

Motyakin Artem Laboratornaya 1 var 15

1

Создано системой Doxygen 1.10.0

1 Алфавитный указатель пространств имен	1
1.1 Package List	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Список файлов	5
3.1 Файлы	5
4 Пространства имен	7
4.1 Пространство имен v15	7
4.1.1 Функции	8
4.1.1.1 partition()	8
4.1.1.2 QuickSort()	8
4.1.1.3 SelectSort()	8
4.1.1.4 ShakerSort()	8
4.1.2 Переменные	8
4.1.2.1 arr1	8
4.1.2.2 arr2	8
4.1.2.3 arr3	9
4.1.2.4 arr_names_man	9
4.1.2.5 arr_names_woman	9
4.1.2.6 arr_quick	9
4.1.2.7 arr_select	9
4.1.2.8 arr_shaker	9
4.1.2.9 arr_sizes	9
4.1.2.10 d	9
4.1.2.11 date_bride	9
4.1.2.12 date_groom	10
4.1.2.13 date_wedding	10
4.1.2.14 days_birthday	10
4.1.2.15 days_wedding	10
4.1.2.16 df	10
4.1.2.17 df1	10
4.1.2.18 df2	10
4.1.2.19 df3	10
4.1.2.20 end_date_birthday	10
4.1.2.21 end_date_wedding	10
4.1.2.22 label	11
4.1.2.23 name_bride	11
4.1.2.24 name_groom	11
4.1.2.25 num_registry	11
4.1.2.26 rotation	11
4.1.2.27 start_date_birthday	11

4.1.2.28 start_date_wedding	11
4.1.2.29 start_time	11
4.1.2.30 table	11
5 Классы	13
5.1 Класс Obj	13
5.1.1 Конструктор(ы)	13
5.1.1.1 __init__()	13
5.1.2 Методы	13
5.1.2.1 __ge__()	13
5.1.2.2 __gt__()	14
5.1.2.3 __le__()	14
5.1.2.4 __lt__()	14
5.1.3 Данные класса	14
5.1.3.1 date_w	14
5.1.3.2 gr_fname	14
5.1.3.3 num_reg	14
6 Файлы	15
6.1 Файл labal/v15.py	15
Предметный указатель	17

Глава 1

Алфавитный указатель пространств имен

1.1 Package List

Полный список документированных пакетов.

[v15](#) [7](#)

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Obj	13
-------------------------------	--------------------

Глава 3

Список файлов

3.1 Файлы

Полный список файлов.

laba1/ v15.py	15
---	----

Глава 4

Пространства имен

4.1 Пространство имен v15

Классы

- class [Obj](#)

Функции

- [partition](#) (arr, left, right)
- [QuickSort](#) (arr, left, right)
- [SelectSort](#) (arr)
- [ShakerSort](#) (arr)

Переменные

- [arr1](#) = df.to_numpy().tolist()[1:]
- [arr2](#) = df.to_numpy().tolist()[1:]
- [arr3](#) = df.to_numpy().tolist()[1:]
- [arr_names_man](#) = np.array([table](#)[0])
- [arr_names_woman](#) = np.array([table](#)[1])
- list [arr_quick](#) = []
- list [arr_select](#) = []
- list [arr_shaker](#) = []
- [arr_sizes](#) = np.array([100, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000])
- dict [d](#)
- list [date_bride](#) = []
- list [date_groom](#) = []
- list [date_wedding](#) = []
- tuple [days_birthday](#) = ([end_date_birthday](#) - [start_date_birthday](#)).days
- tuple [days_wedding](#) = ([end_date_wedding](#) - [start_date_wedding](#)).days
- [df](#) = pd.DataFrame(data=[d](#))
- [df1](#) = pd.DataFrame(data=[arr1](#))
- [df2](#) = pd.DataFrame(data=[arr2](#))
- [df3](#) = pd.DataFrame(data=[arr3](#))
- [end_date_birthday](#) = datetime.strptime("01-01-2000", "%d-%m-%Y")
- [end_date_wedding](#) = datetime.strptime("01-01-2024", "%d-%m-%Y")

- `label`
- `name_bride` = `np.random.choice(arr_names_woman, size=i)`
- `name_groom` = `np.random.choice(arr_names_man, size=i)`
- `num_registry` = `np.random.randint(1, high=i+1, size=i)`
- `rotation`
- `start_date_birthday` = `datetime.strptime("01-01-1980", "%d-%m-%Y")`
- `start_date_wedding` = `datetime.strptime("01-01-2018", "%d-%m-%Y")`
- `int start_time` = `time.time_ns() / 1000000`
- `table` = `pd.read_csv('Names.csv', sep=';', index_col=False, header=None)`

4.1.1 Функции

4.1.1.1 `partition()`

```
partition (
    arr,
    left,
    right )
```

4.1.1.2 `QuickSort()`

```
QuickSort (
    arr,
    left,
    right )
```

4.1.1.3 `SelectSort()`

```
SelectSort (
    arr )
```

4.1.1.4 `ShakerSort()`

```
ShakerSort (
    arr )
```

4.1.2 Переменные

4.1.2.1 `arr1`

```
arr1 = df.to_numpy().tolist()[1:]
```

4.1.2.2 `arr2`

```
arr2 = df.to_numpy().tolist()[1:]
```

4.1.2.3 arr3

```
arr3 = df.to_numpy().tolist()[1:]
```

4.1.2.4 arr_names_man

```
arr_names_man = np.array(table[0])
```

4.1.2.5 arr_names_woman

```
arr_names_woman = np.array(table[1])
```

4.1.2.6 arr_quick

```
arr_quick = []
```

4.1.2.7 arr_select

```
arr_select = []
```

4.1.2.8 arr_shaker

```
arr_shaker = []
```

4.1.2.9 arr_sizes

```
arr_sizes = np.array([100, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000])
```

4.1.2.10 d

```
dict d
```

Инициализатор

```
00001 = {"Groom fullname": name_groom,  
00002      "Groom birth date": np.array(date_groom),  
00003      "Bride fullname": name_bride,  
00004      "Bride birth date": np.array(date_bride),  
00005      "Wedding date": np.array(date_wedding),  
00006      "Registry office number": num_registry}
```

4.1.2.11 date_bride

```
list date_bride = []
```

4.1.2.12 date_groom

```
list date_groom = []
```

4.1.2.13 date_wedding

```
list date_wedding = []
```

4.1.2.14 days_birthday

```
tuple days_birthday = (end_date_birthday - start_date_birthday).days
```

4.1.2.15 days_wedding

```
tuple days_wedding = (end_date_wedding - start_date_wedding).days
```

4.1.2.16 df

```
df = pd.DataFrame(data=d)
```

4.1.2.17 df1

```
df1 = pd.DataFrame(data=arr1)
```

4.1.2.18 df2

```
df2 = pd.DataFrame(data=arr2)
```

4.1.2.19 df3

```
df3 = pd.DataFrame(data=arr3)
```

4.1.2.20 end_date_birthday

```
end_date_birthday = datetime.strptime("01-01-2000", "%d-%m-%Y")
```

4.1.2.21 end_date_wedding

```
end_date_wedding = datetime.strptime("01-01-2024", "%d-%m-%Y")
```

4.1.2.22 label

label

4.1.2.23 name_bride

```
name_bride = np.random.choice(arr_names_woman, size=i)
```

4.1.2.24 name_groom

```
name_groom = np.random.choice(arr_names_man, size=i)
```

4.1.2.25 num_registry

```
num_registry = np.random.randint(1, high=i+1, size=i)
```

4.1.2.26 rotation

rotation

4.1.2.27 start_date_birthday

```
start_date_birthday = datetime.strptime("01-01-1980", "%d-%m-%Y")
```

4.1.2.28 start_date_wedding

```
start_date_wedding = datetime.strptime("01-01-2018", "%d-%m-%Y")
```

4.1.2.29 start_time

```
int start_time = time.time_ns() / 1000000
```

4.1.2.30 table

```
table = pd.read_csv('Names.csv', sep=';', index_col=False, header=None)
```


Глава 5

Классы

5.1 Класс Obj

Открытые члены

- `__ge__` (self, other)
- `__gt__` (self, other)
- `__init__` (self, arr)
- `__le__` (self, other)
- `__lt__` (self, other)

Открытые атрибуты

- `date_w`
- `gr_fname`
- `num_reg`

5.1.1 Конструктор(ы)

5.1.1.1 `__init__`()

```
__init__ (
    self,
    arr )
```

5.1.2 Методы

5.1.2.1 `__ge__`()

```
__ge__ (
    self,
    other )
```

5.1.2.2 `__gt__()`

```
__gt__ (
    self,
    other )
```

5.1.2.3 `__le__()`

```
__le__ (
    self,
    other )
```

5.1.2.4 `__lt__()`

```
__lt__ (
    self,
    other )
```

5.1.3 Данные класса

5.1.3.1 `date_w`

```
date_w
```

5.1.3.2 `gr_fname`

```
gr_fname
```

5.1.3.3 `num_reg`

```
num_reg
```

Объявления и описания членов класса находятся в файле:

- `laba1/v15.py`

Глава 6

Файлы

6.1 Файл laba1/v15.py

Классы

- class [Obj](#)

Пространства имен

- namespace [v15](#)

Функции

- [partition](#) (arr, left, right)
- [QuickSort](#) (arr, left, right)
- [SelectSort](#) (arr)
- [ShakerSort](#) (arr)

Переменные

- [arr1](#) = df.to_numpy().tolist()[1:]
- [arr2](#) = df.to_numpy().tolist()[1:]
- [arr3](#) = df.to_numpy().tolist()[1:]
- [arr_names_man](#) = np.array([table](#)[0])
- [arr_names_woman](#) = np.array([table](#)[1])
- list [arr_quick](#) = []
- list [arr_select](#) = []
- list [arr_shaker](#) = []
- [arr_sizes](#) = np.array([100, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000])
- dict [d](#)
- list [date_bride](#) = []
- list [date_groom](#) = []
- list [date_wedding](#) = []
- tuple [days_birthday](#) = ([end_date_birthday](#) - [start_date_birthday](#)).days
- tuple [days_wedding](#) = ([end_date_wedding](#) - [start_date_wedding](#)).days
- [df](#) = pd.DataFrame(data=[d](#))

- `df1 = pd.DataFrame(data=arr1)`
- `df2 = pd.DataFrame(data=arr2)`
- `df3 = pd.DataFrame(data=arr3)`
- `end_date_birthday = datetime.strptime("01-01-2000", "%d-%m-%Y")`
- `end_date_wedding = datetime.strptime("01-01-2024", "%d-%m-%Y")`
- `label`
- `name_bride = np.random.choice(arr_names_woman, size=i)`
- `name_groom = np.random.choice(arr_names_man, size=i)`
- `num_registry = np.random.randint(1, high=i+1, size=i)`
- `rotation`
- `start_date_birthday = datetime.strptime("01-01-1980", "%d-%m-%Y")`
- `start_date_wedding = datetime.strptime("01-01-2018", "%d-%m-%Y")`
- `int start_time = time.time_ns() / 1000000`
- `table = pd.read_csv('Names.csv', sep=';', index_col=False, header=None)`

Предметный указатель

`--ge__` [Obj, 13](#)
`--gt__` [Obj, 13](#)
`--init__` [Obj, 13](#)
`--le__` [Obj, 14](#)
`--lt__` [Obj, 14](#)

`arr1`
 [v15, 8](#)
`arr2`
 [v15, 8](#)
`arr3`
 [v15, 8](#)
`arr_names_man`
 [v15, 9](#)
`arr_names_woman`
 [v15, 9](#)
`arr_quick`
 [v15, 9](#)
`arr_select`
 [v15, 9](#)
`arr_shaker`
 [v15, 9](#)
`arr_sizes`
 [v15, 9](#)

`d`
 [v15, 9](#)
`date_bride`
 [v15, 9](#)
`date_groom`
 [v15, 9](#)
`date_w`
 [Obj, 14](#)
`date_wedding`
 [v15, 10](#)
`days_birthday`
 [v15, 10](#)
`days_wedding`
 [v15, 10](#)
`df`
 [v15, 10](#)
`df1`
 [v15, 10](#)
`df2`
 [v15, 10](#)
`df3`
 [v15, 10](#)

`end_date_birthday`
 [v15, 10](#)
`end_date_wedding`
 [v15, 10](#)

`gr_fname`
 [Obj, 14](#)

`laba1/v15.py`, [15](#)
`label`
 [v15, 10](#)

`name_bride`
 [v15, 11](#)
`name_groom`
 [v15, 11](#)
`num_reg`
 [Obj, 14](#)
`num_registry`
 [v15, 11](#)

`Obj`, [13](#)
 [__ge__](#), [13](#)
 [__gt__](#), [13](#)
 [__init__](#), [13](#)
 [__le__](#), [14](#)
 [__lt__](#), [14](#)
 [date_w](#), [14](#)
 [gr_fname](#), [14](#)
 [num_reg](#), [14](#)

`partition`
 [v15, 8](#)

`QuickSort`
 [v15, 8](#)

`rotation`
 [v15, 11](#)

`SelectSort`
 [v15, 8](#)
`ShakerSort`
 [v15, 8](#)
`start_date_birthday`
 [v15, 11](#)
`start_date_wedding`

- v15, [11](#)
- start_time
 - v15, [11](#)
- table
 - v15, [11](#)
- v15, [7](#)
 - arr1, [8](#)
 - arr2, [8](#)
 - arr3, [8](#)
 - arr_names_man, [9](#)
 - arr_names_woman, [9](#)
 - arr_quick, [9](#)
 - arr_select, [9](#)
 - arr_shaker, [9](#)
 - arr_sizes, [9](#)
 - d, [9](#)
 - date_bride, [9](#)
 - date_groom, [9](#)
 - date_wedding, [10](#)
 - days_birthday, [10](#)
 - days_wedding, [10](#)
 - df, [10](#)
 - df1, [10](#)
 - df2, [10](#)
 - df3, [10](#)
 - end_date_birthday, [10](#)
 - end_date_wedding, [10](#)
 - label, [10](#)
 - name_bride, [11](#)
 - name_groom, [11](#)
 - num_registry, [11](#)
 - partition, [8](#)
 - QuickSort, [8](#)
 - rotation, [11](#)
 - SelectSort, [8](#)
 - ShakerSort, [8](#)
 - start_date_birthday, [11](#)
 - start_date_wedding, [11](#)
 - start_time, [11](#)
 - table, [11](#)