

Działania na macierzach z zastosowaniem Github Copilot

Generated by Doxygen 1.12.0

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 matrix Class Reference	5
3.1.1 Detailed Description	7
3.1.2 Constructor & Destructor Documentation	7
3.1.2.1 matrix() [1/4]	7
3.1.2.2 matrix() [2/4]	7
3.1.2.3 matrix() [3/4]	7
3.1.2.4 matrix() [4/4]	7
3.1.3 Member Function Documentation	8
3.1.3.1 alokuj()	8
3.1.3.2 diagonalna()	8
3.1.3.3 diagonalna_k()	8
3.1.3.4 dowroc()	9
3.1.3.5 kolumna()	9
3.1.3.6 losuj() [1/2]	10
3.1.3.7 losuj() [2/2]	10
3.1.3.8 nad_przekatna()	11
3.1.3.9 operator()	11
3.1.3.10 operator*() [1/2]	11
3.1.3.11 operator*() [2/2]	12
3.1.3.12 operator*=()	12
3.1.3.13 operator+() [1/2]	13
3.1.3.14 operator+() [2/2]	13
3.1.3.15 operator++()	13
3.1.3.16 operator+=()	13
3.1.3.17 operator-()	14
3.1.3.18 operator--()	14
3.1.3.19 operator-=()	14
3.1.3.20 operator<()	15
3.1.3.21 operator==()	15
3.1.3.22 operator>()	16
3.1.3.23 pod_przekatna()	16
3.1.3.24 pokaz()	17
3.1.3.25 przekatna()	17
3.1.3.26 szachownica()	17
3.1.3.27 wiersz()	18
3.1.3.28 wstaw()	19

3.1.4 Friends And Related Symbol Documentation	20
3.1.4.1 operator*	20
3.1.4.2 operator+	21
3.1.4.3 operator-	21
3.1.4.4 operator<<	22
4 File Documentation	23
4.1 C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/class.cpp	
File Reference	23
4.1.1 Detailed Description	23
4.1.2 Function Documentation	23
4.1.2.1 operator*()	23
4.1.2.2 operator+()	24
4.1.2.3 operator-()	24
4.1.2.4 operator<<()	24
4.2 class.h	25
4.3 C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/zaawansowane4.cpp	
File Reference	26
4.3.1 Detailed Description	26
4.3.2 Function Documentation	26
4.3.2.1 main()	26
Index	27

Chapter 1

Class Index

1.1 Class List

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

matrix	Klasa reprezentuj¹ca macierz kwadratow¹	5
------------------------	---	-------------------

Chapter 2

File Index

2.1 File List

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

C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/ class.cpp	
Implementacja klasy matrix	23
C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/ class.h . .	25
C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/ zaawansowane4.cpp	
Program demonstracyjny dla klasy matrix	26

Chapter 3

Class Documentation

3.1 matrix Class Reference

Klasa reprezentuj¹ca macierz kwadratow¹.

```
#include <class.h>
```

Public Member Functions

- `matrix` (void)
Konstruktor domylny klasy matrix.
- `matrix` (int n)
Konstruktor klasy matrix, który alokuje macierz o rozmiarze $n \times n$.
- `matrix` (int n, int *t)
Konstruktor klasy matrix, który alokuje macierz o rozmiarze $n \times n$ i wype³nia j¹ wartociami z tablicy t.
- `matrix` (matrix &m)
Konstruktor kopiuj¹cy klasy matrix.
- `~matrix` (void)
Destruktor klasy matrix.
- `matrix & alokuj` (int n)
Alokuje pamieæ dla macierzy o rozmiarze $n \times n$.
- `matrix & wstaw` (int x, int y, int wartosc)
Wstawia wartoæ do macierzy w okrelonej pozycji.
- `int pokaz` (int x, int y)
Zwraca wartoæ z macierzy z okrelonej pozycji.
- `matrix & dowroc` (void)
Odwraca macierz (transpozycja).
- `matrix & losuj` (void)
Wype³nia macierz losowymi wartociami.
- `matrix & losuj` (int x)
Wype³nia macierz losowymi wartociami w okrelonej liczbie komórek.
- `matrix & diagonalna` (int *t)
Ustawia wartoci na przek¹tnej macierzy.
- `matrix & diagonalna_k` (int k, int *t)
Ustawia wartoci na przek¹tnej macierzy z przesuniêciem.

- `matrix & kolumna (int x, int *t)`
Ustawia wartosci w kolumnie macierzy.
- `matrix & wiersz (int y, int *t)`
Ustawia wartosci w wierszu macierzy.
- `matrix & przekatna (void)`
Ustawia wartosci na przekatnej macierzy na 1, a pozostale na 0.
- `matrix & pod_przekatna (void)`
Ustawia wartosci pod przekatna macierzy na 1, a pozostale na 0.
- `matrix & nad_przekatna (void)`
Ustawia wartosci nad przekatna macierzy na 1, a pozostale na 0.
- `matrix & szachownica (void)`
Ustawia wartosci w macierzy w formie szachownicy.
- `matrix & operator+ (matrix &m)`
Operator dodawania macierzy.
- `matrix & operator* (matrix &m)`
Operator mnozenia macierzy.
- `matrix & operator+ (int a)`
Operator dodawania liczby do macierzy.
- `matrix & operator* (int a)`
Operator mnozenia macierzy przez liczbe.
- `matrix & operator- (int a)`
Operator odejmowania liczby od macierzy.
- `matrix & operator++ (int)`
Operator inkrementacji macierzy.
- `matrix & operator-- (int)`
Operator dekrementacji macierzy.
- `matrix & operator+= (int a)`
Operator dodawania liczby do macierzy.
- `matrix & operator-= (int a)`
Operator odejmowania liczby od macierzy.
- `matrix & operator*= (int a)`
Operator mnozenia macierzy przez liczbe.
- `matrix & operator() (double value)`
Operator dodawania liczby do macierzy.
- `bool operator== (const matrix &m)`
Operator porownania rownoci macierzy.
- `bool operator> (const matrix &m)`
Operator porownania wiekszoci macierzy.
- `bool operator< (const matrix &m)`
Operator porownania mniejszoci macierzy.

Friends

- `matrix operator+ (int a, matrix &m)`
Operator dodawania liczby do macierzy (globalny).
- `matrix operator* (int a, matrix &m)`
Operator mnozenia liczby przez macierz (globalny).
- `matrix operator- (int a, matrix &m)`
Operator odejmowania liczby od macierzy (globalny).
- `ostream & operator<< (ostream &o, matrix &m)`
Operator wyjcia strumienia dla obiektu matrix.

3.1.1 Detailed Description

Klasa reprezentująca macierz kwadratową¹.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 matrix() [1/4]

```
matrix::matrix (
    void )
```

Konstruktor domylny klasy matrix.

Konstruktor domylny klasy matrix.

3.1.2.2 matrix() [2/4]

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

Konstruktor klasy matrix, który alokuje macierz o rozmiarze $n \times n$.

Konstruktor klasy matrix, który alokuje macierz o rozmiarze $n \times n$.

Parameters

n	Rozmiar macierzy.
-----	-------------------

3.1.2.3 matrix() [3/4]

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

Konstruktor klasy matrix, który alokuje macierz o rozmiarze $n \times n$ i wypełnia ją wartościami z tablicy t .

Konstruktor klasy matrix, który alokuje macierz o rozmiarze $n \times n$ i wypełnia ją wartościami z tablicy t .

Parameters

n	Rozmiar macierzy.
t	Tablica wartości do wypełnienia macierzy.
n	Rozmiar macierzy.
t	Tablica wartości do wypełnienia macierzy.

3.1.2.4 matrix() [4/4]

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

Konstruktor kopiujący klasy matrix.

Konstruktor kopiujący klasy matrix.

Parameters

<i>m</i>	Obiekt matrix, który ma zostać skopiowany.
<i>m</i>	Obiekt matrix, który ma zostać skopiowany.

3.1.3 Member Function Documentation

3.1.3.1 alokuj()

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

Alokuje pamięć dla macierzy o rozmiarze $n \times n$.

Alokuje pamięć dla macierzy o rozmiarze $n \times n$.

Parameters

<i>n</i>	Rozmiar macierzy.
----------	-------------------

Returns

Referencja do obiektu matrix.

3.1.3.2 diagonalna()

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

Ustawia wartości na przekątnej macierzy.

Ustawia wartości na przekątnej macierzy.

Parameters

<i>t</i>	Tablica wartości do ustawienia na przekątnej.
----------	---

Returns

Referencja do obiektu matrix.

Parameters

<i>t</i>	Tablica wartości do ustawienia na przekątnej.
----------	---

Returns

Referencja do obiektu matrix.

3.1.3.3 diagonalna_k()

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

Ustawia wartości na przekątnej macierzy z przesunięciem.

Ustawia wartości na przekątnej macierzy z przesunięciem.

Parameters

k	Przesunięcie przekątnej.
t	Tablica wartości do ustawienia na przekątnej.

Returns

Referencja do obiektu matrix.

Parameters

k	Przesunięcie przekątnej.
t	Tablica wartości do ustawienia na przekątnej.

Returns

Referencja do obiektu matrix.

3.1.3.4 dowroc()

```
matrix & matrix::dowroc (
    void )
```

Odwraca macierz (transpozycja).

Returns

Referencja do obiektu matrix.

3.1.3.5 kolumna()

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

Ustawia wartości w kolumnie macierzy.

Ustawia wartości w kolumnie macierzy.

Parameters

x	Numer kolumny.
t	Tablica wartości do ustawienia w kolumnie.

Returns

Referencja do obiektu matrix.

Parameters

<i>x</i>	Numer kolumny.
<i>t</i>	Tablica wartości do ustawienia w kolumnie.

Returns

Referencja do obiektu matrix.

3.1.3.6 losuj() [1/2]

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

Wypełnia macierz losowymi wartościami w określonej liczbie komórek.

Wypełnia macierz losowymi wartościami w określonej liczbie komórek.

Parameters

<i>x</i>	Liczba komórek do wypełnienia.
----------	--------------------------------

Returns

Referencja do obiektu matrix.

Parameters

<i>x</i>	Liczba komórek do wypełnienia.
----------	--------------------------------

Returns

Referencja do obiektu matrix.

3.1.3.7 losuj() [2/2]

```
matrix & matrix::losuj (  
    void )
```

Wypełnia macierz losowymi wartościami.

Wypełnia macierz losowymi wartościami.

Returns

Referencja do obiektu matrix.

3.1.3.8 nad_przekatna()

```
matrix & matrix::nad_przekatna (
    void )
```

Ustawia wartosci nad przekatna macierzy na 1, a pozostale na 0.

Ustawia wartosci nad przekatna macierzy na 1, a pozostale na 0.

Returns

Referencja do obiektu matrix.

3.1.3.9 operator()

```
matrix & matrix::operator() (
    double value)
```

Operator dodawania liczby do macierzy.

Parameters

<i>value</i>	Wartość do dodania.
--------------	---------------------

Returns

Referencja do obiektu matrix.

Parameters

<i>value</i>	Wartość do dodania.
--------------	---------------------

Returns

Referencja do obiektu matrix.

3.1.3.10 operator*() [1/2]

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

Operator mnożenia macierzy przez liczbę.

Operator mnożenia macierzy przez liczbę.

Parameters

<i>a</i>	Liczba do pomnożenia.
----------	-----------------------

Returns

Referencja do obiektu matrix.

Parameters

<i>a</i>	Liczba do pomnożenia.
----------	-----------------------

Returns

Referencja do obiektu matrix.

3.1.3.11 operator*() [2/2]

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

Operator mnożenia macierzy.

Operator mnożenia macierzy.

Parameters

<i>m</i>	Macierz do pomnożenia.
----------	------------------------

Returns

Referencja do obiektu matrix.

Parameters

<i>m</i>	Macierz do pomnożenia.
----------	------------------------

Returns

Referencja do obiektu matrix.

3.1.3.12 operator*=()

```
matrix & matrix::operator*= (  
    int a)
```

Operator mnożenia macierzy przez liczbę.

Operator mnożenia macierzy przez liczbę.

Parameters

<i>a</i>	Liczba do pomnożenia.
----------	-----------------------

Returns

Referencja do obiektu matrix.

Parameters

<i>a</i>	Liczba do pomnożenia.
----------	-----------------------

Returns

Referencja do obiektu matrix.

3.1.3.13 operator+() [1/2]

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

Operator dodawania liczby do macierzy.

Parameters

<i>a</i>	Liczba do dodania.
----------	--------------------

Returns

Referencja do obiektu matrix.

3.1.3.14 operator+() [2/2]

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

Operator dodawania macierzy.

Parameters

<i>m</i>	Macierz do dodania.
----------	---------------------

Returns

Referencja do obiektu matrix.

3.1.3.15 operator++()

```
matrix & matrix::operator++ (  
    int )
```

Operator inkrementacji macierzy.

Returns

Referencja do obiektu matrix.

3.1.3.16 operator+=()

```
matrix & matrix::operator+= (  
    int a)
```

Operator dodawania liczby do macierzy.

Parameters

<i>a</i>	Liczba do dodania.
----------	--------------------

Returns

Referencja do obiektu matrix.

3.1.3.17 operator-()

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

Operator odejmowania liczby od macierzy.

Parameters

<i>a</i>	Liczba do odjęcia.
----------	--------------------

Returns

Referencja do obiektu matrix.

Parameters

<i>a</i>	Liczba do odjęcia.
----------	--------------------

Returns

Referencja do obiektu matrix.

3.1.3.18 operator--()

```
matrix & matrix::operator-- (
    int )
```

Operator dekrementacji macierzy.

Returns

Referencja do obiektu matrix.

3.1.3.19 operator-=()

```
matrix & matrix::operator-= (
    int a)
```

Operator odejmowania liczby od macierzy.

Parameters

<i>a</i>	Liczba do odjęcia.
----------	--------------------

Returns

Referencja do obiektu matrix.

Parameters

<i>a</i>	Liczba do odjęcia.
----------	--------------------

Returns

Referencja do obiektu matrix.

3.1.3.20 operator<()

```
bool matrix::operator< (
    const matrix & m)
```

Operator porównania mniejszoci macierzy.

Operator porównania mniejszości macierzy.

Parameters

<i>m</i>	Macierz do porównania.
----------	------------------------

Returns

true jeśli macierz jest mniejsza, false w przeciwnym razie.

Parameters

<i>m</i>	Macierz do porównania.
----------	------------------------

Returns

true jeśli macierz jest mniejsza, false w przeciwnym razie.

3.1.3.21 operator==()

```
bool matrix::operator==(
    const matrix & m)
```

Operator porównania równości macierzy.

Operator porównania równości macierzy.

Parameters

<i>m</i>	Macierz do porównania.
----------	------------------------

Returns

true jeli macierze s¹ równe, false w przeciwnym razie.

Parameters

<i>m</i>	Macierz do porównania.
----------	------------------------

Returns

true jeÅli macierze s¹ r³wne, false w przeciwnym razie.

3.1.3.22 operator>()

```
bool matrix::operator> (
    const matrix & m)
```

Operator porównania wiêkszości macierzy.

Operator porównania wiêkszości macierzy.

Parameters

<i>m</i>	Macierz do porównania.
----------	------------------------

Returns

true jeli bieÅca macierz jest wiêksza, false w przeciwnym razie.

Parameters

<i>m</i>	Macierz do porównania.
----------	------------------------

Returns

true jeÅli bieÅca macierz jest wiêksza, false w przeciwnym razie.

3.1.3.23 pod_przekatna()

```
matrix & matrix::pod_przekatna (
    void )
```

Ustawia wartosci pod przekatn¹ macierzy na 1, a pozosta³e na 0.

Ustawia wartosci pod przekatn¹ macierzy na 1, a pozosta³e na 0.

Returns

Referencja do obiektu matrix.

3.1.3.24 pokaz()

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

Zwraca wartoœæ z macierzy z okrełonej pozycji.

Zwraca wartoœæ z macierzy z okrełonej pozycji.

Parameters

x	Wiersz.
y	Kolumna.

Returns

Wartoœæ z macierzy.

Parameters

x	Wiersz.
y	Kolumna.

Returns

Wartoœæ z macierzy.

3.1.3.25 przekatna()

```
matrix & matrix::przekatna (
    void )
```

Ustawia wartoci na przekłtnej macierzy na 1, a pozosta³e na 0.

Ustawia wartoœci na przekłtnej macierzy na 1, a pozostaœe na 0.

Returns

Referencja do obiektu matrix.

3.1.3.26 szachownica()

```
matrix & matrix::szachownica (
    void )
```

Ustawia wartoci w macierzy w formie szachownicy.

Ustawia wartoœci w macierzy w formie szachownicy.

Returns

Referencja do obiektu matrix.

3.1.3.27 wiersz()

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

Ustawia wartosci w wierszu macierzy.

Ustawia wartosci w wierszu macierzy.

Parameters

<i>y</i>	Numer wiersza.
<i>t</i>	Tablica wartoci do ustawienia w wierszu.

Returns

Referencja do obiektu matrix.

Parameters

<i>y</i>	Numer wiersza.
<i>t</i>	Tablica wartoÅci do ustawienia w wierszu.

Returns

Referencja do obiektu matrix.

3.1.3.28 wstaw()

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

Wstawia wartoæ do macierzy w okrelonej pozycji.

Wstawia wartoÅÅ do macierzy w okreÅlonej pozycji.

Parameters

<i>x</i>	Wiersz.
<i>y</i>	Kolumna.
<i>wartosc</i>	Wartoæ do wstawienia.

Returns

Referencja do obiektu matrix.

Parameters

<i>x</i>	Wiersz.
<i>y</i>	Kolumna.
<i>wartosc</i>	WartoÅÅ do wstawienia.

Returns

Referencja do obiektu matrix.

3.1.4 Friends And Related Symbol Documentation

3.1.4.1 `operator*`

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

Operator mnożenia liczby przez macierz (globalny).

Parameters

<i>a</i>	Liczba do pomnożenia.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem mnożenia.

Parameters

<i>a</i>	Liczba do pomnożenia.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem mnożenia.

3.1.4.2 operator+

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

Operator dodawania liczby do macierzy (globalny).

Parameters

<i>a</i>	Liczba do dodania.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem dodawania.

Parameters

<i>a</i>	Liczba do dodania.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem dodawania.

3.1.4.3 operator-

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

Operator odejmowania liczby od macierzy (globalny).

Parameters

<i>a</i>	Liczba do odjęcia.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem odejmowania.

Parameters

<i>a</i>	Liczba do odjęcia.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem odejmowania.

3.1.4.4 operator<<

```
ostream & operator<< (
    ostream & o,
    matrix & m) [friend]
```

Operator wyjścia strumienia dla obiektu matrix.

Parameters

<i>o</i>	Strumień wyjściowy.
<i>m</i>	Obiekt matrix do wyświetlenia.

Returns

Strumień wyjściowy.

Parameters

<i>o</i>	Strumień wyjściowy.
<i>m</i>	Obiekt matrix do wyświetlenia.

Returns

Strumień wyjściowy.

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

- C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/class.h
- C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/class.cpp

Chapter 4

File Documentation

4.1 C:/Users/gfiej/OneDrive/↵

Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/class.cpp

File Reference

Implementacja klasy matrix.

```
#include "class.h"
#include <iostream>
#include <ostream>
#include <ctime>
#include <cstdlib>
```

Functions

- `matrix operator+` (int a, `matrix` &m)
Operator dodawania liczby do macierzy (globalny).
- `matrix operator*` (int a, `matrix` &m)
Operator mnożenia liczby przez macierz (globalny).
- `matrix operator-` (int a, `matrix` &m)
Operator odejmowania liczby od macierzy (globalny).
- `ostream & operator<<` (ostream &o, `matrix` &m)
Operator wyjścia strumienia dla obiektu matrix.

4.1.1 Detailed Description

Implementacja klasy matrix.

4.1.2 Function Documentation

4.1.2.1 `operator*()`

```
matrix operator* (  
    int a,  
    matrix & m)
```

Operator mnożenia liczby przez macierz (globalny).

Operator mnożenia liczby przez macierz (globalny).

Parameters

<i>a</i>	Liczba do pomnożenia.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem mnożenia.

4.1.2.2 operator+()

```
matrix operator+ (
    int a,
    matrix & m)
```

Operator dodawania liczby do macierzy (globalny).

Parameters

<i>a</i>	Liczba do dodania.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem dodawania.

4.1.2.3 operator-()

```
matrix operator- (
    int a,
    matrix & m)
```

Operator odejmowania liczby od macierzy (globalny).

Parameters

<i>a</i>	Liczba do odjęcia.
<i>m</i>	Macierz.

Returns

Nowa macierz będąca wynikiem odejmowania.

4.1.2.4 operator<<()

```
ostream & operator<< (
    ostream & o,
    matrix & m)
```

Operator wyjścia strumienia dla obiektu matrix.

Operator wyjścia strumienia dla obiektu matrix.

Parameters

<i>o</i>	Strumień wyjściowy.
<i>m</i>	Obiekt matrix do wyświetlenia.

Returns

Strumień wyjściowy.

4.2 class.h

```

00001 #pragma once
00002 #include <iostream>
00003 #include <ostream>
00004 using namespace std;
00005
00010 class matrix {
00011 private:
00012     int* dlug;
00013     int** mac;
00014
00015 public:
00019     matrix(void);
00020
00025     matrix(int n);
00026
00032     matrix(int n, int* t);
00033
00038     matrix(matrix& m);
00039
00043     ~matrix(void);
00044
00050     matrix& alokuj(int n);
00051
00059     matrix& wstaw(int x, int y, int wartosc);
00060
00067     int pokaz(int x, int y);
00068
00073     matrix& dowroc(void);
00074
00079     matrix& losuj(void);
00080
00086     matrix& losuj(int x);
00087
00093     matrix& diagonalna(int* t);
00094
00101     matrix& diagonalna_k(int k, int* t);
00102
00109     matrix& kolumna(int x, int* t);
00110
00117     matrix& wiersz(int y, int* t);
00118
00123     matrix& przekatna(void);
00124
00129     matrix& pod_przekatna(void);
00130
00135     matrix& nad_przekatna(void);
00136
00141     matrix& szachownica(void);
00142
00148     matrix& operator+(matrix& m);
00149
00155     matrix& operator*(matrix& m);
00156
00162     matrix& operator+(int a);
00163
00169     matrix& operator*(int a);
00170
00176     matrix& operator-(int a);
00177
00184     friend matrix operator+(int a, matrix& m);
00185
00192     friend matrix operator*(int a, matrix& m);
00193
00200     friend matrix operator-(int a, matrix& m);
00201
00206     matrix& operator++(int);

```

```

00207
00212     matrix& operator--(int);
00213
00219     matrix& operator+=(int a);
00220
00226     matrix& operator-=(int a);
00227
00233     matrix& operator*=(int a);
00234
00240     matrix& operator() (double value);
00241
00248     friend ostream& operator<<(ostream& o, matrix& m);
00249
00255     bool operator==(const matrix& m);
00256
00262     bool operator>(const matrix& m);
00263
00269     bool operator<(const matrix& m);
00270 };

```

4.3 C:/Users/gfiej/OneDrive/↵

Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/zaawansowa File Reference

Program demonstracyjny dla klasy matrix.

```

#include <iostream>
#include <fstream>
#include <ostream>
#include <string>
#include "class.h"

```

Functions

- int `main` ()
Funkcja główna programu.

4.3.1 Detailed Description

Program demonstracyjny dla klasy matrix.

4.3.2 Function Documentation

4.3.2.1 main()

```
int main ()
```

Funkcja główna programu.

Returns

Kod zakończenia programu.

Index

alokuj
 matrix, 8

C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/class.cpp, 23

C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/class.h, 25

C:/Users/gfiej/OneDrive/Desktop/projects/zaawansowane4/zaawansowane4/zaawansowane4/zaawansowane4.cpp, 26

class.cpp
 operator<<, 24
 operator+, 24
 operator-, 24
 operator*, 23

diagonalna
 matrix, 8

diagonalna_k
 matrix, 8

dowroc
 matrix, 9

kolumna
 matrix, 9

losuj
 matrix, 10

main
 zaawansowane4.cpp, 26

matrix, 5
 alokuj, 8
 diagonalna, 8
 diagonalna_k, 8
 dowroc, 9
 kolumna, 9
 losuj, 10
 matrix, 7
 nad_przekatna, 10
 operator<, 15
 operator<<, 22
 operator>, 16
 operator(), 11
 operator+, 13, 21
 operator++, 13
 operator+=, 13
 operator-, 14, 21
 operator--, 14
 operator-=, 14
 operator==, 15
 operator*, 11, 12, 20
 operator*==, 12
 pod_przekatna, 16
 pokaz, 16
 przekatna, 17
 szachownica

operator*==, 12
pod_przekatna, 16
pokaz, 16
przekatna, 17
szachownica, 17
wstaw, 19
nad_przekatna
 matrix, 10

operator<
 matrix, 15

operator<<
 class.cpp, 24
 matrix, 22

operator>
 matrix, 16

operator()
 matrix, 11

operator+
 class.cpp, 24
 matrix, 13, 21

operator++
 matrix, 13

operator+=
 matrix, 13

operator-
 class.cpp, 24
 matrix, 14, 21

operator--
 matrix, 14

operator=
 matrix, 14

operator==
 matrix, 15

operator*
 class.cpp, 23
 matrix, 11, 12, 20

operator*==
 matrix, 12

pod_przekatna
 matrix, 16

pokaz
 matrix, 16

przekatna
 matrix, 17

szachownica

matrix, [17](#)

wiersz

matrix, [17](#)

wstaw

matrix, [19](#)

zaawansowane4.cpp

main, [26](#)