

## My Project

Generated by Doxygen 1.8.17



<b>1 README</b>	<b>1</b>
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 File Index</b>	<b>7</b>
4.1 File List . . . . .	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 arkusz Class Reference . . . . .	9
5.1.1 Member Function Documentation . . . . .	9
5.1.1.1 getszerokosc() . . . . .	9
5.1.1.2 gettyp() . . . . .	9
5.1.1.3 getwysokosc() . . . . .	10
5.1.1.4 tworzenie_arkusza() . . . . .	10
5.1.1.5 uzyskaj_komorke() . . . . .	10
5.2 komorka Class Reference . . . . .	11
5.2.1 Member Function Documentation . . . . .	11
5.2.1.1 uzyskaj_war() . . . . .	11
5.3 komorka_double Class Reference . . . . .	12
5.3.1 Member Function Documentation . . . . .	12
5.3.1.1 dodaj_war() . . . . .	12
5.3.1.2 uzyskaj_war() . . . . .	12
5.4 komorka_string Class Reference . . . . .	13
5.4.1 Member Function Documentation . . . . .	13
5.4.1.1 dodaj_war() . . . . .	13
5.4.1.2 uzyskaj_war() . . . . .	13
<b>6 File Documentation</b>	<b>15</b>
6.1 main.cpp File Reference . . . . .	15
6.2 menu.cpp File Reference . . . . .	15
6.3 operacje_mat.cpp File Reference . . . . .	15
6.3.1 Function Documentation . . . . .	16
6.3.1.1 maximum() . . . . .	16
6.3.1.2 minimum() . . . . .	16
6.3.1.3 mnozenie_arkusza() . . . . .	17
6.3.1.4 srednia1() . . . . .	17
6.3.1.5 sumowanie_kolumn1() . . . . .	17
6.4 operacje_mat.h File Reference . . . . .	17
6.4.1 Function Documentation . . . . .	18
6.4.1.1 maximum() . . . . .	18

6.4.1.2 minimum()	18
6.4.1.3 mnozenie_arkusza()	19
6.4.1.4 srednia1()	19
6.4.1.5 sumowanie_kolumn1()	19
6.5 tablica.cpp File Reference	20
6.5.1 Function Documentation	20
6.5.1.1 nowatablica()	20
6.5.1.2 zapisanie_do_pliku()	20
6.6 tablica.h File Reference	21
6.6.1 Function Documentation	21
6.6.1.1 nowatablica()	21
6.6.1.2 zapisanie_do_pliku()	21
6.7 tablica_class.cpp File Reference	22
6.8 tablica_class.h File Reference	22
6.9 tablica_wysw.cpp File Reference	22
6.9.1 Function Documentation	22
6.9.1.1 wyswietltablica()	22
6.10 tablica_wysw.h File Reference	23
6.10.1 Function Documentation	23
6.10.1.1 wyswietltablica()	23
6.11 zapisz_arkusz.cpp File Reference	23
6.11.1 Function Documentation	24
6.11.1.1 wczytaj_arkusz()	24
6.11.1.2 zapisz_arkusz()	24
6.12 zapisz_arkusz.h File Reference	24
6.12.1 Function Documentation	24
6.12.1.1 wczytaj_arkusz()	24
6.12.1.2 zapisz_arkusz()	25
6.13 zmien_komorke.cpp File Reference	25
6.13.1 Function Documentation	25
6.13.1.1 zmien_komorke()	25
6.14 zmien_komorke.h File Reference	25
6.14.1 Function Documentation	26
6.14.1.1 zmien_komorke()	26

# Chapter 1

## README

Interfejs użytkownika Wytłumacz do czego program służy Wybierz opcje co chcesz zrobić (menu) Wpisz dane które chcesz użyć (tabela komórek) Co chcesz z nimi zrobić wybierz opcje (edytowanie komórek) Program po paru sekundach zwraca wynik operacji (edycja w tablicy i utworzenie nowej komórki z wynikiem) Zapyta czy chcesz powtórzyć daną czynność (możliwość wykonania programu jeszcze raz) Jeśli nie program, się zamknie nie zapisując danych trwale (za każdym razem cała tablica będzie resetowana)

Co program robi? Pobiera dane od użytkownika ile wierszy ile kolumn Tworzy plik [main.cpp](#) Odczytaj wybór z menu ([menu.cpp](#)) i zrób to co należy zrobić Użyj funkcje by łatwiej korzystać z programu (wyswietl dane)(funkcja.↔h)([tablica\\_wysw.cpp](#)) Wpisz dane podane przez użytkownika zapytaj czy są poprawne Wyślij dane do programu ([tablica.cpp](#)) Wyślane dane użyte w programie Przetwórz dane w programie tak by użytkownik był zadowolony Zwroc wynik użytkownikowi Zapytaj czy chce powtórzyć proces

#ProgramowanieObiektowe



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

arkusz . . . . .	9
komorka . . . . .	11
komorka_double . . . . .	12
komorka_string . . . . .	13





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">arkusz</a> . . . . .	9
<a href="#">komorka</a> . . . . .	11
<a href="#">komorka_double</a> . . . . .	12
<a href="#">komorka_string</a> . . . . .	13



## Chapter 4

# File Index

### 4.1 File List

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

<a href="#">main.cpp</a>	15
<a href="#">menu.cpp</a>	15
<b>menu.h</b>	<b>??</b>
<a href="#">operacje_mat.cpp</a>	15
<a href="#">operacje_mat.h</a>	17
<a href="#">tablica.cpp</a>	20
<a href="#">tablica.h</a>	21
<a href="#">tablica_class.cpp</a>	22
<a href="#">tablica_class.h</a>	22
<a href="#">tablica_wysw.cpp</a>	22
<a href="#">tablica_wysw.h</a>	23
<a href="#">zapisz_arkusz.cpp</a>	23
<a href="#">zapisz_arkusz.h</a>	24
<a href="#">zmien_komorke.cpp</a>	25
<a href="#">zmien_komorke.h</a>	25



## Chapter 5

# Class Documentation

### 5.1 arkusz Class Reference

#### Public Member Functions

- void [tworzenie\\_arkusza](#) (bool \*typy\_kolumn, int szerokosc, int wysokosc)  
*Tworzy arkusz.*
- [komorka](#) \* [uzyskaj\\_komorkę](#) (int x, int y)
- int [getszerokosc](#) ()
- int [getwysokosc](#) ()  
*użycie getera do sprowadzenia szerokości*
- bool [gettyp](#) (int x)  
*użycie getera do sprowadzenia wysokości*

#### 5.1.1 Member Function Documentation

##### 5.1.1.1 getszerokosc()

```
int arkusz::getszerokosc ( )
```

#### Returns

int uzyskuje komórke

użycie getera do sprowadzenia szerokości

##### 5.1.1.2 gettyp()

```
bool arkusz::gettyp (
    int x )
```

użycie getera do sprowadzenia wysokości

**Parameters**

<i>x</i>	wartość sprawdzana czy kolumna jest równa double'owi czy stringowi
----------	--

**Returns**

true jeśli kolumna double, zwróć wartość  
false jeśli kolumna string, pomiń wartość w obliczeniach

użycie getera do sprowadzenia typu komórki

**5.1.1.3 getwysokosc()**

```
int arkusz::getwysokosc ( )
```

użycie getera do sprowadzenia szerokości

**Returns**

int zwróć szerokość

użycie getera do sprowadzenia wysokości

**5.1.1.4 tworzenie\_arkusza()**

```
void arkusz::tworzenie_arkusza (
    bool * typy_kolumn,
    int szerokosc,
    int wysokosc )
```

Tworzy arkusz.

**Parameters**

<i>typy_kolumn</i>	wykorzystuje parametr z klasy arkusz
<i>szerokosc</i>	wykorzystuje parametr z klasy arkusz
<i>wysokosc</i>	wykorzystuje parametr z klasy arkusz

**5.1.1.5 uzyskaj\_komorke()**

```
komorka * arkusz::uzyskaj_komorke (
    int x,
    int y )
```

## Parameters

<i>x</i>	wiersze - ilosc wierszy
<i>y</i>	kolumny - iloc kolumn

## Returns

komorka\* wykorzystuje to inna funkcja

wyluskuje komórkę

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

- [tablica\\_class.h](#)
- [tablica\\_class.cpp](#)

## 5.2 komorka Class Reference

Inheritance diagram for komorka:

### Public Member Functions

- virtual void **dodaj\_war** (std::string wartosc)=0
- virtual std::string **uzyskaj\_war** ()=0

### 5.2.1 Member Function Documentation

#### 5.2.1.1 uzyskaj\_war()

```
virtual std::string komorka::uzyskaj_war ( ) [pure virtual]
```

## Returns

std::string

Implemented in [komorka\\_double](#), and [komorka\\_string](#).

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

- [tablica\\_class.h](#)

## 5.3 komorka\_double Class Reference

Inheritance diagram for komorka\_double:

Collaboration diagram for komorka\_double:

### Public Member Functions

- void [dodaj\\_war](#) (std::string wartosc)
- std::string [uzyskaj\\_war](#) ()

### 5.3.1 Member Function Documentation

#### 5.3.1.1 dodaj\_war()

```
void komorka_double::dodaj_war (
    std::string wartosc ) [virtual]
```

funkcja zmieniająca stringa na double

Implements [komorka](#).

#### 5.3.1.2 uzyskaj\_war()

```
std::string komorka_double::uzyskaj_war ( ) [virtual]
```

##### Returns

std::string

##### Returns

return zwraca wartość w postaci stringa

Implements [komorka](#).

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

- [tablica\\_class.h](#)
- [tablica\\_class.cpp](#)



## 5.4 komorka\_string Class Reference

Inheritance diagram for komorka\_string:

Collaboration diagram for komorka\_string:

### Public Member Functions

- void [dodaj\\_war](#) (std::string wartosc)  
*jeśli nie wpisujemy wartości do komórki i wyświetlimy jej zawartość ukażą się nam wykrzykniki*
- std::string [uzyskaj\\_war](#) ()  
*pozwala dodać wartość za pomocą funkcji która wyluskuje samą wartość dla kalsy string*

### 5.4.1 Member Function Documentation

#### 5.4.1.1 dodaj\_war()

```
void komorka_string::dodaj_war (
    std::string wartosc ) [virtual]
```

jeśli nie wpisujemy wartości do komórki i wyświetlimy jej zawartość ukażą się nam wykrzykniki

funkcja wyluskuje wartość z komórki

Implements [komorka](#).

#### 5.4.1.2 uzyskaj\_war()

```
std::string komorka_string::uzyskaj_war ( ) [virtual]
```

pozwala dodać wartość za pomocą funkcji która wyluskuje samą wartość dla kalsy string

#### Returns

std::string dlatego że tworzymy stringi w tabeli i klasa odpowiada stringowi

Construct a new return object Uzyskuje wartość poprzez funkcje

Implements [komorka](#).

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

- [tablica\\_class.h](#)
- [tablica\\_class.cpp](#)



## Chapter 6

# File Documentation

### 6.1 main.cpp File Reference

```
#include "menu.h"  
#include "tablica_class.h"  
#include <iostream>  
Include dependency graph for main.cpp:
```

### 6.2 menu.cpp File Reference

```
#include <iostream>  
#include "tablica.h"  
#include "tablica_wysw.h"  
#include "tablica_class.h"  
#include "zapisz_arkusz.h"  
#include "zmien_komorke.h"  
#include "operacje_mat.h"  
#include <fstream>  
#include <cstdlib>  
Include dependency graph for menu.cpp:
```

### Functions

- void **wypiszmenu** ()
- void **menu** ()

### 6.3 operacje\_mat.cpp File Reference

```
#include <iostream>  
#include "tablica_class.h"  
Include dependency graph for operacje_mat.cpp:
```

## Functions

- void `mnozenie_arkusza` (`arkusz` \*mnozenie)  
*funkcja przemnaża wszystkie wartości w tabelce*
- double `sumowanie_kolumn1` (`arkusz` \*sumowanie\_kolumn)  
*funkcja sumuje kolumny od góry do dołu a następnie dzięki temu że jest double zwraca informacje o sumie w kolumnach i cała funkcja może zostać wykorzystana w obliczeniach*
- void `minimum` (`arkusz` \*min)  
*funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest mniejsza jeśli tak to zostawia ją jeśli nie to zmienia*
- void `maximum` (`arkusz` \*max)  
*funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest większa jeśli tak to zostawia ją jeśli nie to zmienia*
- void `srednia1` (`arkusz` \*srednia)  
*funkcja wywoływana jest poprzez obliczenia proste oraz funkcje sumowanie kolumn gdzie wylicza średnią ze wszystkich kolumn*

### 6.3.1 Function Documentation

#### 6.3.1.1 maximum()

```
void maximum (
    arkusz * max )
```

funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest większa jeśli tak to zostawia ją jeśli nie to zmienia

##### Parameters

<code>max</code>	parametr bazujący na prostej komendzie logicznej
------------------	--

#### 6.3.1.2 minimum()

```
void minimum (
    arkusz * min )
```

funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest mniejsza jeśli tak to zostawia ją jeśli nie to zmienia

##### Parameters

<code>min</code>	parametr który obliczany jest poprzez prostą komendę logiczną
------------------	---

### 6.3.1.3 mnozenie\_arkusza()

```
void mnozenie_arkusza (
    arkusz * mnozenie )
```

funkcja przemnaża wszystkie wartości w tabelce

funkcja przemnaża każdą komórkę przez siebie

#### Parameters

<i>mnozenie</i>	parametr który jest odpowiedzialny za mnożenie
-----------------	--

### 6.3.1.4 srednia1()

```
void srednia1 (
    arkusz * srednia )
```

funkcja wywoływana jest poprzez obliczenia proste oraz funkcje sumowanie kolumn gdzie wylicza średnią ze wszystkich kolumn

#### Parameters

<i>srednia</i>	parametr który wykorzystywany jest do obliczenia średniej poprzez sume kolumn i podzielenia przez ilość wszystkich komórek
----------------	--

### 6.3.1.5 sumowanie\_kolumn1()

```
double sumowanie_kolumn1 (
    arkusz * sumowanie_kolumn )
```

funkcja sumuje kolumny od góry do dołu a następnie dzięki temu że jest double zwraca informacje o sumie w kolumnach i cała funkcja może zostać wykorzystana w obliczeniach

#### Parameters

<i>sumowanie_kolumn</i>	
-------------------------	--

## 6.4 operacje\_mat.h File Reference

```
#include "tablica_class.h"
#include "zmien_komorka.h"
```

Include dependency graph for operacje\_mat.h: This graph shows which files directly or indirectly include this file:

## Functions

- void `mnozenie_arkusza` (`arkusz` \*mnozenie)  
*funkcja przemnaża każdą komórkę przez siebie*
- double `sumowanie_kolumn1` (`arkusz` \*sumowanie\_kolumn)  
*funkcja sumuje kolumny od góry do dołu a następnie dzięki temu że jest double zwraca informacje o sumie w kolumnach i cała funkcja może zostać wykorzystana w obliczeniach*
- void `srednia1` (`arkusz` \*srednia)  
*funkcja wywoływana jest poprzez obliczenia proste oraz funkcje sumowanie kolumn gdzie wylicza średnią ze wszystkich kolumn*
- void `minimum` (`arkusz` \*min)  
*funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest mniejsza jesli tak to zostawia ją jeśli nie to zmienia*
- void `maximum` (`arkusz` \*max)  
*funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest większa jesli tak to zostawia ją jeśli nie to zmienia*

### 6.4.1 Function Documentation

#### 6.4.1.1 maximum()

```
void maximum (
    arkusz * max )
```

funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest większa jesli tak to zostawia ją jeśli nie to zmienia

##### Parameters

<code>max</code>	parametr bazujący na prostej komendzie logicznej
------------------	--

#### 6.4.1.2 minimum()

```
void minimum (
    arkusz * min )
```

funkcja przeszukuje dane poprzez porównanie ze wcześniejszą komórką i sprawdzeniu czy dana komórka jest mniejsza jesli tak to zostawia ją jeśli nie to zmienia

##### Parameters

<code>min</code>	parametr który obliczany jest poprzez prostą komendę logiczną
------------------	---

#### 6.4.1.3 mnozenie\_arkusza()

```
void mnozenie_arkusza (
    arkusz * mnozenie )
```

funkcja przemnaża każdą komórkę przez siebie

##### Parameters

<i>mnozenie</i>	
-----------------	--

funkcja przemnaża każdą komórkę przez siebie

##### Parameters

<i>mnozenie</i>	parametr który jest odpowiedzialny za mnożenie
-----------------	--

#### 6.4.1.4 srednia1()

```
void srednial (
    arkusz * srednia )
```

funkcja wywoływana jest poprzez obliczenia proste oraz funkcje sumowanie kolumn gdzie wylicza średnią ze wszystkich kolumn

##### Parameters

<i>srednia</i>	parametr który wykorzystywany jest do obliczenia średniej poprzez sume kolumn i podzielenia przez ilość wszystkich komórek
----------------	--

#### 6.4.1.5 sumowanie\_kolumn1()

```
double sumowanie_kolumn1 (
    arkusz * sumowanie_kolumn )
```

funkcja sumuje kolumny od góry do dołu a następnie dzięki temu że jest double zwraca informację o sumie w kolumnach i cała funkcja może zostać wykorzystana w obliczeniach

##### Parameters

<i>sumowanie_kolumn</i>	
-------------------------	--

## 6.5 tablica.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include "tablica.h"
```

Include dependency graph for tablica.cpp:

### Functions

- int\*\* [nowatablica](#) (int x, int y)
- int\*\* [zapisanie\\_do\\_pliku](#) (int\*\*tablica, int \*x, int \*y)

### 6.5.1 Function Documentation

#### 6.5.1.1 nowatablica()

```
int** nowatablica (
    int x,
    int y )
```

##### Parameters

<i>x</i>	Wiersze - ilość wierszy
<i>y</i>	Kolumny - ilość kolumn

##### Returns

int\*\* tablica funkcja nie jest używana fizycznie zastąpiła ją funkcja tworzenie arkusza

#### 6.5.1.2 zapisanie\_do\_pliku()

```
int** zapisanie_do_pliku (
    int ** tablica,
    int * x,
    int * y )
```

##### Parameters

<i>tablica</i>	
<i>x</i>	
<i>y</i>	



## Returns

`int**`

## 6.6 `tablica.h` File Reference

This graph shows which files directly or indirectly include this file:

### Functions

- `int** nowatablica` (`int x`, `int y`)
- `int** zapisanie_do_pliku` (`int** tablica`, `int *x`, `int *y`)

### 6.6.1 Function Documentation

#### 6.6.1.1 `nowatablica()`

```
int** nowatablica (  
    int x,  
    int y )
```

## Parameters

<i>x</i>	Wiersze - ilość wierszy
<i>y</i>	Kolumny - ilość kolumn

## Returns

`int** tablica` funkcja nie jest używana fizycznie zastąpiła ją funkcja tworzenie arkusza

#### 6.6.1.2 `zapisanie_do_pliku()`

```
int** zapisanie_do_pliku (  
    int ** tablica,  
    int * x,  
    int * y )
```

## Parameters

<i>tablica</i>	
<i>x</i>	
<i>y</i>	

**Returns**

int\*\*

## 6.7 tablica\_class.cpp File Reference

```
#include "tablica_class.h"
```

Include dependency graph for tablica\_class.cpp:

## 6.8 tablica\_class.h File Reference

```
#include <string>
```

Include dependency graph for tablica\_class.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [komorka](#)
- class [komorka\\_string](#)
- class [komorka\\_double](#)
- class [arkusz](#)

## 6.9 tablica\_wysw.cpp File Reference

```
#include "tablica_wysw.h"
```

```
#include <iostream>
```

Include dependency graph for tablica\_wysw.cpp:

### Functions

- void [wyswietltablica](#) ([arkusz](#) tablica)  
*wyświetla tablice za pomocą wielu funkcji samych w sobie*

### 6.9.1 Function Documentation

#### 6.9.1.1 wyswietltablica()

```
void wyswietltablica (  
    arkusz tablica )
```

wyświetla tablice za pomocą wielu funkcji samych w sobie

## Parameters

tablica	
---------	--

## 6.10 tablica\_wysw.h File Reference

```
#include "tablica_class.h"
```

Include dependency graph for tablica\_wysw.h: This graph shows which files directly or indirectly include this file:

### Functions

- void [wyswietltablica](#) ([arkusz](#) tablica)  
*wyświetla tablice za pomocą wielu funkcji samych w sobie*

#### 6.10.1 Function Documentation

##### 6.10.1.1 wyswietltablica()

```
void wyswietltablica (
    arkusz tablica )
```

wyświetla tablice za pomocą wielu funkcji samych w sobie

## Parameters

tablica	
---------	--

## 6.11 zapisz\_arkusz.cpp File Reference

```
#include <fstream>
#include "tablica_class.h"
#include <string>
```

Include dependency graph for zapisz\_arkusz.cpp:

### Functions

- void [zapisz\\_arkusz](#) ([arkusz](#) zapisz)  
*tworzy plik zapisztablice.txt zapisuje tam wszystkie najważniejsze parametry takie tjak ilosc kolumn ilosc wierszy rodzaj kolumn i wartosc komorek*
- void [wczytaj\\_arkusz](#) ([arkusz](#) \*wczytaj)  
*tworzy od nowa tablice na bazie pliku zapisztablice.txt z wartościami takimi jakie się znajdują w pliku*

## 6.11.1 Function Documentation

### 6.11.1.1 wczytaj\_arkusz()

```
void wczytaj_arkusz (
    arkusz * )
```

tworzy od nowa tablice na bazie pliku zapisztablice.txt z wartościami takimi jakie się znajdują w pliku

### 6.11.1.2 zapisz\_arkusz()

```
void zapisz_arkusz (
    arkusz )
```

tworzy plik zapisztablice.txt zapisuje tam wszystkie najważniejsze parametry takie tjak ilosc kolumn ilosc wierszy rodzaj kolumn i wartosc komorek

## 6.12 zapisz\_arkusz.h File Reference

```
#include "tablica_class.h"
```

Include dependency graph for zapisz\_arkusz.h: This graph shows which files directly or indirectly include this file:

### Functions

- void [zapisz\\_arkusz](#) ([arkusz](#))  
*tworzy plik zapisztablice.txt zapisuje tam wszystkie najważniejsze parametry takie tjak ilosc kolumn ilosc wierszy rodzaj kolumn i wartosc komorek*
- void [wczytaj\\_arkusz](#) ([arkusz](#) \*)  
*tworzy od nowa tablice na bazie pliku zapisztablice.txt z wartościami takimi jakie się znajdują w pliku*

## 6.12.1 Function Documentation

### 6.12.1.1 wczytaj\_arkusz()

```
void wczytaj_arkusz (
    arkusz * )
```

tworzy od nowa tablice na bazie pliku zapisztablice.txt z wartościami takimi jakie się znajdują w pliku

### 6.12.1.2 zapisz\_arkusz()

```
void zapisz_arkusz (
    arkusz )
```

tworzy plik zapisztabelle.txt zapisuje tam wszystkie najważniejsze parametry takie tjak ilosc kolumn ilosc wierszy rodzaj kolumn i wartosc komorek

## 6.13 zmien\_komorke.cpp File Reference

```
#include <iostream>
#include "tablica_class.h"
Include dependency graph for zmien_komorke.cpp:
```

### Functions

- void [zmien\\_komorke](#) ([arkusz](#) zmien\_kom)  
*funkcja uzyskaj komorke uzyskuje komorke przez co mozna do niej to wpisac*

### 6.13.1 Function Documentation

#### 6.13.1.1 zmien\_komorke()

```
void zmien_komorke (
    arkusz zmien_kom )
```

funkcja uzyskaj komorke uzyskuje komorke przez co mozna do niej to wpisac

#### Parameters

<a href="#">zmien_kom</a>	
---------------------------	--

## 6.14 zmien\_komorke.h File Reference

```
#include "tablica_class.h"
Include dependency graph for zmien_komorke.h: This graph shows which files directly or indirectly include this file:
```

### Functions

- void [zmien\\_komorke](#) ([arkusz](#) zmien\_kom)  
*funkcja uzyskaj komorke uzyskuje komorke przez co mozna do niej to wpisac*

## 6.14.1 Function Documentation

### 6.14.1.1 zmien\_komorke()

```
void zmien_komorke (
    arkusz zmien_kom )
```

funkcja uzyskaj komorke uzyskuje komorke przez co mozna do niej to wpisac

#### Parameters

<i>zmien_kom</i>	
------------------	--

# Index

- arkusz, [9](#)
  - getszerokosc, [9](#)
  - gettyp, [9](#)
  - getwysokosc, [10](#)
  - tworzenie\_arkusza, [10](#)
  - uzyskaj\_komorce, [10](#)
- dodaj\_war
  - komorka\_double, [12](#)
  - komorka\_string, [13](#)
- getszerokosc
  - arkusz, [9](#)
- gettyp
  - arkusz, [9](#)
- getwysokosc
  - arkusz, [10](#)
- komorka, [11](#)
  - uzyskaj\_war, [11](#)
- komorka\_double, [12](#)
  - dodaj\_war, [12](#)
  - uzyskaj\_war, [12](#)
- komorka\_string, [13](#)
  - dodaj\_war, [13](#)
  - uzyskaj\_war, [13](#)
- main.cpp, [15](#)
- maximum
  - operacje\_mat.cpp, [16](#)
  - operacje\_mat.h, [18](#)
- menu.cpp, [15](#)
- minimum
  - operacje\_mat.cpp, [16](#)
  - operacje\_mat.h, [18](#)
- mnozenie\_arkusza
  - operacje\_mat.cpp, [16](#)
  - operacje\_mat.h, [18](#)
- nowatablica
  - tablica.cpp, [20](#)
  - tablica.h, [21](#)
- operacje\_mat.cpp, [15](#)
  - maximum, [16](#)
  - minimum, [16](#)
  - mnozenie\_arkusza, [16](#)
  - srednia1, [17](#)
  - sumowanie\_kolumn1, [17](#)
- operacje\_mat.h, [17](#)
  - maximum, [18](#)
- minimum, [18](#)
- mnozenie\_arkusza, [18](#)
- srednia1, [19](#)
- sumowanie\_kolumn1, [19](#)
- srednia1
  - operacje\_mat.cpp, [17](#)
  - operacje\_mat.h, [19](#)
- sumowanie\_kolumn1
  - operacje\_mat.cpp, [17](#)
  - operacje\_mat.h, [19](#)
- tablica.cpp, [20](#)
  - nowatablica, [20](#)
  - zapisanie\_do\_pliku, [20](#)
- tablica.h, [21](#)
  - nowatablica, [21](#)
  - zapisanie\_do\_pliku, [21](#)
- tablica\_class.cpp, [22](#)
- tablica\_class.h, [22](#)
- tablica\_wysw.cpp, [22](#)
  - wyswietltablica, [22](#)
- tablica\_wysw.h, [23](#)
  - wyswietltablica, [23](#)
- tworzenie\_arkusza
  - arkusz, [10](#)
- uzyskaj\_komorce
  - arkusz, [10](#)
- uzyskaj\_war
  - komorka, [11](#)
  - komorka\_double, [12](#)
  - komorka\_string, [13](#)
- wczytaj\_arkusz
  - zapisz\_arkusz.cpp, [24](#)
  - zapisz\_arkusz.h, [24](#)
- wyswietltablica
  - tablica\_wysw.cpp, [22](#)
  - tablica\_wysw.h, [23](#)
- zapisanie\_do\_pliku
  - tablica.cpp, [20](#)
  - tablica.h, [21](#)
- zapisz\_arkusz
  - zapisz\_arkusz.cpp, [24](#)
  - zapisz\_arkusz.h, [24](#)
- zapisz\_arkusz.cpp, [23](#)
  - wczytaj\_arkusz, [24](#)
- zapisz\_arkusz, [24](#)
- zapisz\_arkusz.h, [24](#)

wczytaj\_arkusz, [24](#)  
zapisz\_arkusz, [24](#)  
zmien\_komorke  
    zmien\_komorke.cpp, [25](#)  
    zmien\_komorke.h, [26](#)  
zmien\_komorke.cpp, [25](#)  
    zmien\_komorke, [25](#)  
zmien\_komorke.h, [25](#)  
    zmien\_komorke, [26](#)