

Scatter Search

Generated by Doxygen 1.9.3

1 Scatter Search	1
1.1 Description	1
1.2 Usage	1
1.3 Author	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 Block< Key > Class Template Reference	9
5.1.1 Constructor & Destructor Documentation	10
5.1.1.1 Block() [1/2]	10
5.1.1.2 Block() [2/2]	10
5.1.2 Member Function Documentation	11
5.1.2.1 Insert()	11
5.1.2.2 IsFull()	11
5.1.2.3 Search()	12
5.1.2.4 Write()	12
5.2 DispersionFunction< Key > Class Template Reference	13
5.2.1 Member Function Documentation	13
5.2.1.1 operator()	13
5.3 ExplorationFunction< Key > Class Template Reference	13
5.3.1 Member Function Documentation	14
5.3.1.1 operator()	14
5.4 FdModule< Key > Class Template Reference	14
5.4.1 Constructor & Destructor Documentation	15
5.4.1.1 FdModule()	15
5.4.2 Member Function Documentation	15
5.4.2.1 operator()	15
5.5 FdRandom< Key > Class Template Reference	16
5.5.1 Constructor & Destructor Documentation	17
5.5.1.1 FdRandom()	17
5.5.2 Member Function Documentation	17
5.5.2.1 operator()	17
5.6 FdSum< Key > Class Template Reference	18
5.6.1 Constructor & Destructor Documentation	18
5.6.1.1 FdSum()	18
5.6.2 Member Function Documentation	19

5.6.2.1 operator()	19
5.7 FeDoubleDispersion< Key > Class Template Reference	20
5.7.1 Constructor & Destructor Documentation	20
5.7.1.1 FeDoubleDispersion() [1/2]	21
5.7.1.2 FeDoubleDispersion() [2/2]	21
5.7.2 Member Function Documentation	21
5.7.2.1 operator()	21
5.8 FeLineal< Key > Class Template Reference	22
5.8.1 Constructor & Destructor Documentation	23
5.8.1.1 FeLineal()	23
5.8.2 Member Function Documentation	23
5.8.2.1 operator()	23
5.9 FeQuadratic< Key > Class Template Reference	24
5.9.1 Constructor & Destructor Documentation	25
5.9.1.1 FeQuadratic()	25
5.9.2 Member Function Documentation	25
5.9.2.1 operator()	25
5.10 FeRedispersion< Key > Class Template Reference	26
5.10.1 Constructor & Destructor Documentation	26
5.10.1.1 FeRedispersion()	26
5.10.2 Member Function Documentation	27
5.10.2.1 operator()	27
5.11 HashTable< Key > Class Template Reference	27
5.11.1 Constructor & Destructor Documentation	28
5.11.1.1 HashTable() [1/2]	28
5.11.1.2 HashTable() [2/2]	28
5.11.1.3 ~HashTable()	29
5.11.2 Member Function Documentation	29
5.11.2.1 Insert()	29
5.11.2.2 Search()	30
5.11.2.3 Write()	30
5.12 List< Key > Class Template Reference	31
5.12.1 Constructor & Destructor Documentation	31
5.12.1.1 List()	32
5.12.2 Member Function Documentation	32
5.12.2.1 Insert()	32
5.12.2.2 IsFull()	32
5.12.2.3 Search()	33
5.12.2.4 Write()	33
5.13 Sequence< Key > Class Template Reference	34
5.13.1 Member Function Documentation	34
5.13.1.1 Insert()	35

5.13.1.2 IsFull()	35
5.13.1.3 Search()	35
5.13.1.4 Write()	35
6 File Documentation	37
6.1 include/Block.h File Reference	37
6.1.1 Detailed Description	38
6.2 Block.h	39
6.3 include/DispersionFunction.h File Reference	39
6.3.1 Detailed Description	40
6.4 DispersionFunction.h	40
6.5 include/ExplorationFunction.h File Reference	41
6.5.1 Detailed Description	41
6.6 ExplorationFunction.h	41
6.7 include/FdModule.h File Reference	42
6.7.1 Detailed Description	42
6.8 FdModule.h	43
6.9 include/FdRandom.h File Reference	43
6.9.1 Detailed Description	44
6.10 FdRandom.h	44
6.11 include/FdSum.h File Reference	45
6.11.1 Detailed Description	45
6.12 FdSum.h	46
6.13 include/FeDoubleDispersion.h File Reference	46
6.13.1 Detailed Description	47
6.14 FeDoubleDispersion.h	48
6.15 include/FeLineal.h File Reference	48
6.15.1 Detailed Description	49
6.16 FeLineal.h	49
6.17 include/FeQuadratic.h File Reference	50
6.17.1 Detailed Description	50
6.18 FeQuadratic.h	51
6.19 include/FeRedispersion.h File Reference	51
6.19.1 Detailed Description	52
6.20 FeRedispersion.h	53
6.21 include/HashTable.h File Reference	53
6.21.1 Detailed Description	54
6.21.2 Function Documentation	54
6.21.2.1 operator<<()	54
6.22 HashTable.h	55
6.23 include/List.h File Reference	57
6.23.1 Detailed Description	58

6.24 List.h	59
6.25 include/Sequence.h File Reference	59
6.25.1 Detailed Description	60
6.26 Sequence.h	61
6.27 src/main.cc File Reference	61
6.27.1 Detailed Description	62
Index	63

Chapter 1

Scatter Search

1.1 Description

The main goal is to implement a hash table. In this case, open scattering and closed scattering can be used, also using different scattering techniques (Module, Sum, Pseudorandom) and different exploration techniques (linear, quadratic, double scattering, redispersion). Hash table is a data structure used to store and retrieve values associated with a key. It works through a hash function that takes a key as input and produces an address in the table where the corresponding value is stored.

This program was developed in C++, for more information the documentation is in the /doc directory

1.2 Usage

To compile the program, the *make* command is used in the main directory and the executable of the program is located in the /bin/main directory.

1.3 Author

Fabrizio Daniell Perilli Martín – alu0101138589@ull.edu.es

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

DispersionFunction< Key >	13
FdModule< Key >	14
FdRandom< Key >	16
FdSum< Key >	18
ExplorationFunction< Key >	13
FeDoubleDispersion< Key >	20
FeLineal< Key >	22
FeQuadratic< Key >	24
FeRedispersion< Key >	26
HashTable< Key >	27
Sequence< Key >	34
Block< Key >	9
List< Key >	31

Chapter 3

Class Index

3.1 Class List

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

Block< Key >	9
DispersionFunction< Key >	13
ExplorationFunction< Key >	13
FdModule< Key >	14
FdRandom< Key >	16
FdSum< Key >	18
FeDoubleDispersion< Key >	20
FeLineal< Key >	22
FeQuadratic< Key >	24
FeRedispersion< Key >	26
HashTable< Key >	27
List< Key >	31
Sequence< Key >	34

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

include/ Block.h	This is a sequence-derived class that contains a vector where the keys will be stored when closed dispersion is used	37
include/ DispersionFunction.h	It is an abstract class, it has a pure virtual method that is the overload of the operator() that will be implemented in the derived classes	39
include/ ExplorationFunction.h	It is an abstract class, it has a pure virtual method that is the overload of the operator() that will be implemented in the derived classes	41
include/ FdModule.h	It is a class derived from DispersionFunction that implements the operator() for the module function	42
include/ FdRandom.h	It is a class derived from DispersionFunction that implements the operator() for the pseudorandom function	43
include/ FdSum.h	It is a class derived from DispersionFunction that implements the operator() for the sum function	45
include/ FeDoubleDispersion.h	It is a class derived from ExplorationFunction.h that implements the pure virtual method to perform the double hash exploration function	46
include/ FeLineal.h	It is a class derived from ExplorationFunction.h that implements the pure virtual method to perform the linear exploration function	48
include/ FeQuadratic.h	It is a class derived from ExplorationFunction.h that implements the pure virtual method to perform the quadratic exploration function	50
include/ FeRedispersion.h	It is a class derived from ExplorationFunction.h that implements the pure virtual method to perform the redispersion exploration function	51
include/ HashTable.h	Represents the HashTable class that allows searching and inserting elements and displaying them	53
include/ List.h	This is a sequence derived class that contains a linked list where the keys will be stored when using open dispersion	57

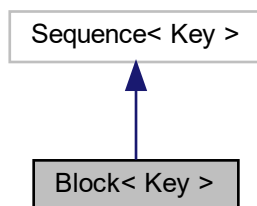
include/ Sequence.h	
Sequence is an abstract class that contains the pure virtual methods that will be implemented in the derived classes	59
src/ main.cc	
This is the main program	61

Chapter 5

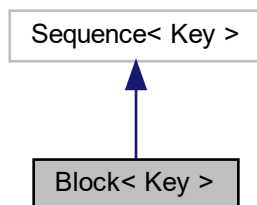
Class Documentation

5.1 Block< Key > Class Template Reference

Inheritance diagram for Block< Key >:



Collaboration diagram for Block< Key >:



Public Member Functions

- [Block](#) ()
Construct a new Block< Key>:: Block object.
- [Block](#) (const unsigned)
Construct a new Block< Key>:: Block object.
- bool [Search](#) (const Key &) const override
Search a key in the block.
- bool [Insert](#) (const Key &) override
Insert a key in the block.
- bool [IsFull](#) () const override
Check if the block is full.
- std::ostream & [Write](#) (std::ostream &) const override
Write the block in the output stream.

5.1.1 Constructor & Destructor Documentation

5.1.1.1 Block() [1/2]

```
template<class Key >
Block< Key >::Block
```

Construct a new Block< Key>:: Block object.

Template Parameters

<i>Key</i>	
------------	--

5.1.1.2 Block() [2/2]

```
template<class Key >
Block< Key >::Block (
    const unsigned size )
```

Construct a new Block< Key>:: Block object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>size</i>	
-------------	--

5.1.2 Member Function Documentation

5.1.2.1 Insert()

```
template<class Key >
bool Block< Key >::Insert (
    const Key & k ) [override], [virtual]
```

Insert a key in the block.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

true
false

Implements [Sequence< Key >](#).

5.1.2.2 IsFull()

```
template<class Key >
bool Block< Key >::IsFull [override], [virtual]
```

Check if the block is full.

Template Parameters

<i>Key</i>	
------------	--

Returns

true
false

Implements [Sequence< Key >](#).

5.1.2.3 Search()

```
template<class Key >
bool Block< Key >::Search (
    const Key & k ) const [override], [virtual]
```

Search a key in the block.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

true
false

Implements [Sequence< Key >](#).

5.1.2.4 Write()

```
template<class Key >
std::ostream & Block< Key >::Write (
    std::ostream & os ) const [override], [virtual]
```

Write the block in the output stream.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>os</i>	
-----------	--

Returns

std::ostream&

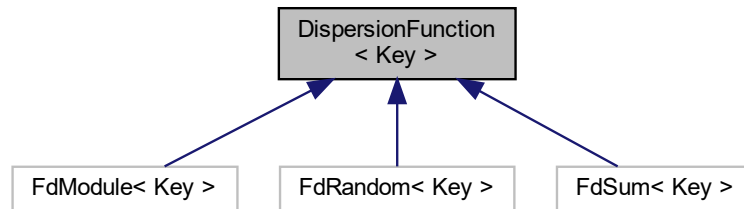
Implements [Sequence< Key >](#).

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

- include/[Block.h](#)

5.2 DispersionFunction< Key > Class Template Reference

Inheritance diagram for DispersionFunction< Key >:



Public Member Functions

- virtual unsigned [operator\(\)](#) (const Key &) const =0

5.2.1 Member Function Documentation

5.2.1.1 operator()

```

template<class Key >
virtual unsigned DispersionFunction< Key >::operator() (
    const Key & ) const [pure virtual]
  
```

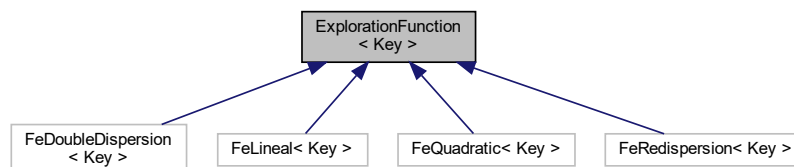
Implemented in [FdModule< Key >](#), [FdRandom< Key >](#), and [FdSum< Key >](#).

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

- include/[DispersionFunction.h](#)

5.3 ExplorationFunction< Key > Class Template Reference

Inheritance diagram for ExplorationFunction< Key >:



Public Member Functions

- virtual unsigned [operator\(\)](#) (const Key &, unsigned) const =0

5.3.1 Member Function Documentation

5.3.1.1 [operator\(\)](#)

```
template<class Key >
virtual unsigned ExplorationFunction< Key >::operator() (
    const Key & ,
    unsigned ) const [pure virtual]
```

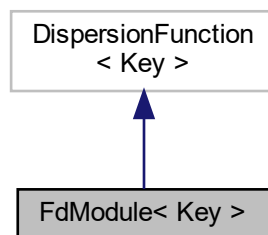
Implemented in [FeDoubleDispersion< Key >](#), [FeLineal< Key >](#), [FeQuadratic< Key >](#), and [FeRedispersion< Key >](#).

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

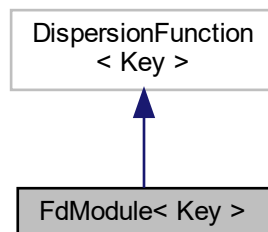
- include/[ExplorationFunction.h](#)

5.4 [FdModule< Key >](#) Class Template Reference

Inheritance diagram for [FdModule< Key >](#):



Collaboration diagram for [FdModule< Key >](#):



Public Member Functions

- [FdModule](#) (const unsigned)
Construct a new Fd Module< Key>:: Fd Module object.
- unsigned [operator\(\)](#) (const Key &) const override
It is the operator() that performs the module function.

5.4.1 Constructor & Destructor Documentation

5.4.1.1 FdModule()

```
template<class Key >
FdModule< Key >::FdModule (
    const unsigned n )
```

Construct a new Fd Module< Key>:: Fd Module object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>n</i>	
----------	--

5.4.2 Member Function Documentation

5.4.2.1 operator()()

```
template<class Key >
unsigned FdModule< Key >::operator() (
    const Key & k ) const [override], [virtual]
```

It is the operator() that performs the module function.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

unsigned

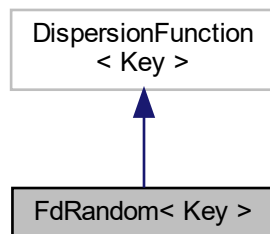
Implements [DispersionFunction< Key >](#).

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

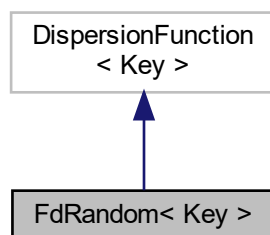
- [include/FdModule.h](#)

5.5 FdRandom< Key > Class Template Reference

Inheritance diagram for FdRandom< Key >:



Collaboration diagram for FdRandom< Key >:



Public Member Functions

- [FdRandom](#) (const unsigned)
Construct a new Fd Random< Key>:: Fd Random object.
- unsigned [operator\(\)](#) (const Key &) const override
It is the operator() that performs the pseudorandom function.

5.5.1 Constructor & Destructor Documentation

5.5.1.1 FdRandom()

```
template<class Key >
FdRandom< Key >::FdRandom (
    const unsigned n )
```

Construct a new Fd Random< Key>:: Fd Random object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>n</i>	
----------	--

5.5.2 Member Function Documentation

5.5.2.1 operator>()

```
template<class Key >
unsigned FdRandom< Key >::operator() (
    const Key & k ) const [override], [virtual]
```

It is the operator() that performs the pseudorandom function.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

unsigned

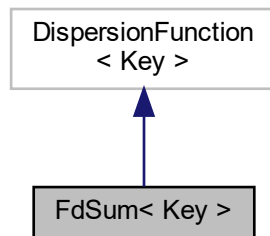
Implements [DispersionFunction< Key >](#).

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

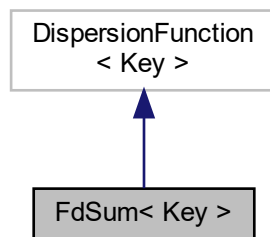
- include/[FdRandom.h](#)

5.6 FdSum< Key > Class Template Reference

Inheritance diagram for FdSum< Key >:



Collaboration diagram for FdSum< Key >:



Public Member Functions

- `FdSum` (const unsigned)
Construct a new Fd Sum< Key>:: Fd Sum object.
- unsigned `operator()` (const Key &) const override
It is the operator() that performs the sum function.

5.6.1 Constructor & Destructor Documentation

5.6.1.1 FdSum()

```

template<class Key >
FdSum< Key >::FdSum (
    const unsigned n )
  
```

Construct a new Fd Sum< Key>:: Fd Sum object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>n</i>	
----------	--

5.6.2 Member Function Documentation

5.6.2.1 operator()()

```
template<class Key >
unsigned FdSum< Key >::operator() (
    const Key & k ) const [override], [virtual]
```

It is the operator() that performs the sum function.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

unsigned

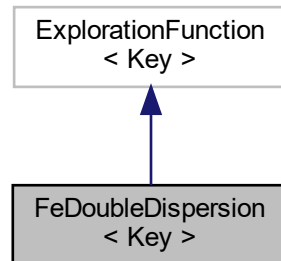
Implements [DispersionFunction< Key >](#).

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

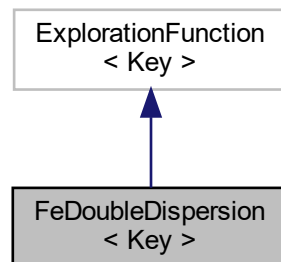
- [include/FdSum.h](#)

5.7 FeDoubleDispersion< Key > Class Template Reference

Inheritance diagram for FeDoubleDispersion< Key >:



Collaboration diagram for FeDoubleDispersion< Key >:



Public Member Functions

- [FeDoubleDispersion](#) ()
Construct a new Fe Double Dispersion< Key>:: Fe Double Dispersion object.
- [FeDoubleDispersion](#) ([DispersionFunction](#)< Key > &)
Construct a new Fe Double Dispersion< Key>:: Fe Double Dispersion object.
- unsigned [operator\(\)](#) (const Key &, unsigned) const override
It is a method that performs the double hash exploration function.

5.7.1 Constructor & Destructor Documentation

5.7.1.1 FeDoubleDispersion() [1/2]

```
template<class Key >
FeDoubleDispersion< Key >::FeDoubleDispersion
```

Construct a new Fe Double Dispersion< Key>:: Fe Double Dispersion object.

Template Parameters

<i>Key</i>	
------------	--

5.7.1.2 FeDoubleDispersion() [2/2]

```
template<class Key >
FeDoubleDispersion< Key >::FeDoubleDispersion (
    DispersionFunction< Key > & function )
```

Construct a new Fe Double Dispersion< Key>:: Fe Double Dispersion object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>function</i>	
-----------------	--

5.7.2 Member Function Documentation

5.7.2.1 operator()()

```
template<class Key >
unsigned FeDoubleDispersion< Key >::operator() (
    const Key & k,
    unsigned i ) const [override], [virtual]
```

It is a method that performs the double hash exploration function.

Template Parameters

<i>Key</i>	
------------	--

Parameters

k	
i	

Returns

unsigned

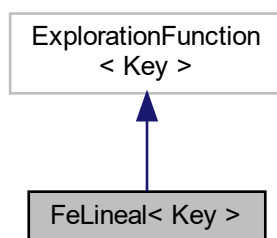
Implements [ExplorationFunction< Key >](#).

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

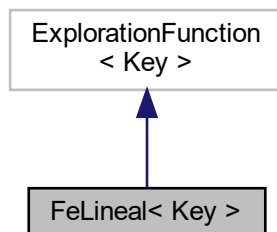
- [include/FeDoubleDispersion.h](#)

5.8 FeLineal< Key > Class Template Reference

Inheritance diagram for FeLineal< Key >:



Collaboration diagram for FeLineal< Key >:



Public Member Functions

- [FeLineal](#) ()
Construct a new Fe Lineal< Key>:: Fe Lineal object.
- unsigned [operator\(\)](#) (const Key &, unsigned) const override
It is a method that performs the linear exploration function.

5.8.1 Constructor & Destructor Documentation

5.8.1.1 FeLineal()

```
template<class Key >
FeLineal< Key >::FeLineal
```

Construct a new Fe Lineal< Key>:: Fe Lineal object.

Template Parameters

<i>Key</i>	
------------	--

5.8.2 Member Function Documentation

5.8.2.1 operator()()

```
template<class Key >
unsigned FeLineal< Key >::operator() (
    const Key & k,
    unsigned i ) const [override], [virtual]
```

It is a method that performs the linear exploration function.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
<i>i</i>	

Returns

unsigned

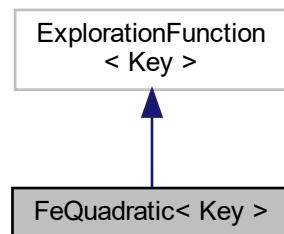
Implements [ExplorationFunction< Key >](#).

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

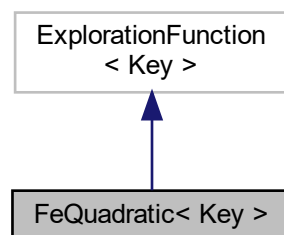
- [include/FeLineal.h](#)

5.9 FeQuadratic< Key > Class Template Reference

Inheritance diagram for FeQuadratic< Key >:



Collaboration diagram for FeQuadratic< Key >:



Public Member Functions

- [FeQuadratic](#) ()
Construct a new Fe Quadratic< Key>:: Fe Quadratic object.
- unsigned [operator\(\)](#) (const Key &, unsigned) const override
It is a method that performs the quadratic exploration function.

5.9.1 Constructor & Destructor Documentation

5.9.1.1 FeQuadratic()

```
template<class Key >
FeQuadratic< Key >::FeQuadratic
```

Construct a new Fe Quadratic< Key>:: Fe Quadratic object.

Template Parameters

<i>Key</i>	
------------	--

5.9.2 Member Function Documentation

5.9.2.1 operator>()

```
template<class Key >
unsigned FeQuadratic< Key >::operator() (
    const Key & k,
    unsigned i ) const [override], [virtual]
```

It is a method that performs the quadratic exploration function.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
<i>i</i>	

Returns

unsigned

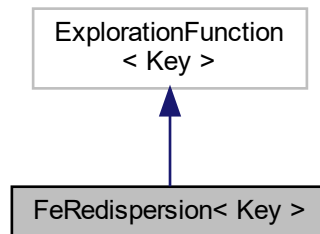
Implements [ExplorationFunction< Key >](#).

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

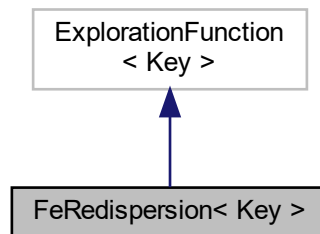
- [include/FeQuadratic.h](#)

5.10 FeRedispersion< Key > Class Template Reference

Inheritance diagram for FeRedispersion< Key >:



Collaboration diagram for FeRedispersion< Key >:



Public Member Functions

- `FeRedispersion()`
Construct a new `Fe Redispersion< Key>:: Fe Redispersion` object.
- unsigned `operator()` (const Key &, unsigned) const override
It is a method that performs the redispersion exploration function.

5.10.1 Constructor & Destructor Documentation

5.10.1.1 FeRedispersion()

```
template<class Key >
FeRedispersion< Key >::FeRedispersion
```

Construct a new `Fe Redispersion< Key>:: Fe Redispersion` object.

Template Parameters

Key	
-----	--

5.10.2 Member Function Documentation

5.10.2.1 operator()

```
template<class Key >
unsigned FeRedispersion< Key >::operator() (
    const Key & k,
    unsigned i ) const [override], [virtual]
```

It is a method that performs the redisdispersion exploration function.

Template Parameters

Key	
-----	--

Parameters

<i>k</i>	
<i>i</i>	

Returns

unsigned

Implements [ExplorationFunction< Key >](#).

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

- include/[FeRedispersion.h](#)

5.11 HashTable< Key > Class Template Reference

Public Member Functions

- [HashTable](#) ()
Construct a new Hash Table< Key>:: Hash Table object.
- [HashTable](#) (unsigned, [DispersionFunction](#)< Key > *, [ExplorationFunction](#)< Key > * = nullptr, unsigned = 0)
Construct a new Hash Table< Key>:: Hash Table object.
- [~HashTable](#) ()

Destroy the Hash Table< Key>:: Hash Table object.

- bool [Insert](#) (const Key &) const
Insert a key in the hash table.
- bool [Search](#) (const Key &) const
Search a key in the hash table.
- std::ostream & [Write](#) (std::ostream &) const
Write the hash table.

Friends

- std::ostream & **operator** (std::ostream &, const [HashTable](#)< Key > &)

5.11.1 Constructor & Destructor Documentation

5.11.1.1 HashTable() [1/2]

```
template<class Key >
HashTable< Key >::HashTable
```

Construct a new Hash Table< Key>:: Hash Table object.

Template Parameters

<i>Key</i>	
------------	--

5.11.1.2 HashTable() [2/2]

```
template<class Key >
HashTable< Key >::HashTable (
    unsigned table_size,
    DispersionFunction< Key > * fd,
    ExplorationFunction< Key > * fe = nullptr,
    unsigned block_size = 0 )
```

Construct a new Hash Table< Key>:: Hash Table object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>table_size</i>	
-------------------	--

Parameters

<i>fd</i>	
<i>fe</i>	
<i>block_size</i>	

5.11.1.3 ~HashTable()

```
template<class Key >
HashTable< Key >::~~HashTable
```

Destroy the Hash Table< Key>:: Hash Table object.

Template Parameters

<i>Key</i>	
------------	--

5.11.2 Member Function Documentation

5.11.2.1 Insert()

```
template<class Key >
bool HashTable< Key >::Insert (
    const Key & k ) const
```

Insert a key in the hash table.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

true
false

5.11.2.2 Search()

```
template<class Key >
bool HashTable< Key >::Search (
    const Key & k ) const
```

Search a key in the hash table.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

true
false

5.11.2.3 Write()

```
template<class Key >
std::ostream & HashTable< Key >::Write (
    std::ostream & os ) const
```

Write the hash table.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>os</i>	
-----------	--

Returns

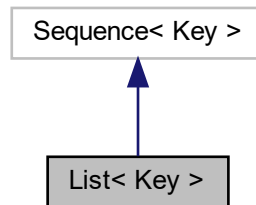
std::ostream&

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

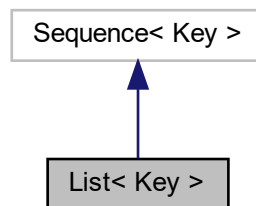
- include/[HashTable.h](#)

5.12 List< Key > Class Template Reference

Inheritance diagram for List< Key >:



Collaboration diagram for List< Key >:



Public Member Functions

- `List ()`
Construct a new List< Key>::List object.
- `bool Search (const Key &) const` override
Search a key in the list.
- `bool Insert (const Key &) override`
Insert a key in the list.
- `bool IsFull () const` override
Check if the list is full.
- `std::ostream & Write (std::ostream &) const` override
Write the list in the output stream.

5.12.1 Constructor & Destructor Documentation

5.12.1.1 List()

```
template<class Key >
List< Key >::List
```

Construct a new List< Key>:: List object.

Template Parameters

Key	
-----	--

5.12.2 Member Function Documentation

5.12.2.1 Insert()

```
template<class Key >
bool List< Key >::Insert (
    const Key & k )  [override], [virtual]
```

Insert a key in the list.

Template Parameters

Key	
-----	--

Parameters

<i>k</i>	
----------	--

Returns

true
false

Implements [Sequence< Key >](#).

5.12.2.2 IsFull()

```
template<class Key >
bool List< Key >::IsFull  [override], [virtual]
```

Check if the list is full.

Template Parameters

<i>Key</i>	
------------	--

Returns

true
false

Implements [Sequence< Key >](#).

5.12.2.3 Search()

```
template<class Key >  
bool List< Key >::Search (   
    const Key & k ) const [override], [virtual]
```

Search a key in the list.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>k</i>	
----------	--

Returns

true
false

Implements [Sequence< Key >](#).

5.12.2.4 Write()

```
template<class Key >  
std::ostream & List< Key >::Write (   
    std::ostream & os ) const [override], [virtual]
```

Write the list in the output stream.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>os</i>	
-----------	--

Returns

std::ostream&

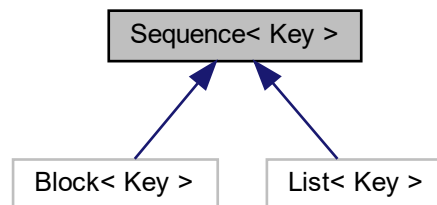
Implements [Sequence< Key >](#).

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

- include/[List.h](#)

5.13 Sequence< Key > Class Template Reference

Inheritance diagram for Sequence< Key >:



Public Member Functions

- virtual bool [Search](#) (const Key &) const =0
- virtual bool [Insert](#) (const Key &)=0
- virtual bool [IsFull](#) () const =0
- virtual std::ostream & [Write](#) (std::ostream &os) const =0

5.13.1 Member Function Documentation

5.13.1.1 Insert()

```
template<class Key >
virtual bool Sequence< Key >::Insert (
    const Key & ) [pure virtual]
```

Implemented in [Block< Key >](#), and [List< Key >](#).

5.13.1.2 IsFull()

```
template<class Key >
virtual bool Sequence< Key >::IsFull ( ) const [pure virtual]
```

Implemented in [Block< Key >](#), and [List< Key >](#).

5.13.1.3 Search()

```
template<class Key >
virtual bool Sequence< Key >::Search (
    const Key & ) const [pure virtual]
```

Implemented in [Block< Key >](#), and [List< Key >](#).

5.13.1.4 Write()

```
template<class Key >
virtual std::ostream & Sequence< Key >::Write (
    std::ostream & os ) const [pure virtual]
```

Implemented in [Block< Key >](#), and [List< Key >](#).

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

- include/[Sequence.h](#)

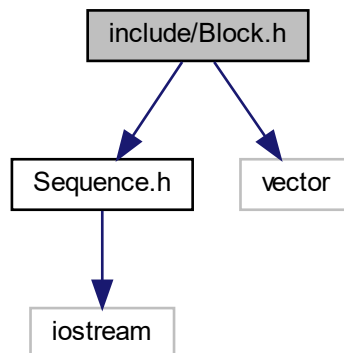
Chapter 6

File Documentation

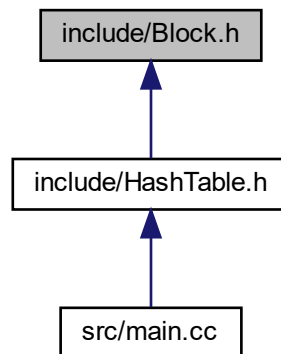
6.1 include/Block.h File Reference

This is a sequence-derived class that contains a vector where the keys will be stored when closed dispersion is used.

```
#include "Sequence.h"  
#include <vector>  
Include dependency graph for Block.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Block< Key >](#)

6.1.1 Detailed Description

This is a sequence-derived class that contains a vector where the keys will be stored when closed dispersion is used.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.2 Block.h

[Go to the documentation of this file.](#)

```

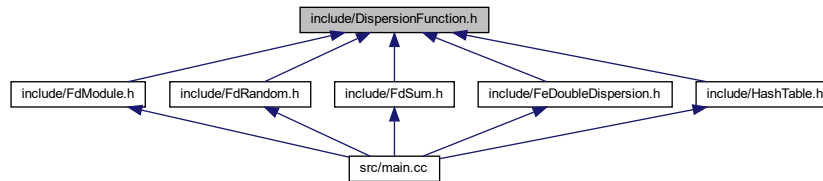
1
12 #include "Sequence.h"
13 #include <vector>
14
15 template <class Key>
16 class Block : public Sequence<Key>
17 {
18 public:
19     Block();
20     Block(const unsigned);
21     bool Search(const Key &) const override;
22     bool Insert(const Key &) override;
23     bool IsFull() const override;
24     std::ostream &Write(std::ostream &) const override;
25
26 private:
27     unsigned block_size_;
28     std::vector<Key> block_;
29 };
30
36 template <class Key>
37 Block<Key>::Block() : block_size_(0), block_(0) {}
38
45 template <class Key>
46 Block<Key>::Block(const unsigned size) : block_size_(size) {}
47
56 template <class Key>
57 bool Block<Key>::Search(const Key &k) const
58 {
59     for (auto &i : block_)
60         if (i == k)
61             return true;
62     return false;
63 }
64
73 template <class Key>
74 bool Block<Key>::Insert(const Key &k)
75 {
76     if (IsFull())
77         return false;
78     if (Search(k))
79         return false;
80     block_.push_back(k);
81     return true;
82 }
83
91 template <class Key>
92 bool Block<Key>::IsFull() const
93 {
94     return block_.size() == block_size_;
95 }
96
104 template <class Key>
105 std::ostream &Block<Key>::Write(std::ostream &os) const
106 {
107     for (auto &i : block_)
108         os << i << " ";
109     return os;
110 }

```

6.3 include/DispersionFunction.h File Reference

It is an abstract class, it has a pure virtual method that is the overload of the operator() that will be implemented in the derived classes.

This graph shows which files directly or indirectly include this file:



Classes

- class [DispersionFunction< Key >](#)

6.3.1 Detailed Description

It is an abstract class, it has a pure virtual method that is the overload of the operator() that will be implemented in the derived classes.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.4 DispersionFunction.h

[Go to the documentation of this file.](#)

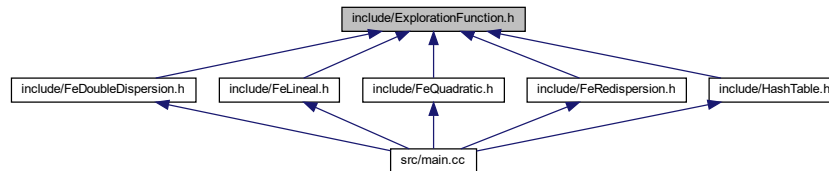
```

1
12 #pragma once
13
14 template <class Key>
15 class DispersionFunction
16 {
17 public:
18     virtual unsigned operator()(const Key &) const = 0;
19 };
  
```

6.5 include/ExplorationFunction.h File Reference

It is an abstract class, it has a pure virtual method that is the overload of the operator() that will be implemented in the derived classes.

This graph shows which files directly or indirectly include this file:



Classes

- class [ExplorationFunction< Key >](#)

6.5.1 Detailed Description

It is an abstract class, it has a pure virtual method that is the overload of the operator() that will be implemented in the derived classes.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.6 ExplorationFunction.h

[Go to the documentation of this file.](#)

```

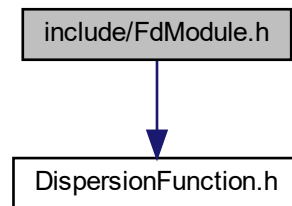
1
12 #pragma once
13
14 template <class Key>
15 class ExplorationFunction
16 {
17 public:
18     virtual unsigned operator()(const Key &, unsigned) const = 0;
19 };
  
```

6.7 include/FdModule.h File Reference

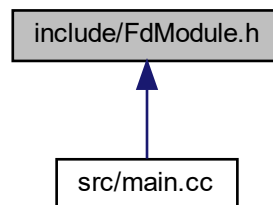
It is a class derived from DispersionFunction that implements the operator() for the module function.

```
#include "DispersionFunction.h"
```

Include dependency graph for FdModule.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [FdModule< Key >](#)

6.7.1 Detailed Description

It is a class derived from DispersionFunction that implements the operator() for the module function.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.8 FdModule.h

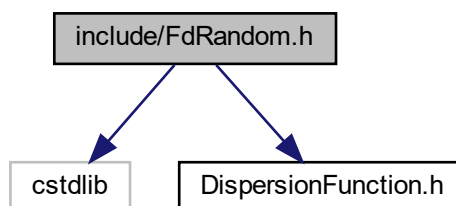
[Go to the documentation of this file.](#)

```
1
12 #include "DispersionFunction.h"
13
14 template <class Key>
15 class FdModule : public DispersionFunction<Key>
16 {
17 public:
18     FdModule(const unsigned);
19     unsigned operator()(const Key &) const override;
20
21 private:
22     unsigned table_size;
23 };
24
31 template <class Key>
32 FdModule<Key>::FdModule(const unsigned n) : table_size(n) {}
33
41 template <class Key>
42 unsigned FdModule<Key>::operator()(const Key &k) const
43 {
44     return k % table_size;
45 }
```

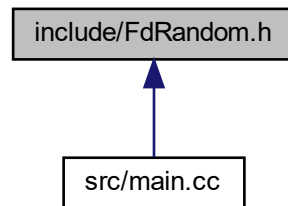
6.9 include/FdRandom.h File Reference

It is a class derived from DispersionFunction that implements the operator() for the pseudorandom function.

```
#include <cstdlib>
#include "DispersionFunction.h"
Include dependency graph for FdRandom.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [FdRandom< Key >](#)

6.9.1 Detailed Description

It is a class derived from DispersionFunction that implements the operator() for the pseudorandom function.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.10 FdRandom.h

[Go to the documentation of this file.](#)

```

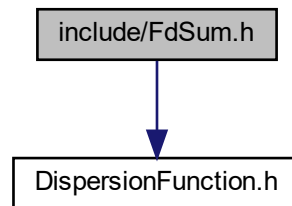
1 #include <cstdlib>
13 #include "DispersionFunction.h"
14
15 template <class Key>
16 class FdRandom : public DispersionFunction<Key>
17 {
18 public:
19     FdRandom(const unsigned);
20     unsigned operator() (const Key &) const override;
21
22 private:
23     unsigned table_size;
24 };
25
32 template <class Key>
33 FdRandom<Key>::FdRandom(const unsigned n) : table_size(n) {}
34
42 template <class Key>
43 unsigned FdRandom<Key>::operator() (const Key &k) const
44 {
45     srand(k);
46     return rand() % table_size;
47 }
  
```

6.11 include/FdSum.h File Reference

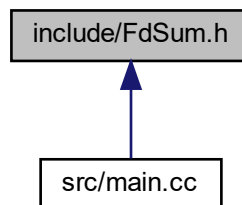
It is a class derived from DispersionFunction that implements the operator() for the sum function.

```
#include "DispersionFunction.h"
```

Include dependency graph for FdSum.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [FdSum< Key >](#)

6.11.1 Detailed Description

It is a class derived from DispersionFunction that implements the operator() for the sum function.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.12 FdSum.h

[Go to the documentation of this file.](#)

```

1
11 #include "DispersionFunction.h"
12
13 template <class Key>
14 class FdSum : public DispersionFunction<Key>
15 {
16 public:
17     FdSum(const unsigned);
18     unsigned operator()(const Key &) const override;
19
20 private:
21     unsigned table_size;
22 };
23
30 template <class Key>
31 FdSum<Key>::FdSum(const unsigned n) : table_size(n) {}
32
40 template <class Key>
41 unsigned FdSum<Key>::operator()(const Key &k) const
42 {
43     unsigned sum = 0;
44     Key temp = k;
45     while (temp > 0)
46     {
47         sum += temp % 10;
48         temp /= 10;
49     }
50     return sum % table_size;
51 }

```

6.13 include/FeDoubleDispersion.h File Reference

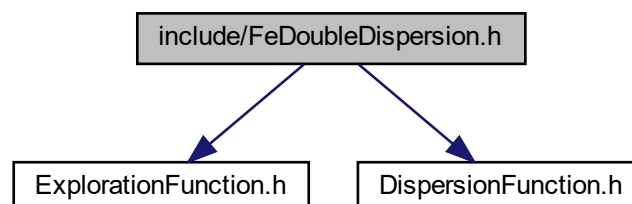
It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the double hash exploration function.

```

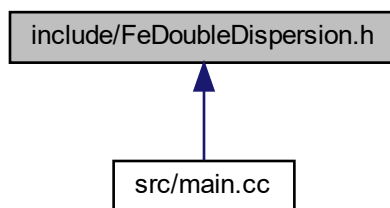
#include "ExplorationFunction.h"
#include "DispersionFunction.h"

```

Include dependency graph for FeDoubleDispersion.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [FeDoubleDispersion](#)< [Key](#) >

6.13.1 Detailed Description

It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the double hash exploration function.

Author

Fabrizio Daniell Perilli Martin

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.14 FeDoubleDispersion.h

[Go to the documentation of this file.](#)

```

1
12 #include "ExplorationFunction.h"
13 #include "DispersionFunction.h"
14
15 template <class Key>
16 class FeDoubleDispersion : public ExplorationFunction<Key>
17 {
18 public:
19     FeDoubleDispersion();
20     FeDoubleDispersion(DispersionFunction<Key> &);
21     unsigned operator()(const Key &, unsigned) const override;
22
23 private:
24     DispersionFunction<Key> *fd_;
25 };
31 template <class Key>
32 FeDoubleDispersion<Key>::FeDoubleDispersion() {}
33
40 template <class Key>
41 FeDoubleDispersion<Key>::FeDoubleDispersion(DispersionFunction<Key> &function)
42 {
43     fd_ = &function;
44 }
45
54 template <class Key>
55 unsigned FeDoubleDispersion<Key>::operator()(const Key &k, unsigned i) const
56 {
57     return fd_>operator()(k) * i;
58 }

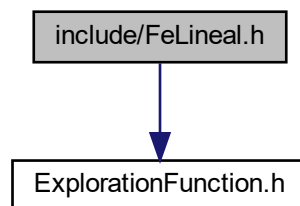
```

6.15 include/FeLineal.h File Reference

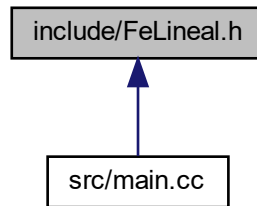
It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the linear exploration function.

```
#include "ExplorationFunction.h"
```

Include dependency graph for FeLineal.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [FeLineal< Key >](#)

6.15.1 Detailed Description

It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the linear exploration function.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.16 FeLineal.h

[Go to the documentation of this file.](#)

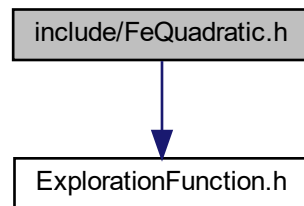
```
1
12 #include "ExplorationFunction.h"
13
14 template <class Key>
15 class FeLineal : public ExplorationFunction<Key>
16 {
17 public:
18     FeLineal();
19     unsigned operator()(const Key &, unsigned) const override;
20 };
21
27 template <class Key>
28 FeLineal<Key>::FeLineal() {}
29
38 template <class Key>
39 unsigned FeLineal<Key>::operator()(const Key &k, unsigned i) const
40 {
41     return i;
42 }
```

6.17 include/FeQuadratic.h File Reference

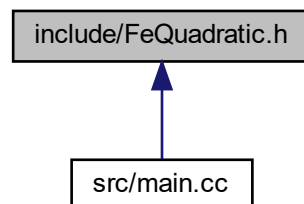
It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the quadratic exploration function.

```
#include "ExplorationFunction.h"
```

Include dependency graph for FeQuadratic.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [FeQuadratic< Key >](#)

6.17.1 Detailed Description

It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the quadratic exploration function.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.18 FeQuadratic.h

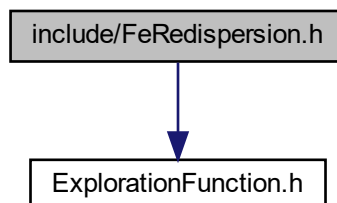
[Go to the documentation of this file.](#)

```
1
12 #include "ExplorationFunction.h"
13
14 template <class Key>
15 class FeQuadratic : public ExplorationFunction<Key>
16 {
17 public:
18     FeQuadratic();
19     unsigned operator()(const Key &, unsigned) const override;
20 };
21
27 template <class Key>
28 FeQuadratic<Key>::FeQuadratic() {}
29
38 template <class Key>
39 unsigned FeQuadratic<Key>::operator()(const Key &k, unsigned i) const
40 {
41     return i * i;
42 }
```

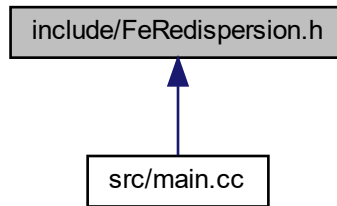
6.19 include/FeRedispersion.h File Reference

It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the redisdispersion exploration function.

```
#include "ExplorationFunction.h"
Include dependency graph for FeRedispersion.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [FeRedispersions< Key >](#)

6.19.1 Detailed Description

It is a class derived from [ExplorationFunction.h](#) that implements the pure virtual method to perform the redispersions exploration function.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.20 FeRedispersion.h

[Go to the documentation of this file.](#)

```

1
11 #include "ExplorationFunction.h"
12
13 template <class Key>
14 class FeRedispersion : public ExplorationFunction<Key>
15 {
16 public:
17     FeRedispersion();
18     unsigned operator()(const Key &, unsigned) const override;
19 };
20
26 template <class Key>
27 FeRedispersion<Key>::FeRedispersion() {}
28
37 template <class Key>
38 unsigned FeRedispersion<Key>::operator()(const Key &k, unsigned i) const
39 {
40     srand(k);
41     for (size_t j = 0; j < i; j++)
42         rand();
43     return rand();
44 }

```

6.21 include/HashTable.h File Reference

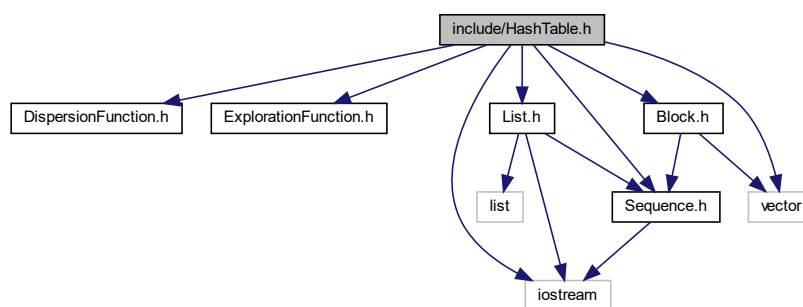
Represents the [HashTable](#) class that allows searching and inserting elements and displaying them.

```

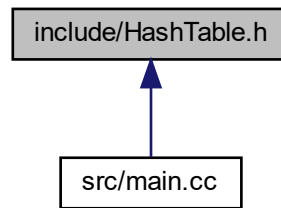
#include "DispersionFunction.h"
#include "ExplorationFunction.h"
#include "Sequence.h"
#include "List.h"
#include "Block.h"
#include <iostream>
#include <vector>

```

Include dependency graph for HashTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [HashTable< Key >](#)

Functions

- `template<class Key >`
`std::ostream & operator<< (std::ostream &os, const HashTable< Key > &ht)`
Overload the operator << to write the hash table.

6.21.1 Detailed Description

Represents the [HashTable](#) class that allows searching and inserting elements and displaying them.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.21.2 Function Documentation

6.21.2.1 `operator<<()`

```
template<class Key >
std::ostream & operator<< (
    std::ostream & os,
    const HashTable< Key > & ht )
```

Overload the operator << to write the hash table.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>os</i>	
<i>ht</i>	

Returns

std::ostream&

6.22 HashTable.h

[Go to the documentation of this file.](#)

```

1
12 #pragma once
13 #include "DispersionFunction.h"
14 #include "ExplorationFunction.h"
15 #include "Sequence.h"
16 #include "List.h"
17 #include "Block.h"
18 #include <iostream>
19 #include <vector>
20
21 template <class Key>
22 class HashTable;
23
24 template <class Key>
25 std::ostream &operator<<(std::ostream &, const HashTable<Key> &);
26
27 template <class Key>
28 class HashTable
29 {
30 public:
31     HashTable();
32     HashTable(unsigned, DispersionFunction<Key> *, ExplorationFunction<Key> * = nullptr, unsigned = 0);
33     ~HashTable();
34     bool Insert(const Key &) const;
35     bool Search(const Key &) const;
36     std::ostream &Write(std::ostream &) const;
37     friend std::ostream &operator<< <Key>(std::ostream &, const HashTable<Key> &);
38
39 private:
40     DispersionFunction<Key> *fd_;
41     ExplorationFunction<Key> *fe_;
42     unsigned table_size_;
43     std::vector<Sequence<Key> *> table_;
44     unsigned block_size_;
45 };
46
47
48
49
50
51
52 template <class Key>
53 HashTable<Key>::HashTable() {}
54
55
56
57
58
59
60
61
62
63
64 template <class Key>
65 HashTable<Key>::HashTable(unsigned table_size, DispersionFunction<Key> *fd, ExplorationFunction<Key> *fe,
66     unsigned block_size)
67 {
68     table_size_ = table_size;
69     block_size_ = block_size;
70     fd_ = fd;
71     fe_ = fe;
72
73     for (size_t i = 0; i < table_size_; i++)
74     {
75         if (fe_ == nullptr)
76             table_.push_back(new List<Key>());
77         else
78             table_.push_back(new Block<Key>(block_size_));
79     }
80 }

```

```

80
86 template <class Key>
87 HashTable<Key>::~~HashTable()
88 {
89     for (size_t i = 0; i < table_size_; i++)
90         delete table_[i];
91 }
92
101 template <class Key>
102 bool HashTable<Key>::Insert(const Key &k) const
103 {
104     unsigned index = fd_>operator()(k);
105
106     if (fe_ == nullptr)
107     {
108         if (table_[index]>Insert(k))
109         {
110             std::cout << "Key inserted in position: " << index << std::endl;
111             return true;
112         }
113         else
114         {
115             std::cout << "Key already exists" << std::endl;
116             return false;
117         }
118     }
119     else
120     {
121         for (size_t i = 0; i < table_size_; ++i)
122         {
123             unsigned exploration_index = (index + fe_>operator()(k, i)) % table_size_;
124             if (table_[exploration_index]>Insert(k))
125             {
126                 std::cout << "Key inserted in position: " << exploration_index << std::endl;
127                 return true;
128             }
129             else if (table_[exploration_index]>IsFull())
130                 continue;
131             else
132             {
133                 std::cout << "Key already exists" << std::endl;
134                 return false;
135             }
136         }
137     }
138     std::cout << "Key not inserted table is full" << std::endl;
139     return false;
140 }
141
150 template <class Key>
151 bool HashTable<Key>::Search(const Key &k) const
152 {
153     unsigned index = fd_>operator()(k);
154
155     if (fe_ == nullptr)
156     {
157         if (table_[index]>Search(k))
158         {
159             std::cout << "Key found in position: " << index << std::endl;
160             return true;
161         }
162         else
163         {
164             std::cout << "Key not found" << std::endl;
165             return false;
166         }
167     }
168     else
169     {
170         for (size_t i = 0; i < table_size_; ++i)
171         {
172             unsigned exploration_index = (index + fe_>operator()(k, i)) % table_size_;
173             if (table_[exploration_index]>Search(k))
174             {
175                 std::cout << "Key found in position: " << exploration_index << std::endl;
176                 return true;
177             }
178             else if (table_[exploration_index]>IsFull())
179                 continue;
180             else
181             {
182                 std::cout << "Key not found" << std::endl;
183                 return false;
184             }
185         }
186     }
187     return false;

```

```

188 }
189
197 template <class Key>
198 std::ostream &HashTable<Key>::Write(std::ostream &os) const
199 {
200     for (size_t i = 0; i < table_size_; i++)
201     {
202         os << "\ni: " << i << " --> ";
203         table_[i]->Write(os);
204     }
205
206     return os;
207 }
208
217 template <class Key>
218 std::ostream &operator<<(std::ostream &os, const HashTable<Key> &ht)
219 {
220     return ht.Write(os);
221 }

```

6.23 include/List.h File Reference

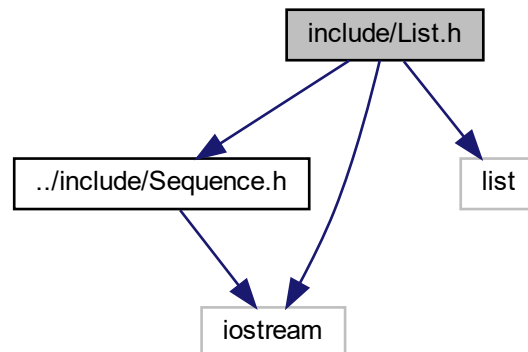
This is a sequence derived class that contains a linked list where the keys will be stored when using open dispersion.

```

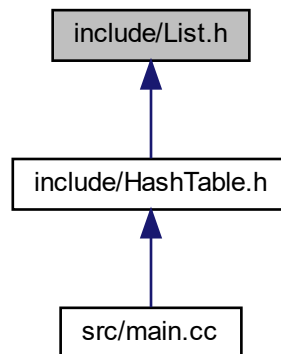
#include "../include/Sequence.h"
#include <list>
#include <iostream>

```

Include dependency graph for List.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [List< Key >](#)

6.23.1 Detailed Description

This is a sequence derived class that contains a linked list where the keys will be stored when using open dispersion.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.24 List.h

[Go to the documentation of this file.](#)

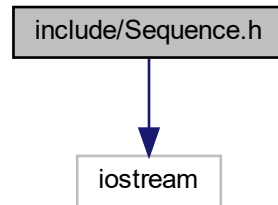
```
1
12 #include "../include/Sequence.h"
13 #include <list>
14 #include <iostream>
15
16 template <class Key>
17 class List : public Sequence<Key>
18 {
19 public:
20     List();
21     bool Search(const Key &) const override;
22     bool Insert(const Key &) override;
23     bool IsFull() const override;
24     std::ostream &Write(std::ostream &) const override;
25
26 private:
27     std::list<Key> list_;
28 };
29
30 template <class Key>
31 List<Key>::List() {}
32
33 template <class Key>
34 bool List<Key>::Search(const Key &k) const
35 {
36     for (auto &i : list_)
37         if (i == k)
38             return true;
39     return false;
40 }
41
42 template <class Key>
43 bool List<Key>::Insert(const Key &k)
44 {
45     if (IsFull())
46         return false;
47     if (Search(k))
48         return false;
49     list_.push_back(k);
50     return true;
51 }
52
53 template <class Key>
54 bool List<Key>::IsFull() const
55 {
56     return false;
57 }
58
59 template <class Key>
60 std::ostream &List<Key>::Write(std::ostream &os) const
61 {
62     for (auto &i : list_)
63         os << i << " ";
64     return os;
65 }
```

6.25 include/Sequence.h File Reference

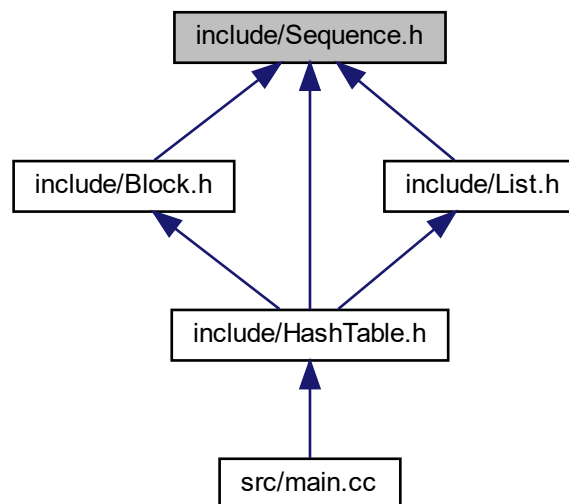
[Sequence](#) is an abstract class that contains the pure virtual methods that will be implemented in the derived classes.

```
#include <iostream>
```

Include dependency graph for Sequence.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Sequence< Key >](#)

6.25.1 Detailed Description

[Sequence](#) is an abstract class that contains the pure virtual methods that will be implemented in the derived classes.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

6.26 Sequence.h

[Go to the documentation of this file.](#)

```

1
12 #pragma once
13 #include <iostream>
14
15 template <class Key>
16 class Sequence
17 {
18 public:
19     virtual bool Search(const Key &) const = 0;
20     virtual bool Insert(const Key &) = 0;
21     virtual bool IsFull() const = 0;
22     virtual std::ostream &Write(std::ostream &os) const = 0;
23 };

```

6.27 src/main.cc File Reference

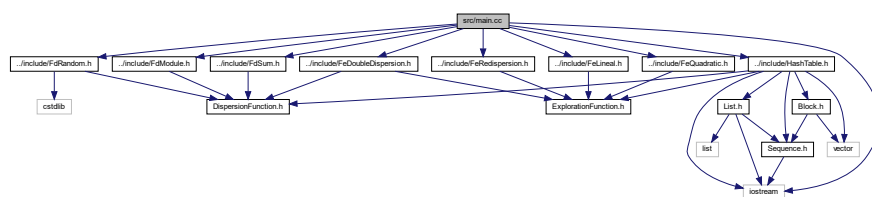
This is the main program.

```

#include <iostream>
#include "../include/FdModule.h"
#include "../include/FdSum.h"
#include "../include/FdRandom.h"
#include "../include/FeLineal.h"
#include "../include/FeQuadratic.h"
#include "../include/FeDoubleDispersion.h"
#include "../include/FeRedispersion.h"
#include "../include/HashTable.h"

```

Include dependency graph for main.cc:



Functions

- `int main ()`

6.27.1 Detailed Description

This is the main program.

Author

Fabrizio Daniell Perilli Martín

Version

0.1

Date

2023-03-12

Copyright

Copyright (c) 2023

Index

~HashTable
 HashTable< Key >, 29

Block
 Block< Key >, 10

Block< Key >, 9
 Block, 10
 Insert, 11
 IsFull, 11
 Search, 11
 Write, 12

DispersionFunction< Key >, 13
 operator(), 13

ExplorationFunction< Key >, 13
 operator(), 14

FdModule
 FdModule< Key >, 15
FdModule< Key >, 14
 FdModule, 15
 operator(), 15

FdRandom
 FdRandom< Key >, 17
FdRandom< Key >, 16
 FdRandom, 17
 operator(), 17

FdSum
 FdSum< Key >, 18
FdSum< Key >, 18
 FdSum, 18
 operator(), 19

FeDoubleDispersion
 FeDoubleDispersion< Key >, 20, 21
FeDoubleDispersion< Key >, 20
 FeDoubleDispersion, 20, 21
 operator(), 21

FeLineal
 FeLineal< Key >, 23
FeLineal< Key >, 22
 FeLineal, 23
 operator(), 23

FeQuadratic
 FeQuadratic< Key >, 25
FeQuadratic< Key >, 24
 FeQuadratic, 25
 operator(), 25

FeRedispersion
 FeRedispersion< Key >, 26

FeRedispersion< Key >, 26
 FeRedispersion, 26
 operator(), 27

HashTable
 HashTable< Key >, 28
HashTable< Key >, 27
 ~HashTable, 29
 HashTable, 28
 Insert, 29
 Search, 29
 Write, 30

HashTable.h
 operator<<, 54

include/Block.h, 37, 39
include/DispersionFunction.h, 39, 40
include/ExplorationFunction.h, 41
include/FdModule.h, 42, 43
include/FdRandom.h, 43, 44
include/FdSum.h, 45, 46
include/FeDoubleDispersion.h, 46, 48
include/FeLineal.h, 48, 49
include/FeQuadratic.h, 50, 51
include/FeRedispersion.h, 51, 53
include/HashTable.h, 53, 55
include/List.h, 57, 59
include/Sequence.h, 59, 61

Insert
 Block< Key >, 11
 HashTable< Key >, 29
 List< Key >, 32
 Sequence< Key >, 34

IsFull
 Block< Key >, 11
 List< Key >, 32
 Sequence< Key >, 35

List
 List< Key >, 31
List< Key >, 31
 Insert, 32
 IsFull, 32
 List, 31
 Search, 33
 Write, 33

operator<<
 HashTable.h, 54
operator()

DispersionFunction< Key >, [13](#)
ExplorationFunction< Key >, [14](#)
FdModule< Key >, [15](#)
FdRandom< Key >, [17](#)
FdSum< Key >, [19](#)
FeDoubleDispersion< Key >, [21](#)
FeLineal< Key >, [23](#)
FeQuadratic< Key >, [25](#)
FeRedispersion< Key >, [27](#)

Search

Block< Key >, [11](#)
HashTable< Key >, [29](#)
List< Key >, [33](#)
Sequence< Key >, [35](#)

Sequence< Key >, [34](#)

Insert, [34](#)
IsFull, [35](#)
Search, [35](#)
Write, [35](#)

src/main.cc, [61](#)

Write

Block< Key >, [12](#)
HashTable< Key >, [30](#)
List< Key >, [33](#)
Sequence< Key >, [35](#)