

Lab 2 Silin Ivan

Generated by Doxygen 1.9.6

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 housing_estate Struct Reference	5
3.1.1 Constructor & Destructor Documentation	5
3.1.1.1 housing_estate() [1/2]	5
3.1.1.2 housing_estate() [2/2]	6
3.1.2 Member Function Documentation	6
3.1.2.1 operator<()	6
3.1.2.2 operator<=()	6
3.1.2.3 operator>()	6
3.1.2.4 operator>=()	6
3.1.3 Member Data Documentation	6
3.1.3.1 name	6
3.1.3.2 number_of_residents	7
3.1.3.3 number_of_rooms	7
3.1.3.4 numberApartment	7
3.1.3.5 numberHouse	7
3.1.3.6 square	7
4 File Documentation	9
4.1 D:/23/labpr_2/main.cpp File Reference	9
4.1.1 Function Documentation	9
4.1.1.1 BinSearch()	10
4.1.1.2 File()	10
4.1.1.3 LinearSearch()	10
4.1.1.4 main()	10
4.1.1.5 operator<<()	10
4.1.1.6 partition()	10
4.1.1.7 quickSort()	11
Index	13

Chapter 1

Class Index

1.1 Class List

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

housing_estate	5
--	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

D:/23/labpr_2/ main.cpp	9
---	---

Chapter 3

Class Documentation

3.1 housing_estate Struct Reference

Public Member Functions

- [housing_estate](#) ()
- [housing_estate](#) (int [numberHouse](#), int [numberApartment](#), float [square](#), string [name](#))
- bool [operator>](#) ([housing_estate](#) &h2)
- bool [operator>=](#) ([housing_estate](#) &h2)
- bool [operator<](#) ([housing_estate](#) &h2)
- bool [operator<=](#) ([housing_estate](#) &h2)

Public Attributes

- int [numberHouse](#)
- int [numberApartment](#)
- int [number_of_rooms](#)
- float [square](#)
- string [name](#)
- int [number_of_residents](#)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 housing_estate() [1/2]

```
housing_estate::housing_estate ( ) [inline]
```

3.1.1.2 housing_estate() [2/2]

```
housing_estate::housing_estate (
    int numberHouse,
    int numberApartment,
    float square,
    string name ) [inline]
```

3.1.2 Member Function Documentation

3.1.2.1 operator<()

```
bool housing_estate::operator< (
    housing_estate & h2 ) [inline]
```

3.1.2.2 operator<=()

```
bool housing_estate::operator<= (
    housing_estate & h2 ) [inline]
```

3.1.2.3 operator>()

```
bool housing_estate::operator> (
    housing_estate & h2 ) [inline]
```

3.1.2.4 operator>=()

```
bool housing_estate::operator>= (
    housing_estate & h2 ) [inline]
```

3.1.3 Member Data Documentation

3.1.3.1 name

```
string housing_estate::name
```

3.1.3.2 number_of_residents

```
int housing_estate::number_of_residents
```

3.1.3.3 number_of_rooms

```
int housing_estate::number_of_rooms
```

3.1.3.4 numberApartment

```
int housing_estate::numberApartment
```

3.1.3.5 numberHouse

```
int housing_estate::numberHouse
```

3.1.3.6 square

```
float housing_estate::square
```

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

- D:/23/labpr_2/[main.cpp](#)

Chapter 4

File Documentation

4.1 D:/23/labpr_2/main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string.h>
#include <stdio.h>
#include <vector>
#include <sstream>
#include <chrono>
#include <iterator>
#include <algorithm>
#include <map>
Include dependency graph for main.cpp:
```

Classes

- struct [housing_estate](#)

Functions

- std::ostream & [operator<<](#) (std::ostream &os, const [housing_estate](#) &h)
- template<class T >
std::vector< int > [LinearSearch](#) (vector< T > &a, int size, string key)
- template<class T >
int [partition](#) (vector< T > &nums, int low, int high)
- template<class T >
void [quickSort](#) (vector< T > &nums, int low, int high)
- template<class T >
std::pair< int, int > [BinSearch](#) (vector< T > a, string key, int low, int high)
- void [File](#) (string file, int sort, string k)
- int [main](#) ()

4.1.1 Function Documentation

4.1.1.1 BinSearch()

```
template<class T >
std::pair< int, int > BinSearch (
    vector< T > a,
    string key,
    int low,
    int high )
```

4.1.1.2 File()

```
void File (
    string file,
    int sort,
    string k )
```

4.1.1.3 LinearSearch()

```
template<class T >
std::vector< int > LinearSearch (
    vector< T > & a,
    int size,
    string key )
```

4.1.1.4 main()

```
int main ( )
```

4.1.1.5 operator<<()

```
std::ostream & operator<< (
    std::ostream & os,
    const housing_estate & h )
```

4.1.1.6 partition()

```
template<class T >
int partition (
    vector< T > & nums,
    int low,
    int high )
```

4.1.1.7 quickSort()

```
template<class T >
void quickSort (
    vector< T > & nums,
    int low,
    int high )
```


Index

- BinSearch
 - main.cpp, [9](#)
- D:/23/labpr_2/main.cpp, [9](#)
- File
 - main.cpp, [10](#)
- housing_estate, [5](#)
 - housing_estate, [5](#)
 - name, [6](#)
 - number_of_residents, [6](#)
 - number_of_rooms, [7](#)
 - numberApartment, [7](#)
 - numberHouse, [7](#)
 - operator<, [6](#)
 - operator<=, [6](#)
 - operator>, [6](#)
 - operator>=, [6](#)
 - square, [7](#)
- LinearSearch
 - main.cpp, [10](#)
- main
 - main.cpp, [10](#)
- main.cpp
 - BinSearch, [9](#)
 - File, [10](#)
 - LinearSearch, [10](#)
 - main, [10](#)
 - operator<<, [10](#)
 - partition, [10](#)
 - quickSort, [10](#)
- name
 - housing_estate, [6](#)
- number_of_residents
 - housing_estate, [6](#)
- number_of_rooms
 - housing_estate, [7](#)
- numberApartment
 - housing_estate, [7](#)
- numberHouse
 - housing_estate, [7](#)
- operator<
 - housing_estate, [6](#)
- operator<<
 - main.cpp, [10](#)
- operator<=
- housing_estate, [6](#)
- operator>
 - housing_estate, [6](#)
- operator>=
 - housing_estate, [6](#)
- partition
 - main.cpp, [10](#)
- quickSort
 - main.cpp, [10](#)
- square
 - housing_estate, [7](#)