

Лабораторная работа №2

Дисциплина: Компьютерный практикум по статистическому анализу данных

Манаева Варвара Евгеньевна.

18 ноября 2023

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

Изучить несколько структур данных, реализованных в Julia, научиться применять их и операции над ними для решения задач.

1. Используя Jupyter Lab, повторите примеры из раздела 2.2.
2. Выполните задания для самостоятельной работы (раздел 2.4).

Выполнение лабораторной работы

Повторение примеров

Повторение примеров

1. Реализация кортежей
2. Реализация словарей
3. Реализация множеств
4. Реализация массивов

Реализация кортежей

```
[1]: # создать словарь с именем phonebook:  
phonebook = Dict("Иванов И.И." => ("867-5309", "333-5544"), "Бухгалтерия" => "555-2368")  
phonebook
```

```
[1]: Dict{String, Any} with 2 entries:  
  "Бухгалтерия" => "555-2368"  
  "Иванов И.И." => ("867-5309", "333-5544")
```

```
[2]: # вывести ключи словаря:  
keys(phonebook)
```

```
[2]: KeySet for a Dict{String, Any} with 2 entries. Keys:  
  "Бухгалтерия"  
  "Иванов И.И."
```

```
[3]: # вывести значения элементов словаря:  
values(phonebook)
```

```
[3]: ValueIterator for a Dict{String, Any} with 2 entries. Values:  
  "555-2368"  
  ("867-5309", "333-5544")
```

```
[4]: # вывести заданные в словаре пары "ключ - значение":  
pairs(phonebook)
```

```
[4]: Dict{String, Any} with 2 entries:  
  "Бухгалтерия" => "555-2368"  
  "Иванов И.И." => ("867-5309", "333-5544")
```

```
[5]: # проверка вхождения ключа в словарь:  
haskey(phonebook, "Иванов И.И.")
```

Самостоятельная работа

1. Пересечение и объединение множеств $A = \{0, 3, 4, 9\}$, $B = \{1, 3, 4, 7\}$ и $C = \{0, 1, 2, 4, 7, 8, 9\}$

```
[93]: function peresech(x, y)
      z = []
      for i in x
        for j in y
          if i == j
            append!(z, i)
          end
        end
      end
      return z
end

function obyedin(x, y...)
  z = x
  for i in y
    for j in i
      if findfirst(isequal(j), z) == nothing
        append!(z, j)
      end
    end
  end
  return sort(z)
end

A = [0, 3, 4, 9]
B = [1, 3, 4, 7]
C = [0, 1, 2, 4, 7, 8, 9]
peresech(A,B), obyedin(A,B)
```

```
[93]: (Any[3, 4], [0, 1, 3, 4, 7, 9])
```

Вычисляем формулу из задания: $P = (A \cap B) \cup (A \cap C) \cup (B \cap C)$ (разделена скобками для упрощения восприятия)

```
[94]: t1 = peresech(A,B)
      t2 = peresech(A,C)
      t3 = peresech(B,C)
      P = obyedin(t1,t2,t3)
```

```
[94]: 6-element Vector{Any}:
      0
      1
      3
      4
      7
```

2. Примеры с выполнением операций над множествами элементов разных типов (1)

2. Примеры с выполнением операций над множествами элементов разных типов

```
[63]: t = Set([1 2 3; 4 5 6; "By" "Ka" "Бака"])
```

```
[63]: Set{Any} with 9 elements:  
5  
4  
"Ka"  
6  
2  
"Бака"  
"By"  
3  
1
```

```
[64]: 1 in t
```

```
[64]: true
```

```
[65]: 0 in t
```

```
[65]: false
```

```
[66]: [1, 2, 3] in t
```

```
[66]: false
```

```
[67]: intersect(Set([1, 2, 3, 4, 7, 8, 10]), t)
```

```
[67]: Set{Any} with 4 elements:  
4  
2  
3  
1
```

```
[68]: union(Set([1, 2, 3, 4, 7, 8, 10]), t)
```

```
[68]: Set{Any} with 12 elements:  
5  
"Ka"  
7  
"Бака"  
8  
"By"  
1  
4
```

2. Примеры с выполнением операций над множествами элементов разных типов (2)

```
[69]: setdiff(Set([1, 2, 3, 4, 7, 8, 10]), t)
[69]: Set{Int64} with 3 elements:
      7
      10
      8

[70]: setdiff(t, Set([1, 2, 3, 4, 7, 8, 10]))
[70]: Set{Any} with 5 elements:
      5
      "Ka"
      6
      "Бака"
      "By"
```

Рис. 3: Примеры выполнения операций

3. Создать разными способами несколько видов массивов

1. массив $(1, 2, 3, \dots, N-1, N)$. N выберите больше 20

```
[70]: N = 100
      t = zeros(Int64, N)
      for i in 1:N
          t[i] = i
      end
      t
```

```
[70]: 100-element Vector{Int64}:
       1
       2
       3
       4
       5
       6
       7
       8
       9
      10
      11
      12
      13
       ⋮
      89
      90
      91
      92
      93
      94
      95
      96
      97
      98
      99
     100
```

```
[71]: t = collect(1:N)
      t
```

```
[71]: 100-element Vector{Int64}:
       1
       2
       3
       4
       5
       6
       7
       ⋮
```

4. Создать массив квадратов натуральных чисел от 1 до 100

4. Создать массив квадратов натуральных чисел от 1 до 100

```
[153]: t = [i^2 for i in 1:100]
```

```
[153]: 100-element Vector{Int64}:
```

```
 1  
 4  
 9  
16  
25  
36  
49  
64  
81  
100  
121  
144  
169  
⋮  
7921  
8100  
8281  
8464  
8649  
8836  
9025  
9216  
9409  
9604  
9801  
10000
```

Рис. 5: Массив натуральных чисел

5. С помощью пакета `Primes` сгенерировать массив первых 168 простых чисел, определим 89-е простое число, создадим срез массива с 89-го по 99-й элементов

5. С помощью пакета `Primes` сгенерировать массив первых 168 простых чисел, определим 89-е простое число, создадим срез массива с 89-го по 99-й элементов.

```
[156]: using Primes  
t = [prime(i) for i in 1:168]
```

```
[156]: 168-element Vector{Int64}:
```

```
 2  
 3  
 5  
 7  
11  
13  
17  
19  
23  
29  
31  
37  
41  
:  
:  
919  
929  
937  
941  
947  
953  
967  
971  
977  
983  
991  
997
```

```
[159]: t[89], t[89:99]
```

```
[159]: (461, [461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523])
```

Рис. 6: Работа с простыми числами

6. Реализовать формулы (1)

6. Реализовать формулы

- $\sum_{i=10}^{100} (i^3 + 4i^2)$;
- $\sum_{i=1}^M (\frac{2^i}{i} + \frac{3^i}{i^2})$, $M = 25$;
- $1 + \frac{2}{3} + (\frac{2}{3}\frac{4}{5}) + \dots + (\frac{2}{3}\frac{4}{5} \dots \frac{38}{39}) = 1 + \sum_{i=1}^{19} \prod_{j=1}^i \frac{2j}{2j+1}$

```
[130]: t = [i^3+4*i^2 for i in 10:100]
```

```
[130]: 91-element Vector{Int64}:
```

```
1400
1815
2304
2873
3528
4275
5120
6069
7128
8303
9600
11025
12584
⋮
736653
761400
786695
812544
838953
865928
893475
921600
950309
979608
1009503
1040000
```

```
[131]: sum(t)
```

```
[131]: 26852735
```

```
[132]: M = 25
```

```
      t = [2^i/i + 3^i/(i^2) for i in 1:M]
```

```
[132]: 25-element Vector{Float64}:
```

```
5.0
4.25
5.6666666666666666
```

6. Реализовать формулы (2)

```
299.8888888888889
692.89
1650.206611570248
4031.8958333333335
10064.01775147929
25573.188775510203
65957.45333333334
172247.25390625
454561.89273356396
1.2103058055555555e6
3.24715495567867e6
8.7693898025e6
2.3819486156462584e7
6.502755020041322e7
1.7832914331001893e8
4.910281070572917e8
1.3570039523888e9
```

```
[133]: sum(t)
```

```
[133]: 2.1291704368143802e9
```

```
[134]: t = cat([1], [prod([2j/(2j+1) for j in 1:i]) for i in 1:19], dims=1)
```

```
[134]: 20-element Vector{Float64}:
```

```
1.0
0.6666666666666666
0.5333333333333333
0.45714285714285713
0.4063492063492063
0.36940836940836935
0.34099234099234094
0.3182595182595182
0.2995383701266054
0.2837731927515209
0.27026018357287707
0.25850974080883893
0.24816935117648536
0.2389778937255044
0.23073727670048702
0.2232941387424068
0.2165276496896066
0.2103411454127607
0.20465624959079418
0.19940865344744046
```

```
[135]: sum(t)
```

```
[135]: 6.976346137897619
```


Выводы по проделанной работе

В результате выполнения работы мы изучили несколько структур данных, реализованных в Julia, и научились применять их и операции над ними для решения задач.

Были записаны скринкасты выполнения и защиты лабораторной работы.