

Лабораторная работа № 2

Структуры данных

Шияпова Д.И.

25 сентября 2025

Российский университет дружбы народов, Москва, Россия

- Шияпова Дарина Илдаровна
- Студентка
- Российский университет дружбы народов
- 1132226458@pfur.ru



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

[5]:

```
# кортеж из элементов типа String:  
favoritelang = ("Python", "Julia", "R")
```

[5]:

```
("Python", "Julia", "R")
```

[7]:

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

[7]:

```
(1, 2, 3)
```

[9]:

```
# кортеж из элементов разных типов:  
x2 = (1, 2.0, "tmp")
```

[9]:

```
(1, 2.0, "tmp")
```

```
# обратиться к элементам кортежа x2:
```

```
x2[1]
```

```
[17]:
```

```
1
```

```
[19]:
```

```
x2[2]
```

```
[19]:
```

```
2.0
```

```
[23]:
```

```
c = x1[2] + x1[3]
```

```
[23]:
```

```
5
```

Рис. 2: Примеры использования кортежей

[29]:

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

[29]:

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

[31]:

```
# вывести ключи словаря:  
keys(phonebook)
```

[31]:

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

[*]:

Рис. 3: Примеры использования словарей

Выполнение лабораторной работы

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

```
[35]:
```

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

```
[41]:
```

```
haskey(phonebook, "Иванов И.И.")
```

```
[41]:
```

```
true
```

```
[43]:
```

```
# добавить элемент в словарь:  
phonebook["Сидоров П.С."] = "555-3344"
```

```
[43]:
```

```
"555-3344"
```

Рис. 4: Примеры использования словарей

[69]:

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

[69]:

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

[71]:

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

[71]:

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

Рис. 5: Примеры использования множеств

[79]:

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

[79]:

false

[81]:

```
S3 = Set([1,2,2,3,1,2,3,2,1]);  
S4 = Set([2,3,1]);  
issetequal(S3,S4)
```

[81]:

true

[83]:

Рис. 6: Примеры использования множеств

Выполнение лабораторной работы

```
In [112.] # создание пустого массива с абстрактным типом:
empty_array_1 = []

Out[112.] Any[]

In [114.] # создание пустого массива с конкретным типом:
empty_array_2 = (Int64[])
empty_array_3 = (Float64[])

Out[114.] Float64[]

In [116.] # Вектор-столбец:
a = [1, 2, 3]

Out[116.] 3-element Vector{Int64}:
 1
 2
 3

In [119.] # Вектор-строка:
b = [1 2 3]

Out[119.] 1×3 Matrix{Int64}:
 1  2  3

In [121.] # многомерные массивы (матрицы):
A = [[1, 2, 3] [4, 5, 6] [7, 8, 9]]
B = [[1 2 3]; [4 5 6]; [7 8 9]]

Out[121.] 3×3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

In [123.] A

Out[123.] 3×3 Matrix{Int64}:
 1  4  7
 2  5  8
 3  6  9

In [125.] B

Out[125.] 3×3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

In [144.] # одномерный массив из 8 элементов (массив $1 \times 8$)
# со значениями, случайно распределёнными на интервале [0, 1]:
c = rand(1,8)

Out[144.] 1×8 Matrix{Float64}:
 0.796904  0.960987  0.465085  0.69098  ...  0.986977  0.775043  0.465763

In [146.] # многомерный массив $2 \times 3$ (2 строки, 3 столбца) элементов
# со значениями, случайно распределёнными на интервале [0, 1]:
C = rand(2,3);

In [148.] C
```

Выполнение лабораторной работы

```
In [150.] # трёхмерный массив:
D = rand(4, 3, 2)

Out[150.] 4x3x2 Array{Float64, 3}:
[:, :, 1] =
 0.713405  0.432203  0.404381
 0.109381  0.536658  0.921419
 0.186919  0.730975  0.475927
 0.216667  0.261629  0.63092

[:, :, 2] =
 0.0173873  0.26553  0.181904
 0.5598  0.439747  0.789622
 0.276212  0.536542  0.465716
 0.104413  0.577015  0.31033

In [154.] # массив из квадратных корней всех целых чисел от 1 до 10:
roots = [sqrt(i) for i in 1:10]

Out[154.] 10-element Vector{Float64}:
 1.0
 1.4142135623730951
 1.7320508075688772
 2.0
 2.23606797749979
 2.449489742783178
 2.6457513110645907
 2.8284271247461903
 3.0
 3.1622776601683795

In [156.] # массив с элементами вида 3^i*x^2,
# где x - нечётное число от 1 до 9 (включительно)
ar_1 = [3^i*2 for i in 1:2:9]

Out[156.] 5-element Vector{Int64}:
 3
 27
 75
 147
 243

In [160.] # массив квадратов элементов, если квадрат не делится на 5 или 4:
ar_2=[i^2 for i=1:10 if (i^2%5!=0 && i^2%4!=0)]

Out[160.] 4-element Vector{Int64}:
 1
 9
 49
 81

In [167.] # одномерный массив из пяти единиц:
ones(5)

Out[167.] 5-element Vector{Float64}:
 1.0
 1.0
 1.0
 1.0
 1.0
```

Выполнение лабораторной работы

```
In [171.. # одномерный массив из 4 нулей:
zeros(4)

Out[171.. 4-element Vector{Float64}:
 0.0
 0.0
 0.0
 0.0

In [173.. # заполнить массив 3х2 цифрами 3.5
fill(3.5,(3,2))

Out[173.. 3x2 Matrix{Float64}:
 3.5 3.5
 3.5 3.5
 3.5 3.5

In [183.. # заполнение массива посредством функции repeat():
repeat([1,2],3,3)
repeat([1 2],3,3)

Out[183.. 3x6 Matrix{Int64}:
 1 2 1 2 1 2
 1 2 1 2 1 2
 1 2 1 2 1 2

In [185.. # преобразование одномерного массива из целых чисел от 1 до 12
# в 6-двумерный массив 2х6
a = collect(1:12)
b = reshape(a,(2,6))

Out[185.. 2x6 Matrix{Int64}:
 1 3 5 7 9 11
 2 4 6 8 10 12

In [187.. # транспонирование
b'

Out[187.. 6x2 adjoint{::Matrix{Int64}} with eltype Int64:
 1 2
 3 4
 5 6
 7 8
 9 10
 11 12

In [189.. # транспонирование
c = transpose(b)

Out[189.. 6x2 transpose{::Matrix{Int64}} with eltype Int64:
 1 2
 3 4
 5 6
 7 8
 9 10
 11 12

In [194.. # массив 10х5 целых чисел в диапазоне [10, 20]:
ar = rand(10:20, 10, 5)

Out[194.. 10x5 Matrix{Int64}:
 18 11 15 12 19
 18 12 20 19 18
 16 14 19 11 15
```

№1

```
In [233... a = Set([0, 3, 4, 9]); b = Set([1, 3, 4, 7]); c = Set([0, 1, 2, 4, 7, 8, 9]);
```

```
In [235... union(intersect(a, b), intersect(a, c), intersect(b, c))
```

```
Out[235... Set{Int64} with 6 elements:
```

```
0  
4  
7  
9  
3  
1
```

Рис. 10: Задание №1. Работа с множествами

```
In [237.. s1 = Set([1, 2.3, "Privett"])

Out[237.. Set{Any} with 3 elements:
          2.3
          1
          "Privett"

In [241.. for i in s1
          println(i)
          end

          2.3
          1
          Privett

In [243.. in(1, s1)

Out[243.. true

In [247.. in(1.1, s1)

Out[247.. false

In [255.. s1 = push!(s1, "poka")

Out[255.. Set{Any} with 4 elements:
          "poka"
          2.3
          1
          "Privett"
```

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

Выполнение лабораторной работы

[illegible]

Рис. 12: Задание №3. Работа с массивами

```
In [339..  
# 3.9  
tt = []  
for i in 1:3  
    push!(tt, 2^tmp[i])  
end  
print(vcat(tt, repeat([tt[3]], 3)), "\n")  
k = 0  
for i in tt  
    if '6' in string(i)  
        k+=1  
    end  
end  
print(k)  
Any[16, 64, 0, 0, 0, 8]  
2  
  
In [347..  
# 3.10  
using Statistics  
y(x) = exp(x) * cos(x)  
y1 = [y(x) for x in 3:0.1:6]  
mean(y1)  
Out[347.. 53.11374594642971
```

Рис. 13: Задание №3. Работа с массивами

Выполнение лабораторной работы

[illegible]

Рис. 14: Задание №3. Работа с массивами

Выполнение лабораторной работы

[illegible]

Рис. 15: Задание №3. Работа с массивами

Выполнение лабораторной работы

```
In [105]: for i in range(1, 1000):
          j = 1
          while True:
              if j > 1000:
                  break
              if i % j == 0:
                  break
              j += 1
          print(i)
```

1000 991 983 977 971 967 961 955 951 947 941 937 931 927 921 917 911 907 901 897 891 887 881 877 871 867 861 857 851 847 841 837 831 827 821 817 811 807 801 797 791 787 781 777 771 767 761 757 751 747 741 737 731 727 721 717 711 707 701 697 691 687 681 677 671 667 661 657 651 647 641 637 631 627 621 617 611 607 601 597 591 587 581 577 571 567 561 557 551 547 541 537 531 527 521 517 511 507 501 497 491 487 481 477 471 467 461 457 451 447 441 437 431 427 421 417 411 407 401 397 391 387 381 377 371 367 361 357 351 347 341 337 331 327 321 317 311 307 301 297 291 287 281 277 271 267 261 257 251 247 241 237 231 227 221 217 211 207 201 197 191 187 181 177 171 167 161 157 151 147 141 137 131 127 121 117 111 107 101 97 91 87 81 77 71 67 61 57 51 47 41 37 31 27 21 17 11 7 5 4 3 2 1

In [106]: for i in range(1, 1000):
 j = 1
 while True:
 if j > 1000:
 break
 if i % j == 0:
 break
 j += 1
 print(i)

Рис. 16: Задание №3. Работа с векторами

```
In [388]: v3 = [sin(yy[i])/cos(x[i+1]) for i in 1:249]
```

```
Out[388]: 249-element Vector{Float64}:
```

```
0.2749317292348841
0.12762999037442216
-1.773155702778313
-2.9404149631430325
0.9556718292300051
1.6685243721888336
0.21172616989152088
0.27247590097310664
-7.109027655181126
-1.7209623006722765
0.6179721088796811
1.3358670493488118
2.8365690611460055
!
-1.0200199268206551
-1.095336990483831
-1.6340787522883482
-0.21801431199909593
2.3291327971867877
-1.3277493034642984
6.493293427584357
0.6338928291431104
1.021959685670216
0.8316520359898227
-0.6960368448325855
0.5675323400317531
```

Рис. 17: Задание №3. Работа с векторами

Выполнение лабораторной работы

[illegible]

Рис. 18: Задание №3. Работа с векторами

Выполнение лабораторной работы

[illegible]

Рис. 19: Задание №3. Работа с векторами

Выполнение лабораторной работы

```
2> [40]. [length(i) for i in yy if abs(maxsum(yy) - i) <= 0.002]
Out[40]: 82

2> [41]. print(countTrue for i in x if i % 2 == 0)
print("\n")
print(countTrue for i in x if i % 2 != 0)
138
139

2> [41]. print(countTrue for i in x if i % 2 == 0)
44

2> [42]. count_x = x[startpaw(yy)]
print(f"Получено: {x[startpaw(yy)]}, count_x")
Определение 1: def(x): 137, 181, 180, 236, 672, 951, 177, 124, 916, 259, 267, 989, 936, 67, 989, 656, 0, 989, 216, 951, 659, 269, 176, 652, 917, 126, 759, 216, 689, 127, 219, 959, 16, 976, 65, 959, 269, 922, 959, 711, 252, 689, 656, 687, 651, 912, 929, 127, 618, 618, 941, 759, 672, 818, 66, 941, 491, 496, 997, 61, 199, 759, 989, 499, 657, 262, 7, 195, 499, 259, 621, 629, 949, 126, 199, 762, 273, 761, 612, 669, 651, 761, 261, 681, 219, 969, 931, 119, 912, 109, 279, 989, 92, 761, 22, 611, 941, 216, 876, 689, 765, 269, 751, 989, 759, 629, 91, 679, 249, 239, 611, 24, 689, 212, 716, 989, 499, 941, 759, 989, 761, 657, 76, 199, 618, 759, 195, 876, 759, 109, 269, 191, 261, 656, 961, 976, 259, 989, 299, 194, 989, 681, 716, 989, 611, 659, 611, 959, 499, 95, 618, 618, 199, 77, 959, 687, 126, 109, 656, 751, 761, 989, 671, 65, 656, 959, 611, 989, 269, 212, 161, 761, 689, 652, 109, 912, 617, 989, 949, 989, 6, 1, 829, 677, 618, 247, 14, 981, 989, 144, 799, 132, 912, 912, 219, 219, 129, 678, 939, 499, 199, 989, 969, 212, 297, 989, 987, 189, 118, 44, 912, 911, 497, 976, 919, 189, 212, 689, 969, 759, 139, 497, 689, 269, 922, 717, 269, 124, 494, 989, 799, 779, 939, 941, 912, 911, 499, 652, 247, 124, 9, 76, 967, 951, 761, 917, 967, 776, 49, 769]
```

Рис. 20: Задание №3. Работа с векторами

```
In [432_ reverse(last(sort(x), 10))
```

```
Out[432_ 10-element Vector{Any}:
```

```
997
```

```
997
```

```
993
```

```
980
```

```
976
```

```
973
```

```
964
```

```
956
```

```
952
```

```
951
```

Рис. 21: Задание №3. Работа с векторами

[illegible]

Рис. 22: Задание №3. Работа с векторами

```
In [456_ using Primes  
myprime = primes(prime(168))
```

```
Out[456_ 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
```

```
In [459_ prime(89)

Out[459_ 461

In [461_ myprime1 = [prime(i) for i in 89:99]

Out[461_ 11-element Vector{Int64}:
         461
         463
         467
         479
         487
         491
         499
         503
         509
         521
         523
```

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

Выполнение лабораторной работы

```
In [473_ s = 0
for i in 10:100
    s = s + (i^3 + 4*i^2)
end
print(s)
```

26852735

```
In [475_ s = 0
for i in 1:25
    s = s + ((2^i)/(i) + (3^i)/(i^2))
end
print(s)
```

2.1291704368143802e9

```
In [477_ s = 1
a1 = 1
for i in 2:2:38
    a1 = a1 * i/(i+1)
    s = s + a1
end
print(s)
```

6.97634613789762

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