

lab3

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 hash	6
3.1.3.2 name	7
3.1.3.3 number_of_residents	7
3.1.3.4 number_of_rooms	7
3.1.3.5 numberApartment	7
3.1.3.6 numberHouse	7
3.1.3.7 square	7
3.2 HousingEstateHashTable Class Reference	7
3.2.1 Constructor & Destructor Documentation	8
3.2.1.1 HousingEstateHashTable() [1/2]	8
3.2.1.2 HousingEstateHashTable() [2/2]	8
3.2.2 Member Function Documentation	8
3.2.2.1 addElement()	8
3.2.2.2 countCollisions()	8
3.2.2.3 curHash() [1/2]	8
3.2.2.4 curHash() [2/2]	8
3.2.2.5 findElement()	8
4 File Documentation	9
4.1 D://main.cpp File Reference	9
4.1.1 Function Documentation	10
4.1.1.1 BinSearch()	10
4.1.1.2 File()	10
4.1.1.3 hashFunctionComplicated()	10
4.1.1.4 hashFunctionSimple()	10
4.1.1.5 LinearSearch()	10

4.1.1.6 main()	11
4.1.1.7 operator<<()	11
4.1.1.8 partition()	11
4.1.1.9 quickSort()	11
4.1.2 Variable Documentation	11
4.1.2.1 meth	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
HousingEstateHashTable	7

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

D:// 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](#)
- int [hash](#)

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 hash

```
int housing_estate::hash
```

3.1.3.2 name

```
string housing_estate::name
```

3.1.3.3 number_of_residents

```
int housing_estate::number_of_residents
```

3.1.3.4 number_of_rooms

```
int housing_estate::number_of_rooms
```

3.1.3.5 numberApartment

```
int housing_estate::numberApartment
```

3.1.3.6 numberHouse

```
int housing_estate::numberHouse
```

3.1.3.7 square

```
float housing_estate::square
```

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

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

3.2 HousingEstateHashTable Class Reference

Public Member Functions

- [HousingEstateHashTable](#) ()
- [HousingEstateHashTable](#) (const int a)
- void [addElement](#) ([housing_estate](#) &object)
- int [countCollisions](#) () const
- void [findElement](#) (const std::string &name) const
- long long [curHash](#) (const [housing_estate](#) &object) const
- long long [curHash](#) (const std::string &name) const

3.2.1 Constructor & Destructor Documentation

3.2.1.1 HousingEstateHashTable() [1/2]

```
HousingEstateHashTable::HousingEstateHashTable ( ) [inline]
```

3.2.1.2 HousingEstateHashTable() [2/2]

```
HousingEstateHashTable::HousingEstateHashTable (
    const int a ) [inline]
```

3.2.2 Member Function Documentation

3.2.2.1 addElement()

```
void HousingEstateHashTable::addElement (
    housing_estate & object ) [inline]
```

3.2.2.2 countCollisions()

```
int HousingEstateHashTable::countCollisions ( ) const [inline]
```

3.2.2.3 curHash() [1/2]

```
long long HousingEstateHashTable::curHash (
    const housing_estate & object ) const [inline]
```

3.2.2.4 curHash() [2/2]

```
long long HousingEstateHashTable::curHash (
    const std::string & name ) const [inline]
```

3.2.2.5 findElement()

```
void HousingEstateHashTable::findElement (
    const std::string & name ) const [inline]
```

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

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

Chapter 4

File Documentation

4.1 D://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 <set>
```

Classes

- struct [housing_estate](#)
- class [HousingEstateHashTable](#)

Functions

- long long [hashFunctionSimple](#) (string str)
- long long [hashFunctionComplicated](#) (string str)
- 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 method, string key)
- int [main](#) ()

Variables

- int `meth` =1

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 method,
    string key )
```

4.1.1.3 hashFunctionComplicated()

```
long long hashFunctionComplicated (
    string str )
```

4.1.1.4 hashFunctionSimple()

```
long long hashFunctionSimple (
    string str )
```

4.1.1.5 LinearSearch()

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

4.1.1.6 main()

```
int main ( )
```

4.1.1.7 operator<<()

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

4.1.1.8 partition()

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

4.1.1.9 quickSort()

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

4.1.2 Variable Documentation

4.1.2.1 meth

```
int meth =1
```


Index

- addElement
 - HousingEstateHashTable, 8
- BinSearch
 - main.cpp, 10
- countCollisions
 - HousingEstateHashTable, 8
- curHash
 - HousingEstateHashTable, 8
- D://main.cpp, 9
- File
 - main.cpp, 10
- findElement
 - HousingEstateHashTable, 8
- hash
 - housing_estate, 6
- hashFunctionComplicated
 - main.cpp, 10
- hashFunctionSimple
 - main.cpp, 10
- housing_estate, 5
 - hash, 6
 - housing_estate, 5
 - name, 6
 - number_of_residents, 7
 - number_of_rooms, 7
 - numberApartment, 7
 - numberHouse, 7
 - operator<, 6
 - operator<=, 6
 - operator>, 6
 - operator>=, 6
 - square, 7
- HousingEstateHashTable, 7
 - addElement, 8
 - countCollisions, 8
 - curHash, 8
 - findElement, 8
 - HousingEstateHashTable, 8
- LinearSearch
 - main.cpp, 10
- main
 - main.cpp, 10
- main.cpp
 - BinSearch, 10
 - File, 10
 - hashFunctionComplicated, 10
 - hashFunctionSimple, 10
 - LinearSearch, 10
 - main, 10
 - meth, 11
 - operator<<, 11
 - partition, 11
 - quickSort, 11
- meth
 - main.cpp, 11
- name
 - housing_estate, 6
- number_of_residents
 - housing_estate, 7
- number_of_rooms
 - housing_estate, 7
- numberApartment
 - housing_estate, 7
- numberHouse
 - housing_estate, 7
- operator<
 - housing_estate, 6
- operator<<
 - main.cpp, 11
- operator<=
 - housing_estate, 6
- operator>
 - housing_estate, 6
- operator>=
 - housing_estate, 6
- partition
 - main.cpp, 11
- quickSort
 - main.cpp, 11
- square
 - housing_estate, 7