Лабораторная работа №1.

Julia. Установка и настройка. Основные принципы.

Ишанова А.И. группа НФИ-02-19

Содержание

# Цель работы

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

# Задание

1. Установите под свою операционную систему Julia, Jupyter.
2. Используя JupyterLab, повторите примеры из раздела 1.3.3.
3. Выполните задания для самостоятельной работы.[1]

# Теоретическое введение

Julia — высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков (например, MATLAB и Octave), однако имеет некоторые существенные отличия. Julia написан на Си, C++ и Scheme. Имеет встроенную поддержку многопоточности и распределённых вычислений, реализованные в том числе в стандартных конструкциях. [2]

Язык является динамическим, при этом поддерживает JIT-компиляцию (JIT-компилятор на основе LLVM входит в стандартный комплект), благодаря чему, по утверждению авторов языка, приложения, полностью написанные на языке (без использование низкоуровневых библиотек и векторных операций) практически не уступают в производительности приложениям, написанным на статически компилируемых языках, таких как Си или C++. Большая часть стандартной библиотеки языка написана на нём же. [2]

# Выполнение лабораторной работы

## Установка Julia и Jupyter

В ходе выполнения других курсов, мною уже была установлена Julia с IJulia для работы с Jupyter Notebook, поэтому этот пункт задания был мною опущен.

## Повторение примеров

1. Повторяем примеры с определением типа числовой величины. (fig. 1)

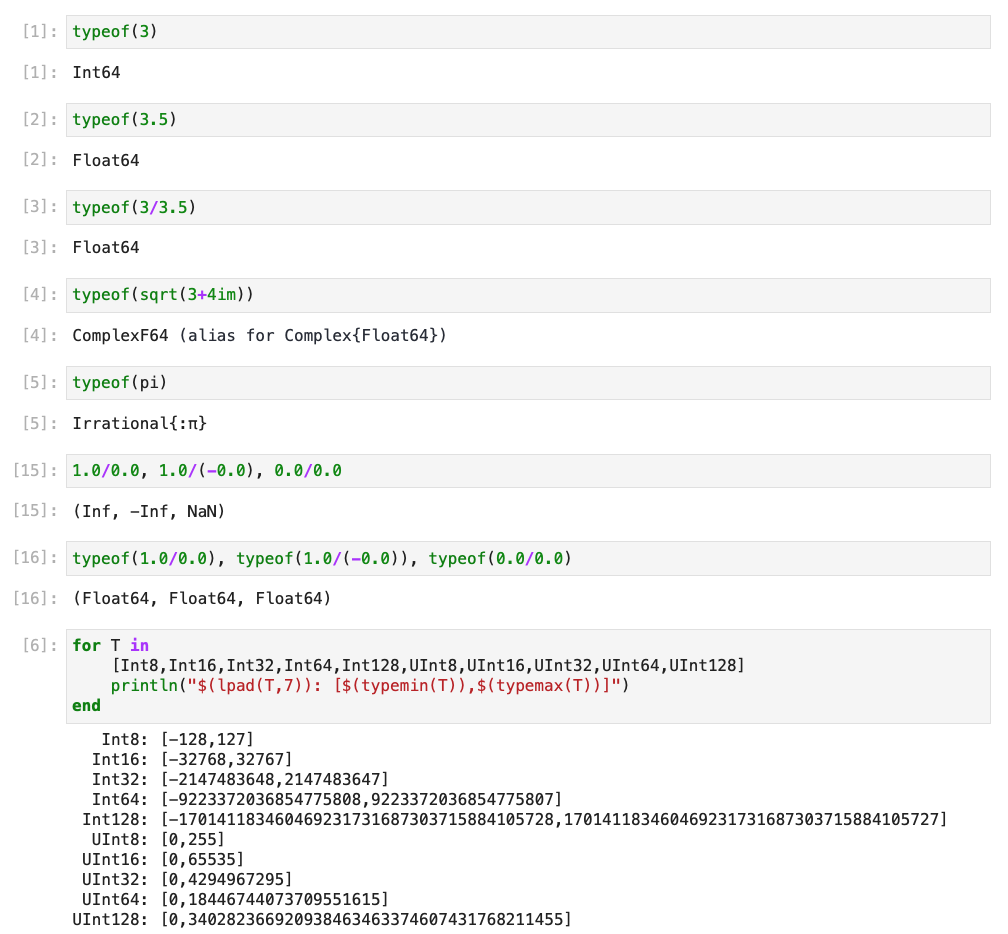


Figure 1: Примеры определения типа числовых величин

1. Потворяем примеры приведения аргументов к одному типу. (fig. 2)

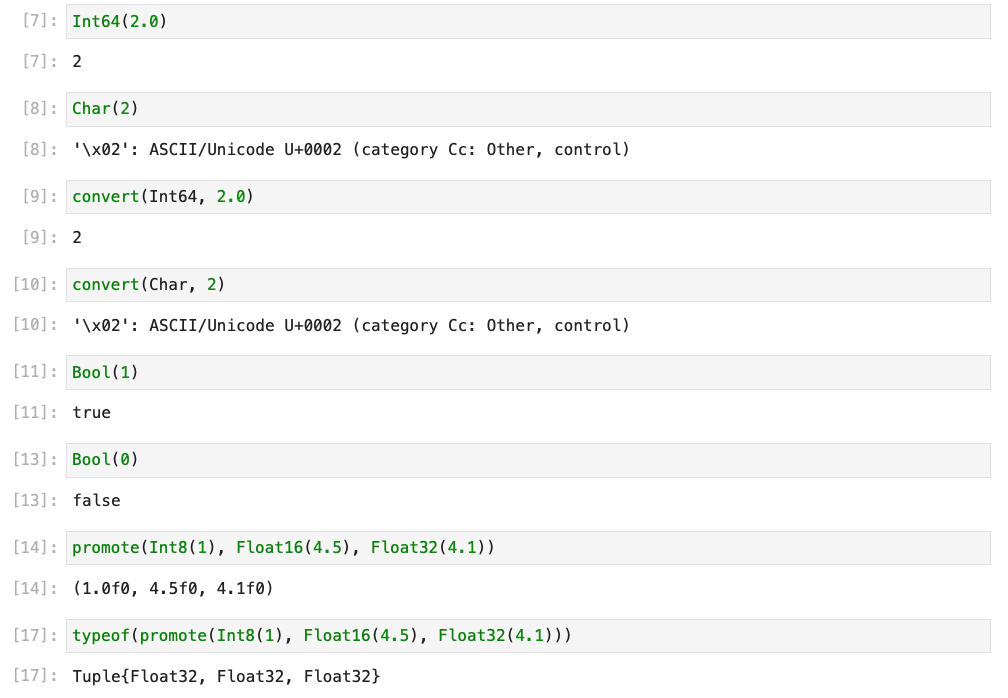


Figure 2: Примеры приведения аргументов к одному типу

1. Повторяем примеры определения функций. (fig. 3)

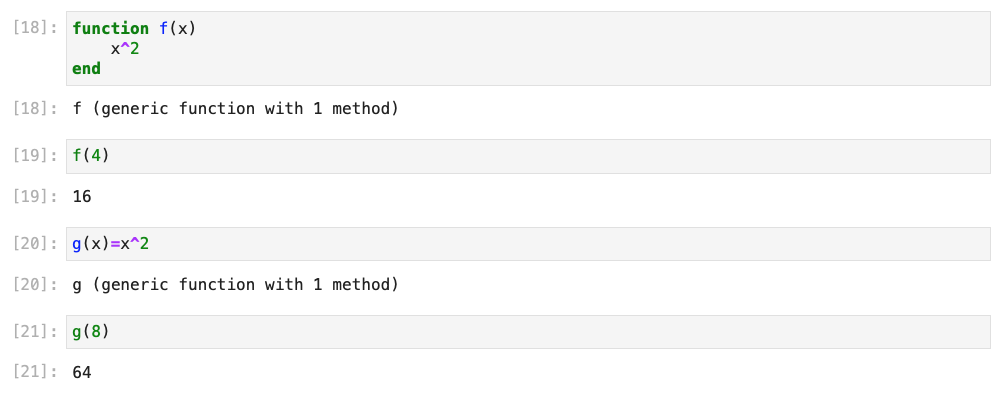


Figure 3: Примеры определения функций

1. Повторяем примеры работы с массивами. (fig. 4)

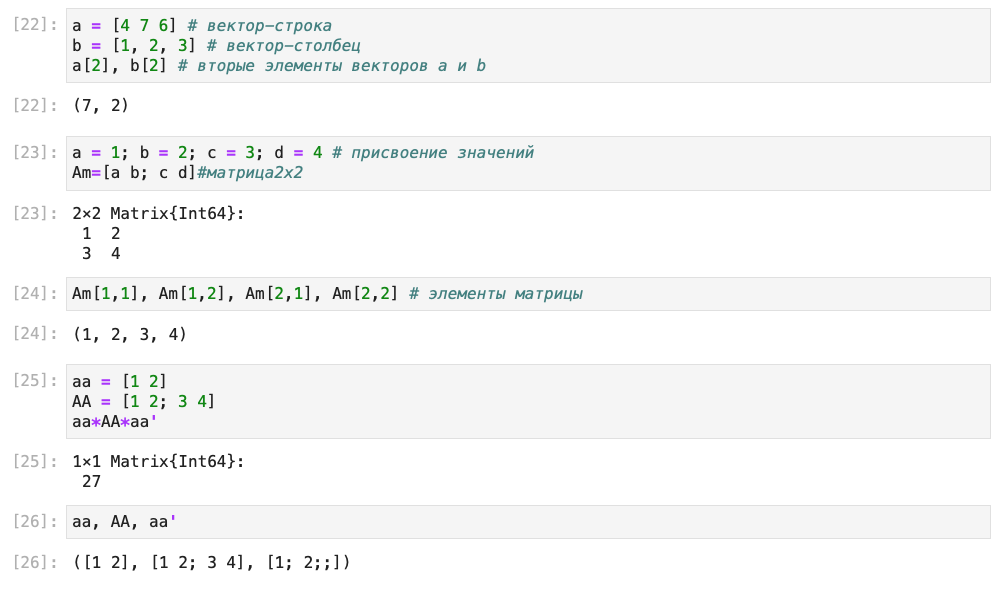


Figure 4: Примеры работы с массивами

## Задания для самостоятельной работы

1. Изучили документацию по основным функциям Julia для чтения / записи / вывода информации на экран: read(), readline(), readlines(), readdlm(), print(), println(), show(), write(). Привели свои примеры их использования. (fig. 5 - fig. 16)

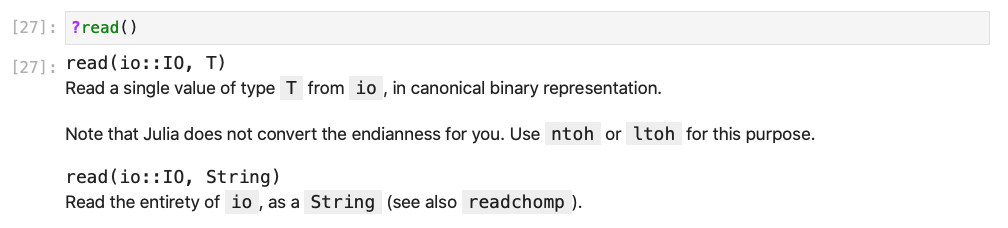


Figure 5: Фрагмент документации по команде read()

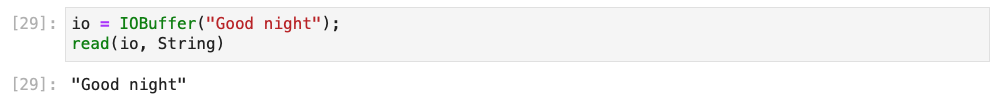


Figure 6: Пример по команде read()

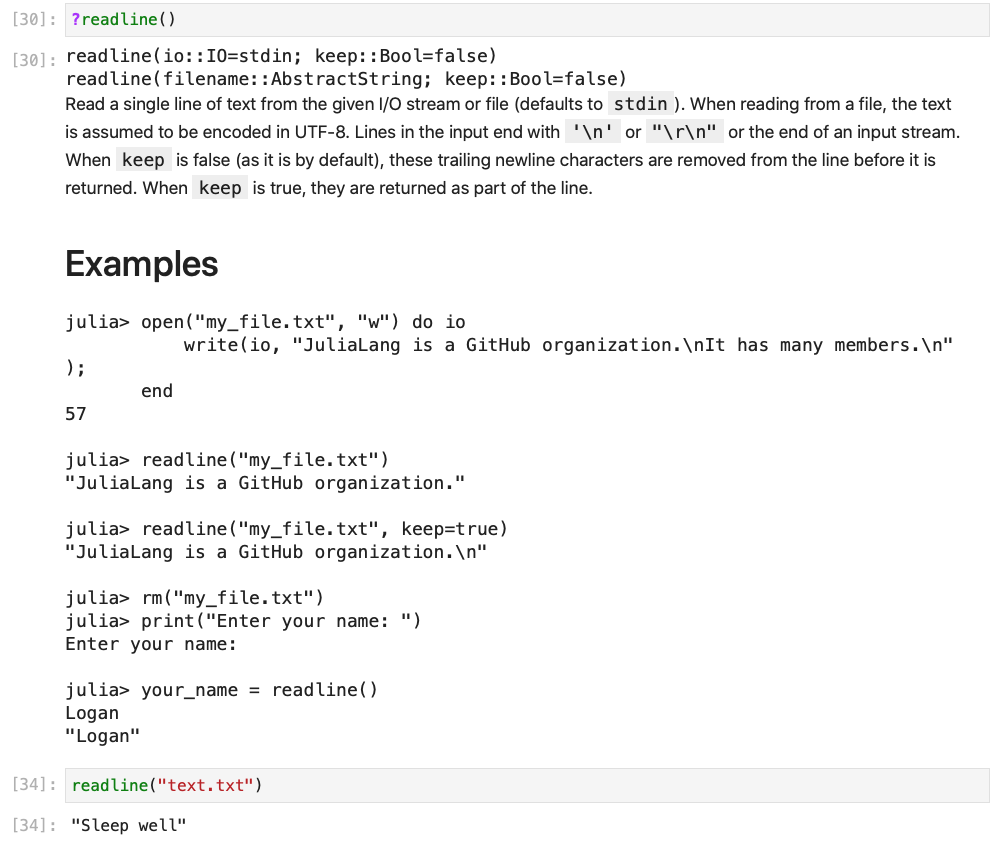


Figure 7: Документация по команде readline() и пример использования



Figure 8: Документация по команде readlines() и пример использования

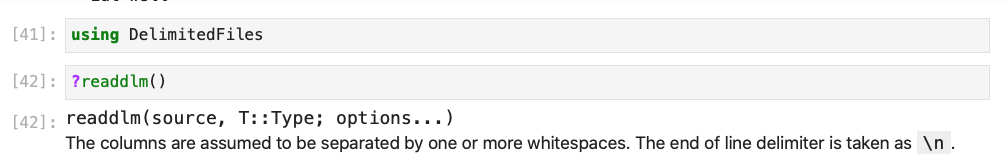


Figure 9: Фрагмент документации по команде readlm()

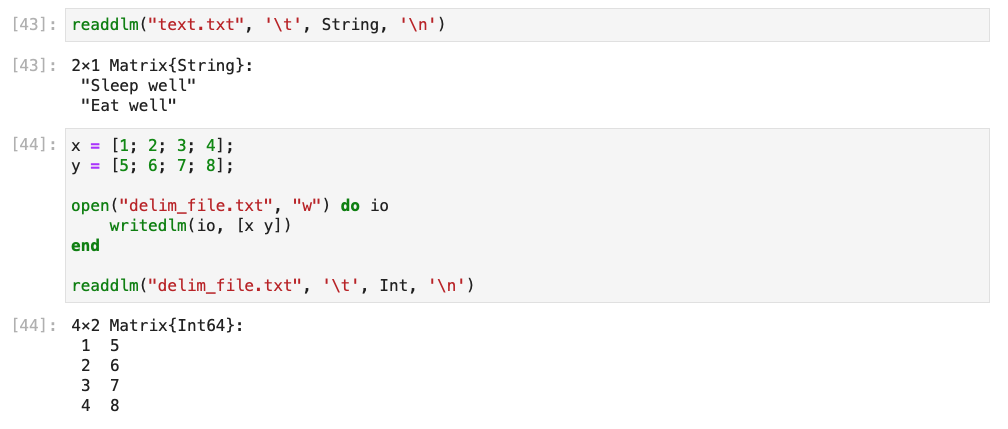


Figure 10: Примеры по команде readlm()

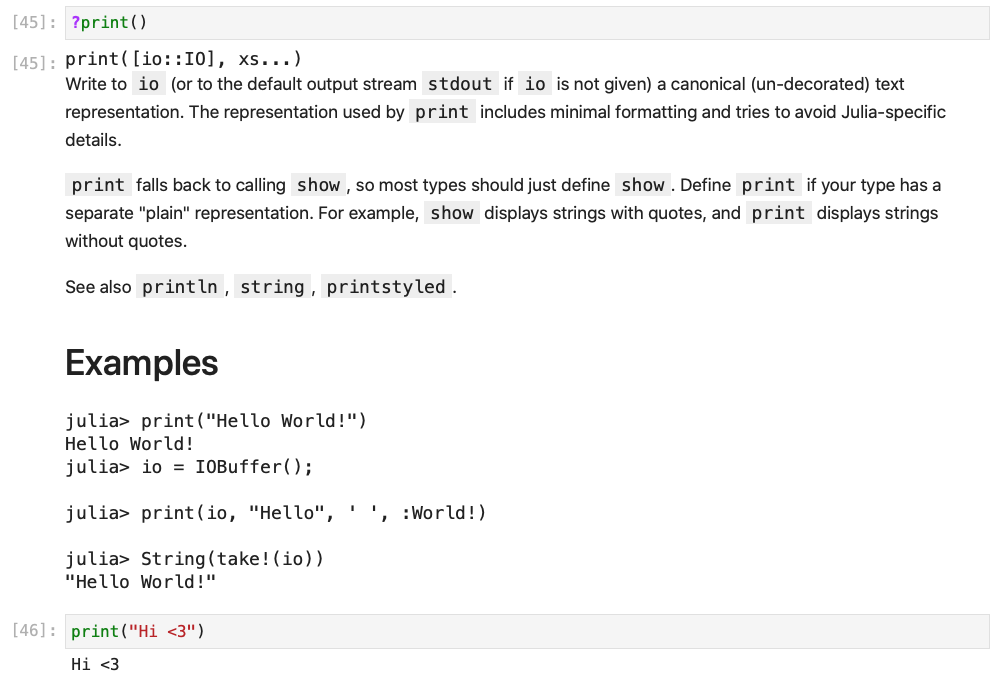


Figure 11: Документация по команде print() и пример использования



Figure 12: Документация по команде println() и пример использования

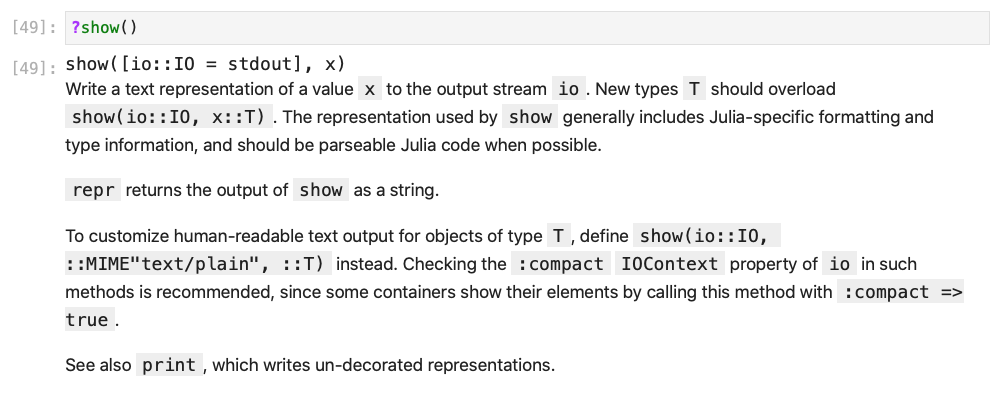


Figure 13: Фрагмент документации по команде show()

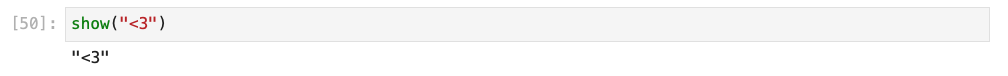


Figure 14: Пример по команде show()

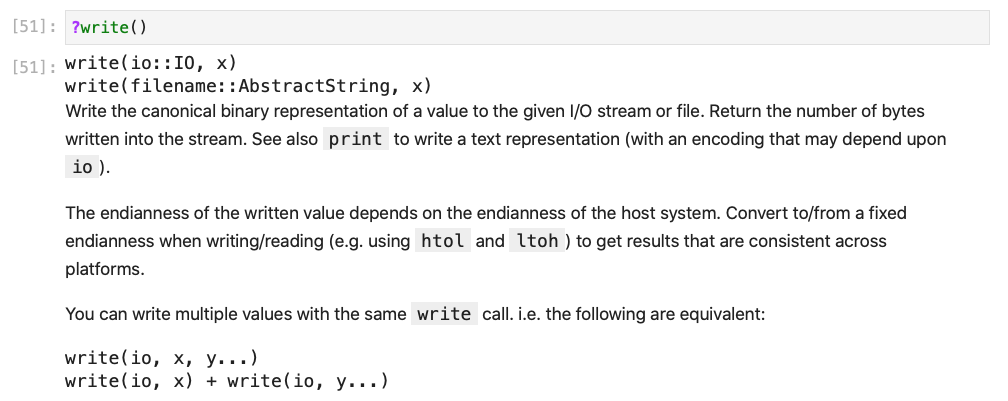


Figure 15: Фрагмент документации по команде write()

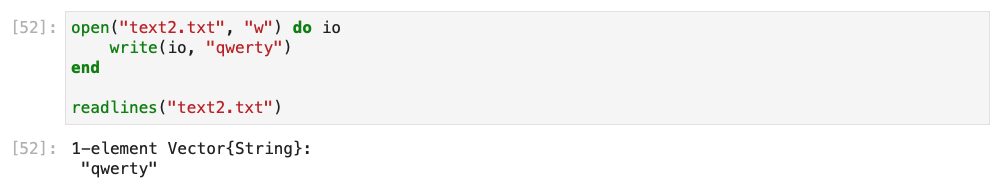


Figure 16: Пример по команде write()

1. Изучили документацию по функции parse(). Привели свой пример её использования. (fig. 17)



Figure 17: Документация по команде parse() и пример использования

1. Изучили синтаксис Julia для базовых математических операций с разным типом переменных: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, сравнение, логические операции. Привели свои примеры. (fig. 18 - fig. 22)

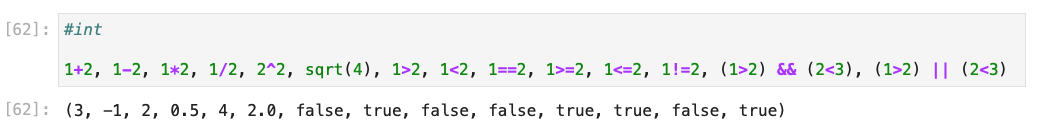


Figure 18: Примеры операций с типом int

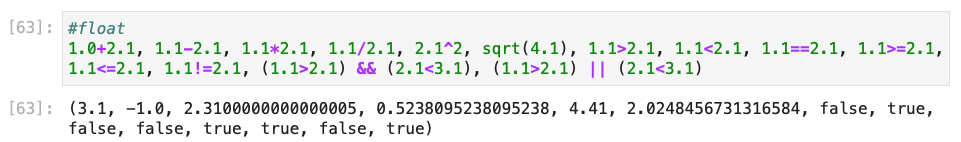


Figure 19: Примеры операций с типом float

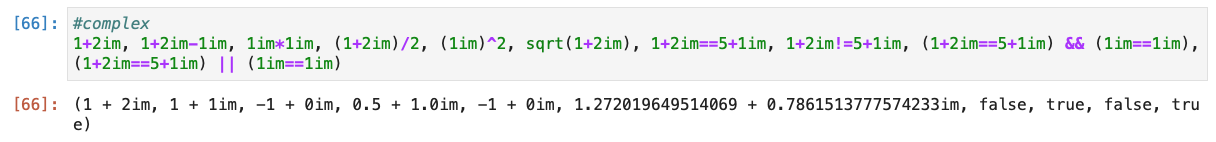


Figure 20: Примеры операций с типом complex

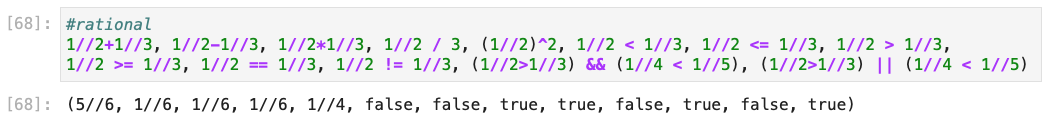


Figure 21: Примеры операций с типом rational

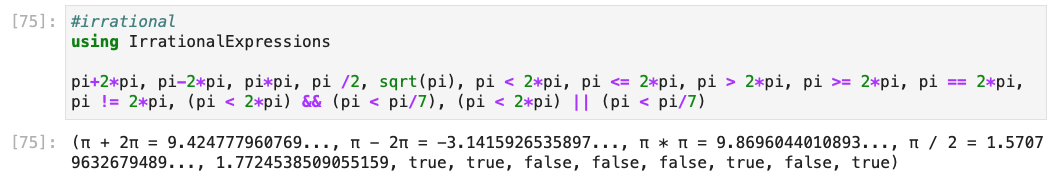


Figure 22: Примеры операций с типом irrational

1. Привели несколько своих примеров с операциями над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр. (fig. 23, fig. 24)

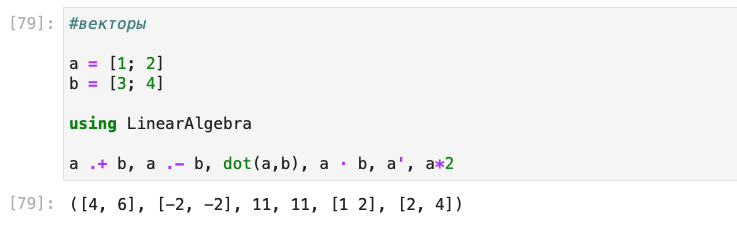


Figure 23: Примеры операций с векторами

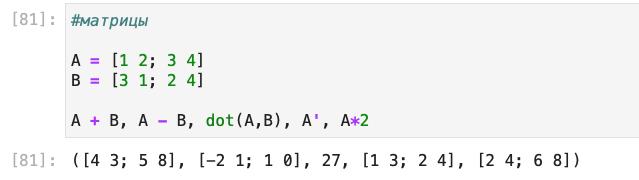


Figure 24: Примеры операций с матрицами

# Листинг

# -\*- coding: utf-8 -\*-  
# ---  
# jupyter:  
# jupytext:  
# text\_representation:  
# extension: .jl  
# format\_name: light  
# format\_version: '1.5'  
# jupytext\_version: 1.14.1  
# kernelspec:  
# display\_name: Julia 1.8.2  
# language: julia  
# name: julia-1.8  
# ---  
  
typeof(3)  
  
typeof(3.5)  
  
typeof(3/3.5)  
  
typeof(sqrt(3+4im))  
  
typeof(pi)  
  
1.0/0.0, 1.0/(-0.0), 0.0/0.0  
  
typeof(1.0/0.0), typeof(1.0/(-0.0)), typeof(0.0/0.0)  
  
for T in  
 [Int8,Int16,Int32,Int64,Int128,UInt8,UInt16,UInt32,UInt64,UInt128]  
 println("$(lpad(T,7)): [$(typemin(T)),$(typemax(T))]")  
end  
  
Int64(2.0)  
  
Char(2)  
  
convert(Int64, 2.0)  
  
convert(Char, 2)  
  
Bool(1)  
  
Bool(0)  
  
promote(Int8(1), Float16(4.5), Float32(4.1))  
  
typeof(promote(Int8(1), Float16(4.5), Float32(4.1)))  
  
function f(x)  
 x^2  
end  
  
f(4)  
  
g(x)=x^2  
  
g(8)  
  
a = [4 7 6] # вектор-строка  
b = [1, 2, 3] # вектор-столбец  
a[2], b[2] # вторые элементы векторов a и b  
  
a = 1; b = 2; c = 3; d = 4 # присвоение значений  
Am=[a b; c d]#матрица2х2  
  
Am[1,1], Am[1,2], Am[2,1], Am[2,2] # элементы матрицы  
  
aa = [1 2]  
AA = [1 2; 3 4]  
aa\*AA\*aa'  
  
aa, AA, aa'  
  
?read()  
  
io = IOBuffer("Good night");  
read(io, String)  
  
?readline()  
  
readline("text.txt")  
  
?readlines()  
  
readlines("text.txt")  
  
using DelimitedFiles  
  
?readdlm()  
  
readdlm("text.txt", '\t', String, '\n')  
  
# +  
x = [1; 2; 3; 4];  
y = [5; 6; 7; 8];  
  
open("delim\_file.txt", "w") do io  
 writedlm(io, [x y])  
end  
  
readdlm("delim\_file.txt", '\t', Int, '\n')  
# -  
  
?print()  
  
print("Hi <3")  
  
?println()  
  
println("How are you?", "Good")  
  
?show()  
  
show("<3")  
  
?write()  
  
# +  
open("text2.txt", "w") do io  
 write(io, "qwerty")  
end  
  
readlines("text2.txt")  
# -  
  
?parse()  
  
parse(Int64, "8")  
  
typeof(parse(Int64, "8"))  
  
parse(Int, "f", base=16)  
  
# + tags=[]  
#int  
  
1+2, 1-2, 1\*2, 1/2, 2^2, sqrt(4), 1>2, 1<2, 1==2, 1>=2, 1<=2, 1!=2, (1>2) && (2<3), (1>2) || (2<3)  
# -  
  
#float  
1.0+2.1, 1.1-2.1, 1.1\*2.1, 1.1/2.1, 2.1^2, sqrt(4.1), 1.1>2.1, 1.1<2.1, 1.1==2.1, 1.1>=2.1,  
1.1<=2.1, 1.1!=2.1, (1.1>2.1) && (2.1<3.1), (1.1>2.1) || (2.1<3.1)  
  
#complex  
1+2im, 1+2im-1im, 1im\*1im, (1+2im)/2, (1im)^2, sqrt(1+2im), 1+2im==5+1im, 1+2im!=5+1im, (1+2im==5+1im) && (1im==1im),  
(1+2im==5+1im) || (1im==1im)  
  
#rational  
1//2+1//3, 1//2-1//3, 1//2\*1//3, 1//2 / 3, (1//2)^2, 1//2 < 1//3, 1//2 <= 1//3, 1//2 > 1//3,  
1//2 >= 1//3, 1//2 == 1//3, 1//2 != 1//3, (1//2>1//3) && (1//4 < 1//5), (1//2>1//3) || (1//4 < 1//5)  
  
# +  
#irrational  
using IrrationalExpressions  
  
pi+2\*pi, pi-2\*pi, pi\*pi, pi /2, sqrt(pi), pi < 2\*pi, pi <= 2\*pi, pi > 2\*pi, pi >= 2\*pi, pi == 2\*pi,  
pi != 2\*pi, (pi < 2\*pi) && (pi < pi/7), (pi < 2\*pi) || (pi < pi/7)  
  
# +  
#векторы  
  
a = [1; 2]  
b = [3; 4]  
  
using LinearAlgebra  
  
a .+ b, a .- b, dot(a,b), a ⋅ b, a', a\*2  
  
# +  
#матрицы  
  
A = [1 2; 3 4]  
B = [3 1; 2 4]  
  
A + B, A - B, dot(A,B), A', A\*2  
# -

# Вывод

В ходе выполнения лабораторной работы на примерах были изучены основы синтаксиса в Julia.

# Библиография

1. Методические материалы курса.
2. Wikipedia: Julia (язык программирования). (https://ru.wikipedia.org/wiki/Julia\_(%D1%8F%D0%B7%D1%8B%D0%BA\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)