
[78]: String(take!(io))
[78]: "first line\nsecond line"

```

2. Изучил документацию к функции `parse()` и привел пример ее использования:

```

[81]: parse(Float64, "69.420")

```

```

[81]: 69.42

```

3. Изучил синтаксис базовых математических операций и проверил их работу с разными типами данных:

```

[83]: x=1
[83]: 1
[84]: y=2
[84]: 2
[91]: k=1.10
[91]: 1.1
[92]: l=2.20
[92]: 2.2
[93]: x+y, k+1, x+k
[93]: (3, 3.3000000000000003, 2.1)

```

```
[94]: x-y, k-l, x-k
[94]: (-1, -1.1, -0.10000000000000009)
[95]: x*y, k*l, x*k
[95]: (2, 2.4200000000000004, 1.1)
[97]: x/y, k/l, x/k, l/y
[97]: (0.5, 0.5, 0.9090909090909091, 1.1)
[98]: x^y, k^l, x^k, l^y
[98]: (1, 1.2332863005546628, 1.0, 4.840000000000001)
[99]: sqrt(x), sqrt(k)
[99]: (1.0, 1.0488088481701516)
[100]: x==y, k==l, x==k
[100]: (false, false, false)
[101]: x>0 && k%l==0
[101]: false
```

#### 4. Привел примеры базовых операций над матрицами:

```
[102]: n=[1 2;3 4]
[102]: 2x2 Matrix{Int64}:
 1  2
 3  4
[103]: m=[5 6; 7 8]
[103]: 2x2 Matrix{Int64}:
 5  6
 7  8
[104]: n+m
[104]: 2x2 Matrix{Int64}:
 6  8
10 12
[105]: m-n
[105]: 2x2 Matrix{Int64}:
 4  4
 4  4
[108]: using LinearAlgebra
[109]: dot(n,m)
[109]: 70
[111]: transpose(n)
[111]: 2x2 transpose(::Matrix{Int64}) with eltype Int64:
 1  3
 2  4
[112]: n*2
[112]: 2x2 Matrix{Int64}:
 2  4
 6  8
```

## **Выводы**

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