Lab1

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 real_matrix Class Reference

Public Member Functions

- **real_matrix** ()
  Implementation of the real matrix class.
- **real_matrix** (int _rows, int _cols)
  Class constructor with parameters.
- **real_matrix** (const char *file_name)
  Class constructor with parameters.
- **real_matrix** (const **real_matrix** &other)
  Copy constructor.
- ~**real_matrix** ()
  Class destructor.
- **real_matrix** & **operator++** ()
  preincrement operator. Changes content returns option after change
- **real_matrix** **operator++** (int)
  post-increment operator. Changes the content returns the option before the change
- **real_matrix** & **operator--** ()
  predecrement operator. Changes content returns option after change
- **real_matrix** **operator--** (int)
  postdecrement operator. Changes the content returns the option before the change
- int **get_rows** () const
  row getter
- int **get_cols** () const
  column getter
- double **get_element** (int _i, int _j) const
  element getter
- bool **quad_check** ()
  Helper method for checking if a matrix is square.
- bool **diagonal_check** ()
  Helper method for checking if a matrix is diagonal.
- bool **zero_check** ()
  Helper method for checking if a matrix is zero.
- bool **identity_check** ()

Helper method for checking if a matrix is identity.

- bool **symmetrical_check** ()

  Helper method for checking if a matrix is symmetrical.
- bool **uptriang_check** ()

  Helper method for checking if a matrix is upper triangle.
- bool **lowtriang_check** ()

  Helper method for checking if a matrix is lower triangle.
- void **determine_type** ()

  the main method for determining the type of matrix. Calls helper methods and prints the type to the console
- void **transponse** ()

  matrix transposition method <3
- void **change_form** (int _new_rows, int _new_cols)

  method for changing the shape of the matrix. If the new matrix is smaller, then the original matrix is cut off from the upper left corner to accommodate the new values; if more, then the new rows and columns are filled with zeros. :^)
- **real_matrix sub_matrix** (int _new_rows, int _new_cols)

  method for extracting a submatrix. If the new matrix is smaller, then the original matrix is cut off from the upper left corner to accommodate the new values, if >= then I return the original matrix.

Friends

- std::ostream & **operator<<** (std::ostream &os, const **real_matrix** &rm)

  Overloading the stream output operator.

3.1.1 Constructor & Destructor Documentation

3.1.1.1 real_matrix() [1/3]

```
real_matrix::real_matrix ( )
```

Implementation of the real matrix class.

Author

Aliteya

Version

2.5

Date

September 2023

Contains a default constructor, with specified dimensions and reading a matrix from a file; operators ++ and – for a given matrix; cout thread overload; transposition methods, type definitions and selecting a submatrix of a given size.

default class constructor 0 rows, 0 columns

3.1.1.2 real_matrix() [2/3]

```
real_matrix::real_matrix (
            int _rows,
            int _cols )
```

Class constructor with parameters.

Parameters

_rows,_cols	Rows, columns.

3.1.1.3 real_matrix() [3/3]

```
real_matrix::real_matrix (
            const char * file_name )
```

Class constructor with parameters.

Parameters

filename	the file from which the matrix is read

3.1.2 Member Function Documentation

3.1.2.1 change_form()

```
void real_matrix::change_form (
            int _new_rows,
            int _new_cols )
```

method for changing the shape of the matrix. If the new matrix is smaller, then the original matrix is cut off from the upper left corner to accommodate the new values; if more, then the new rows and columns are filled with zeros. :^)

Parameters

_new_rows,_new_cols	new matrix form

Warning

!!!THE MATRIX ITSELF IS CHANGING!!!

3.1.2.2 diagonal_check()

```
bool real_matrix::diagonal_check ( )
```

Helper method for checking if a matrix is diagonal.

Returns

True if yes, False if no

3.1.2.3 identity_check()

```
bool real_matrix::identity_check ( )
```

Helper method for checking if a matrix is identity.

Returns

True if yes, False if no

3.1.2.4 lowtriang_check()

```
bool real_matrix::lowtriang_check ( )
```

Helper method for checking if a matrix is lower triangle.

Returns

True if yes, False if no

3.1.2.5 operator++() [1/2]

```
real_matrix & real_matrix::operator++ ( )
```

preincrement operator. Changes content returns option after change

Returns

∗this is the modified object itself

3.1.2.6 operator++() [2/2]

```
real_matrix real_matrix::operator++ (
            int  )
```

post-increment operator. Changes the content returns the option before the change

Returns

copy_matr copy of the object before modification

3.1.2.7 operator--() [1/2]

```
real_matrix & real_matrix::operator-- ( )
```

predecrement operator. Changes content returns option after change

Returns

∗this is the modified object itself

3.1.2.8 operator--() [2/2]

```
real_matrix real_matrix::operator-- (
            int  )
```

postdecrement operator. Changes the content returns the option before the change

Returns

copy_matr copy of the object before modification

3.1.2.9 quad_check()

```
bool real_matrix::quad_check ( )
```

Helper method for checking if a matrix is square.

Returns

True if yes, False if no

3.1.2.10 sub_matrix()

```
real_matrix real_matrix::sub_matrix (
            int _new_rows,
            int _new_cols )
```

method for extracting a submatrix. If the new matrix is smaller, then the original matrix is cut off from the upper left corner to accommodate the new values, if >= then I return the original matrix.

Parameters

_new_rows,_new_cols	new matrix form

Returns

changed_matr submatrix cut from a copy of the original

Warning

!!!THE COPY OF THE MATRIX IS CHANGING!!!

3.1.2.11 symmetrical_check()

```
bool real_matrix::symmetrical_check ( )
```

Helper method for checking if a matrix is symmetrical.

Returns

True if yes, False if no

3.1.2.12 transponse()

```
void real_matrix::transponse ( )
```

matrix transposition method <3

Warning

!!!THE MATRIX ITSELF IS CHANGING!!!

3.1.2.13 uptriang_check()

```
bool real_matrix::uptriang_check ( )
```

Helper method for checking if a matrix is upper triangle.

Returns

True if yes, False if no

3.1.2.14 zero_check()

```
bool real_matrix::zero_check ( )
```

Helper method for checking if a matrix is zero.

Returns

True if yes, False if no

3.1.3 Friends And Related Symbol Documentation

3.1.3.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const  real_matrix & rm )  [friend]
```

Overloading the stream output operator.

Parameters

os	An instance of the ostream class
rm	An instance of the class **real_matrix** (p. 5)

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

- real_matrix.h
- real_matrix.cpp

Chapter 4

File Documentation

4.1 real_matrix.h

```
00001 #pragma once
00002 #ifndef REAL_MATRIX_H
00003 #define REAL_MATRIX_H
00004
00005 #include <iostream>
00006 #include <fstream>
00007
00008 class real_matrix {
00009 private:
00010     int rows;
00011     int cols;
00012     double** content;
00013 public:
00014     real_matrix();
00015     real_matrix(int _rows, int _cols);
00016     real_matrix(const char* file_name);
00017     real_matrix(const real_matrix& other);
00018     ~real_matrix();
00019     real_matrix& operator++();
00020     real_matrix operator++(int);
00021     real_matrix& operator--();
00022     real_matrix operator--(int);
00023     int get_rows() const;
00024     int get_cols() const;
00025     double get_element(int _i, int _j) const;
00026     friend std::ostream& operator«(std::ostream& os, const real_matrix& rm);
00027     bool quad_check();
00028     bool diagonal_check();
00029     bool zero_check();
00030     bool identity_check();
00031     bool symmetrical_check();
00032     bool uptriang_check();
00033     bool lowtriang_check();
00034     void determine_type();
00035     void transponse();
00036     void change_form(int _new_rows, int _new_cols);
00037     real_matrix sub_matrix(int _new_rows, int _new_cols);
00038
00039 };
00040
00041 #endif
```

Index