Matryca

Generated by Doxygen 1.12.0

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 matrix Class Reference

The class that is responsible for the matrix.

```
#include <matrix.h>
```

Public Member Functions

- int **rozmiar** () const
- **matrix** ()

  The constructor of the class without parameters.
- matrix (int n)

  The constructor of the class.
- matrix (int n, int ∗t)

  The constructor of the class.
- matrix (const matrix &m)

  The constructor of the class.
- matrix & alokuj (int n)

  Allocate a matrix for the given size.
- matrix & **wstaw** (int x, int y, int wartosc)

  Inserts a value into the matrix at the designated coordinates.
- int **pokaz** (int x, int y) const

  Print the field at the given coordinates.
- matrix & **odwroc** ()

  Invert the matrix.
- matrix & **losuj** ()

  Fill the matrix with random numbers.
- matrix & losuj (int x)

  Fill the array with an amount of random numbers in random positions.
- matrix & diagonalna (int ∗t)

  Fill the diagonal with the values from the array.
- matrix & diagonalna_k (int k, int ∗t)

  Fill the diagonal with the values from the array at an offset line.
- matrix & kolumna (int x, int ∗t)

Draw a column within the matrix.

- matrix & wiersz (int y, int ∗t)

    Draw a row within the matrix.
- matrix & **przekatna** ()

    Fills the matrix with ones on the diagonal.
- matrix & **pod_przekatna** ()

    Fills the matrix with ones below the diagonal.
- matrix & **nad_przekatna** ()

    Fills the matrix with ones above the diagonal.
- matrix & **szachownica** ()

    Fills the matrix with a checkerboard of ones and zeroes.
- matrix & **ustal** (int ∗t)

    Fills the matrix from the given array.
- matrix & **zero** ()

    Clear the matrix.
- matrix **operator+** (const matrix &m) const

    Operator overloading for addition.
- matrix **operator∗** (matrix &m)

    Operator overloading for multiplication.
- matrix **operator+** (int a) const

    Operator overloading for pre-incrementation.
- matrix **operator∗** (int a) const

    Operator overloading for multiplying with intigers.
- matrix **operator-** (int a) const

    Operator overloading for subtraction.
- matrix **operator++** (int)

    Operator overloading for incrementation.
- matrix **operator--** (int)

    Operator overloading for derementation.
- matrix & **operator+=** (int a)

    Operator overloading for additive assigning.
- matrix & **operator-=** (int a)

    Operator overloading for subtractive asigning.
- matrix & **operator∗=** (int a)

    Operator overloading for multiplicative assigning.
- matrix & **operator()** (double a)

    Operator overloading for brackets, in this case adding the whole part of the double.
- bool **operator==** (const matrix &m) const

    Operator overloading for comparison.
- bool **operator>** (const matrix &m) const

    Operator overloading for the greater comparison.
- bool **operator<** (const matrix &m) const

    Operator overloading for the lesser comparison.

Friends

- matrix **operator+** (int a, const matrix &m)

    Operator overloading for addition with intigers.
- matrix **operator∗** (int a, const matrix &m)

    Operator overloading for multiplying with intiger.
- matrix **operator-** (int a, const matrix &m)

    Operator overloading for subtraction with intigers.
- std::ostream & **operator<<** (std::ostream &o, const matrix &m)

    Operator overloading for printing a whole matrix.

3.1.1 Detailed Description

The class that is responsible for the matrix.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 matrix() [1/3]

```
matrix::matrix (
              int n)
```

The constructor of the class.

Parameters

	The	desired size

3.1.2.2 matrix() [2/3]

```
matrix::matrix (
              int n,
              int * t)
```

The constructor of the class.

Parameters

	The	desired size
	The	array to draw from

3.1.2.3 matrix() [3/3]

```
matrix::matrix (
              const matrix & m)
```

The constructor of the class.

Parameters

	The	matrix to copy from

3.1.3 Member Function Documentation

3.1.3.1 alokuj()

```
matrix & matrix::alokuj (
              int n)
```

Allocate a matrix for the given size.

Parameters

	The	desired size

3.1.3.2 diagonalna()

```
matrix & matrix::diagonalna (
             int * t)
```

Fill the diagonal with the values from the array.

Parameters

	The	array to draw from

3.1.3.3 diagonalna_k()

```
matrix & matrix::diagonalna_k (
             int k,
             int * t)
```

Fill the diagonal with the values from the array at an offset line.

Parameters

	The	offset
	The	array to draw from

3.1.3.4 kolumna()

```
matrix & matrix::kolumna (
             int x,
             int * t)
```

Draw a column within the matrix.

Parameters

	The	column number
	The	array to draw from

3.1.3.5 losuj()

```
matrix & matrix::losuj (
             int x)
```

Fill the array with an amount of random numbers in random positions.

Parameters

	The	amount of numbers to fill

3.1.3.6 wiersz()

```
matrix & matrix::wiersz (
             int y,
             int * t)
```

Draw a row within the matrix.

Parameters

	The	row number
	The	array to draw from

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

- matrix.h
- alokuj.cpp
- diagonalna.cpp
- diagonalna_k.cpp
- kolumna.cpp
- losuj.cpp
- losuj_x.cpp
- matrix.cpp
- nad_przekatna.cpp
- odwroc.cpp
- operator(wpisana liczba).cpp
- operator++(inkr).cpp
- operator++.cpp
- operator+.cpp
- operator+a.cpp
- operator--(inkr).cpp
- operator-.cpp
- operator-a.cpp
- operator==.cpp
- operator_mniejsze.cpp
- operator_wieksze.cpp
- operatorx.cpp
- operatorxa.cpp
- operatorxx.cpp
- pod_przekatna.cpp
- pokaz.cpp
- przekatna.cpp
- szachownica.cpp
- ustal.cpp
- wiersz.cpp
- wstaw.cpp
- zero.cpp

Chapter 4

File Documentation

4.1 matrix.h

```
#ifndef MATRIX_H
#define MATRIX_H
#include <memory>
#include <iostream>

class matrix {
private:
    int n_;
    std::unique_ptr<int[]> data_;
    inline int idx(int x, int y) const { return x * n_ + y; }
public:
    int rozmiar() const { return n_; }
    matrix();
    matrix(int n);
    matrix(int n, int* t);
    matrix(const matrix& m);
    ~matrix();
    matrix& alokuj(int n);
    matrix& wstaw(int x, int y, int wartosc);
    int pokaz(int x, int y) const;
    matrix& odwroc();
    matrix& losuj();
    matrix& losuj(int x);
    matrix& diagonalna(int* t);
    matrix& diagonalna_k(int k, int* t);
    matrix& kolumna(int x, int* t);
    matrix& wiersz(int y, int* t);
    matrix& przekatna();
    matrix& pod_przekatna();
    matrix& nad_przekatna();
    matrix& szachownica();
    matrix& ustal(int* t);
    matrix& zero();
    matrix operator+(const matrix& m) const;
    matrix operator*(matrix& m);
    matrix operator+(int a) const;
    matrix operator*(int a) const;
    matrix operator-(int a) const;
    friend matrix operator+(int a, const matrix& m);
    friend matrix operator*(int a, const matrix& m);
    friend matrix operator-(int a, const matrix& m);
    matrix operator++(int);
    matrix operator--(int);
    matrix& operator+=(int a);
    matrix& operator-=(int a);
    matrix& operator*=(int a);
    matrix& operator()(double a);
    friend std::ostream& operator«(std::ostream& o, const matrix& m);
    bool operator==(const matrix& m) const;
    bool operator>(const matrix& m) const;
    bool operator<(const matrix& m) const;

};

#endif
```

Index