

Лабораторная №1

Generated by Doxygen 1.10.0

1	README	1
2	Namespace Index	3
2.1	Namespace List	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	algorithms Namespace Reference	9
5.1.1	Detailed Description	9
5.1.2	Function Documentation	9
5.1.2.1	insert_sort()	9
5.1.2.2	merge()	10
5.1.2.3	merge_sort()	10
5.1.2.4	quick_sort()	11
5.2	main Namespace Reference	11
5.2.1	Detailed Description	12
5.2.2	Function Documentation	12
5.2.2.1	rand_schedule_data()	12
5.2.2.2	read_schedule()	12
5.2.2.3	to_need_len()	12
5.2.2.4	to_type()	13
5.2.2.5	write_schedule()	13
5.2.3	Variable Documentation	13
5.2.3.1	figsize	13
5.2.3.2	insert	13
5.2.3.3	label	13
5.2.3.4	merge	13
5.2.3.5	quick	14
5.2.3.6	sizes	14
5.3	schedule Namespace Reference	14
5.3.1	Detailed Description	14
5.3.2	Function Documentation	14
5.3.2.1	date_cmp()	14
5.3.2.2	time_cmp()	15
6	Class Documentation	17
6.1	schedule.Schedule Class Reference	17
6.1.1	Detailed Description	17
6.1.2	Constructor & Destructor Documentation	18
6.1.2.1	__init__()	18

6.1.3 Member Function Documentation	18
6.1.3.1 <code>__eq__()</code>	18
6.1.3.2 <code>__ge__()</code>	18
6.1.3.3 <code>__gt__()</code>	19
6.1.3.4 <code>__le__()</code>	19
6.1.3.5 <code>__lt__()</code>	19
6.1.3.6 <code>__ne__()</code>	19
6.1.3.7 <code>__str__()</code>	19
6.1.4 Member Data Documentation	20
6.1.4.1 <code>date</code>	20
6.1.4.2 <code>dur</code>	20
6.1.4.3 <code>num</code>	20
6.1.4.4 <code>time</code>	20
6.1.4.5 <code>typ</code>	20
7 File Documentation	21
7.1 <code>algorithms.py</code> File Reference	21
7.2 <code>main.py</code> File Reference	21
7.3 <code>README.md</code> File Reference	22
7.4 <code>schedule.py</code> File Reference	22
Предметный указатель	23

Глава 1

README

В репозитории три .py файла:

[main.py](#) - генерация данных, чтение/запись данных из файла/в файл, замеры скорости каждого алгоритма, построение графиков

[algorithms.py](#) - реализация quick_sort, insert_sort, merge_sort

[schedule.py](#) - класс записи в расписании поездов, перегрузка операторов сравнения

Две картинки .png - графики время/размер

Файлы in.txt - сгенерированные данные и out.txt - отсортированные данные

Глава 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

algorithms		
	Algorithms.py	9
main		
	Main.py	11
schedule		
	Schedule.py	14

Глава 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

schedule.Schedule	
Класс расписания поездов	17

Глава 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

algorithms.py	21
main.py	21
schedule.py	22

Глава 5

Namespace Documentation

5.1 algorithms Namespace Reference

[algorithms.py](#)

Functions

- [insert_sort](#) (arr)
Функция сортировки массива 'Вставками'.
- [quick_sort](#) (arr, start=0, end=-1)
Функция сортировки массива методом 'QuickSort'.
- [merge_sort](#) (arr, start=0, end=-1)
Функция сортировки массива методом слияния
- [merge](#) (arr, start, mid, end)
Функция слияния двух частей массива в одну для MergeSort.

5.1.1 Detailed Description

[algorithms.py](#)

Файл с функциями сортировок

5.1.2 Function Documentation

5.1.2.1 insert_sort()

```
algorithms.insert_sort (  
    arr )
```

Функция сортировки массива 'Вставками'.

Parameters

arr	сортируемый список
-----	--------------------

Returns

Ничего не возвращает

5.1.2.2 merge()

```
algorithms.merge (  
    arr,  
    start,  
    mid,  
    end )
```

Функция слияния двух частей массива в одну для MergeSort.

Parameters

arr	сортируемый список
start	индекс начала
mid	индекс середины
end	индекс конца

Returns

Ничего не возвращает

5.1.2.3 merge_sort()

```
algorithms.merge_sort (  
    arr,  
    start = 0,  
    end = -1 )
```

Функция сортировки массива методом слияния

Parameters

arr	сортируемый список
start	индекс начала сортировки
end	индекс конца сортировки

Returns

Ничего не возвращает

5.1.2.4 quick_sort()

```
algorithms.quick_sort (  
    arr,  
    start = 0,  
    end = -1 )
```

Функция сортировки массива методом 'QuickSort'.

Parameters

arr	сортируемый список
start	индекс начала сортировки
end	индекс конца сортировки

Returns

Ничего не возвращает

5.2 main Namespace Reference

[main.py](#)

Functions

- [read_schedule](#) (path, n)
Функция для чтения расписания из текстового файла (разделитель - пробел)
- [write_schedule](#) (schedule_list, path)
Функция для записи расписания в текстовый файл (разделитель - пробел) из списка с объектами класса Schedule.
- [to_type](#) (x)
Функция преобразования числа в тип поезда в зависимости от его остатка на 3 (для случайной генерации данных)
- [to_need_len](#) (s, n)
Функция, дополняющая строку до нужной длины, добавлением нулей в начало
- [rand_schedule_data](#) ()
Функция, возвращающая объект класса Schedule со случайными данными

Variables

- list [sizes](#) = [100, 1000, 5000, 10000, 20000, 50000, 70000, 100000, 120000, 150000]
- list [quick](#)
- list [merge](#)
- list [insert](#)
- [figsize](#)
- [label](#)

5.2.1 Detailed Description

[main.py](#)

Файл для проведения замеров времени сортировки разных данных алгоритмами из [algorithms.py](#)

5.2.2 Function Documentation

5.2.2.1 `rand_schedule_data()`

```
main.rand_schedule_data ( )
```

Функция, возвращающая объект класса `Schedule` со случайными данными

5.2.2.2 `read_schedule()`

```
main.read_schedule (
    path,
    n )
```

Функция для чтения расписания из текстового файла (разделитель - пробел)

Parameters

n	количество строк, которое должно быть прочитано
path	путь к текстовому файлу с расписанием (разделитель - пробел)

Returns

список с объектами класса `Schedule`

5.2.2.3 `to_need_len()`

```
main.to_need_len (
    s,
    n )
```

Функция, дополняющая строку до нужной длины, добавлением нулей в начало

Parameters

s	строка
n	необходимая длина

Returns

возвращает дополненную нулями строку

5.2.2.4 to_type()

```
main.to_type (
    x )
```

Функция преобразования числа в тип поезда в зависимости от его остатка на 3 (для случайной генерации данных)

5.2.2.5 write_schedule()

```
main.write_schedule (
    schedule_list,
    path )
```

Функция для записи расписания в текстовый файл (разделитель - пробел) из списка с объектами класса Schedule.

Parameters

schedule_list	список с объектами Schedule, из которого надо записать данные в файл
path	путь к файлу

5.2.3 Variable Documentation

5.2.3.1 figsize

```
main.figsize
```

5.2.3.2 insert

```
main.insert
```

Initial value:

```
00001 = [0.0019609928131103516, 0.17946696281433105, 4.519682168960571, 18.38012409210205, 78.48876190185547,
00002      529.2682859897614, 490, 1000, 1440, 2250]
```

5.2.3.3 label

```
main.label
```

5.2.3.4 merge

```
main.merge
```

Initial value:

```
00001 = [0.0005178451538085938, 0.008249044418334961, 0.05442500114440918, 0.12309789657592773, 0.275620698928833,
00002      0.7975671291351318, 1.1729211807250977, 1.7789530754089355, 2.1777520179748535, 2.803262948989868]
```

5.2.3.5 quick

main.quick

Initial value:

```
00001 = [0.0009491443634033203, 0.023778915405273438, 0.08445596694946289, 0.18958592414855957, 0.39475512504577637,
00002      1.176802635192871, 2.93355393409729, 2.5385050773620605, 3.0803987979888916, 4.008557081222534]
```

5.2.3.6 sizes

main.sizes = [100, 1000, 5000, 10000, 20000, 50000, 70000, 100000, 120000, 150000]

5.3 schedule Namespace Reference

[schedule.py](#)

Classes

- class [Schedule](#)
Класс расписания поездов

Functions

- [time_cmp](#) (a, b)
Функция сравнения двух строк содержащих время формата чч:мм:сс
- [date_cmp](#) (a, b)
Функция сравнения двух строк содержащих даты формата дд.

5.3.1 Detailed Description

[schedule.py](#)

Класс расписания, его методы, вспомогательные функции (компараторы полей класса)

5.3.2 Function Documentation

5.3.2.1 date_cmp()

```
schedule.date_cmp (  
    a,  
    b )
```

Функция сравнения двух строк содержащих даты формата дд.

мм.гггг

Parameters

a	дата 1 дд.мм.гггг
b	дата 2 дд.мм.гггг

Returns

-1 если $a < b$, 0 если $a = b$, 1 если $a > b$

5.3.2.2 time_cmp()

```
schedule.time_cmp (  
    a,  
    b )
```

Функция сравнения двух строк содержащих время формата чч:мм:сс

Parameters

a	время 1 чч:мм:сс
b	время 2 чч:мм:сс

Returns

-1 если $a < b$, 0 если $a = b$, 1 если $a > b$

Глава 6

Class Documentation

6.1 schedule.Schedule Class Reference

Класс расписания поездов

Public Member Functions

- `__init__` (self, num=0, date='00.00.0000', typ='-', time='00:00:00', dur='00:00:00')
Конструктор
- `__eq__` (self, other)
Оператор сравнения ==.
- `__lt__` (self, other)
Оператор сравнения <.
- `__gt__` (self, other)
Оператор сравнения >.
- `__le__` (self, other)
Оператор сравнения <=.
- `__ge__` (self, other)
Оператор сравнения >=.
- `__ne__` (self, other)
Оператор сравнения !=.
- `__str__` (self)
Метод преобразования в строку

Public Attributes

- num
- date
- typ
- time
- dur

6.1.1 Detailed Description

Класс расписания поездов

6.1.2 Constructor & Destructor Documentation

6.1.2.1 `__init__()`

```
schedule.Schedule.__init__(
    self,
    num = 0,
    date = '00.00.0000',
    typ = '-',
    time = '00:00:00',
    dur = '00:00:00')
```

Конструктор

Parameters

num	номер поезда :int
date	дата отправления дд.мм.гггг :str
typ	тип поезда (скорый, пассажирский, товарный):str
time	время отправления чч:мм:сс :str
dur	время в пути чч:мм:сс :str

6.1.3 Member Function Documentation

6.1.3.1 `__eq__()`

```
schedule.Schedule.__eq__(
    self,
    other)
```

Оператор сравнения `==`.

Сравнение по полям: дата отправления, время отправления, номер поезда, время в пути (по убыванию приоритета)

6.1.3.2 `__ge__()`

```
schedule.Schedule.__ge__(
    self,
    other)
```

Оператор сравнения `>=`.

Сравнение по полям: дата отправления, время отправления, номер поезда, время в пути (по убыванию приоритета)

6.1.3.3 `__gt__()`

```
schedule.Schedule.__gt__(  
    self,  
    other )
```

Оператор сравнения `>`

Сравнение по полям: дата отправления, время отправления, номер поезда, время в пути (по убыванию приоритета)

6.1.3.4 `__le__()`

```
schedule.Schedule.__le__(  
    self,  
    other )
```

Оператор сравнения `<=`.

Сравнение по полям: дата отправления, время отправления, номер поезда, время в пути (по убыванию приоритета)

6.1.3.5 `__lt__()`

```
schedule.Schedule.__lt__(  
    self,  
    other )
```

Оператор сравнения `<`.

Сравнение по полям: дата отправления, время отправления, номер поезда, время в пути (по убыванию приоритета)

6.1.3.6 `__ne__()`

```
schedule.Schedule.__ne__(  
    self,  
    other )
```

Оператор сравнения `!=`.

Сравнение по полям: дата отправления, время отправления, номер поезда, время в пути (по убыванию приоритета)

6.1.3.7 `__str__()`

```
schedule.Schedule.__str__(  
    self )
```

Метод преобразования в строку

6.1.4 Member Data Documentation

6.1.4.1 date

`schedule.Schedule.date`

6.1.4.2 dur

`schedule.Schedule.dur`

6.1.4.3 num

`schedule.Schedule.num`

6.1.4.4 time

`schedule.Schedule.time`

6.1.4.5 typ

`schedule.Schedule.typ`

Глава 7

File Documentation

7.1 algorithms.py File Reference

Namespaces

- namespace [algorithms](#)
[algorithms.py](#)

Functions

- [algorithms.insert_sort](#) (arr)
Функция сортировки массива 'Вставками'.
- [algorithms.quick_sort](#) (arr, start=0, end=-1)
Функция сортировки массива методом 'QuickSort'.
- [algorithms.merge_sort](#) (arr, start=0, end=-1)
Функция сортировки массива методом слияния
- [algorithms.merge](#) (arr, start, mid, end)
Функция слияния двух частей массива в одну для MergeSort.

7.2 main.py File Reference

Namespaces

- namespace [main](#)
[main.py](#)

Functions

- [main.read_schedule](#) (path, n)
Функция для чтения расписания из текстового файла (разделитель - пробел)
- [main.write_schedule](#) (schedule_list, path)
Функция для записи расписания в текстовый файл (разделитель - пробел) из списка с объектами класса Schedule.
- [main.to_type](#) (x)
Функция преобразования числа в тип поезда в зависимости от его остатка на 3 (для случайной генерации данных)
- [main.to_need_len](#) (s, n)
Функция, дополняющая строку до нужной длины, добавлением нулей в начало
- [main.rand_schedule_data](#) ()
Функция, возвращающая объект класса Schedule со случайными данными

Variables

- list [main.sizes](#) = [100, 1000, 5000, 10000, 20000, 50000, 70000, 100000, 120000, 150000]
- list [main.quick](#)
- list [main.merge](#)
- list [main.insert](#)
- [main.figsize](#)
- [main.label](#)

7.3 README.md File Reference

7.4 schedule.py File Reference

Classes

- class [schedule.Schedule](#)
Класс расписания поездов

Namespaces

- namespace [schedule](#)
[schedule.py](#)

Functions

- [schedule.time_cmp](#) (a, b)
Функция сравнения двух строк содержащих время формата чч:мм:сс
- [schedule.date_cmp](#) (a, b)
Функция сравнения двух строк содержащих даты формата дд.

Предметный указатель

- `--eq__`
 - `schedule.Schedule`, 18
- `--ge__`
 - `schedule.Schedule`, 18
- `--gt__`
 - `schedule.Schedule`, 18
- `--init__`
 - `schedule.Schedule`, 18
- `--le__`
 - `schedule.Schedule`, 19
- `--lt__`
 - `schedule.Schedule`, 19
- `--ne__`
 - `schedule.Schedule`, 19
- `--str__`
 - `schedule.Schedule`, 19

- algorithms, 9
 - `insert_sort`, 9
 - `merge`, 10
 - `merge_sort`, 10
 - `quick_sort`, 10
- algorithms.py, 21

- date
 - `schedule.Schedule`, 20
- `date_cmp`
 - `schedule`, 14
- `dur`
 - `schedule.Schedule`, 20

- figsize
 - main, 13

- insert
 - main, 13
- `insert_sort`
 - algorithms, 9

- label
 - main, 13

- main, 11
 - figsize, 13
 - insert, 13
 - label, 13
 - merge, 13
 - quick, 13
 - `rand_schedule_data`, 12
 - `read_schedule`, 12

- `sizes`, 14
- `to_need_len`, 12
- `to_type`, 12
- `write_schedule`, 13
- main.py, 21
- merge
 - algorithms, 10
 - main, 13
- `merge_sort`
 - algorithms, 10
- num
 - `schedule.Schedule`, 20
- quick
 - main, 13
- `quick_sort`
 - algorithms, 10

- `rand_schedule_data`
 - main, 12
- `read_schedule`
 - main, 12
- README, 1
- README.md, 22

- schedule, 14
 - `date_cmp`, 14
 - `time_cmp`, 15
- schedule.py, 22
- `schedule.Schedule`, 17
 - `--eq__`, 18
 - `--ge__`, 18
 - `--gt__`, 18
 - `--init__`, 18
 - `--le__`, 19
 - `--lt__`, 19
 - `--ne__`, 19
 - `--str__`, 19
 - date, 20
 - dur, 20
 - num, 20
 - time, 20
 - typ, 20

- sizes
 - main, 14

- time
 - `schedule.Schedule`, 20
- `time_cmp`

- schedule, [15](#)
- to_need_len
 - main, [12](#)
- to_type
 - main, [12](#)
- typ
 - schedule.Schedule, [20](#)
- write_schedule
 - main, [13](#)