

## S21 Matrix

Generated by Doxygen 1.9.1



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 S21Matrix Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	7
3.1.2.1 S21Matrix() [1/4]	7
3.1.2.2 S21Matrix() [2/4]	7
3.1.2.3 S21Matrix() [3/4]	7
3.1.2.4 S21Matrix() [4/4]	7
3.1.2.5 ~S21Matrix()	8
3.1.3 Member Function Documentation	8
3.1.3.1 CalcComplements()	8
3.1.3.2 checkSize()	9
3.1.3.3 Determinant()	9
3.1.3.4 EqMatrix()	9
3.1.3.5 EqSizeMatrix()	10
3.1.3.6 getCols_()	10
3.1.3.7 getMatrixValue()	10
3.1.3.8 getRows_()	11
3.1.3.9 InverseMatrix()	11
3.1.3.10 MulMatrix()	11
3.1.3.11 MulNumber()	12
3.1.3.12 operator>() [1/2]	12
3.1.3.13 operator>() [2/2]	13
3.1.3.14 operator*() [1/2]	13
3.1.3.15 operator*() [2/2]	13
3.1.3.16 operator*==() [1/2]	14
3.1.3.17 operator*==() [2/2]	14
3.1.3.18 operator+()	15
3.1.3.19 operator+=()	15
3.1.3.20 operator-()	15
3.1.3.21 operator-=()	16
3.1.3.22 operator=() [1/2]	16
3.1.3.23 operator=() [2/2]	16
3.1.3.24 operator==(())	17
3.1.3.25 setCols_()	17
3.1.3.26 setMatrixValue() [1/2]	17
3.1.3.27 setMatrixValue() [2/2]	18

---

3.1.3.28 setRows_()	18
3.1.3.29 SubMatrix()	18
3.1.3.30 SumMatrix()	19
3.1.3.31 Transpose()	19
<b>4 File Documentation</b>	<b>21</b>
4.1 function/Errors.cpp File Reference	21
4.1.1 Detailed Description	22
4.1.2 Function Documentation	22
4.1.2.1 ChecksError()	22
4.1.2.2 errors_()	22
4.1.2.3 getErrorMessage()	23
4.2 function/helpers.cpp File Reference	23
4.2.1 Detailed Description	24
4.3 function/operations.cpp File Reference	24
4.3.1 Detailed Description	25
4.4 function/operators.cpp File Reference	25
4.4.1 Detailed Description	26
4.5 function/struct.cpp File Reference	26
4.5.1 Detailed Description	27
<b>Index</b>	<b>29</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">S21Matrix</a>	Class representing a matrix in a mathematical context . . . . .	<a href="#">5</a>
---------------------------	---	-------------------



## Chapter 2

# File Index

### 2.1 File List

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

<b>s21_matrix_oop.h</b>	??
function/ <a href="#">Errors.cpp</a>	
File for checking errors and throwing them	21
function/ <b>Errors.h</b>	??
function/ <a href="#">helpers.cpp</a>	
Helpers function	23
function/ <a href="#">operations.cpp</a>	
Operations implementation	24
function/ <a href="#">operators.cpp</a>	
Operator implementation	25
function/ <a href="#">struct.cpp</a>	
Structur function in this file	26





## 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

<i>rows</i>	Number of rows in the matrix.
<i>cols</i>	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

<i>other</i>	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

<i>other</i>	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

<i>The</i>	matrix is not square.
<i>The</i>	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

<i>The</i>	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

<i>other</i>	comparison matrix
--------------	-------------------

#### Returns

true or false

### 3.1.3.5 EqSizeMatrix()

```
bool S2lMatrix::EqSizeMatrix (
    const S2lMatrix & other ) const
```

Checking the equality of matrix sizes.

#### Parameters

<i>other</i>	comparison matrix
--------------	-------------------

#### Returns

true or false

### 3.1.3.6 getCols\_()

```
int S2lMatrix::getCols_ ( ) const
```

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

#### Returns

size cols

### 3.1.3.7 getMatrixValue()

```
double S2lMatrix::getMatrixValue (
    int i,
    int j )
```

Function to get value by indexes.

#### Parameters

<i>i</i>	index for rows
<i>j</i>	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

<i>Matrix</i>	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

<i>The</i>	number of columns of the first matrix is not equal to the number of rows of the second matrix.
------------	--

**Parameters**

<i>other</i>	Matrix for multiplication.
--------------	----------------------------

**See also**[ChecksError](#)[deleteMAtrix](#)**3.1.3.11 MulNumber()**

```
void S2lMatrix::MulNumber (
    const double num )
```

Multiplies the current matrix by a number.

**Parameters**

<i>num</i>	number.
------------	---------

**3.1.3.12 operator>() [1/2]**

```
double & S2lMatrix::operator() (
    int i,
    int j )
```

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

**Parameters**

<i>i</i>	The index of the string.
<i>j</i>	Column index.

**Returns**

A reference to a matrix element.

**Exceptions**

<i>std::out_of_range</i>	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>i</i>	The index of the string.
<i>j</i>	Column index.

#### Returns

A constant reference to a matrix element.

#### Exceptions

<code>std::out_of_range</code>	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

<i>number</i>	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

<i>other</i>	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

<i>number</i>	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

<i>other</i>	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

<i>other</i>	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

<i>other</i>	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

<i>other</i>	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**

<i>other</i>	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**

<i>other</i>	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

<i>other</i>	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

<i>other</i>	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

<i>newCols</i>	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>i</i>	index for rows
<i>j</i>	index for cols
<i>value</i>	set value

**3.1.3.27 setMatrixValue()** [2/2]

```
void S2lMatrix::setMatrixValue (
    std::vector< double > & values )
```

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

## Parameters

<i>values</i>	vector of values
---------------	------------------

**3.1.3.28 setRows\_()**

```
void S2lMatrix::setRows_ (
    int newRows )
```

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

## Parameters

<i>newRows</i>	new size rows
----------------	---------------

**3.1.3.29 SubMatrix()**

```
void S2lMatrix::SubMatrix (
    const S2lMatrix & other )
```

Subtracts another matrix from the current one.

## Exceptions

<i>Different</i>	matrix dimensions.
------------------	--------------------

## Parameters

<i>other</i>	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

<i>Different</i>	matrix dimensions.
------------------	--------------------

## Parameters

<i>other</i>	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](#)





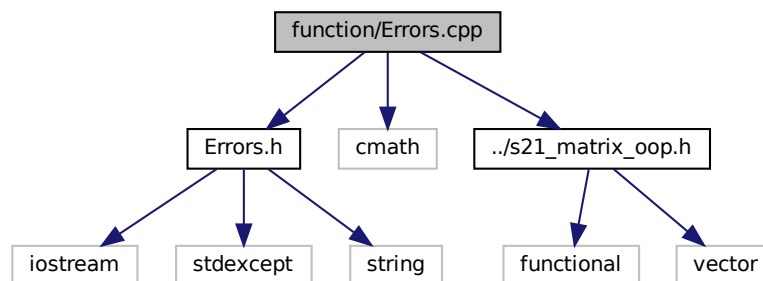
## Chapter 4

# 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 S2lMatrix & first,
    const S2lMatrix * second )
```

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

#### Parameters

<i>code_</i>	error code
<i>first</i>	first matrix, main
<i>second</i>	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

<i>code</i> ↔	error code
—	

## 4.1.2.3 getErrorMessage()

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

Get the Error Message object.

## Parameters

<i>index</i>	index by which the error is selected
--------------	--------------------------------------

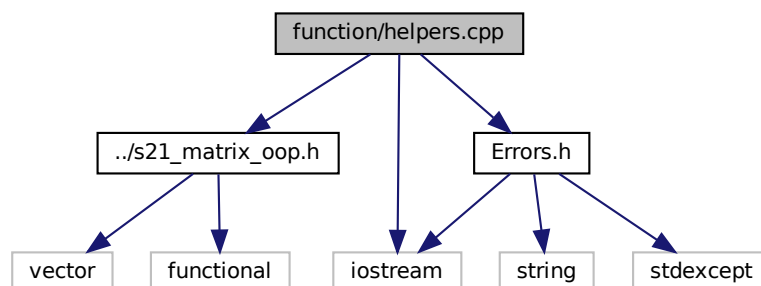
## Returns

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

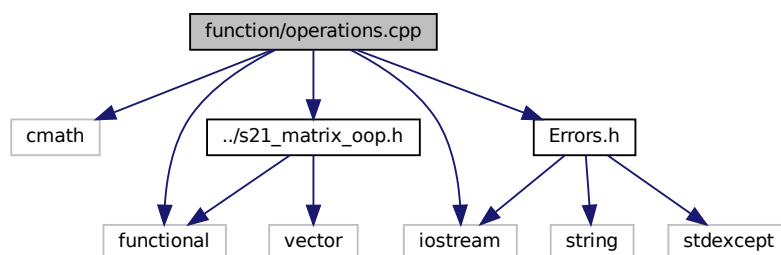
Copyright

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 compare value*

### 4.3.1 Detailed Description

operations implementation

**Author**

nenamaxi

**Version**

0.1

**Date**

2024-02-18

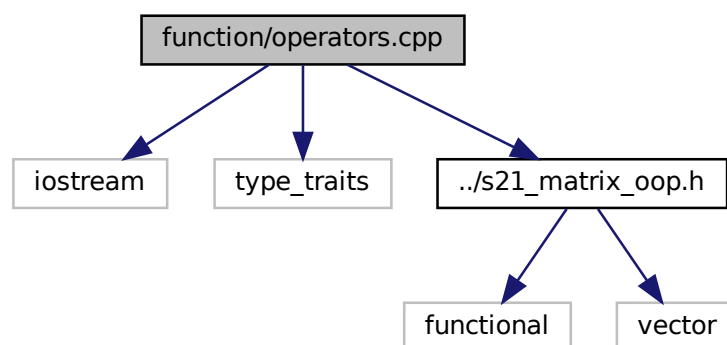
**Copyright**

Copyright (c) 2024

## 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

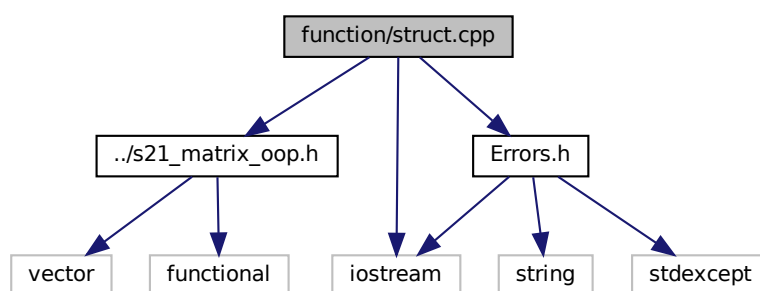
#### Copyright

Copyright (c) 2024

## 4.5 function/struct.cpp File Reference

structur fansion 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 fancement in this file

Author

nenamaxi

Version

0.1

Date

2024-02-18

Copyright

Copyright (c) 2024





# Index

`~S21Matrix`  
    [S21Matrix, 8](#)

`CalcComplements`  
    [S21Matrix, 8](#)

`ChecksError`  
    [Errors.cpp, 22](#)

`checkSize`  
    [S21Matrix, 8](#)

`Determinant`  
    [S21Matrix, 9](#)

`EqMatrix`  
    [S21Matrix, 9](#)

`EqSizeMatrix`  
    [S21Matrix, 10](#)

`Errors.cpp`  
    [ChecksError, 22](#)  
    [errors\\_, 22](#)  
    [getErrorMessage, 23](#)

`errors_`  
    [Errors.cpp, 22](#)

`function/Errors.cpp, 21`  
`function/helpers.cpp, 23`  
`function/operations.cpp, 24`  
`function/operators.cpp, 25`  
`function/struct.cpp, 26`

`getCols_`  
    [S21Matrix, 10](#)

`getErrorMessage`  
    [Errors.cpp, 23](#)

`getMatrixValue`  
    [S21Matrix, 10](#)

`getRows_`  
    [S21Matrix, 10](#)

`InverseMatrix`  
    [S21Matrix, 11](#)

`MulMatrix`  
    [S21Matrix, 11](#)

`MulNumber`  
    [S21Matrix, 12](#)

`operator*`  
    [S21Matrix, 13](#)

`operator*=  
    S21Matrix, 14`

`operator()`  
    [S21Matrix, 12](#)

`operator+`  
    [S21Matrix, 14](#)

`operator+=`  
    [S21Matrix, 15](#)

`operator-`  
    [S21Matrix, 15](#)

`operator-=`  
    [S21Matrix, 16](#)

`operator=`  
    [S21Matrix, 16](#)

`operator==`  
    [S21Matrix, 17](#)

`S21Matrix, 5`  
    [~S21Matrix, 8](#)  
    [CalcComplements, 8](#)  
    [checkSize, 8](#)  
    [Determinant, 9](#)  
    [EqMatrix, 9](#)  
    [EqSizeMatrix, 10](#)  
    [getCols\\_, 10](#)  
    [getMatrixValue, 10](#)  
    [getRows\\_, 10](#)  
    [InverseMatrix, 11](#)  
    [MulMatrix, 11](#)  
    [MulNumber, 12](#)  
    [operator\\*, 13](#)  
    [operator\\*=  
    \[operator\\(\\), 12\]\(#\)  
    \[operator+, 14\]\(#\)  
    \[operator+=, 15\]\(#\)  
    \[operator-, 15\]\(#\)  
    \[operator-=, 16\]\(#\)  
    \[operator=, 16\]\(#\)  
    \[operator==, 17\]\(#\)  
    \[S21Matrix, 7\]\(#\)  
    \[setCols\\\_, 17\]\(#\)  
    \[setMatrixValue, 17, 18\]\(#\)  
    \[setRows\\\_, 18\]\(#\)  
    \[SubMatrix, 18\]\(#\)  
    \[SumMatrix, 19\]\(#\)  
    \[Transpose, 19\]\(#\)  
`setCols\_`  
    \[S21Matrix, 17\]\(#\)  
`setMatrixValue`  
    \[S21Matrix, 17, 18\]\(#\)  
`setRows\_`  
    \[S21Matrix, 18\]\(#\)](#)

SubMatrix  
    S21Matrix, [18](#)  
SumMatrix  
    S21Matrix, [19](#)  
Transpose  
    S21Matrix, [19](#)