S21 Matrix

Generated by Doxygen 1.9.1

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 S21Matrix Class Reference

Class representing a matrix in a mathematical context.

`#include <s21_matrix_oop.h>`

Public Member Functions

- S21Matrix ()
  Default constructor.
- S21Matrix (int rows, int cols)
  Constructor with given dimensions.
- S21Matrix (const S21Matrix &other)
  Copy constructor.
- S21Matrix (S21Matrix &&other) noexcept
  Move constructor.
- ~S21Matrix ()
  Destructor.
- bool EqSizeMatrix (const S21Matrix &other) const
  Checking the equality of matrix sizes.
- bool checkSize () const
  function to check the correctness of the matrix size
- bool EqMatrix (const S21Matrix &other) const
  Checks matrices for equality with each other.
- void SumMatrix (const S21Matrix &other)
  Adds the second matrix to the current one.
- void SubMatrix (const S21Matrix &other)
  Subtracts another matrix from the current one.
- void MulNumber (const double num)
  Multiplies the current matrix by a number.
- void MulMatrix (const S21Matrix &other)
  Multiplies the current matrix by the second matrix.
- S21Matrix Transpose ()
  Creates a new transposed matrix from the current one and returns it.

- S21Matrix CalcComplements ()

  Calculates the algebraic addition matrix of the current one and returns it.
- double Determinant () const

  Calculates and returns the determinant of the current matrix.
- S21Matrix InverseMatrix ()

  Calculates the inverse matrix.
- S21Matrix & operator= (S21Matrix &&other) noexcept

  Moves the contents of a matrix from another object to the current object.
- S21Matrix & operator= (const S21Matrix &other)

  Overloaded assignment operator for copying a matrix.
- S21Matrix operator+ (const S21Matrix &other) const

  Overloaded matrix addition operator.
- S21Matrix operator- (const S21Matrix &other) const

  Overloaded matrix subtraction operator.
- S21Matrix operator∗ (const S21Matrix &other) const

  Overloaded matrix multiplication operator.
- S21Matrix operator∗ (const double &number) const

  Overloaded operator for multiplying a matrix by a number.
- bool operator== (const S21Matrix &other) const

  Overloaded operator for comparing two matrices for equality.
- S21Matrix & operator+= (const S21Matrix &other)

  Overloaded matrix addition operator.
- S21Matrix & operator-= (const S21Matrix &other)

  Overloaded matrix subtraction operator.
- S21Matrix & operator∗= (const S21Matrix &other)

  Overloaded matrix multiplication operator.
- S21Matrix & operator∗= (const double &number)

  Overloaded operator for multiplying a matrix by a number.
- double & operator() (int i, int j)

  Returns a reference to a matrix element at the given indices.
- const double & operator() (int i, int j) const

  Returns a constant reference to the matrix element at the given indices.
- int getRows_ () const

  Accessor method to get the value of a private variable rows.
- int getCols_ () const

  Accessor method to get the value of a private variable cols.
- double getMatrixValue (int i, int j)

  Function to get value by indexes.
- void setRows_ (int newRows)

  Mutator method for setting the value of a private variable rows.
- void setCols_ (int newCols)

  Mutator method for setting the value of a private variable cols.
- void setMatrixValue (int i, int j, double value)

  Setting a value to a matrix.
- void setMatrixValue (std::vector< double > &values)

  Setting values from a vector(values) to a matrix.

3.1.1 Detailed Description

Class representing a matrix in a mathematical context.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 S21Matrix() [1/4]

```
S21Matrix::S21Matrix ( )
```

Default constructor.

Creates a 3x3 square matrix and fills it with zeros.

3.1.2.2 S21Matrix() [2/4]

```
S21Matrix::S21Matrix (
            int rows,
            int cols )
```

Constructor with given dimensions.

Creates a matrix with the specified number of rows and columns and fills it with zeros.

Parameters

rows	Number of rows in the matrix.
cols	The number of columns in the matrix.

3.1.2.3 S21Matrix() [3/4]

```
S21Matrix::S21Matrix (
            const S21Matrix & other )
```

Copy constructor.

Creates a copy of the matrix using deep copy.

Parameters

other	Link to the matrix to be copied.

3.1.2.4 S21Matrix() [4/4]

```
S21Matrix::S21Matrix (
            S21Matrix && other )  [noexcept]
```

Move constructor.

Moves resources from another object to a new object.

Parameters

other	R-value reference to the object being moved.

3.1.2.5 ~S21Matrix()

```
S21Matrix::~S21Matrix ( )
```

Destructor.

Frees memory allocated for the matrix.

3.1.3 Member Function Documentation

3.1.3.1 CalcComplements()

```
S21Matrix S21Matrix::CalcComplements ( )
```

Calculates the algebraic addition matrix of the current one and returns it.

Exceptions

The	matrix is not square.
The	matrix size $<$ 2x2.

Returns

Algebraic addition matrix.

See also

ChecksError

getMatrixMinor

Determinant

3.1.3.2 checkSize()

```
bool S21Matrix::checkSize ( ) const
```

function to check the correctness of the matrix size

Returns

true or false

3.1.3.3 Determinant()

```
double S21Matrix::Determinant ( ) const
```

Calculates and returns the determinant of the current matrix.

Exceptions

The	matrix is not square.

Returns

Determinant.

See also

ChecksError

getMatrixMinor

Determinant

3.1.3.4 EqMatrix()

```
bool S21Matrix::EqMatrix (
            const S21Matrix & other ) const
```

Checks matrices for equality with each other.

Parameters

other	comparison matrix

Returns

true or false

3.1.3.5 EqSizeMatrix()

```
bool S21Matrix::EqSizeMatrix (
            const S21Matrix & other ) const
```

Checking the equality of matrix sizes.

Parameters

other	comparison matrix

Returns

true or false

3.1.3.6 getCols_()

```
int S21Matrix::getCols_ ( ) const
```

Accessor method to get the value of a private variable cols.

Returns

size cols

3.1.3.7 getMatrixValue()

```
double S21Matrix::getMatrixValue (
            int i,
            int j )
```

Function to get value by indexes.

Parameters

i	index for rows
j	index for cols

Returns

values `matrix[i][j]`

3.1.3.8 getRows_()

```
int S21Matrix::getRows_ ( ) const
```

Accessor method to get the value of a private variable rows.

Returns

size rows

3.1.3.9 InverseMatrix()

```
S21Matrix S21Matrix::InverseMatrix ( )
```

Calculates the inverse matrix.

Exceptions

Matrix	determinant is 0.

Returns

inverse matrix.

See also

ChecksError
Determinant
Transpose
CalcComplements
MulNumber

3.1.3.10 MulMatrix()

```
void S21Matrix::MulMatrix (
            const S21Matrix & other )
```

Multiplies the current matrix by the second matrix.

Exceptions

The	number of columns of the first matrix is not equal to the number of rows of the second matrix.

Parameters

other	Matrix for multiplication.

See also

ChecksError

deleteMAtrix

3.1.3.11 MulNumber()

```
void S21Matrix::MulNumber (
            const double num )
```

Multiplies the current matrix by a number.

Parameters

num	number.

3.1.3.12 operator()() [1/2]

```
double & S21Matrix::operator() (
            int i,
            int j )
```

Returns a reference to a matrix element at the given indices.

Parameters

i	The index of the string.
j	Column index.

Returns

A reference to a matrix element.

Exceptions

std::out_of_range	if indexes are outside the matrix size.

3.1.3.13 operator()() [2/2]

```
const double & S21Matrix::operator() (
            int i,
            int j ) const
```

Returns a constant reference to the matrix element at the given indices.

Parameters

i	The index of the string.
j	Column index.

Returns

A constant reference to a matrix element.

Exceptions

std::out_of_range	if indexes are outside the matrix size.

3.1.3.14 operator∗() [1/2]

```
S21Matrix S21Matrix::operator* (
            const double & number ) const
```

Overloaded operator for multiplying a matrix by a number.

Creates a new matrix that is the result of multiplying the current matrix by the specified number.

Parameters

number	The number by which each element of the current matrix is multiplied.

Returns

A new matrix containing the result of multiplication by a number.

3.1.3.15 operator∗() [2/2]

```
S21Matrix S21Matrix::operator* (
            const S21Matrix & other ) const
```

Overloaded matrix multiplication operator.

Creates a new matrix that is the result of multiplying the current matrix by another matrix.

Parameters

other	Another matrix by which the current one is multiplied.

Returns

A new matrix containing the result of the multiplication.

3.1.3.16 operator*=() [1/2]

```
S21Matrix & S21Matrix::operator*= (
            const double & number )
```

Overloaded operator for multiplying a matrix by a number.

Multiplies each element of the current matrix by the specified number and stores the result in the current matrix.

Parameters

number	The number by which each element of the current matrix is multiplied.

Returns

A reference to the current matrix after multiplying by a number.

3.1.3.17 operator*=() [2/2]

```
S21Matrix & S21Matrix::operator*= (
            const S21Matrix & other )
```

Overloaded matrix multiplication operator.

Multiplies the current matrix by another matrix and stores the result in the current matrix.

Parameters

other	Another matrix by which the current one is multiplied.

Returns

A reference to the current matrix after performing the multiplication.

3.1.3.18 operator+()

```
S21Matrix S21Matrix::operator+ (
            const S21Matrix & other ) const
```

Overloaded matrix addition operator.

Creates a new matrix that is the result of adding the current matrix to another matrix.

Parameters

other	Another matrix that adds to the current one.

Returns

A new matrix containing the result of the addition.

3.1.3.19 operator+=()

```
S21Matrix & S21Matrix::operator+= (
            const S21Matrix & other )
```

Overloaded matrix addition operator.

Adds another matrix to the current one and stores the result in the current matrix.

Parameters

other	Another matrix that adds to the current one.

Returns

A reference to the current matrix after the addition is performed.

3.1.3.20 operator-()

```
S21Matrix S21Matrix::operator- (
            const S21Matrix & other ) const
```

Overloaded matrix subtraction operator.

Creates a new matrix that is the result of subtracting another matrix from the current one.

Parameters

other	Another matrix that is subtracted from the current one.

Returns

A new matrix containing the result of the subtraction.

3.1.3.21 operator-=()

```
S21Matrix & S21Matrix::operator-= (
            const S21Matrix & other )
```

Overloaded matrix subtraction operator.

Subtracts another matrix from the current one and stores the result in the current matrix.

Parameters

other	Another matrix that is subtracted from the current one.

Returns

A reference to the current matrix after subtraction is performed.

3.1.3.22 operator=() [1/2]

```
S21Matrix & S21Matrix::operator= (
            const S21Matrix & other )
```

Overloaded assignment operator for copying a matrix.

Copies the contents of another matrix into the current matrix.

Parameters

other	Another matrix from which the data is copied.

Returns

A reference to the current matrix after the assignment has been performed.

3.1.3.23 operator=() [2/2]

```
S21Matrix & S21Matrix::operator= (
            S21Matrix && other )  [noexcept]
```

Moves the contents of a matrix from another object to the current object.

Parameters

other	The matrix from which to move.

Returns

A reference to the current object after the move.

3.1.3.24 operator==()

```
bool S21Matrix::operator== (
            const S21Matrix & other ) const
```

Overloaded operator for comparing two matrices for equality.

Parameters

other	Another matrix with which the current one is compared.

Returns

true if the matrices are equal, false otherwise.

3.1.3.25 setCols_()

```
void S21Matrix::setCols_ (
            int newCols )
```

Mutator method for setting the value of a private variable cols.

Parameters

newCols	new size cols

3.1.3.26 setMatrixValue() [1/2]

```
void S21Matrix::setMatrixValue (
            int i,
            int j,
            double value )
```

Setting a value to a matrix.

Parameters

i	index for rows
j	index for cols
value	set value

3.1.3.27 setMatrixValue() [2/2]

```
void S21Matrix::setMatrixValue (
            std::vector< double > & values )
```

Setting values from a vector(values) to a matrix.

Parameters

values	vector of values

3.1.3.28 setRows_()

```
void S21Matrix::setRows_ (
            int newRows )
```

Mutator method for setting the value of a private variable rows.

Parameters

newRows	new size rows

3.1.3.29 SubMatrix()

```
void S21Matrix::SubMatrix (
            const S21Matrix & other )
```

Subtracts another matrix from the current one.

Exceptions

Different	matrix dimensions.

Parameters

other	Matrix to be subtracted.

See also

PerformMatrixOperation

3.1.3.30 SumMatrix()

```
void S21Matrix::SumMatrix (
            const S21Matrix & other )
```

Adds the second matrix to the current one.

Exceptions

Different	matrix dimensions.

Parameters

other	Another matrix with which addition is performed.

See also

PerformMatrixOperation

3.1.3.31 Transpose()

```
S21Matrix S21Matrix::Transpose ( )
```

Creates a new transposed matrix from the current one and returns it.

Returns

Transposed matrix.

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

- s21_matrix_oop.h
- function/helpers.cpp
- function/operations.cpp
- function/operators.cpp
- function/struct.cpp

Chapter 4

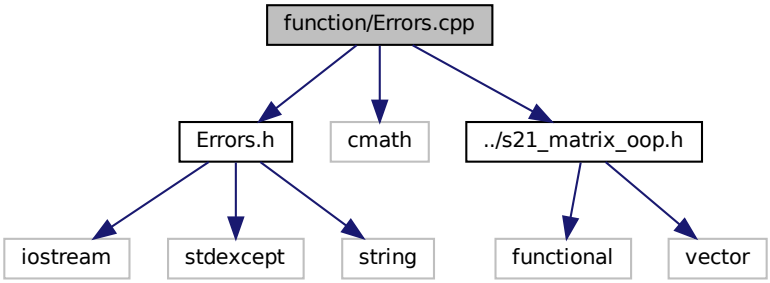
File Documentation

4.1 function/Errors.cpp File Reference

File for checking errors and throwing them.

```
#include "Errors.h"
#include <cmath>
#include "../s21_matrix_oop.h"
```

Include dependency graph for Errors.cpp:



Functions

- void errors_ (const Code code_)

  Function for throwing an error based on a specific code.
- void ChecksError (Code code_, const S21Matrix &first, const S21Matrix *second)

  Function for checking errors in the code between the first and second matrices, or only the first.
- const char * getErrorMessage (size_t index)

  Get the Error Message object.

4.1.1 Detailed Description

File for checking errors and throwing them.

Author

nenamaxi

Version

0.1

Date

2024-02-18

Copyright

Copyright (c) 2024

4.1.2 Function Documentation

4.1.2.1 ChecksError()

```
void ChecksError (
            Code code_,
            const S21Matrix & first,
            const S21Matrix * second )
```

Function for checking errors in the code between the first and second matrices, or only the first.

Parameters

*code_*	error code
first	first matrix, main
second	second matrix to check

4.1.2.2 errors_()

```
void errors_ (
            const Code code_ )
```

Function for throwing an error based on a specific code.

Parameters

*code_*	error code

4.1.2.3 getErrorMessage()

```
const char* getErrorMessage (
            size_t index )
```

Get the Error Message object.

Parameters

index	index by which the error is selected

Returns

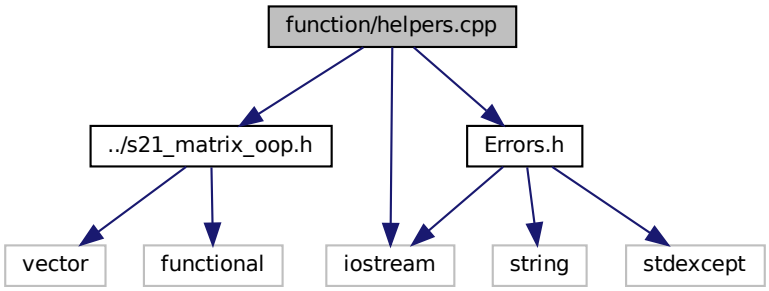
const char*

4.2 function/helpers.cpp File Reference

helpers function

```
#include <iostream>
#include "../s21_matrix_oop.h"
#include "Errors.h"
```

Include dependency graph for helpers.cpp:



4.2.1 Detailed Description

helpers function

Author

nenamaxi

Version

0.1

Date

2024-02-18

Copyright

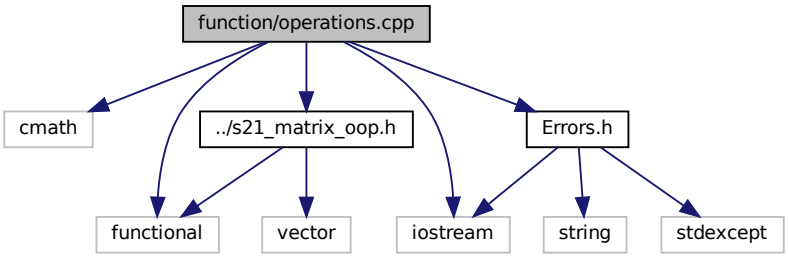
Copyright (c) 2024

4.3 function/operations.cpp File Reference

operations implementation

```
#include <cmath>
#include <functional>
#include <iostream>
#include "../s21_matrix_oop.h"
#include "Errors.h"
```

Include dependency graph for operations.cpp:



Variables

- constexpr double ACCURACY = 0.0000001

  accuracy for compere value

4.3.1 Detailed Description

operations implementation

Author

nenamaxi

Version

0.1

Date

2024-02-18

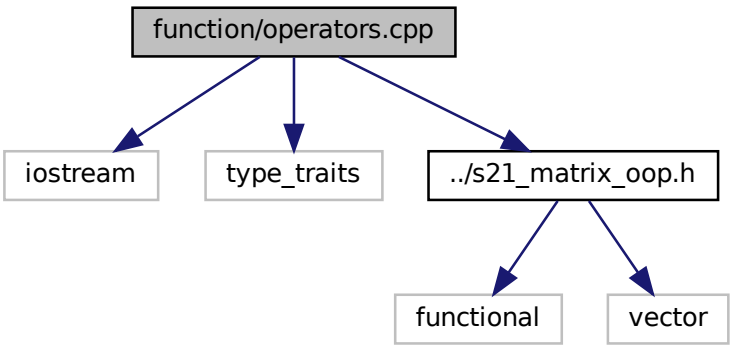
Copyright

4.4 function/operators.cpp File Reference

operator implementation

```
#include <iostream>
#include <type_traits>
#include "../s21_matrix_oop.h"
```

Include dependency graph for operators.cpp:



4.4.1 Detailed Description

operator implementation

Author

nenamaxi

Version

0.1

Date

2024-02-18

Copyright

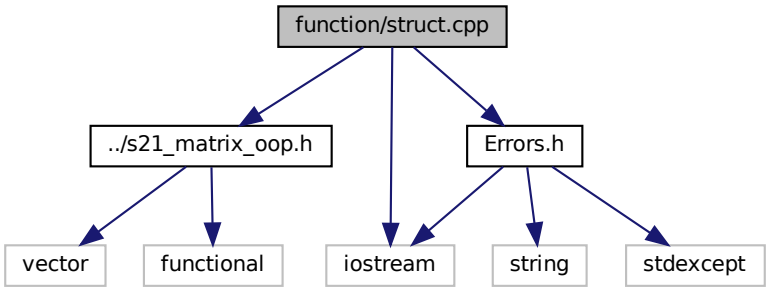
Copyright (c) 2024

4.5 function/struct.cpp File Reference

structur fanction in this file

```
#include <iostream>
#include "../s21_matrix_oop.h"
#include "Errors.h"
```

Include dependency graph for struct.cpp:



4.5.1 Detailed Description

structur fanction in this file

Author

nenamaxi

Version

0.1

Date

2024-02-18

Copyright

Index