

My Project

Generated by Doxygen 1.8.17

1 Namespace Index	1
1 Namespace Index	1
1.1 Namespace List . . . . .	1
2 Class Index	1
2.1 Class List . . . . .	1
3 File Index	2
3.1 File List . . . . .	2
4 Namespace Documentation	2
4.1 graphics Namespace Reference . . . . .	2
5 Class Documentation	2
5.1 flatsData Struct Reference . . . . .	2
5.1.1 Detailed Description . . . . .	2
5.1.2 Constructor & Destructor Documentation . . . . .	3
5.1.3 Friends And Related Function Documentation . . . . .	3
5.1.4 Member Data Documentation . . . . .	5
6 File Documentation	5
6.1 data.txt File Reference . . . . .	5
6.2 graphics.py File Reference . . . . .	5
6.2.1 Detailed Description . . . . .	6
6.3 main.cpp File Reference . . . . .	6
6.4 output.txt File Reference . . . . .	6
6.5 timestamps.txt File Reference . . . . .	6
Index	7

## 1 Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[graphics](#) 2

## 2 Class Index

### 2.1 Class List

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

[flatsData](#)  
Структура для работы с данным о квартирах 2

## 3 File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">graphics.py</a>	5
Файл, который строит графики о времени выполнения сортировки массивов	
<a href="#">main.cpp</a>	6

## 4 Namespace Documentation

### 4.1 graphics Namespace Reference

## 5 Class Documentation

### 5.1 flatsData Struct Reference

Структура для работы с данным о квартирах

#### Public Member Functions

- [flatsData](#) (int [houseNum](#), int [flatNum](#), int [roomCount](#), int [flatSquare](#), string [ownerName](#), int [countResidents](#))

#### Public Attributes

- int [houseNum](#)
- int [flatNum](#)
- int [roomCount](#)
- int [flatSquare](#)
- string [ownerName](#)
- int [countResidents](#)

#### Friends

- bool [operator==](#) (const [flatsData](#) &a, const [flatsData](#) &b)  
Оператор проверки на равенство двух квартир
- bool [operator<](#) (const [flatsData](#) &a, const [flatsData](#) &b)  
Оператор проверки меньше
- bool [operator<=](#) (const [flatsData](#) &a, const [flatsData](#) &b)  
Оператор проверки на меньше или равно
- bool [operator>](#) (const [flatsData](#) &a, const [flatsData](#) &b)  
Оператор проверки на больше
- bool [operator>=](#) (const [flatsData](#) &a, const [flatsData](#) &b)  
Оператор проверки на больше или равно

#### 5.1.1 Detailed Description

Структура для работы с данным о квартирах

## Parameters

houseNum	номер дома
flatNum	номер квартиры
roomCount	количество комнат
flatSquare	площадь квартиры
ownerName	имя владельца
countResidents	количество жителей

## 5.1.2 Constructor &amp; Destructor Documentation

5.1.2.1 flatsData() flatsData::flatsData (

```

    int houseNum,
    int flatNum,
    int roomCount,
    int flatSquare,
    string ownerName,
    int countResidents ) [inline]

```

## 5.1.3 Friends And Related Function Documentation

5.1.3.1 operator< bool operator< (

```

    const flatsData & a,
    const flatsData & b ) [friend]

```

Оператор проверки меньше

## Parameters

a	данные о первой квартире
b	данные о второй квартире

## Returns

Меньше ли данные о первой квартире

5.1.3.2 operator<= bool operator<= (

```

    const flatsData & a,
    const flatsData & b ) [friend]

```

Оператор проверки на меньше или равно

## Parameters

a	данные о первой квартире
b	данные о второй квартире

## Returns

Меньше или равны ли данные о первой

```
5.1.3.3 operator== bool operator== (  
    const flatsData & a,  
    const flatsData & b ) [friend]
```

Оператор проверки на равенство двух квартир

## Parameters

a	данные о первой квартире
b	данные о второй квартире

## Returns

Совпадают ли данные о квартирах

```
5.1.3.4 operator> bool operator> (  
    const flatsData & a,  
    const flatsData & b ) [friend]
```

Оператор проверки на больше

## Parameters

a	данные о первой квартире
b	данные о второй квартире

## Returns

Больше ли данные о первой квартире

```
5.1.3.5 operator>= bool operator>= (  
    const flatsData & a,  
    const flatsData & b ) [friend]
```

Оператор проверки на больше или равно

## Parameters

a	данные о первой квартире
b	данные о второй квартире

## Returns

больше или равны ли данные о первой квартире

## 5.1.4 Member Data Documentation

5.1.4.1 `countResidents` `int flatsData::countResidents`

5.1.4.2 `flatNum` `int flatsData::flatNum`

5.1.4.3 `flatSquare` `int flatsData::flatSquare`

5.1.4.4 `houseNum` `int flatsData::houseNum`

5.1.4.5 `ownerName` `string flatsData::ownerName`

5.1.4.6 `roomCount` `int flatsData::roomCount`

The documentation for this struct was generated from the following file:

- [main.cpp](#)

## 6 File Documentation

### 6.1 data.txt File Reference

### 6.2 graphics.py File Reference

Файл, который строит графики о времени выполнения сортировки массивов

## Namespaces

- [graphics](#)

### 6.2.1 Detailed Description

Файл, который строит графики о времени выполнения сортировки массивов

### 6.3 main.cpp File Reference

```
#include <bits/stdc++.h>
```

Include dependency graph for main.cpp:

### 6.4 output.txt File Reference

### 6.5 timestamps.txt File Reference

## Index

- countResidents
  - flatsData, [5](#)
- data.txt, [5](#)
- flatNum
  - flatsData, [5](#)
- flatsData, [2](#)
  - countResidents, [5](#)
  - flatNum, [5](#)
  - flatsData, [3](#)
  - flatSquare, [5](#)
  - houseNum, [5](#)
  - operator<, [3](#)
  - operator<=, [3](#)
  - operator>, [4](#)
  - operator>=, [4](#)
  - operator==, [4](#)
  - ownerName, [5](#)
  - roomCount, [5](#)
- flatSquare
  - flatsData, [5](#)
- graphics, [2](#)
- graphics.py, [5](#)
- houseNum
  - flatsData, [5](#)
- main.cpp, [6](#)
- operator<
  - flatsData, [3](#)
- operator<=
  - flatsData, [3](#)
- operator>
  - flatsData, [4](#)
- operator>=
  - flatsData, [4](#)
- operator==
  - flatsData, [4](#)
- output.txt, [6](#)
- ownerName
  - flatsData, [5](#)
- roomCount
  - flatsData, [5](#)
- timestamps.txt, [6](#)