Лабораторная работа № 2

Структуры данных

Шияпова Дарина Илдаровна

Содержание

# 1 Цель работы

Основная цель работы – изучить несколько структур данных, реализованных в Julia, научиться применять их и операции над ними для решения задач.

# 2 Задание

1. Используя Jupyter Lab, повторите примеры.
2. Выполните задания для самостоятельной работы.

# 3 Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений [**julialang?**]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia [**juliadoc?**].

Рассмотрим несколько структур данных, реализованных в Julia. Несколько функций (методов), общих для всех структур данных:

* isempty() – проверяет, пуста ли структура данных;
* length() – возвращает длину структуры данных;
* in() – проверяет принадлежность элемента к структуре;
* unique() – возвращает коллекцию уникальных элементов структуры,
* reduce() – свёртывает структуру данных в соответствии с заданным бинарным оператором;
* maximum() (или minimum()) – возвращает наибольший (или наименьший) результат вызова функции для каждого элемента структуры данных.

# 4 Выполнение лабораторной работы

Для начала выполним примеры из раздела про кортежи (рис. 1-2).

Кортеж (Tuple) – структура данных (контейнер) в виде неизменяемой индексируемой последовательности элементов какого-либо типа (элементы индексируются с единицы).

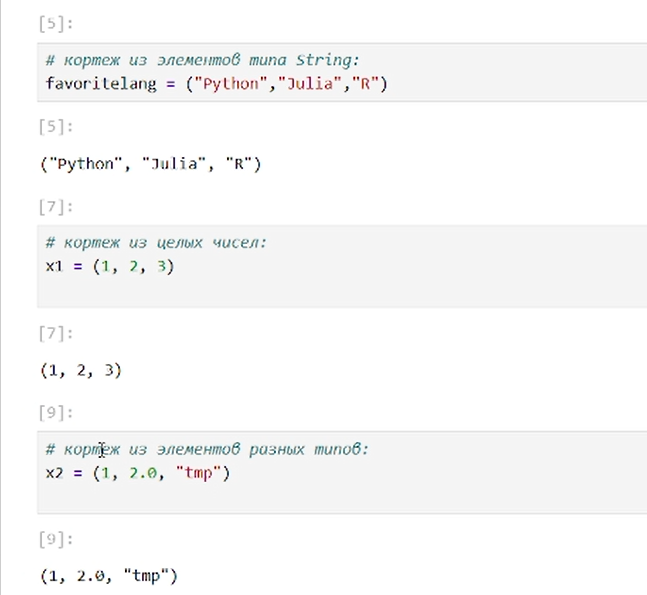


Рис. 1: Примеры использования кортежей

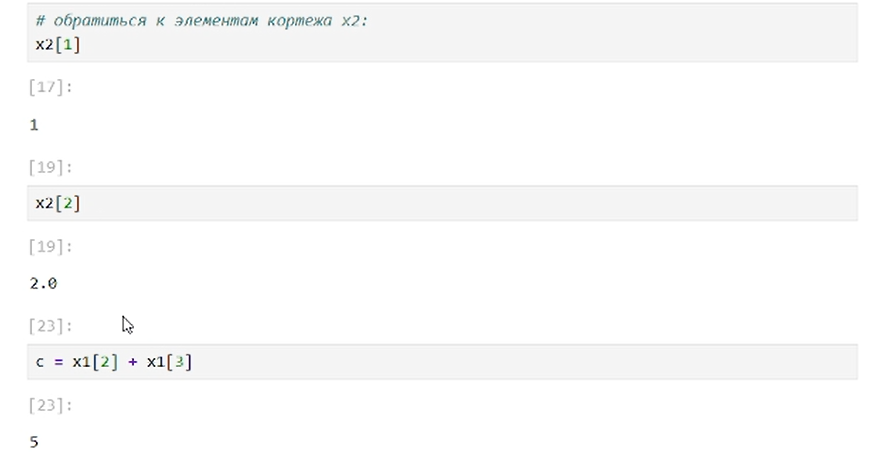


Рис. 2: Примеры использования кортежей

Теперь выполним примеры из раздела про словари (рис. 3-**¿fig:024?**).

Словарь – неупорядоченный набор связанных между собой по ключу данных.

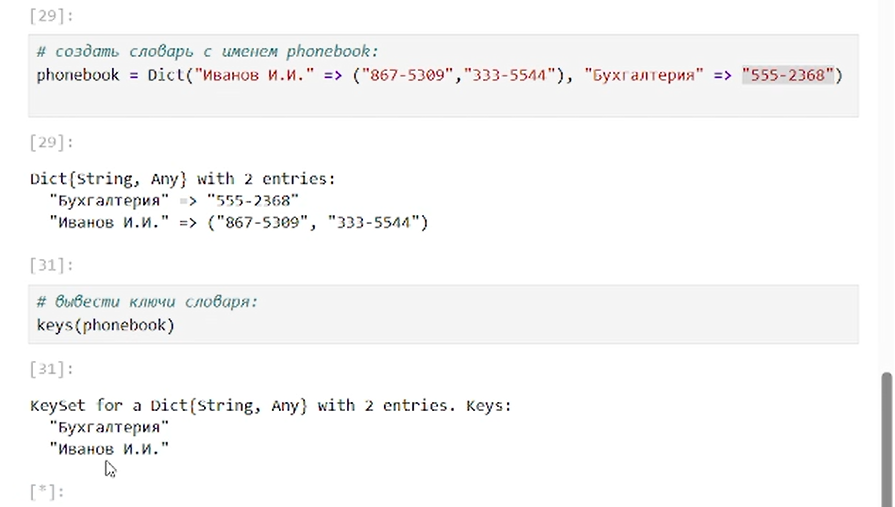


Рис. 3: Примеры использования словарей



Рис. 4: Примеры использования словарей

Выполним примеры из раздела про множества (рис. 5-6).

Множество, как структура данных в Julia, соответствует множеству, как математическому объекту, то есть является неупорядоченной совокупностью элементов какого-либо типа. Возможные операции над множествами: объединение, пересечение, разность; принадлежность элемента множеству.

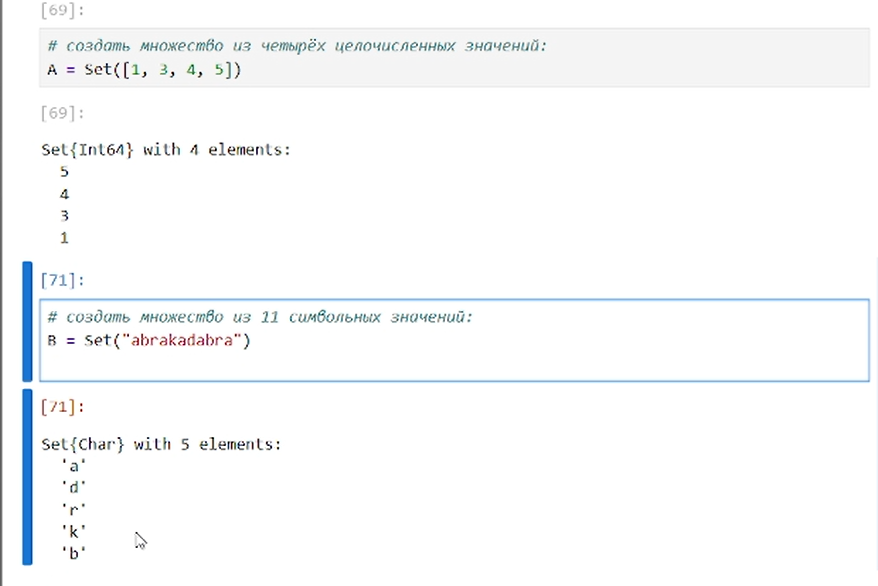


Рис. 5: Примеры использования множеств

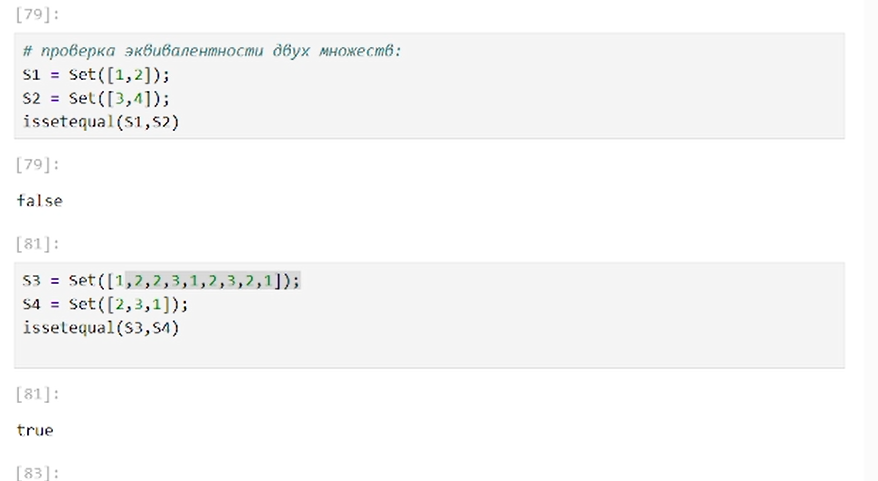


Рис. 6: Примеры использования множеств

Выполним примеры из раздела про массивы (рис. 5-9).

Массив — коллекция упорядоченных элементов, размещённая в многомерной сетке. Векторы и матрицы являются частными случаями массивов.



Рис. 7: Примеры использования массивов

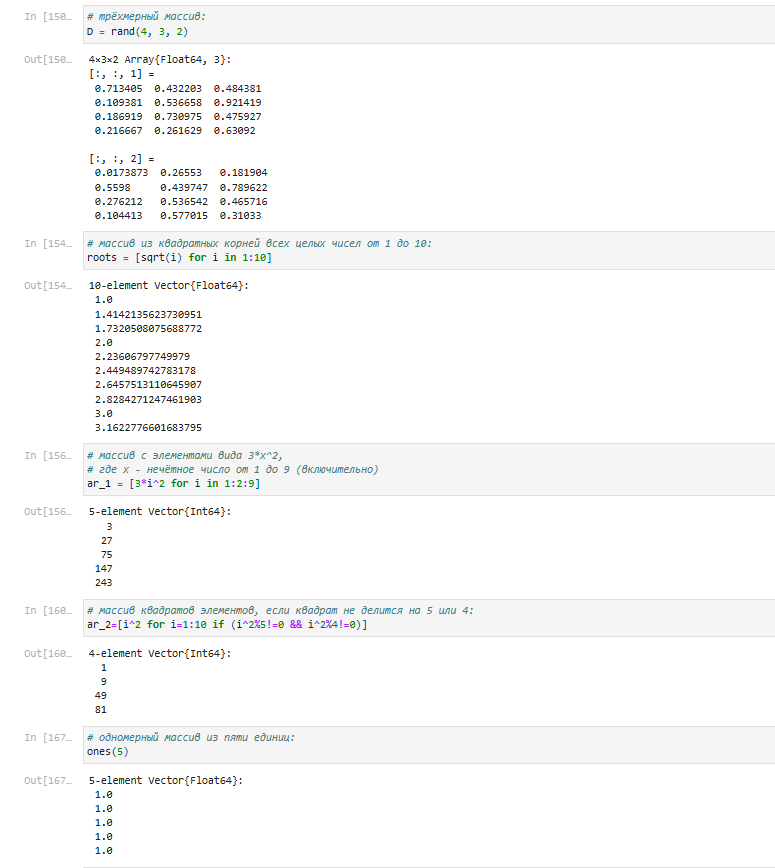


Рис. 8: Примеры использования массивов



Рис. 9: Примеры использования массивов

Перейдем к выполнению заданий.

1. Даны множества: . Найдем (рис. 10).

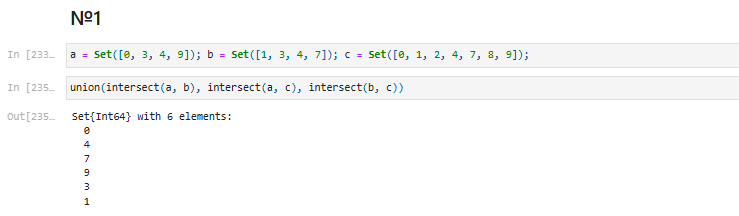


Рис. 10: Задание №1. Работа с множествами

1. Приведем свои примеры с выполнением операций над множествами элементов разных типов (рис. 11).

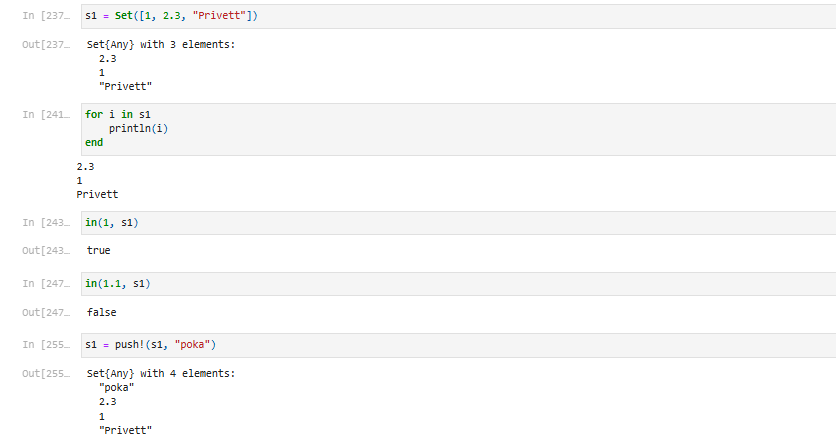


Рис. 11: Задание №2. Примеры операций над множествами элементов разных типов

1. Создадим массивы разными способами, используя циклы (рис. 12-22):

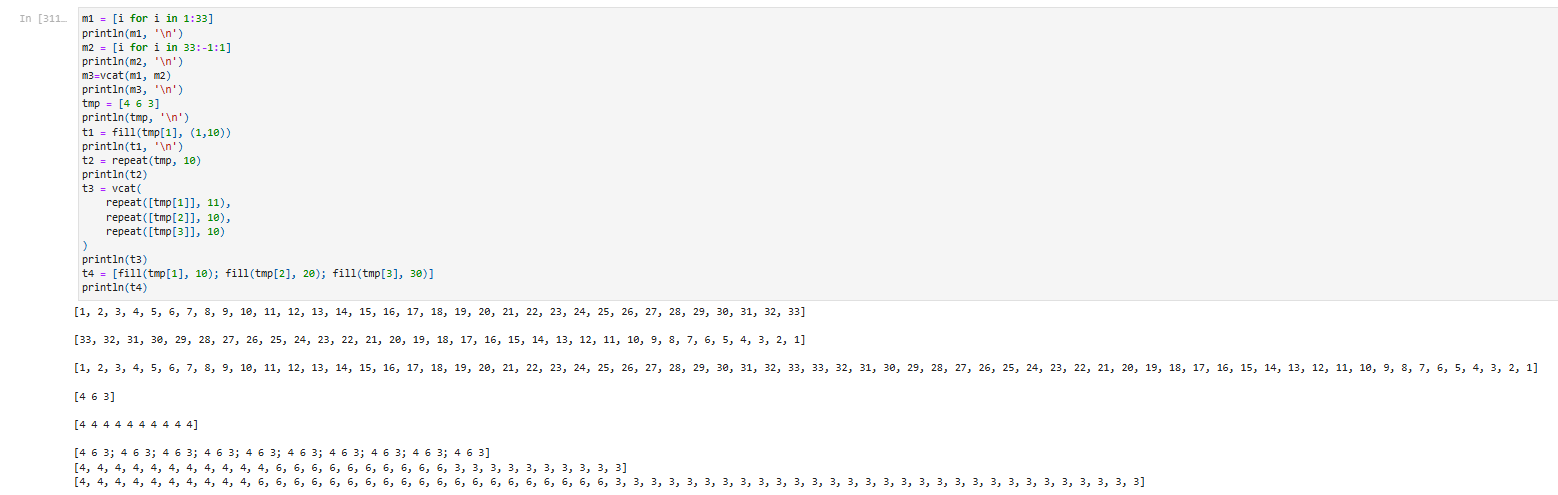


Рис. 12: Задание №3. Работа с массивами

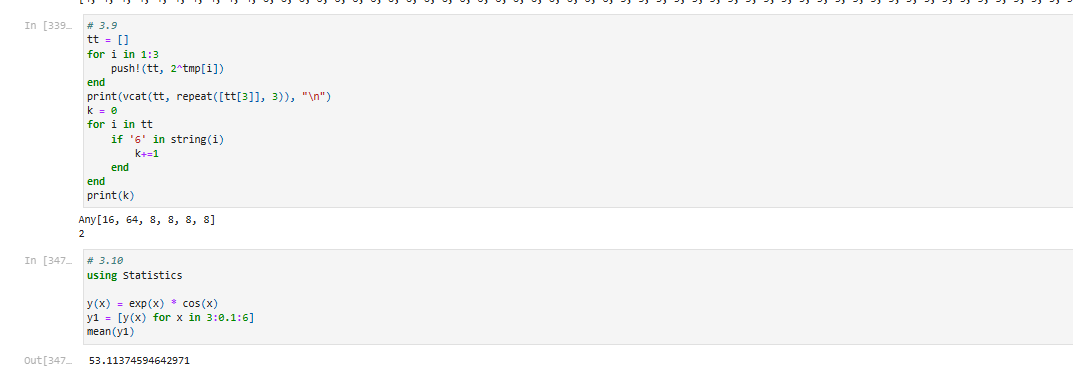


Рис. 13: Задание №3. Работа с массивами

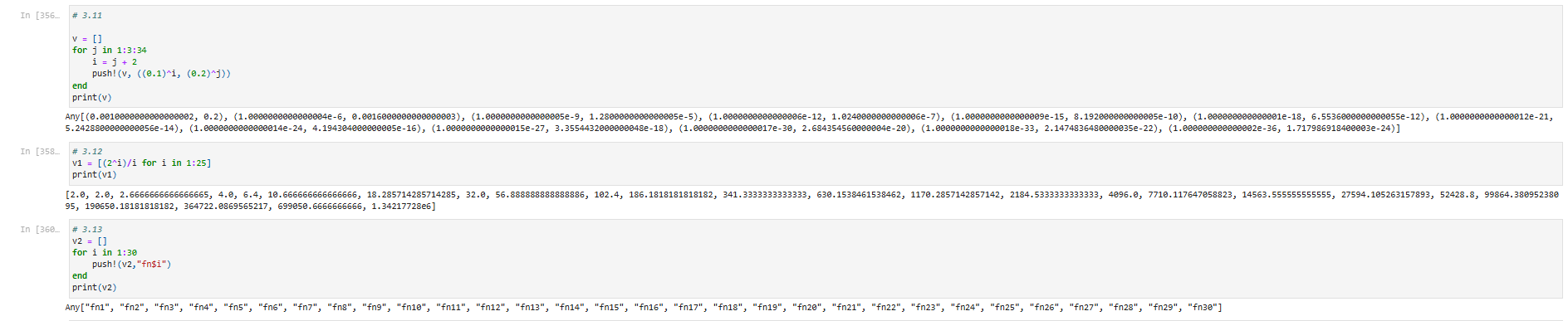


Рис. 14: Задание №3. Работа с массивами

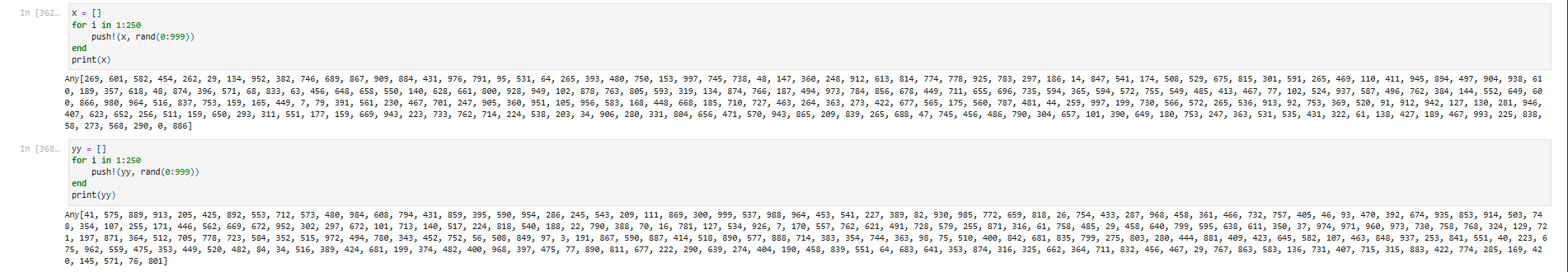


Рис. 15: Задание №3. Работа с массивами

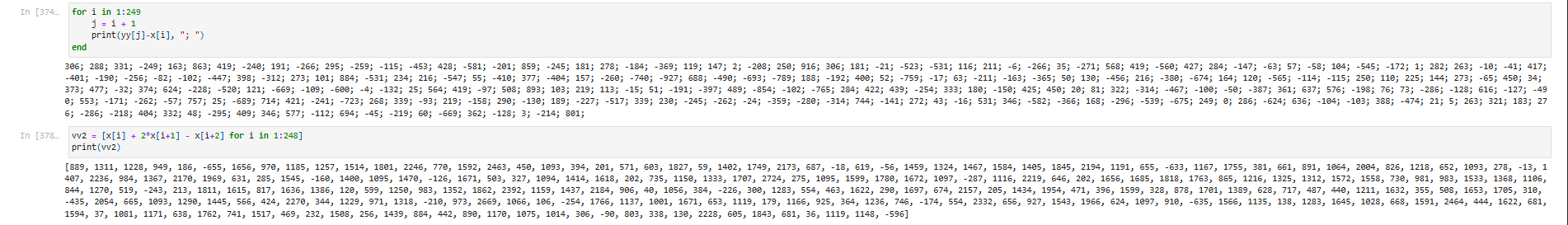


Рис. 16: Задание №3. Работа с векторами

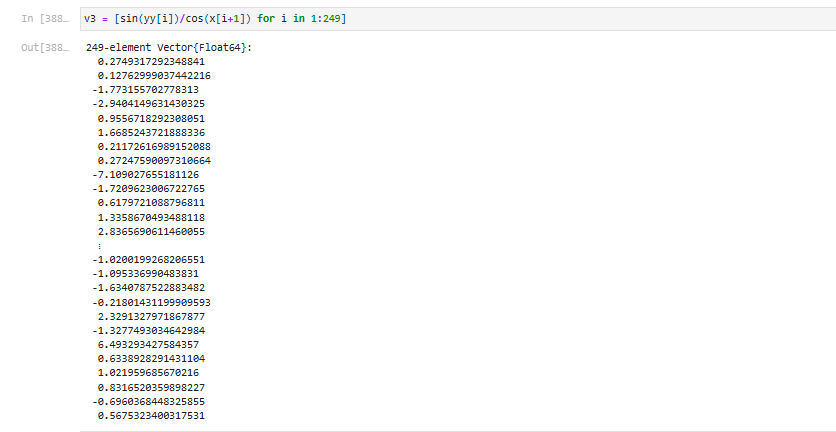


Рис. 17: Задание №3. Работа с векторами

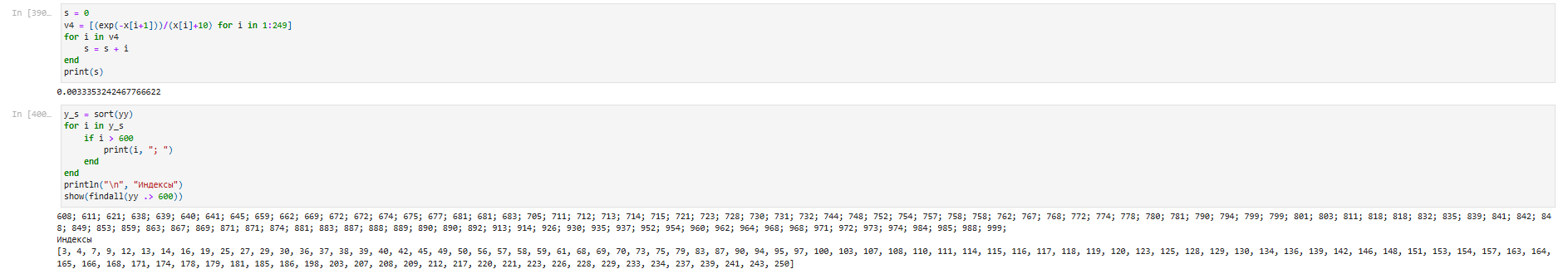


Рис. 18: Задание №3. Работа с векторами

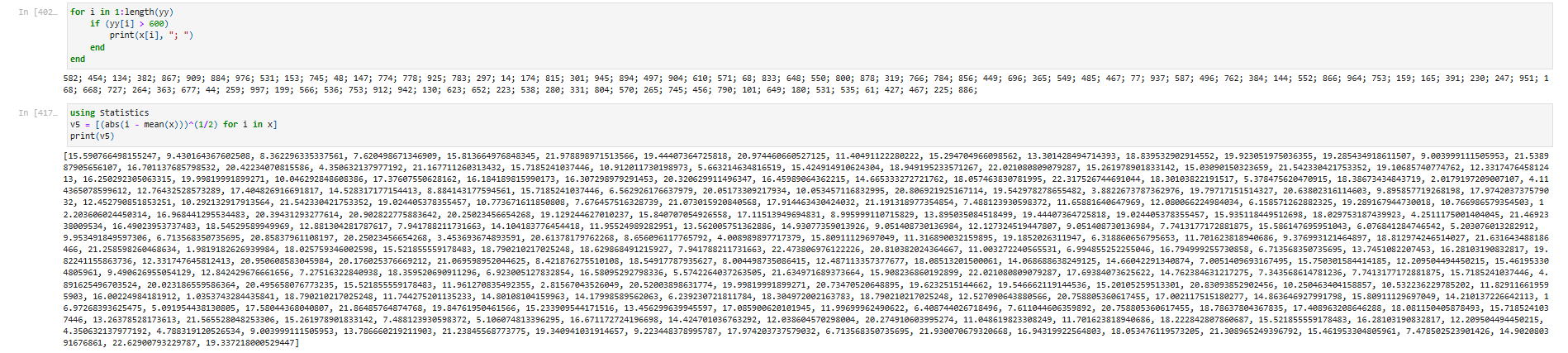


Рис. 19: Задание №3. Работа с векторами

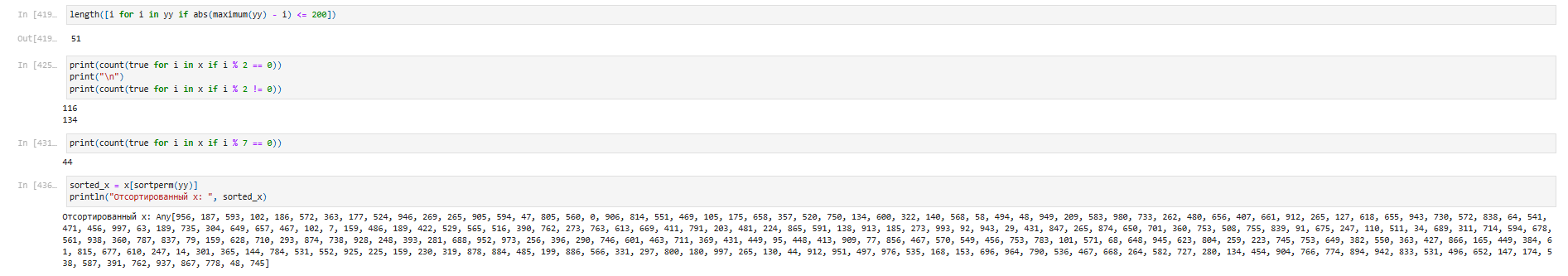


Рис. 20: Задание №3. Работа с векторами

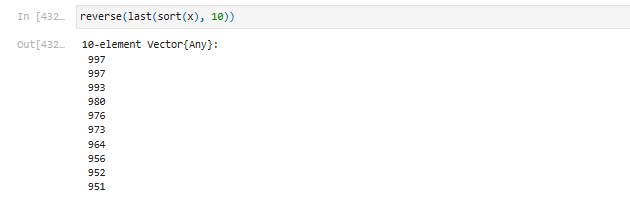


Рис. 21: Задание №3. Работа с векторами

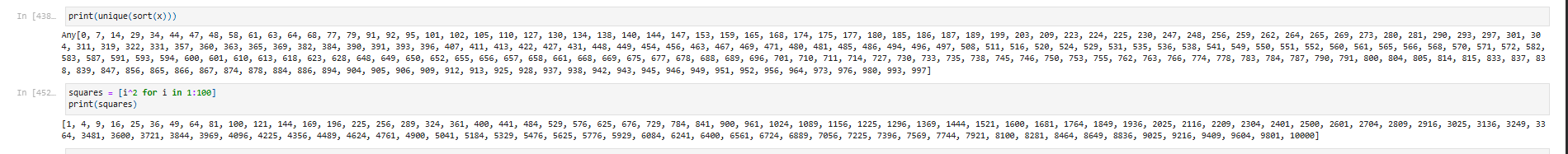


Рис. 22: Задание №3. Работа с векторами

1. Создадим массив squares, в котором будут храниться квадраты всех целых чисел от 1 до 100 (рис. 23).

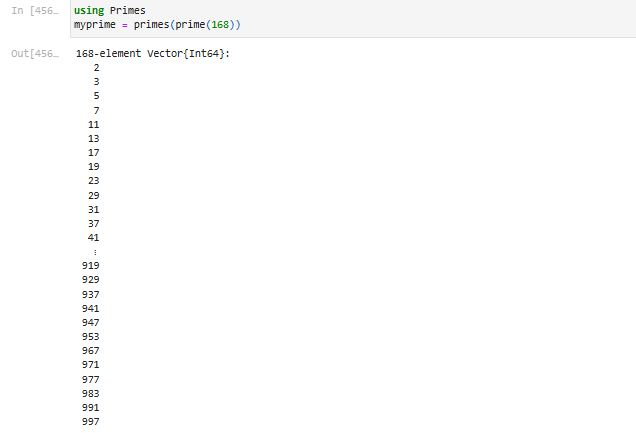


Рис. 23: Задание №4

1. Подключим пакет Primes (функции для вычисления простых чисел). Сгенерируем массив myprimes, в котором будут храниться первые 168 простых чисел. Определим 89-е наименьшее простое число. Получии срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа (рис. 24).

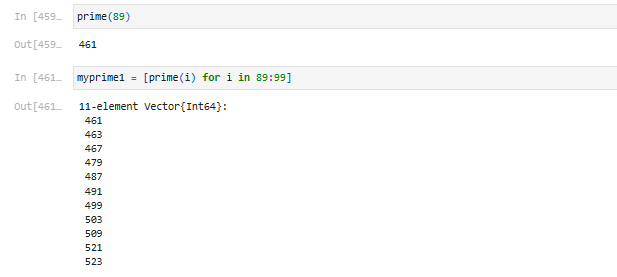


Рис. 24: Задание №5. Работа с пакетом Primes

1. Вычислим следующие выражения (рис. 25).

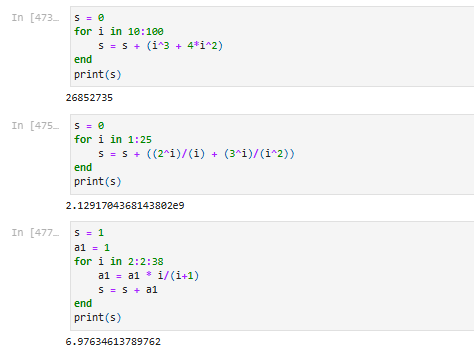


Рис. 25: Задание №6

# 5 Выводы

В результате выполнения данной лабораторной работы я изучила несколько структур данных, реализованных в Julia, научилась применять их и операции над ними для решения задач.

# Список литературы