

Лабораторная работа №2.

Структуры данных

Тазаева А. А.

Российский университет дружбы народов, Москва, Россия

Цели работы

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

Задание

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

Кортежи

Примеры кортежей

```
[3]: () # Пустой кортеж
```

```
[3]: ()
```

```
[5]: favoritelang = ("python", "julia", "R") #Кортеж из элементов типа String
```

```
[5]: ("python", "julia", "R")
```

```
[13]: x1 = (1, 2, 3) # кортеж из целых чисел
```

```
[13]: (1, 2, 3)
```

```
[14]: x2 = (1, 2.0, "tmp") # Кортеж из элементов разных типов
```

```
[14]: (1, 2.0, "tmp")
```

```
[15]: x3 = (a=2, b=1+2) # Именованный кортеж
```

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

Рис. 1: Примеры кортежей

Примеры операций над кортежами

```
[16]: length(x2) # длина кортежа x2
```

```
[16]: 3
```

```
[17]: x2[1], x2[2], x2[3] # обратиться к элементам кортежа x2
```

```
[17]: (1, 2.0, "tmp")
```

```
[18]: c = x1[2] + x1[3] # произвести какую-либо операцию (сложение)  
# с вторым и третьим элементами кортежа x1
```

```
[18]: 5
```

```
[19]: x3.a, x3.b, x3[2] # обращение к элементам именованного кортежа x3
```

```
[19]: (2, 3, 3)
```

```
[21]: # проверка вхождения элементов tmp и 0 в кортеж x2  
# (два способа обращения к методу in()):  
in("tmp", x2), 0 in x2
```

```
[21]: (true, false)
```

Рис. 2: Примеры операций над кортежами

Примеры словарей и операций над ними

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

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

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

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

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

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

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

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

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

```
[29]: true
```


Примеры множеств и операций над ними

```
[34]: # создать множество из четырёх целочисленных значений:  
A = Set([1, 3, 4, 5])
```

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

```
[35]: # создать множество из 11 символьных значений:  
B = Set("abracadabra")
```

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

```
[40]: # проверка эквивалентности двух множеств:  
S1 = Set([1,2]);  
S2 = Set([3,4]);  
issetequal(S1,S2)
```

```
[40]: false
```

```
[41]: # проверка эквивалентности двух множеств:  
S3 = Set([1,2,2,3,1,2,3,2,1]);  
S4 = Set([2,3,1]);  
issetequal(S3,S4)
```

```
[41]: true
```

```
[42]: # объединение множеств:  
C=union(S1,S2)
```

```
[42]: Set{Int64} with 4 elements:  
4  
2  
3  
1
```

```
[43]: # пересечение множеств:  
D = intersect(S1,S3)
```

```
[43]: Set{Int64} with 2 elements:  
2  
1
```

```
[44]: # разность множеств:  
E = setdiff(S3,S1)
```

```
[44]: Set{Int64} with 1 element:  
3
```

```
[45]: # проверка вхождения элементов одного множества в другое:  
issubset(S1,S4)
```

```
[45]: true
```

```
[46]: # добавление элемента в множество:  
push!(S4, 99)
```

```
[46]: Set{Int64} with 4 elements:  
2  
99  
3  
1
```

Примеры массивов

[49]: *# создание пустого массива с абстрактным типом:*

```
empty_array_1 = []
```

[49]: Any[]

[52]: *# создание пустого массива с конкретным типом:*

```
empty_array_2 = (Int64)[]
```

[52]: Int64[]

[51]: empty_array_3 = (Float64)[]

[51]: Float64[]

[53]: *# вектор-столбец:*

```
a = [1, 2, 3]
```

[53]: 3-element Vector{Int64}:

```
1  
2  
3
```

[55]: *# вектор-строка:*

```
b = [1 2 3]
```

[55]: 1×3 Matrix{Int64}:

```
1 2 3
```

[57]: *# многомерные массивы (матрицы):*

```
A = [[1, 2, 3] [4, 5, 6] [7, 8, 9]]
```

[57]: 3×3 Matrix{Int64}:

```
1 4 7  
2 5 8  
3 6 9
```

Некоторые операции для работы с массивами

```
[71]: # одномерный массив из пяти единиц:  
ones(5)
```

```
[71]: 5-element Vector{Float64}:  
 1.0  
 1.0  
 1.0  
 1.0  
 1.0
```

```
[73]: # двумерный массив 2x3 из единиц:  
ones(2,3)
```

```
[73]: 2x3 Matrix{Float64}:  
 1.0  1.0  1.0  
 1.0  1.0  1.0
```

```
[74]: # одномерный массив из 4 нулей:  
zeros(4)
```

```
[74]: 4-element Vector{Float64}:  
 0.0  
 0.0  
 0.0  
 0.0
```

```
[75]: # заполнить массив 3x2 цифрами 3.5  
fill(3.5,(3,2))
```

```
[75]: 3x2 Matrix{Float64}:  
 3.5  3.5  
 3.5  3.5  
 3.5  3.5
```

Самостоятельная работа

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

```
A = Set([0, 3, 4, 9])
B = Set([1, 3, 4, 7])
C = Set([0, 1, 2, 4, 7, 8, 9])
P = union(intersect(A,B), intersect(A,B),intersect(A,C),intersect(B,C))
print(P)

Set([0, 4, 7, 9, 3, 1])
```

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

```
N = Set([2002, 2003, 2004, 2005]) # выпускники кружка Оленьи рога
T = Set([2002, 2000, 2001, 2005, 1998, 1997, 1930, 2009, 1992, 1993, 2003, 2004]) # выпускники школы творчества
# входят ли выпускники кружка Оленьи рога в список выпускников школы творчества
issubset(N,T)

true

N = Set([2002, 2003, 2004, 2005]) # выпускники кружка Оленьи рога
S = Set([2000, 1993, 1994, 1995]) # выпускники кружка Оленьи копыта
R = Set([1930, 2009, 2004, 2005]) # выпускники кружка Оленьи хвостики
T = union(N,S,R) # выпускники школы творчества по годам
print(T)

Set([2004, 2005, 1995, 2002, 1930, 1994, 2003, 1993, 2009, 2000])
```

Рис. 8: Самостоятельная работа. Задание 1 и 2

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

```
N = 29
Arr1 = collect(1:1:N)
print(Arr1)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
```

3.2) массив $(N, N-1, \dots, 2, 1)$, N выберите больше 20;

```
N = 29
Arr2 = collect(N:-1:1)
print(Arr2)
```

```
[29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

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

```
N = 20
Arr3 = cat(collect(1:1:N), collect(N:-1:1), dims=1)
print(Arr3)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

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

```
tmp = [4, 6, 3]
print("tmp = ", tmp)

tmp = [4, 6, 3]
```

Рис. 9: Самостоятельная работа. Задание 3. Подпункты 3.1, 3.2, 3.3, 3.4

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

```
Arr3_5 = fill(tmp[1], 10)
print(Arr3_5)
```

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4]

3.6) массив, в котором все элементы массива tmp повторяются 10 раз;

```
Arr3_6 = repeat(tmp, 10)
print(Arr3_6)
```

[4, 6, 3, 4, 6, 3, 4, 6, 3, 4, 6, 3, 4, 6, 3, 4, 6, 3, 4, 6, 3, 4, 6, 3, 4, 6, 3, 4, 6, 3]

3.7) массив, в котором первый элемент массива `tmp` встречается 11 раз, второй элемент — 10 раз, третий элемент — 10 раз;

```
Arr3_7 = cat(fill(tmp[1],11),fill(tmp[2],10),fill(tmp[3],10), dins = 1)
print(Arr3_7)
```

[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

3.8) массив, в котором первый элемент массива tmp встречается 10 раз подряд, второй элемент — 20 раз подряд, третий элемент — 30 раз подряд;

```
Arr3_8 = cat(fill(tmp[1],10),fill(tmp[2],20),fill(tmp[3],30), dins = 1)
print(Arr3_8)
```

[illegible]

Рис. 10: Самостоятельная работа. Задание 3. Подпункты 3.5, 3.6, 3.7, 3.8

Выводы по проделанной работе

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