Отчёт по лабораторной работе №2 по дисциплине Компьютерный практикум по статистическому анализу данных

Структуры данных

Шаповалова Диана Дмитриевна

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 Выполнение лабораторной работы

## 2.1 Кортежи

Примеры кортежей:

#пустой кортеж:

()

#кортеж из элементов типа String:

favoritelang = (“Python”,“Julia”,“R”)

#кортеж из целых чисел:

x1 = (1, 2, 3)

#кортеж из элементов разных типов:

x2 = (1, 2.0, “tmp”)

#именованный кортеж:

x3 = (a=2, b=1+2)

Примеры операций над кортежами:

#длина кортежа x2:

length(x2)

и т.д.

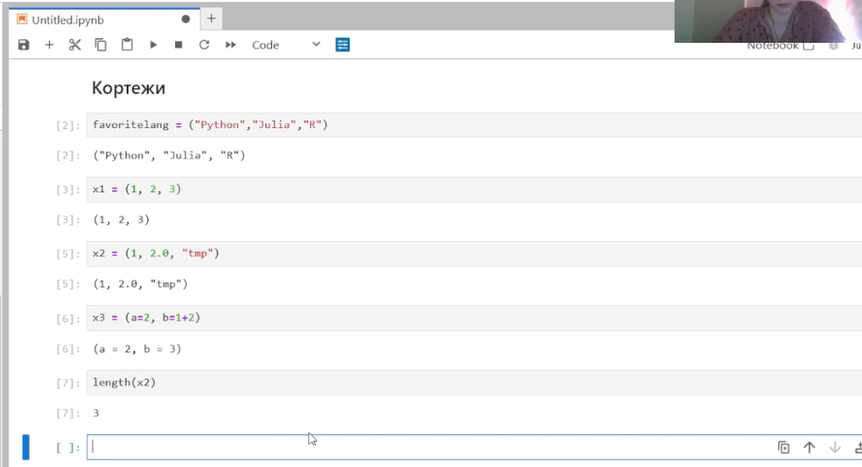


Рис. 1: Выполняем примеры по Кортежам

## 2.2 Словари

Словарь — неупорядоченный набор связанных между собой по ключу данных.

Синтаксис определения словаря:

Dict(key1 => value1, key2 => value2, …)

Примеры словарей и операций над ними:

#создать словарь с именем phonebook:

phonebook = Dict(“Иванов И.И.” => (“867-5309”,“333-5544”), “Бухгалтерия” => “555-2368”)

#вывести ключи словаря:

keys(phonebook)

#вывести значения элементов словаря:

values(phonebook)

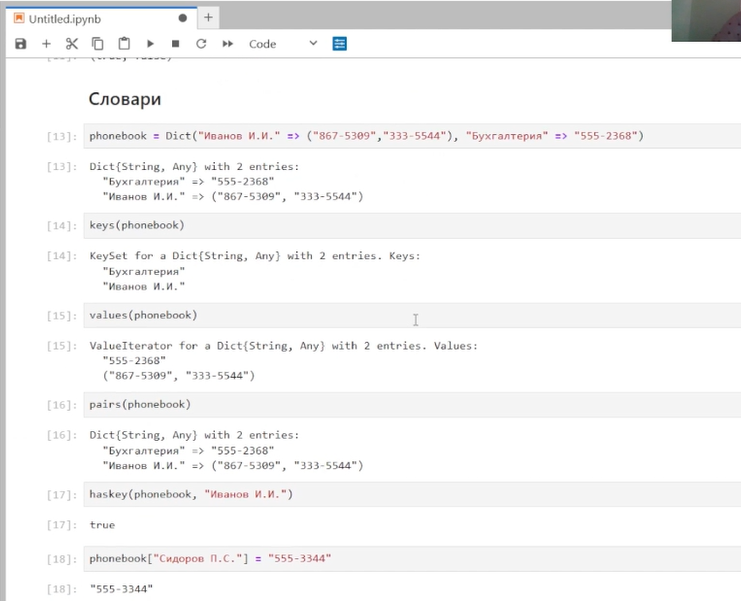


Рис. 2: Выполняем примеры по Словарям

## 2.3 Множества

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.

Примеры множеств и операций над ними:

#создать множество из четырёх целочисленных значений:

A = Set([1, 3, 4, 5])

#создать множество из 11 символьных значений:

B = Set(“abrakadabra”)

#проверка эквивалентности двух множеств:

S1 = Set([1,2]);

S2 = Set([3,4]);

issetequal(S1,S2)

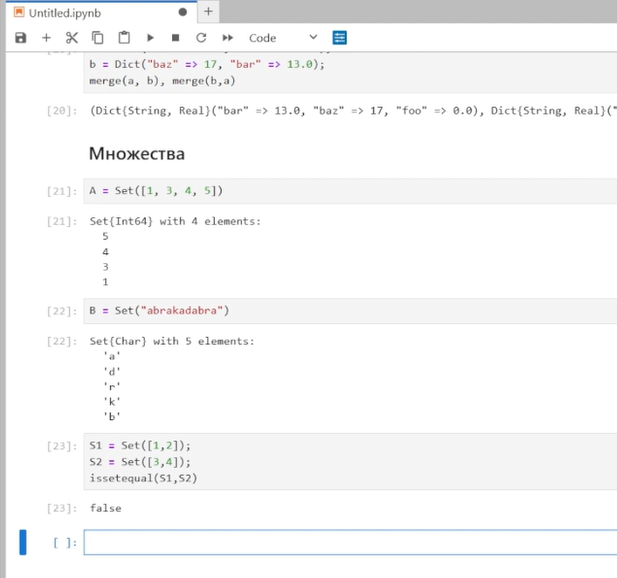


Рис. 3: Выполняем примеры по Множествам

## 2.4 Массивы

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

Общий синтаксис одномерных массивов:

array\_name\_1 = [element1, element2, …]

array\_name\_2 = [element1 element2 …]

Примеры массивов:

#создание пустого массива с абстрактным типом:

empty\_array\_1 = []

#создание пустого массива с конкретным типом:

empty\_array\_2 = (Int64)[]

empty\_array\_3 = (Float64)[]

## 2.5 Задания для самостоятельного выполнения

## 2.6 1. Даны множества: 𝐴 = {0, 3, 4, 9}, 𝐵 = {1, 3, 4, 7}, 𝐶 = {0, 1, 2, 4, 7, 8, 9}. Найти 𝑃 = 𝐴 ∩ 𝐵 ∪ 𝐴 ∩ 𝐵 ∪ 𝐴 ∩ 𝐶 ∪ 𝐵 ∩ 𝐶.

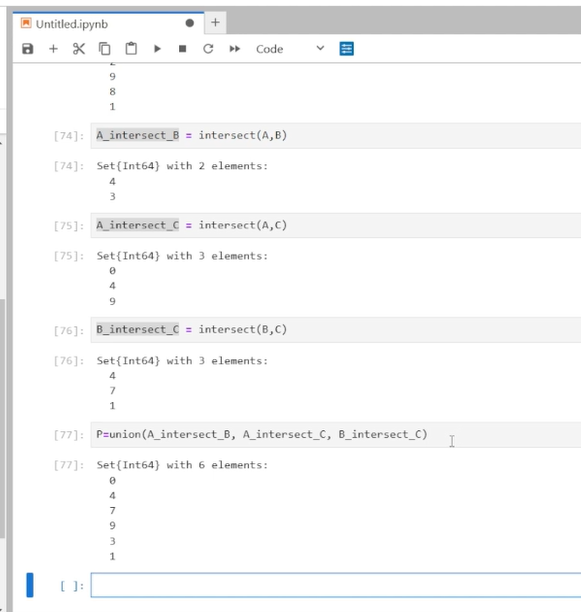


Рис. 4: задание 1

## 2.7 2. Приведите свои примеры с выполнением операций над множествами элементов разных типов

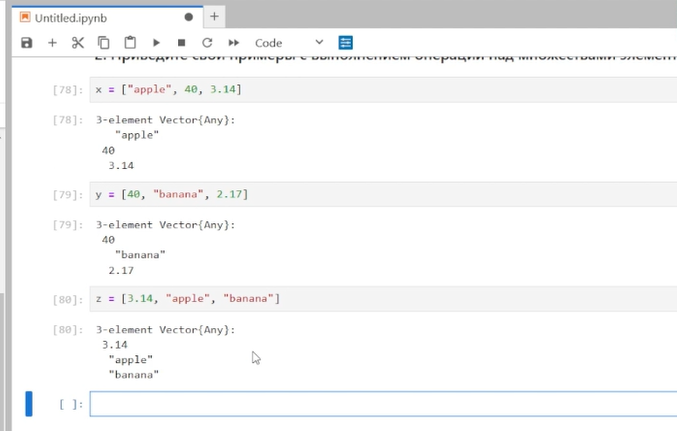


Рис. 5: задание 2



Рис. 6: задание 2

## 2.8 3. Создайте разными способами:

## 2.9 3.1) массив (1, 2, 3, … 𝑁 − 1, 𝑁 ), 𝑁 выберите больше 20;

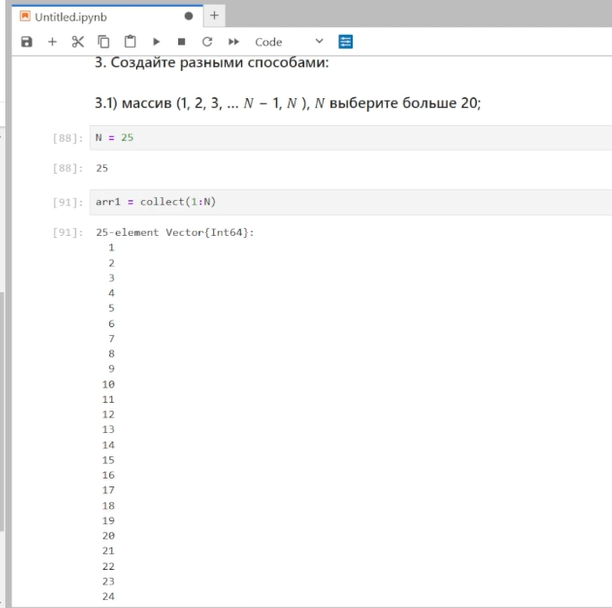


Рис. 7: задание 3.1

## 2.10 3.2) массив (𝑁, 𝑁 − 1 … , 2, 1), 𝑁 выберите больше 20;

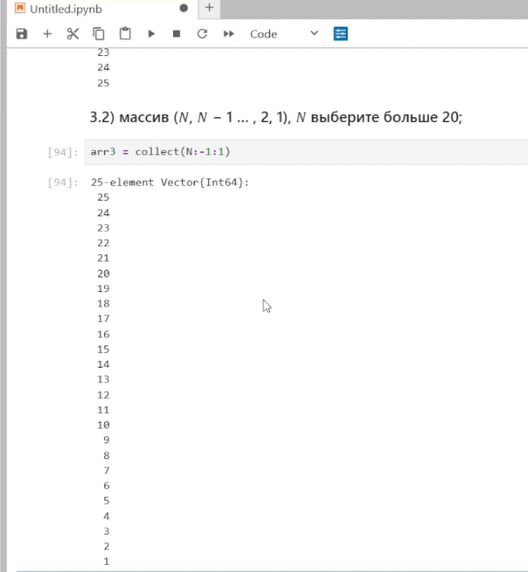


Рис. 8: задание 3.2

## 2.11 3.3) массив (1, 2, 3, … , 𝑁 − 1, 𝑁, 𝑁 − 1, … , 2, 1), 𝑁 выберите больше 20;

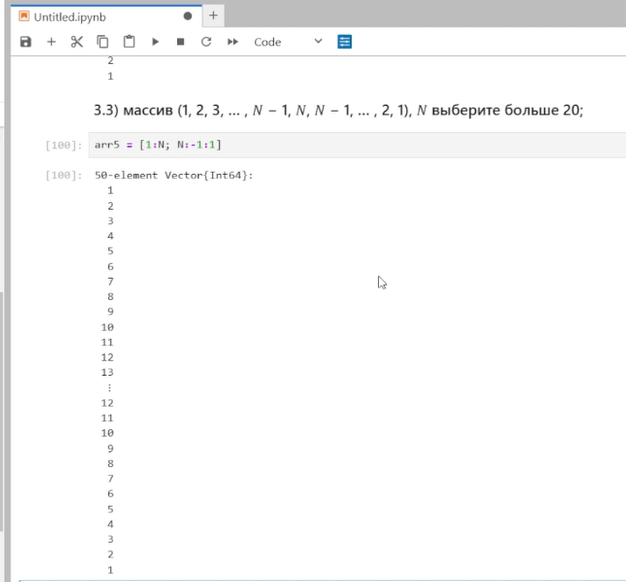


Рис. 9: задание 3.3

## 2.12 3.4) массив с именем tmp вида (4, 6, 3);

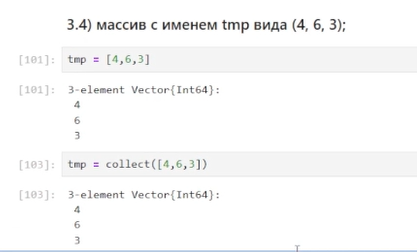


Рис. 10: задание 3.4

## 2.13 3.5) массив, в котором первый элемент массива tmp повторяется 10 раз;

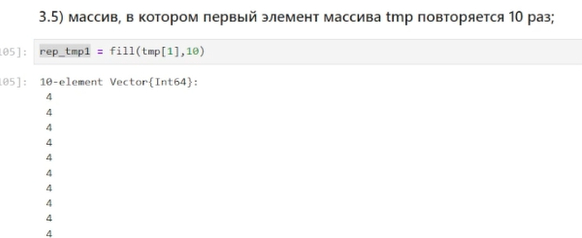


Рис. 11: задание 3.5

(Заданий слишком много в пункте 3, поэтому все остальные задания [находятся тут](https://github.com/ddshapovalova/study_2024-2025_computer-practice/blob/master/labs/lab2/lab2.ipynb) )

## 2.14 5. Подключите пакет Primes (функции для вычисления простых чисел). Сгенерируйте массив myprimes, в котором будут храниться первые 168 простых чисел. Определите 89-е наименьшее простое число. Получите срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа.

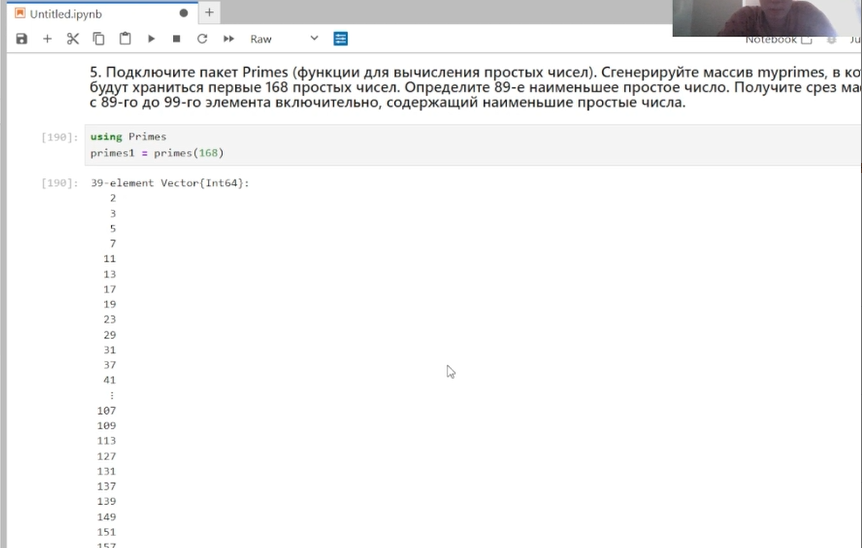


Рис. 12: задание 5

## 2.15 6. Вычислите следующие выражения

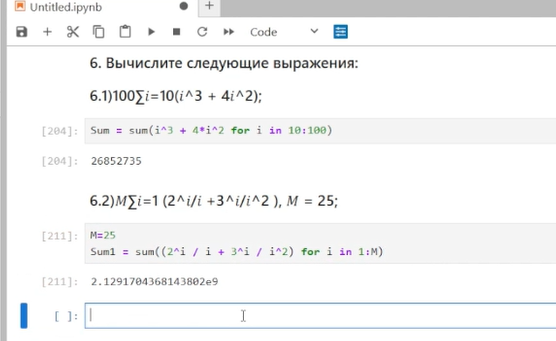


Рис. 13: задание 6

# 3 Выводы

Мы изучили несколько структур данных, реализованных в Julia, и научились применять их и операции над ними для решения задач