

# Компьютерный практикум по статистическому анализу данных. Лаб №2

Структуры данных.

---

Шаповалова Диана Дмитриевна

19 ноября 2024

Российский университет дружбы народов, Москва, Россия

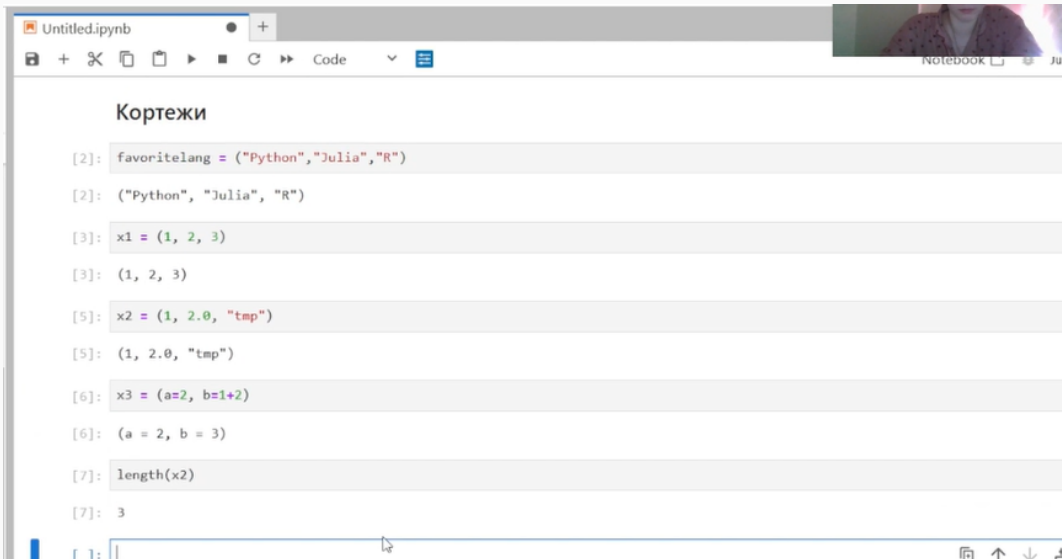
## Вводная часть

---

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

## Выполнение лабораторной работы

---



```
Untitled.ipynb
```

Кортежи

```
[2]: favoritelang = ("Python", "Julia", "R")
```

```
[2]: ("Python", "Julia", "R")
```

```
[3]: x1 = (1, 2, 3)
```

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

```
[5]: x2 = (1, 2.0, "tmp")
```

```
[5]: (1, 2.0, "tmp")
```

```
[6]: x3 = (a=2, b=1+2)
```

```
[6]: (a = 2, b = 3)
```

```
[7]: length(x2)
```

```
[7]: 3
```

```
[ ]:
```

```
Untitled.ipynb
[13]: phonebook = Dict("Иванов И.И." => ("867-5309", "333-5544"), "Бухгалтерия" => "555-2368")
[13]: Dict{String, Any} with 2 entries:
      "Бухгалтерия" => "555-2368"
      "Иванов И.И." => ("867-5309", "333-5544")
[14]: keys(phonebook)
[14]: KeySet for a Dict{String, Any} with 2 entries. Keys:
      "Бухгалтерия"
      "Иванов И.И."
[15]: values(phonebook)
[15]: ValueIterator for a Dict{String, Any} with 2 entries. Values:
      "555-2368"
      ("867-5309", "333-5544")
[16]: pairs(phonebook)
[16]: Dict{String, Any} with 2 entries:
      "Бухгалтерия" => "555-2368"
      "Иванов И.И." => ("867-5309", "333-5544")
[17]: haskey(phonebook, "Иванов И.И.")
[17]: true
[18]: phonebook["Сидоров П.С."] = "555-3344"
[18]: "555-3344"
```

```
Untitled.ipynb
```

```
b = Dict{"baz" => 17, "bar" => 13.0};  
merge(a, b), merge(b,a)
```

```
[20]: (Dict{String, Real}{"bar" => 13.0, "baz" => 17, "foo" => 0.0}, Dict{String, Real}{"
```

### Множества

```
[21]: A = Set{[1, 3, 4, 5]}
```

```
[21]: Set{Int64} with 4 elements:  
5  
4  
3  
1
```

```
[22]: B = Set{"abracadabra"}
```

```
[22]: Set{Char} with 5 elements:  
'a'  
'd'  
'r'  
'k'  
'b'
```

```
[23]: S1 = Set{[1,2]};  
      S2 = Set{[3,4]};  
      issetequal(S1,S2)
```

```
[23]: false
```

```
[ ]:
```

## Задания для самостоятельного выполнения

---



1. Даны множества:  $A = \{0, 3, 4, 9\}$ ,  $B = \{1, 3, 4, 7\}$ ,  $C = \{0, 1, 2, 4, 7, 8, 9\}$ . Найти  $A \cap B \cup A \cap C \cup B \cap C$ .

```
Untitled.ipynb
+
[74]: A_intersect_B = intersect(A,B)
[74]: Set(Int64) with 2 elements:
      4
      3

[75]: A_intersect_C = intersect(A,C)
[75]: Set(Int64) with 3 elements:
      0
      4
      9

[76]: B_intersect_C = intersect(B,C)
[76]: Set(Int64) with 3 elements:
      4
      7
      1

[77]: P=union(A_intersect_B, A_intersect_C, B_intersect_C)
[77]: Set(Int64) with 6 elements:
      0
      4
      7
      9
      3
      1
```

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

Untitled.ipynb



```
[78]: x = ["apple", 40, 3.14]
```

```
[78]: 3-element Vector{Any}:  
      "apple"  
      40  
      3.14
```

```
[79]: y = [40, "banana", 2.17]
```

```
[79]: 3-element Vector{Any}:  
      40  
      "banana"  
      2.17
```

```
[80]: z = [3.14, "apple", "banana"]
```

```
[80]: 3-element Vector{Any}:  
      3.14  
      "apple"  
      "banana"
```

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

```
[81]: # разность множеств

[82]: x_minus_y = setdiff(x,y)

[82]: 2-element Vector{Any}:
      "apple"
      3.14

[83]: x_minus_z = setdiff(x,z)

[83]: 1-element Vector{Any}:
      40

[84]: z_minus_y = setdiff(z,y)

[84]: 2-element Vector{Any}:
      3.14
```

3. Создайте разными способами: 3.1) массив  $(1, 2, 3, \dots, N - 1, N)$ ,  $N$  выберите больше 20;

```
Untitled.ipynb
[88]: N = 25
[88]: 25
[91]: arr1 = collect(1:N)
[91]: 25-element Vector{Int64}:
      1
      2
      3
      4
      5
      6
      7
      8
      9
     10
     11
     12
     13
     14
     15
     16
     17
     18
     19
     20
     21
     22
     23
     24
```

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

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

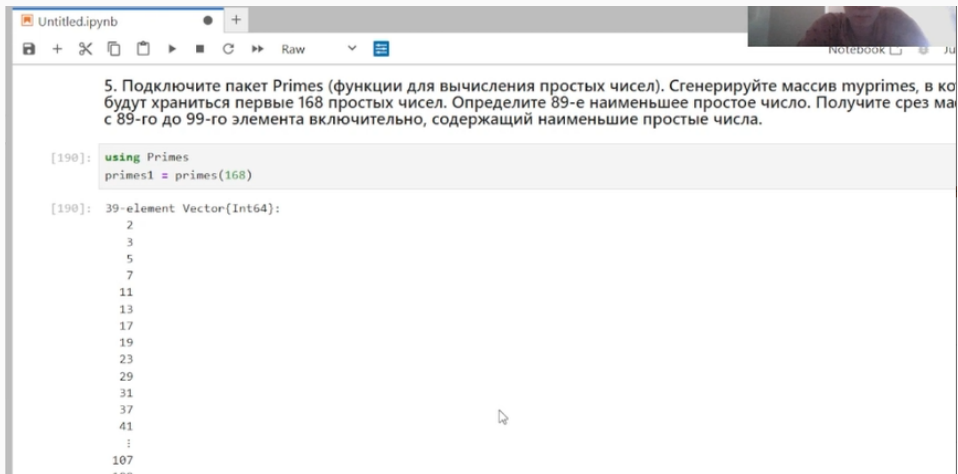
```
[05]: rep_tmp1 = fill(tmp[1],10)
```

```
[05]: 10-element Vector{Int64}:
```

```
 4
 4
 4
 4
 4
 4
 4
 4
 4
 4
 4
```

Рис. 8: задание 3.5

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.



The screenshot shows a Jupyter Notebook window titled "Untitled.ipynb". The interface includes a toolbar with icons for saving, adding cells, deleting, copying, pasting, and running code. The main area contains a text cell with the task description and a code cell with the following code:

```
[190]: using Primes
       primes1 = primes(168)
```

Below the code cell, the output is displayed as a 39-element vector of Int64 values, showing the first 39 prime numbers:

```
[190]: 39-element Vector{Int64}:
       2
       3
       5
       7
      11
      13
      17
      19
      23
      29
      31
      37
      41
       ⋮
     107
      ...
```

## 6. Вычислите следующие выражения

Untitled.ipynb

Save + Copy Paste Run Cell Code ▾

6. Вычислите следующие выражения:

6.1)  $100 \sum_{i=10} (i^3 + 4i^2)$ ;

```
[204]: Sum = sum(i^3 + 4*i^2 for i in 10:100)
```

```
[204]: 26852735
```

6.2)  $M \sum_{i=1} (2^{i/i} + 3^{i/i^2})$ ,  $M = 25$ ;

```
[211]: M=25
Sum1 = sum((2^i / i + 3^i / i^2) for i in 1:M)
```

```
[211]: 2.1291704368143802e9
```

```
[ ]: |
```

## Выводы

---



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