My Project

Generated by Doxygen 1.12.0

Chapter 1

P4

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 Matrix Class Reference

Klasa reprezentujaca macierz kwadratowa.

`#include <Matrix.h>`

Public Member Functions

- Matrix ()
- Matrix (int n)
- Matrix (int n, int ∗t)
- Matrix (const Matrix &m)
- ∼Matrix ()
- Matrix & Macierz_Alokacja (int n)
- Matrix & wstaw (int x, int y, int wartosc)
- int pokaz (int x, int y)
- Matrix & Macierz_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 (void)
- Matrix & pod_przekatna (void)
- Matrix & nad_przekatna (void)
- Matrix & szachownica (void)
- Matrix & operator+ (Matrix &m)
- Matrix & operator∗ (Matrix &m)
- Matrix & operator+ (int a)
- Matrix & operator∗ (int a)
- Matrix & operator- (int a)
- Matrix & operator= (const Matrix &m)

Public Attributes

- int ** **wsm**

  Wskaznik na tablice dwuwymiarowa przechowujaca macierz.
- int **rozmiar**

  Rozmiar macierzy (liczba wierszy/kolumn).

Friends

- Matrix operator+ (int a, Matrix &m)
- Matrix operator* (int a, Matrix &m)
- Matrix operator- (int a, Matrix &m)

4.1.1 Detailed Description

Klasa reprezentujaca macierz kwadratowa.

Klasa ta pozwala na tworzenie i manipulowanie macierzami kwadratowymi o dynamicznie alokowanej pamieci. Zawiera metody umozliwiajace wykonywanie roznych operacji na macierzach, takich jak dodawanie, mnozenie, wstawianie wartosci, czy odwracanie macierzy.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Matrix() [1/4]

```
Matrix::Matrix ()
```

Konstruktor domyslny.

Inicjalizuje wskaznik `wsm` na `nullptr` i rozmiar macierzy na 0. Nie alokuje pamieci dla macierzy.

4.1.2.2 Matrix() [2/4]

```
Matrix::Matrix (
            int n)
```

Konstruktor przeciazeniowy.

Alokuje pamiec dla macierzy o rozmiarze `n x n` i inicjalizuje jej wartosci na 0.

Parameters

n	Rozmiar macierzy.

4.1.2.3 Matrix() [3/4]

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

Konstruktor przeciazeniowy z tablica.

Alokuje pamiec dla macierzy o rozmiarze `n x n` i przepisuje dane z tablicy `t`.

Parameters

n	Rozmiar macierzy.
t	Tablica zawierajaca dane do przypisania do macierzy.

4.1.2.4 Matrix() [4/4]

```
Matrix::Matrix (
            const Matrix & m)
```

Konstruktor kopiujacy.

Tworzy kopie macierzy `m`, alokujac nowa pamiec i kopiujac jej zawartosc.

Parameters

m	Macierz, ktora ma zostac skopiowana.

4.1.2.5 ~Matrix()

```
Matrix::~Matrix ()
```

Destruktor.

Zwalnia pamiec zajmowana przez macierz.

4.1.3 Member Function Documentation

4.1.3.1 diagonalna()

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

Ustawia wartosci na przekatnej macierzy z tablicy `t`, a pozostale elementy ustawia na 0.

Parameters

t	Tablica z danymi do ustawienia na przekatnej.

Returns

Referencja do obiektu macierzy po ustawieniu przekatnej.

4.1.3.2 diagonalna_k()

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

Ustawia wartosci na k-tej przekatnej macierzy z tablicy `t`, a pozostale elementy ustawia na 0. Parametr `k` okresla przesuniecie przekatnej w gore lub w dol.

Parameters

k	Przesuniecie przekatnej.
t	Tablica z danymi do ustawienia na przekatnej.

Returns

Referencja do obiektu macierzy po ustawieniu k-tej przekatnej.

4.1.3.3 kolumna()

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

Wstawia wartosci z tablicy `t` do kolumny o indeksie `x`.

Parameters

x	Indeks kolumny.
t	Tablica z danymi do wstawienia do kolumny.

Returns

Referencja do obiektu macierzy po wstawieniu wartosci do kolumny.

4.1.3.4 losuj() [1/2]

```
Matrix & Matrix::losuj ()
```

Wypelnia macierz losowymi wartosciami w zakresie od 0 do 9.

Returns

Referencja do obiektu macierzy po wypelnieniu.

4.1.3.5 losuj() [2/2]

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

Losuje `x` wartosci w macierzy i umieszcza je w losowych miejscach.

Parameters

x	Liczba losowanych wartosci.

Returns

Referencja do obiektu macierzy po losowaniu.

4.1.3.6 Macierz_Alokacja()

```
Matrix & Matrix::Macierz_Alokacja (
            int n)
```

Alokuje pamiec dla macierzy o rozmiarze `n x n` jesli pamiec nie byla wczesniej alokowana, lub w razie potrzeby zmienia rozmiar juz zaalokowanej macierzy.

Parameters

n	Nowy rozmiar macierzy.

4.1.3.7 Macierz_Odwroc()

```
Matrix & Matrix::Macierz_Odwroc ()
```

Odwraca macierz (zamienia wiersze z kolumnami).

Returns

Referencja do obiektu macierzy po odwróceniu.

4.1.3.8 nad_przekatna()

```
Matrix & Matrix::nad_przekatna (
            void )
```

Wypelnia macierz tak, ze 1 znajduje sie powyzej przekatnej, a pozostale elementy to 0.

Returns

Referencja do obiektu macierzy po uzupelnieniu.

4.1.3.9 operator∗() [1/2]

```
Matrix & Matrix::operator* (
            int a)
```

Operator mnozenia macierzy przez liczbe calkowita.

Parameters

a	Liczba calkowita do pomnozenia.

Returns

Referencja do obiektu macierzy po pomnozeniu przez liczbe.

4.1.3.10 operator∗() [2/2]

```
Matrix & Matrix::operator* (
            Matrix & m)
```

Operator mnozenia dwoch macierzy.

Parameters

m	Druga macierz do mnozenia.

Returns

Referencja do obiektu macierzy po pomnozeniu.

4.1.3.11 operator+() [1/2]

```
Matrix & Matrix::operator+ (
            int a)
```

Operator dodawania macierzy i liczby calkowitej.

Parameters

a	Liczba calkowita do dodania.

Returns

Referencja do obiektu macierzy po dodaniu liczby.

4.1.3.12 operator+() [2/2]

```
Matrix & Matrix::operator+ (
            Matrix & m)
```

Operator dodawania dwoch macierzy.

Parameters

m	Druga macierz do dodania.

Returns

Referencja do obiektu macierzy po dodaniu.

4.1.3.13 operator-()

```
Matrix & Matrix::operator- (
            int a)
```

Operator odejmowania liczby calkowitej od macierzy.

Parameters

a	Liczba calkowita do odjecia.

Returns

Referencja do obiektu macierzy po odjeciu liczby.

4.1.3.14 operator=()

```
Matrix & Matrix::operator= (
            const Matrix & m)
```

Operator przypisania.

Kopiuje zawartosc jednej macierzy do drugiej.

Parameters

m	Macierz do skopiowania.

Returns

Referencja do obiektu po przypisaniu.

4.1.3.15 pod_przekatna()

```
Matrix & Matrix::pod_przekatna (
            void )
```

Wypelnia macierz tak, ze 1 znajduje sie ponizej przekatnej, a pozostale elementy to 0.

Returns

Referencja do obiektu macierzy po uzupelnieniu.

4.1.3.16 pokaz()

```
int Matrix::pokaz (
            int x,
            int y)
```

Zwraca wartosc elementu macierzy na pozycji (x, y).

Parameters

x	Indeks wiersza.
y	Indeks kolumny.

Returns

Wartosc elementu na pozycji (x, y).

4.1.3.17 przekatna()

```
Matrix & Matrix::przekatna (
            void )
```

Wypelnia macierz tak, ze 1 znajduje sie na przekatnej, a pozostale elementy to 0.

Returns

Referencja do obiektu macierzy po uzupelnieniu.

4.1.3.18 szachownica()

```
Matrix & Matrix::szachownica (
            void )
```

Wypelnia macierz w sposob przypominajacy szachownice.

Returns

Referencja do obiektu macierzy po uzupelnieniu.

4.1.3.19 wiersz()

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

Wstawia wartosci z tablicy `t` do wiersza o indeksie `y`.

Parameters

y	Indeks wiersza.
t	Tablica z danymi do wstawienia do wiersza.

Returns

Referencja do obiektu macierzy po wstawieniu wartosci do wiersza.

4.1.3.20 wstaw()

```
Matrix & Matrix::wstaw (
            int x,
            int y,
            int wartosc)
```

Wstawia wartosc `wartosc` do macierzy na pozycje (x, y).

Parameters

x	Indeks wiersza.
y	Indeks kolumny.
wartosc	Wartosc do wstawienia.

4.1.4 Friends And Related Symbol Documentation

4.1.4.1 operator*

```
Matrix operator* (
            int a,
            Matrix & m)  [friend]
```

Operator mnozenia liczby calkowitej przez macierz.

Parameters

a	Liczba calkowita do pomnozenia.
m	Macierz, ktora mnozymy przez liczbe.

Returns

Nowa macierz po pomnozeniu.

4.1.4.2 operator+

```
Matrix operator+ (
            int a,
            Matrix & m)  [friend]
```

Operator dodawania liczby calkowitej do macierzy.

Parameters

a	Liczba calkowita do dodania.
m	Macierz do ktorej dodajemy liczbe.

Returns

Nowa macierz po dodaniu liczby.

4.1.4.3 operator-

```
Matrix operator- (
            int a,
            Matrix & m)  [friend]
```

Operator odejmowania liczby calkowitej od macierzy.

Parameters

a	Liczba calkowita do odjecia.
m	Macierz, od ktorej odejmujemy liczbe.

Returns

Nowa macierz po odjeciu.

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

- Matrix.h
- Matrix.cpp

Chapter 5

File Documentation

5.1 Matrix.h

```
00001 #pragma once
00002 #include <string>
00003
00013 class Matrix {
00014 public:
00015     int** wsm;
00016     int rozmiar;
00017
00022     Matrix();
00023
00029     Matrix(int n);
00030
00037     Matrix(int n, int* t);
00038
00044     Matrix(const Matrix& m);
00045
00049     ~Matrix();
00050
00055     Matrix& Macierz_Alokacja(int n);
00056
00062     Matrix& wstaw(int x, int y, int wartosc);
00063
00069     int pokaz(int x, int y);
00070
00074     Matrix& Macierz_Odwroc();
00075
00079     Matrix& losuj();
00080
00085     Matrix& losuj(int x);
00086
00091     Matrix& diagonalna(int* t);
00092
00099     Matrix& diagonalna_k(int k, int* t);
00100
00106     Matrix& kolumna(int x, int* t);
00107
00113     Matrix& wiersz(int y, int* t);
00114
00118     Matrix& przekatna(void);
00119
00123     Matrix& pod_przekatna(void);
00124
00128     Matrix& nad_przekatna(void);
00129
00133     Matrix& szachownica(void);
00134
00139     Matrix& operator+(Matrix& m);
00140
00145     Matrix& operator*(Matrix& m);
00146
00151     Matrix& operator+(int a);
00152
00157     Matrix& operator*(int a);
00158
00163     Matrix& operator-(int a);
00164
00170     friend Matrix operator+(int a, Matrix& m);
00171
```

```
00177     friend Matrix operator*(int a, Matrix& m);
00178
00184     friend Matrix operator-(int a, Matrix& m);
00185
00192     Matrix& operator=(const Matrix& m);
00193 };
```

Index