Лабораторная работа №1

Julia. Установка и настройка. Основные принципы

Чемоданова Ангелина Александровна

Содержание

# 1 Введение

## 1.1 Цели и задачи

**Цель работы**

Основная цель работы — подготовить рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомиться с основами синтаксиса Julia[[1]][2].

**Задание**

1. Установите под свою операционную систему Julia, Jupyter.
2. Используя Jupyter Lab, повторите примеры.
3. Выполните задания для самостоятельной работы.

# 2 Выполнение лабораторной работы

## 2.1 Подготовка инструментария к работе

На моем компьютере ранее уже была установлена Julia для выполнения работ по “Математическому моделированию”(рис. 1):



Рис. 1: Julia

## 2.2 Основы синтаксиса Julia на примерах

Для начала потренируемся с определением типов числовых величин (рис. 2):

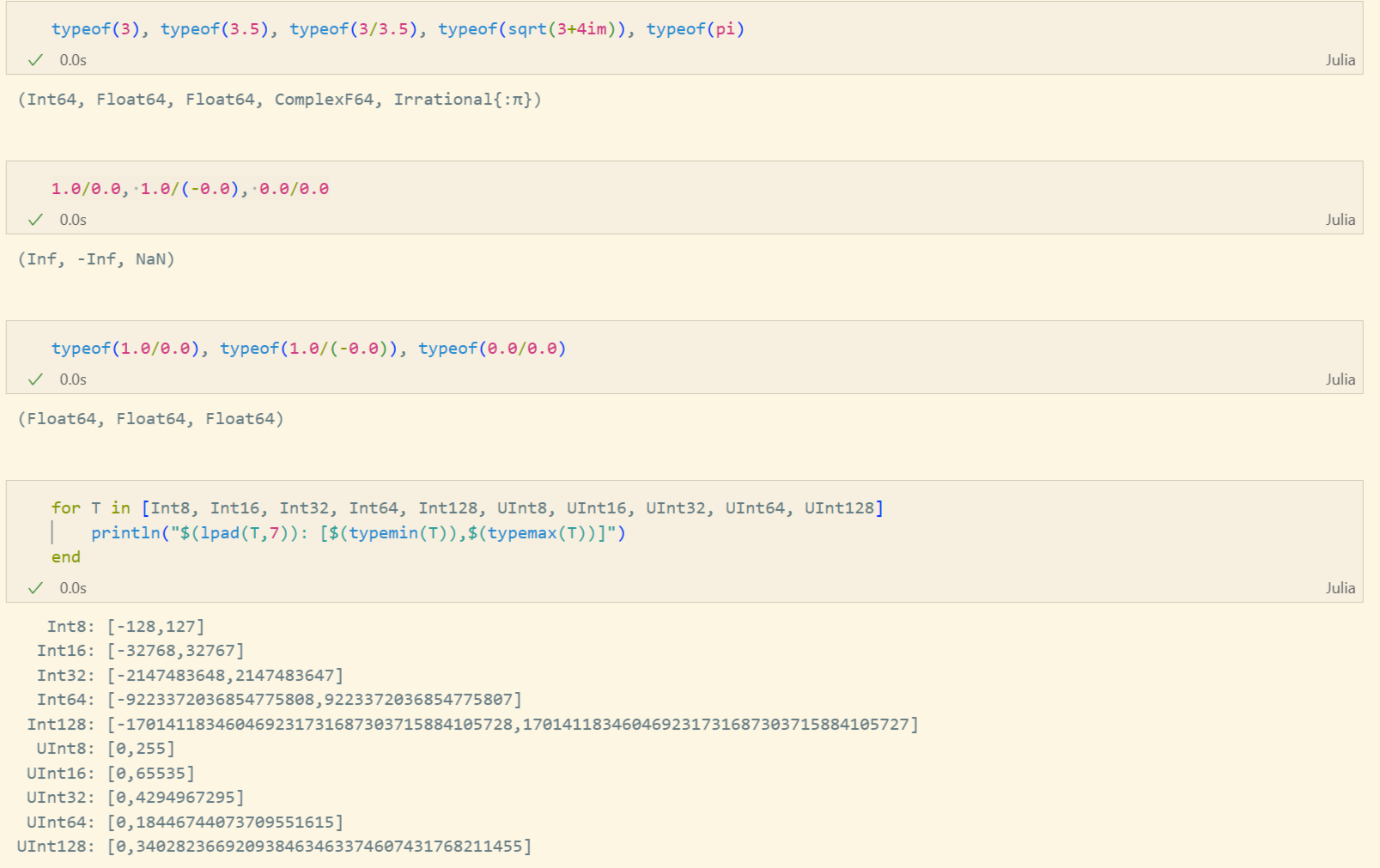


Рис. 2: Примеры определения типа числовых величин

После чего приступим к рассмотрению приведения аргументов к одному типу (рис. 3):

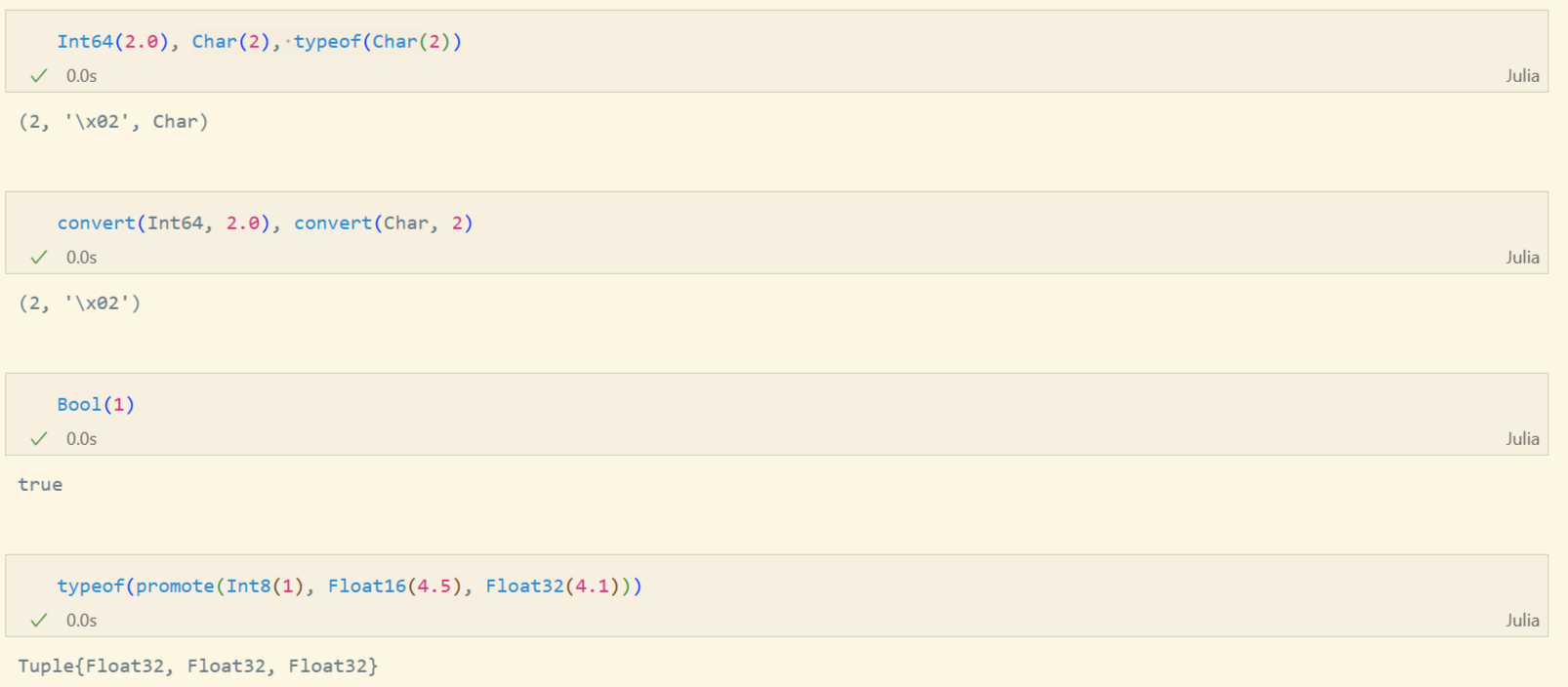


Рис. 3: Примеры приведения аргументов к одному типу

И рассмотрим примеры определения функций (рис. 4), а также работу с массивами (рис. 5):

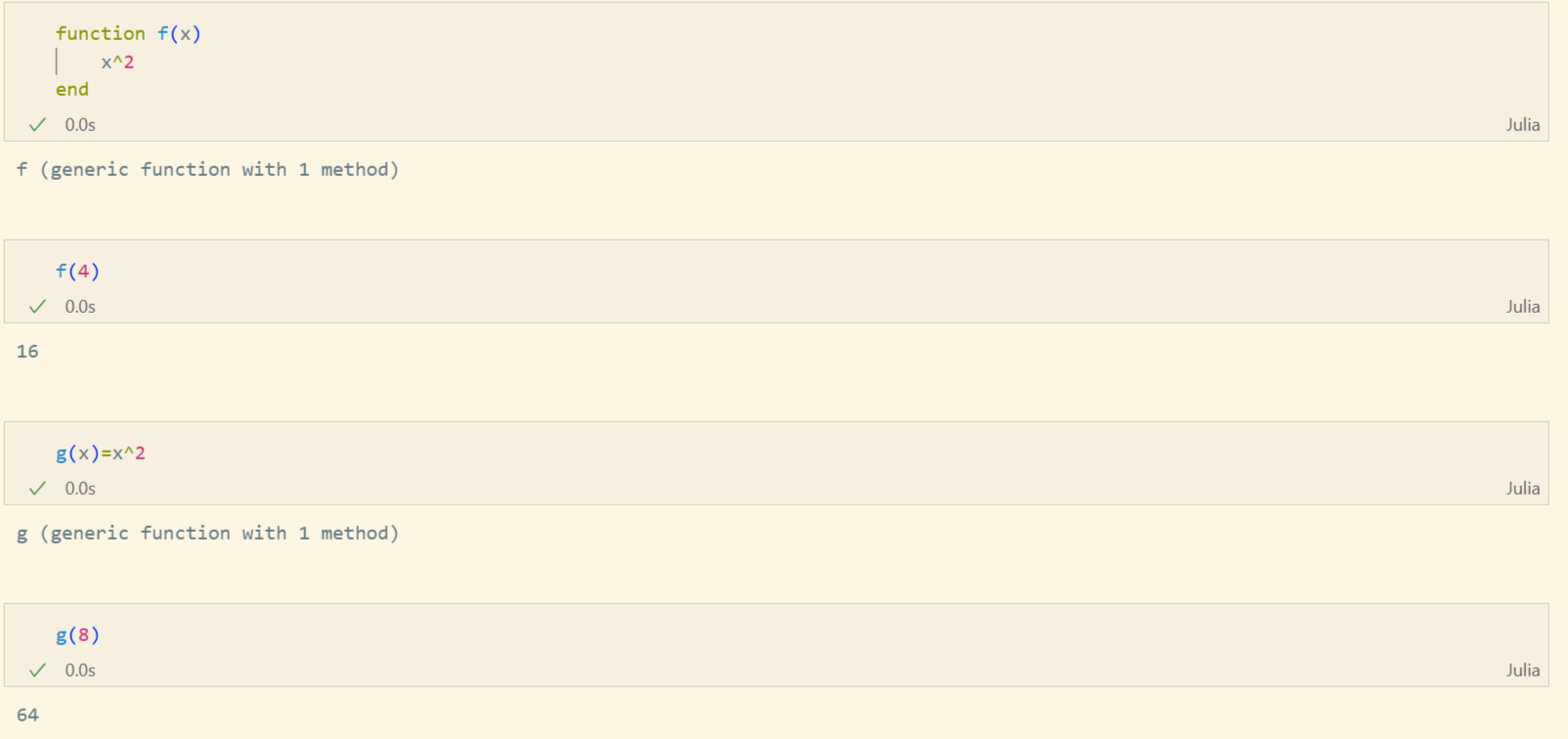


Рис. 4: Примеры определения функций

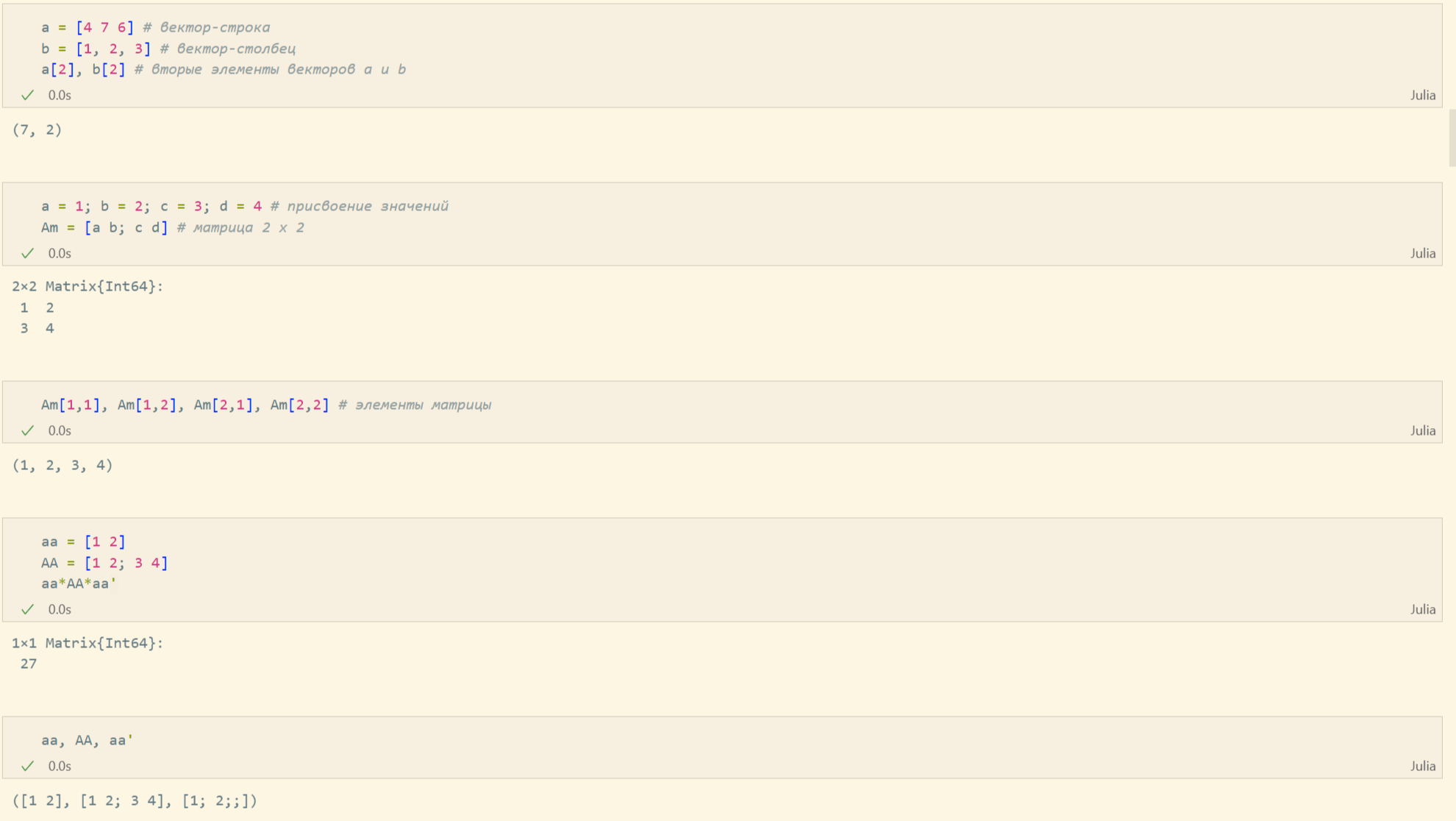


Рис. 5: Примеры работы с массивами

## 2.3 Самостоятельная работа

В первом задании рассмотрим основные функции для чтения / записи / вывода информации на экран. Для этого составим свои примеры (рис. 6 - рис. 8):

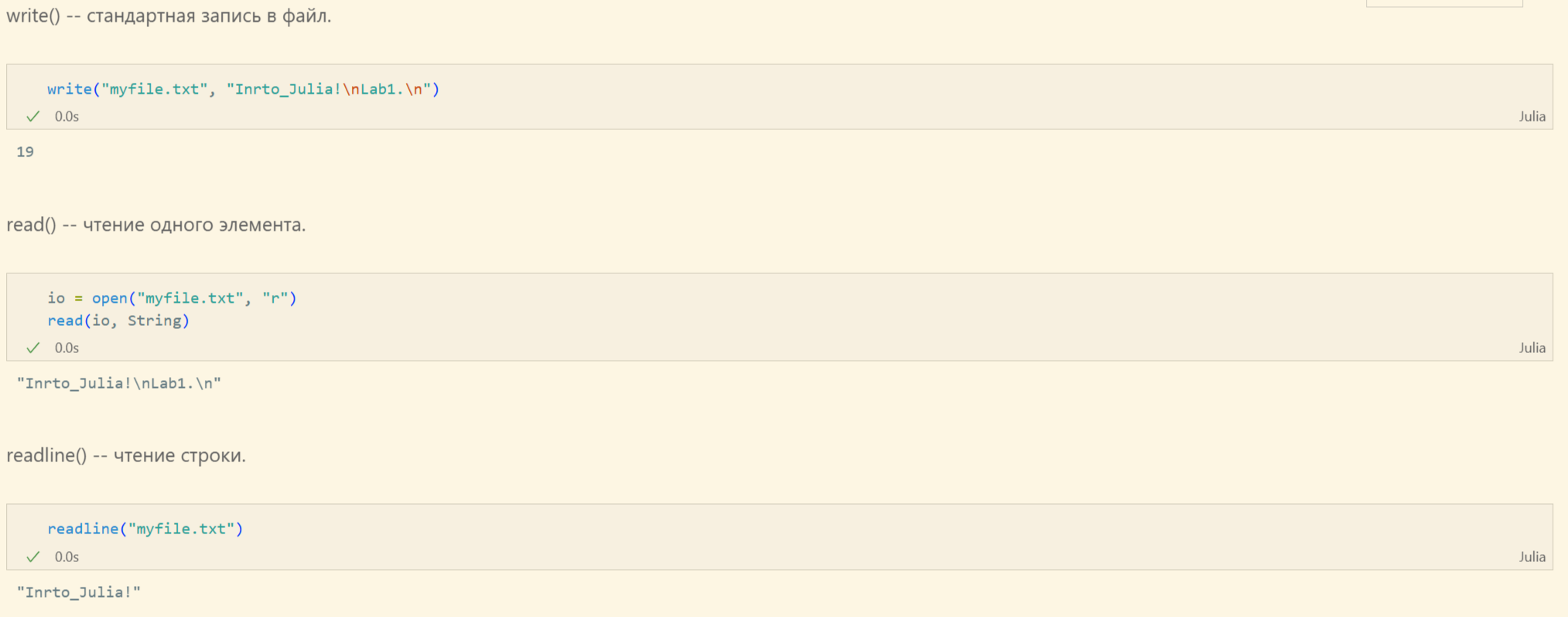


Рис. 6: Примеры работы с функциями для чтения/записи/вывода информации на экран

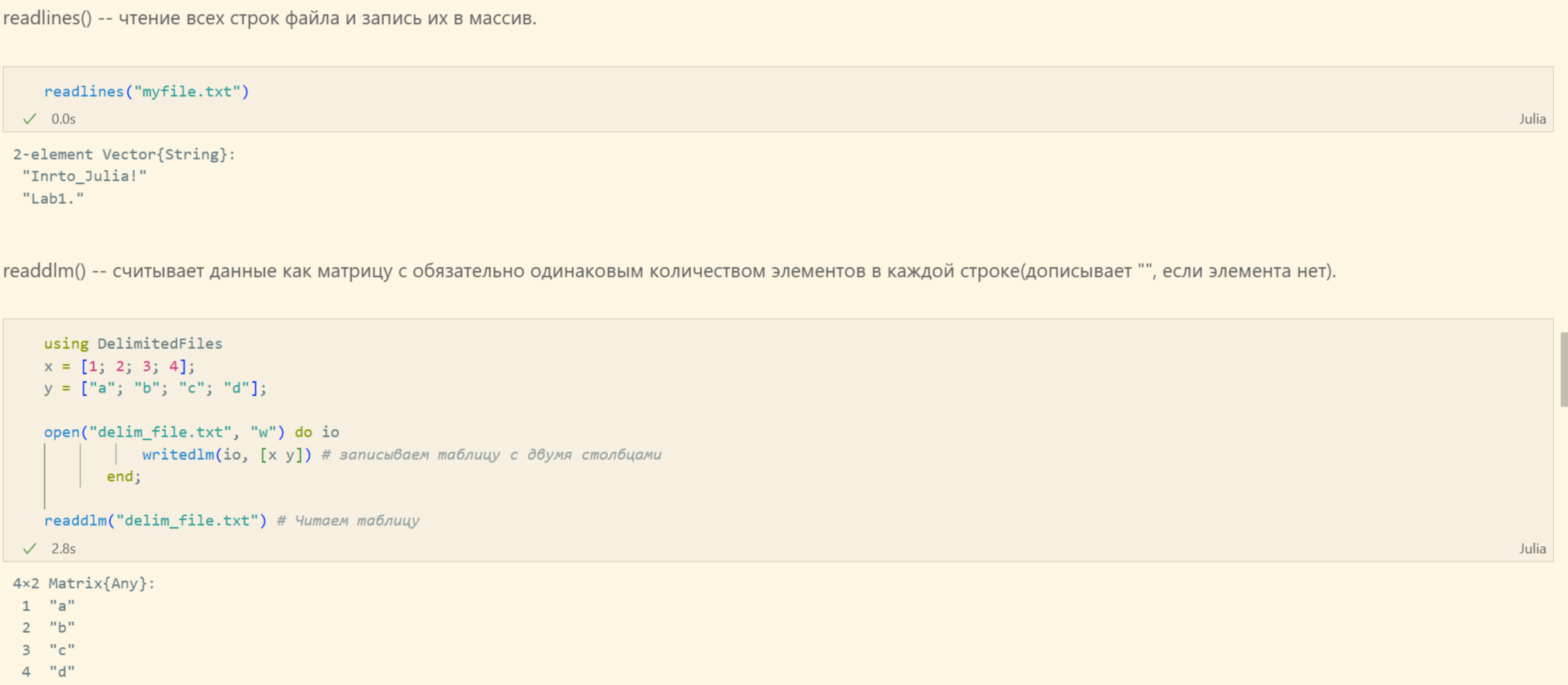


Рис. 7: Примеры работы с функциями для чтения/записи/вывода информации на экран

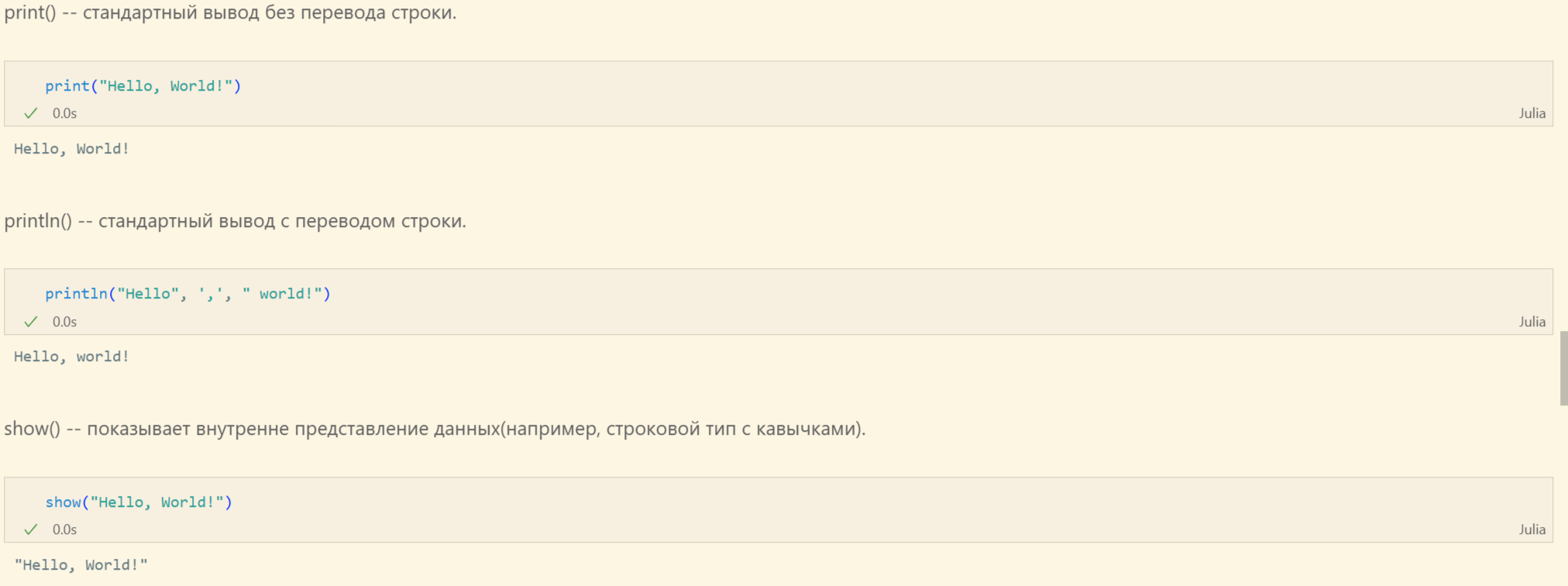


Рис. 8: Примеры работы с функциями для чтения/записи/вывода информации на экран

* read() – чтение одного элемента,
* readline() – чтение строки,
* readlines() – чтение всех строк файла и запись их в массив,
* readdlm() – считывает данные как матрицу с обязательно одинаковым количеством элементов в каждой строке(дописывает "", если элемента нет),
* print() – стандартный вывод без перевода строки,
* println() – стандартный вывод с переводом строки,
* show() – показывает внутренне представление данных(например, строковой тип с кавычками),
* write() – стандартная запись в файл.

Во втором задании состаивим пример для функции parse() (рис. 9):

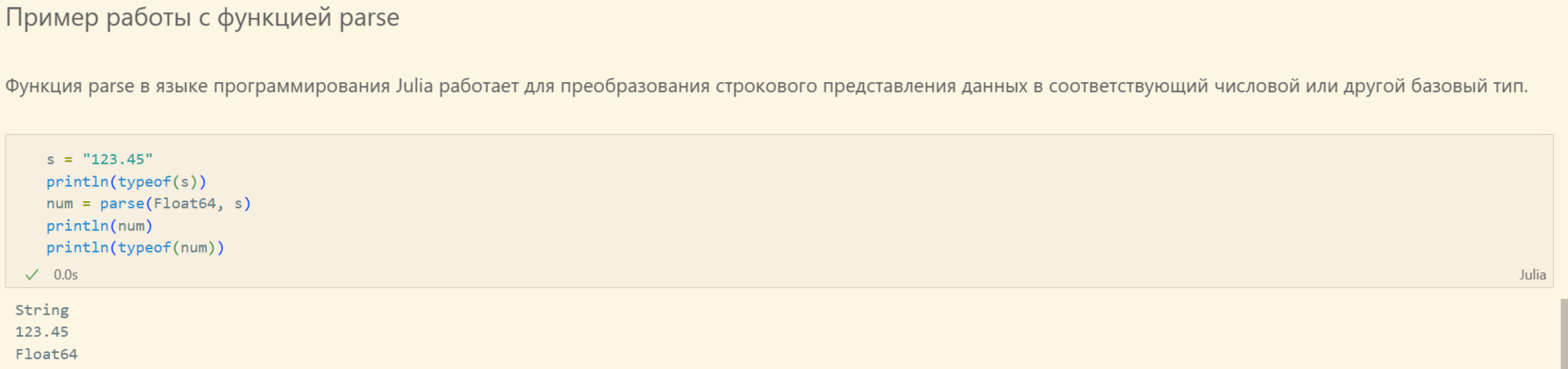


Рис. 9: Пример работы с функцией parse

Далее изучим синтаксис Julia для базовых математических операций с разным типом переменных (рис. 10 - рис. 12):

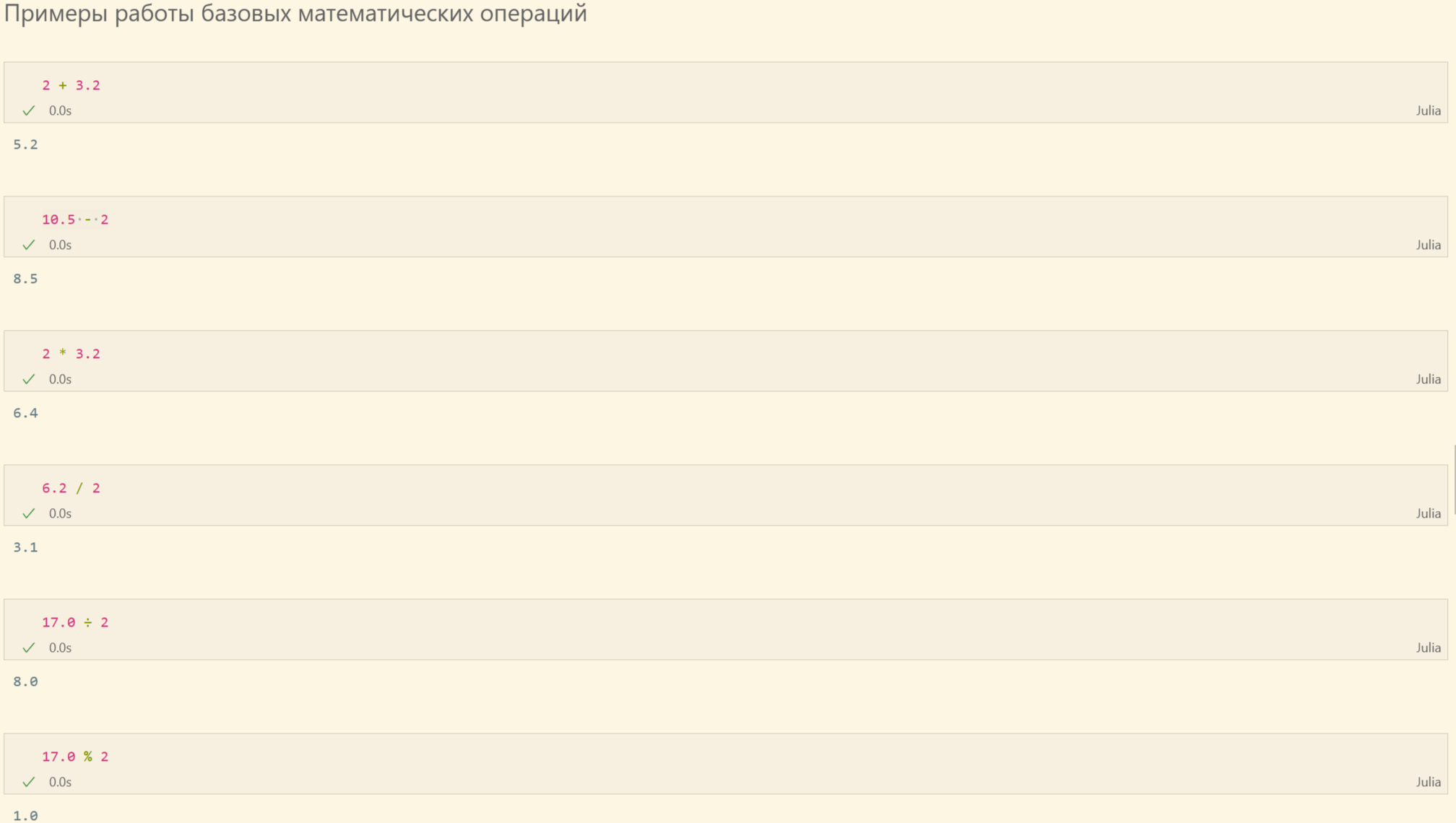


Рис. 10: Примеры работы базовых математических операций

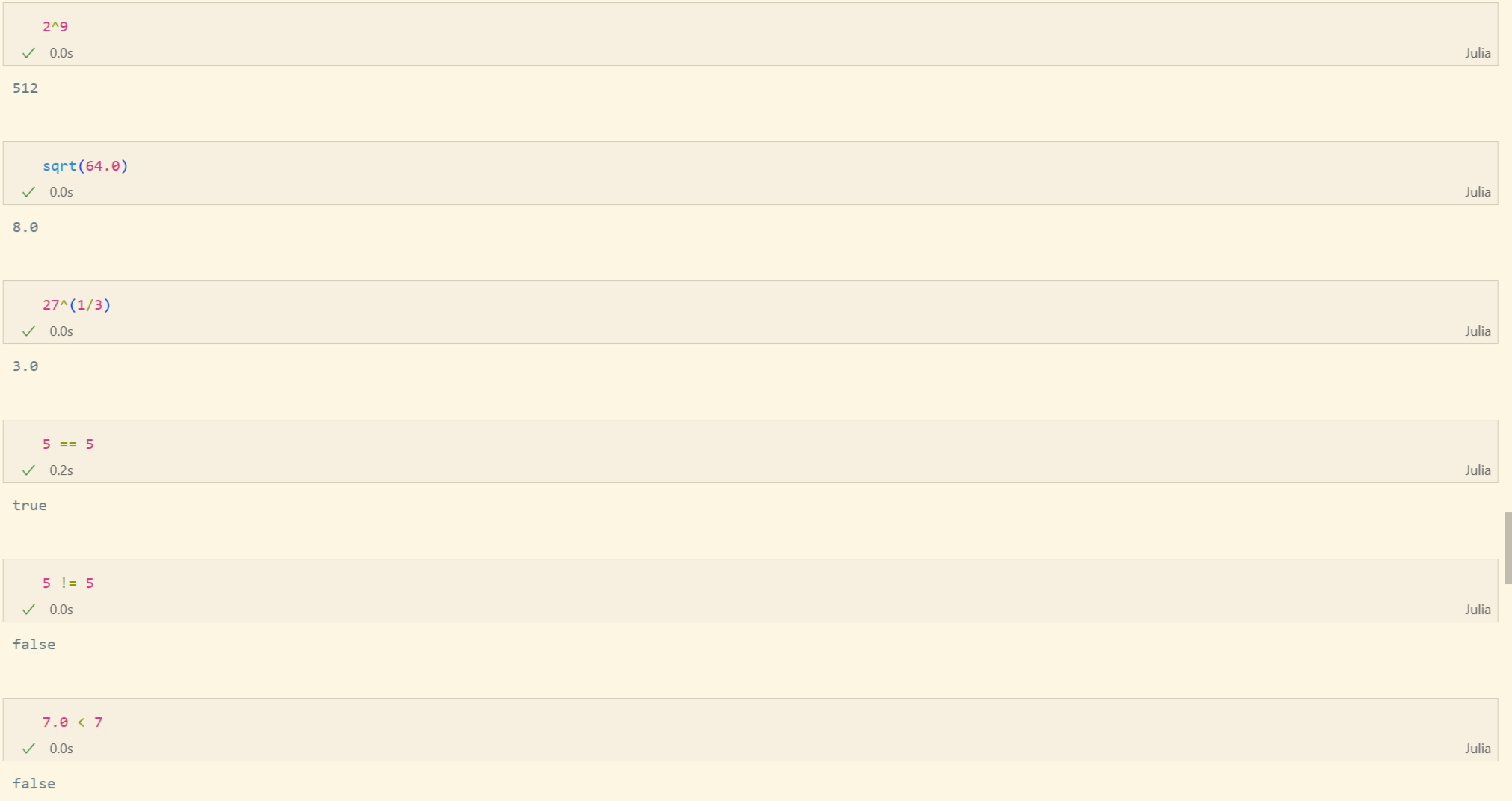


Рис. 11: Примеры работы базовых математических операций

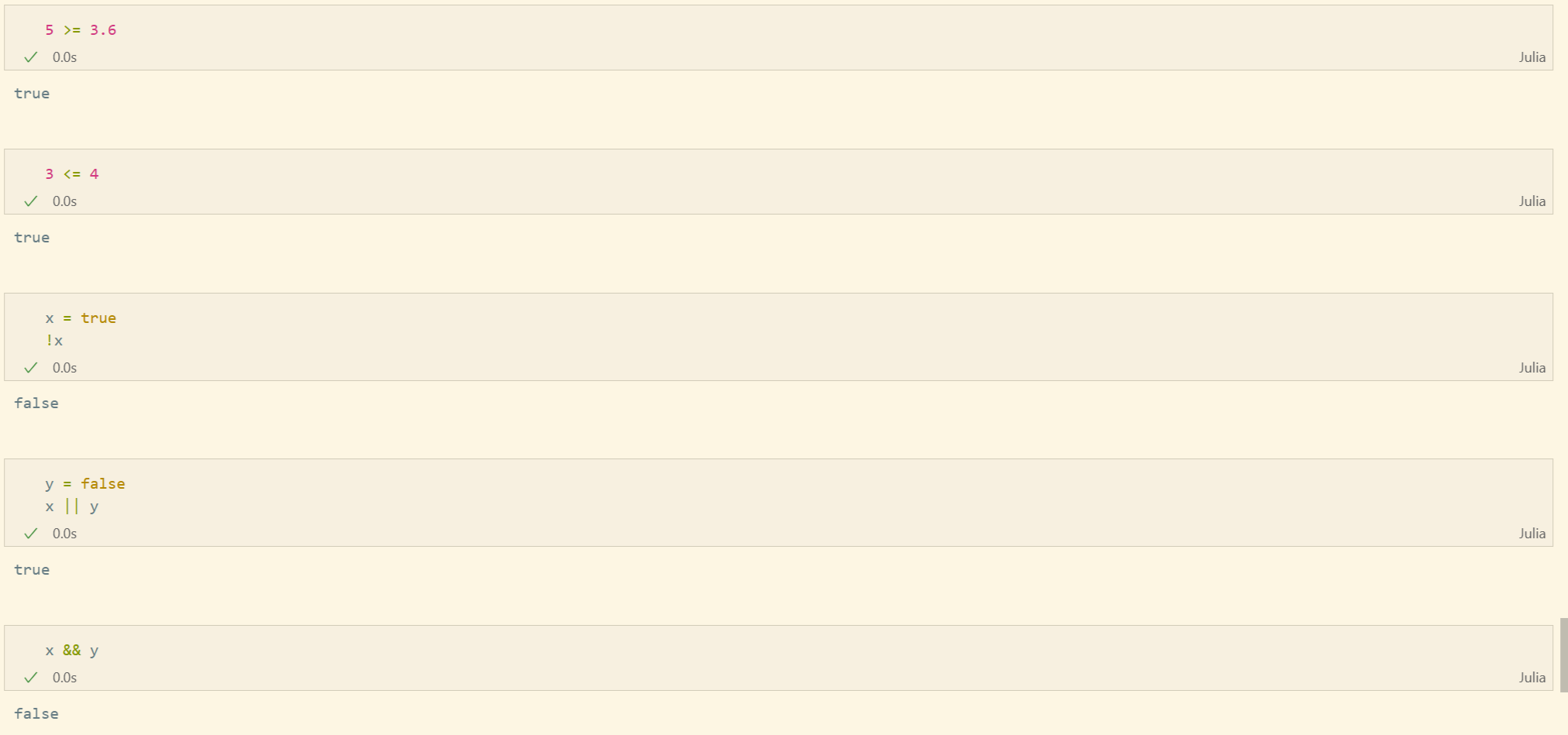


Рис. 12: Примеры работы базовых математических операций

В конце работы приведём несколько примеров с операциями над матрицами (рис. 13 - рис. 14):

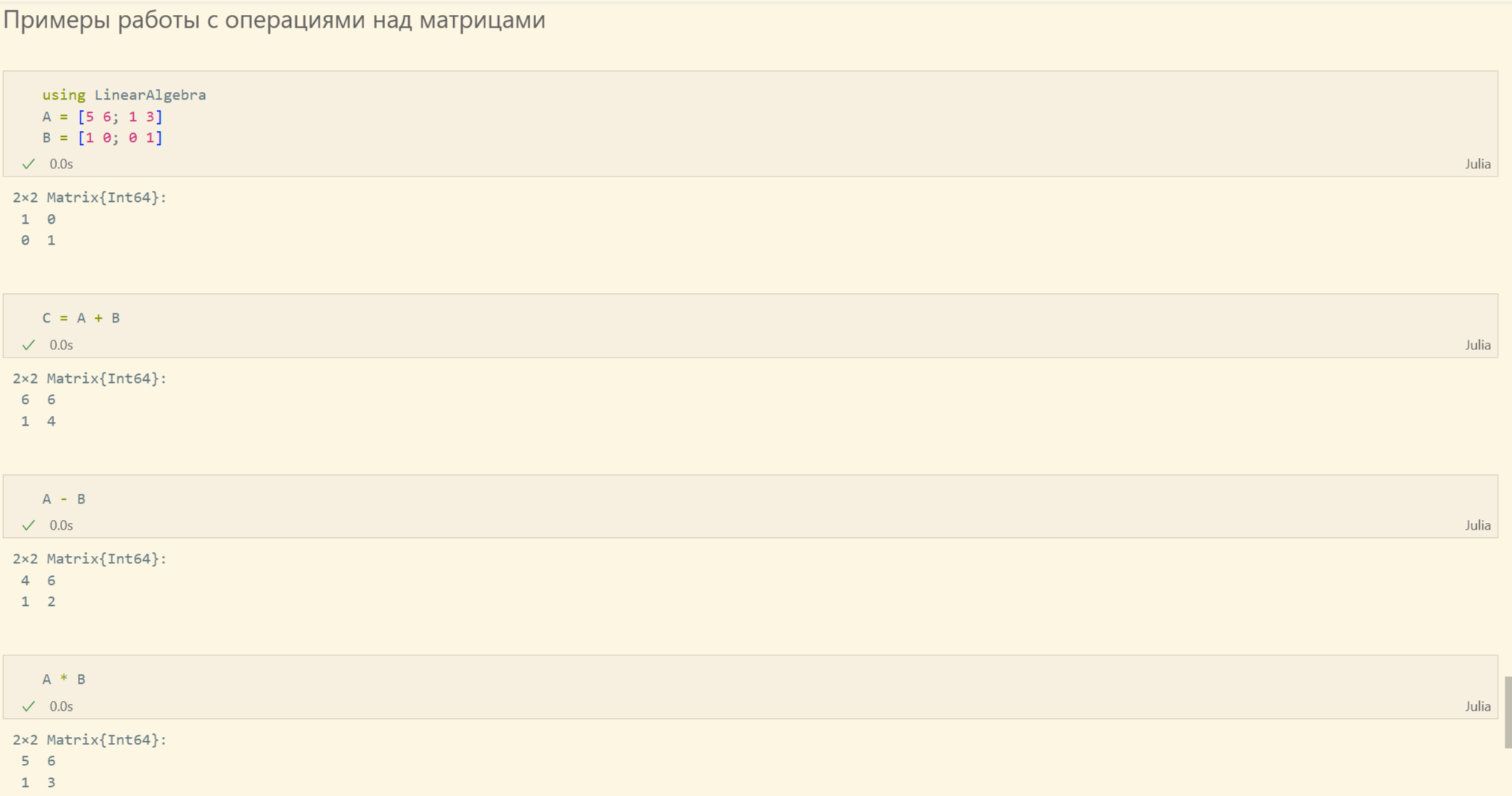


Рис. 13: Примеры работы с операциями над матрицами

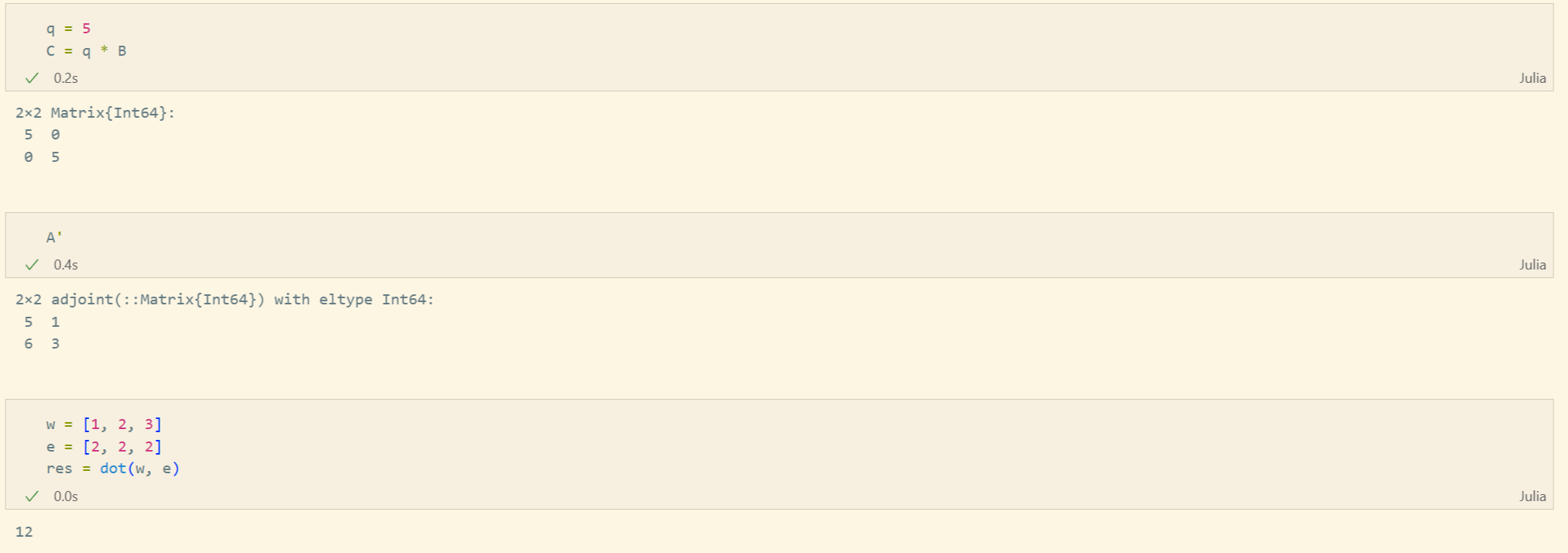


Рис. 14: Примеры работы с операциями над матрицами

# 3 Выводы

В результате выполнения данной лабораторной работы мы подготовили рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомились с основами синтаксиса Julia.

# Список литературы

1. JuliaLang [Электронный ресурс]. 2025 JuliaLang.org contributors. URL: <https://julialang.org/> (дата обращения: 09.09.2025).

2. Julia 1.11 Documentation [Электронный ресурс]. 2025 JuliaLang.org contributors. URL: <https://docs.julialang.org/en/v1/> (дата обращения: 09.09.2025).