My Project

Generated by Doxygen 1.9.8

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 Multiset Class Reference

Implementation of the Multiset class.

```
#include <Multitest.h>
```

Public Member Functions

- void parsStrToSet (const string &elements)
- string toString () const
- size_t countEls (const string &element) const
- size_t countSubs (const Multiset &subset) const
- bool findEls (const string &element) const
- bool findSubs (const Multiset &subset) const
- void addToRes (const string &element, size_t amount)
- void addToRes (const Multiset &set, size_t amount)
- void addAll (const Multiset &set1, const Multiset &set2)
- Multiset ()
- Multiset (const string &elements)
- Multiset (const char ∗elements)
- Multiset (const Multiset &set)
- bool isEmpty () const
- Multiset boolean () const
- size_t getCardinality () const
- void addElement (const string &element)
- void addElement (const char ∗element)
- void addSubset (const Multiset &set)
- void delElement (const string &element)
- void delElement (const char ∗element)
- void delSubset (const Multiset &set)
- bool operator[ ] (const string &element) const
- bool operator[ ] (const char ∗element) const
- bool operator[ ] (const Multiset &subset) const
- bool operator== (const Multiset &Set) const
- bool operator!= (const Multiset &Set) const
- Multiset operator+ (const Multiset &Set) const

- Multiset operator+= (const Multiset &Set)
- Multiset operator∗ (const Multiset &Set) const
- Multiset operator∗= (const Multiset &Set)
- Multiset operator- (const Multiset &Set) const
- Multiset operator-= (const Multiset &Set)
- void printSetInfo () const
- void printSet () const

Public Attributes

- vector< string > **elements_**
- vector< Multiset > **subsets_**
- size_t **cardinality_**

3.1.1 Detailed Description

Implementation of the Multiset class.

Author

Verkovich E. V.

Version

1.0

Date

October 2023

Multiset class contains 2 vector fields for storing multiset elements, a field showing the capacity of the set and its maximum nesting; a constructor with an initialization list, with a parameter, as well as a copy constructor; methods for adding and removing elements; a method for determining the emptiness of the set; a method for forming the booleans of the set; overloaded operators "+", "+=", "∗", "∗=", "-", "-=", "[ ]"; other additional methods.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Multiset() `[1/4]`

```
Multiset::Multiset ( )
```

constructor with initialization list

3.1.2.2 Multiset() `[2/4]`

```
Multiset::Multiset (
             const string & element )
```

constructor with parameter

Parameters

element	

3.1.2.3 Multiset() [3/4]

```
Multiset::Multiset (
             const char * element )
```

constructor with parameter

Parameters

element	

3.1.2.4 Multiset() [4/4]

```
Multiset::Multiset (
             const Multiset & set )
```

copy constructor

Parameters

set	

3.1.3 Member Function Documentation

3.1.3.1 addAll()

```
void Multiset::addAll (
             const Multiset & set1,
             const Multiset & set2 )
```

subfunction for the "+" operator

Parameters

set1,set2	

3.1.3.2 addElement() [1/2]

```
void Multiset::addElement (
             const char * element )
```

adds an element to set

Parameters

element	

3.1.3.3 addElement() [2/2]

```
void Multiset::addElement (
             const string & element )
```

adds an element to set

Parameters

element	

3.1.3.4 addSubset()

```
void Multiset::addSubset (
             const Multiset & set )
```

adds a subset to set

Parameters

set	

3.1.3.5 addToRes() [1/2]

```
void Multiset::addToRes (
             const Multiset & set,
             size_t amount )
```

overloaded function addToRes()

Parameters

set,amount	

3.1.3.6 addToRes() [2/2]

```
void Multiset::addToRes (
             const string & element,
             size_t amount )
```

adds string element to vector <string> elements amount times

Parameters

element,amount	

3.1.3.7 boolean()

```
Multiset Multiset::boolean ( ) const
```

a method for forming the booleans of the set

Returns

result boolean set

3.1.3.8 countEls()

```
size_t Multiset::countEls (
             const string & element ) const
```

counts occurrences of a given element in the set

Parameters

element	an element, the occurrences of which are counted

Returns

result of std::count() function

3.1.3.9 countSubs()

```
size_t Multiset::countSubs (
             const Multiset & subset ) const
```

overloaded function countEls()

Parameters

subset	

Returns

result of std::count() function

3.1.3.10 delElement() [1/2]

```
void Multiset::delElement (
             const char * element )
```

deletes an element from set

Parameters

element	

3.1.3.11 delElement() [2/2]

```
void Multiset::delElement (
             const string & element )
```

deletes an element from set

Parameters

element	

3.1.3.12 delSubset()

```
void Multiset::delSubset (
             const Multiset & set )
```

deletes a subset from set

Parameters

set	

3.1.3.13 findEls()

```
bool Multiset::findEls (
             const string & element ) const
```

determines whether an element in the set

Parameters

element	an element, the occurrences of which are counted

Returns

result of std::find() function

3.1.3.14 findSubs()

```
bool Multiset::findSubs (
             const Multiset & subset ) const
```

overloaded function findEls()

Parameters

subset	

Returns

result of std::find() function

3.1.3.15 getCardinality()

```
size_t Multiset::getCardinality ( ) const
```

cardinality_ getter

3.1.3.16 isEmpty()

```
bool Multiset::isEmpty ( ) const
```

a method for determining the emptiness of the set

Returns

result for std::empty() finction for elements_ and subsets_

3.1.3.17 operator"!=()

```
bool Multiset::operator!= (
             const Multiset & set ) const
```

overloaded operator "!="

Parameters

set	

Returns

true if two sets are not equal

3.1.3.18 operator∗()

```
Multiset Multiset::operator∗ (
            const Multiset & set ) const
```

overloaded operator "∗"

Parameters

set	

Returns

intersection of current set and parameter-set

3.1.3.19 operator∗=()

```
Multiset Multiset::operator∗= (
            const Multiset & set )
```

overloaded operator "∗="

Parameters

set	

Returns

intersection of current set and parameter-set that assigned to the current set

3.1.3.20 operator+()

```
Multiset Multiset::operator+ (
            const Multiset & set ) const
```

overloaded operator "+"

Parameters

set	

Returns

union of current set and parameter-set

3.1.3.21 operator+=()

```
Multiset Multiset::operator+= (
             const Multiset & set )
```

overloaded operator "+="

Parameters

set	

Returns

union of current set and parameter-set that assigned to the current set

3.1.3.22 operator-()

```
Multiset Multiset::operator- (
             const Multiset & set ) const
```

overloaded operator "-"

Parameters

set	

Returns

difference of current set and parameter-set

3.1.3.23 operator-=()

```
Multiset Multiset::operator-= (
             const Multiset & set )
```

overloaded operator "-="

Parameters

set	

Returns

difference of current set and parameter-set that assigned to the current set

3.1.3.24 operator==()

```
bool Multiset::operator== (
             const Multiset & set ) const
```

overloaded operator "=="

Parameters

set	

Returns

true if two sets are equal

3.1.3.25 operator[]() [1/3]

```
bool Multiset::operator[] (
            const char * element ) const
```

overloaded operator "[ ]"

Parameters

element	

Returns

result of findEls() function

3.1.3.26 operator[]() [2/3]

```
bool Multiset::operator[] (
            const Multiset & subset ) const
```

overloaded operator "[ ]"

Parameters

subset	

Returns

result of findEls() function

3.1.3.27 operator[]() [3/3]

```
bool Multiset::operator[] (
            const string & element ) const
```

overloaded operator "[ ]"

Parameters

element	

Returns

result of findEls() function

3.1.3.28 parsStrToSet()

```
void Multiset::parsStrToSet (
             const string & setString )
```

parsing a string into a set

Parameters

setString	a string from which a set is formed

3.1.3.29 printSet()

```
void Multiset::printSet ( ) const
```

Console output of a set

3.1.3.30 printSetInfo()

```
void Multiset::printSetInfo ( ) const
```

Console output of a set, its capacity and whether it is empty

3.1.3.31 toString()

```
string Multiset::toString ( ) const
```

forms a string from the class object

Returns

a string

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

- Multitest.h
- mmultisset.cpp

Chapter 4

File Documentation

4.1 Multitest.h

```
#pragma once
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
using namespace std;
class Multiset
{
public:
    vector<string>elements_;
    vector<Multiset>subsets_;
    size_t cardinality_;

    void parsStrToSet(const string& elements);
    string toString() const;
    size_t countEls(const string& element) const;
    size_t countSubs(const Multiset& subset) const;
    bool findEls(const string& element) const;
    bool findSubs(const Multiset& subset) const;

    void addToRes(const string& element, size_t amount);
    void addToRes(const Multiset& set, size_t amount);
    void addAll(const Multiset& set1, const Multiset& set2);
public:
    Multiset();
    Multiset(const string& elements);
    Multiset(const char* elements);
    Multiset(const Multiset& set);

    bool isEmpty() const;
    Multiset boolean() const;
    size_t getCardinality() const;

    void addElement(const string& element);
    void addElement(const char* element);
    void addSubset(const Multiset& set);
    void delElement(const string& element);
    void delElement(const char* element);
    void delSubset(const Multiset& set);

    bool operator[](const string& element) const;
    bool operator[](const char* element) const;
    bool operator[](const Multiset& subset) const;
    bool operator==(const Multiset& Set) const;
    bool operator!=(const Multiset& Set) const;
    Multiset operator+(const Multiset& Set) const;
    Multiset operator+= (const Multiset& Set);
    Multiset operator*(const Multiset& Set) const;
    Multiset operator*= (const Multiset& Set);
    Multiset operator-(const Multiset& Set) const;
    Multiset operator-= (const Multiset& Set);

    void printSetInfo() const;
    void printSet() const;
};
```

Index