Program

Wygenerowano przez Doxygen 1.8.17

Chapter 1

README

1. Interfejs uzytkownika :

- wprowadzenie do programu, wyswietlenie czym program jest - cout << "Program polega na..."
- wprowadzenie informacji do programu - cin >> tablica[i] w petli zaleznej od wielkosci tablicy podanej przez uzytkownika
- wyswietlenie informacji jakich operacji na danych mozemy wykonac - "cout" do wyswietlenia mozliwosci, wybor za pomoca "switch"
- po wykonaniu operacji wyswietlenie pytania czy program ma sie wylaczyc czy tez nie - program polega na petli ktora powtarza sie do momentu wybrania opcji wylaczenia
- wyswietlenie zaktualizowanych danych - "cout" zamieszczone w petli zaleznej od wielkosci tablicy
- kontynuowanie programu do momentu az uzytkownik go wylaczy - po probie wylaczenia programu petla na ktorej opiera sie program przestaje sie powtarzac, program zakonczy sie komenda return 0;

1. Arkusz :

- Program glowny :
- zapis informacji podanych przez uzytkownika - poprzez komunikacje z interfejsem uzytkownika, zapis za pomoca tabeli
- zapis danych powstalych w wyniku operacji - ewentualne usuniecie juz niepotrzebnych komponentow pomocniczych takich jak np tablica dynamiczna
- Funkcje :
- przeprowadzenie operacji podanej przez uzytkownika - wlaczenie funkcji odpowiedzialnej za dane zadanie (fukcje : zmieniajaca wielkosc tablicy, aktualizujaca zawartosc tablicy, wyswietlajaca zawartosc tablicy)
- przekazanie przetworzonych informacji do programu glownego

1. Pliki :

- Wprowadzanie danych :
- w przypadku gdy arkusz nie został utworzony tworzymy pusty arkusz ktorego wymiary mamy podane w pliku

- w przypadku gdy posiadamy już arkusz mamy 2 opcje :
  - wielkość jest zgodna z rozmiarami podanymi w pliku, wtedy nastepuje zerowanie arkusza i wprowadzanie wartosci podanych w pliku
  - wielkosc nie jest zgodna z rozmiarami podanymi w pliku, wtedy nastepuje usuniecie arkusza, stworzenie nowego arkusza który następnie zostanie "wyzerowany", ostatnia czynnoscia bedzie wczytanie wartosci z pliku do nowo utworzonego arkusza
- Zapisanie danych :
- do pliku tekstowego, za pomoca funkcji, zapisujemy kolejno wielkosc arkusza a nastepnie jego wartosci

Chapter 2

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Chapter 3

Indeks klas

3.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Chapter 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

Chapter 5

Dokumentacja klas

5.1 Dokumentacja klasy Arkusz

Metody publiczne

- int tworzenie_arkusza (int szerokosc, int wysokosc)

  Funkcja tworząca arkusz.
- void zmiana_rozmiaru (int szerokosc, int wysokosc)

  Funkcja modyfikująca wielkość arkusza.
- void zmiana_kolumny (int szerokosc, int typ)

  Funkcja zmieniająca typ komórek w danej kolumnie.
- Komorka * zwracanie_komorki (int szerokosc, int wysokosc)

  Funkcja zwracająca komórkę na podanych koordynatach.
- int zwracanie_szerokosci ()

  Funkcja zwracająca szerokość
- int zwracanie_wysokosci ()

  Funkcja zwracająca szerokość
- int zwracanie_typow (int szerokosc)

  Funkcja sprawdzająca typ komórek w kolumnie.
- void nadawanie_szerokosci (int szerokosc)

  Funkcja nadająca wartość szerokości arkusza.
- void nadawanie_wysokosci (int wysokosc)

  Funkcja nadająca wartość wysokości arkusza.
- void nadawanie_typow (int *typy)

  Funkcja nadająca wartość typów arkusza.
- double najwieksza_wartosc_kolumna (int szerokosc)

  Funkcja pokazująca największą wartość w kolumnie.
- double najmniejsza_wartosc_kolumna (int szerokosc)

  Funkcja pokazująca najmniejszą wartość w kolumnie.
- double suma_wartosc_kolumna (int szerokosc)

  Funkcja pokazująca sumę wartości w kolumnie.
- double srednia_wartosc_kolumna (int szerokosc)

  Funkcja pokazująca średnią wartości w kolumny.

5.1.1 Opis szczegółowy

Definicja w linii 23 pliku tablica.h.

5.1.2 Dokumentacja funkcji składowych

5.1.2.1 nadawanie_szerokosci()

```
void Arkusz::nadawanie_szerokosci (
            int szerokosc )
```

Funkcja nadająca wartość szerokości arkusza.

Parametry

in	*Szerokość*	Nadawana szerokość

Definicja w linii 159 pliku tablica.cpp.

```
160 {
161     Arkusz::szerokosc = szerokosc;
162 }
```

5.1.2.2 nadawanie_typow()

```
void Arkusz::nadawanie_typow (
            int * typy )
```

Funkcja nadająca wartość typów arkusza.

Parametry

in	*Typy*	Nadawane typy

Definicja w linii 169 pliku tablica.cpp.

```
170 {
171     Arkusz::typy = typy;
172 }
```

5.1.2.3 nadawanie_wysokosci()

```
void Arkusz::nadawanie_wysokosci (
            int wysokosc )
```

Funkcja nadająca wartość wysokości arkusza.

Parametry

in	*Wysokość*	Nadawana wysokość

Definicja w linii 164 pliku tablica.cpp.

```
165 {
166     Arkusz::wysokosc = wysokosc;
167 }
```

5.1.2.4 najmniejsza_wartosc_kolumna()

```
double Arkusz::najmniejsza_wartosc_kolumna (
            int szerokosc )
```

Funkcja pokazująca najmniejszą wartość w kolumnie.

Parametry

in	*Szerokość*	Szerokość na której szukana jest wartość
out	*Wartość*	Najmniejsza wartość w kolumnie

Definicja w linii 200 pliku tablica.cpp.

```
201 {
202     double wynik;
203     try
204     {
205         wynik = stod(Arkusz::zwracanie_komorki(szerokosc, 0)->zwracanie_wartosci());
206         for (int i = 0; i < Arkusz::zwracanie_wysokosci(); i++)
207         {
208             if (wynik > stod(Arkusz::zwracanie_komorki(szerokosc, i)->zwracanie_wartosci()))
209             {
210                 wynik = stod(Arkusz::zwracanie_komorki(szerokosc, i)->zwracanie_wartosci());
211             }
212         }
213         return wynik;
214     }
215     catch(...)
216     {
217         std::cout « "W podanej kolumnie nie ma liczb" « "\n";
218     }
219 }
```

5.1.2.5 najwieksza_wartosc_kolumna()

```
double Arkusz::najwieksza_wartosc_kolumna (
            int szerokosc )
```

Funkcja pokazująca największą wartość w kolumnie.

Parametry

in	*Szerokość*	Szerokość na której szukana jest wartość
out	*Wartość*	Największa wartość w kolumnie

Definicja w linii 177 pliku tablica.cpp.

```
178 {
179     double wynik;
180     try
181     {
182         wynik = stod(Arkusz::zwracanie_komorki(szerokosc, 0)->zwracanie_wartosci());
183         for (int i = 0; i < Arkusz::zwracanie_wysokosci(); i++)
184         {
185             if (wynik < stod(Arkusz::zwracanie_komorki(szerokosc, i)->zwracanie_wartosci()))
186             {
187                 wynik = stod(Arkusz::zwracanie_komorki(szerokosc, i)->zwracanie_wartosci());
188             }
189         }
190         return wynik;
191     }
192     catch(...)
193     {
194         std::cout « "W podanej kolumnie nie ma liczb" « "\n";
195     }
196 }
```

5.1.2.6 srednia_wartosc_kolumna()

```
double Arkusz::srednia_wartosc_kolumna (
            int szerokosc )
```

Funkcja pokazująca średnią wartości w kolumny.

Parametry

in	*Szerokość*	Szerokość z której wartości jest wyliczana średnia
out	*Wartość*	Średnia wartość w kolumnie

Definicja w linii 243 pliku tablica.cpp.

```
244 {
245     double wynik = 0;
246     try
247     {
248         for (int i = 0; i < Arkusz::zwracanie_wysokosci(); i++)
249         {
250             wynik += stod(Arkusz::zwracanie_komorki(szerokosc, i)->zwracanie_wartosci());
251         }
252         wynik = wynik / (Arkusz::wysokosc);
253         return wynik;
254     }
255     catch(...)
256     {
257         std::cout « "W podanej kolumnie nie ma liczb" « "\n";
258     }
259 }
```

5.1.2.7 suma_wartosc_kolumna()

```
double Arkusz::suma_wartosc_kolumna (
            int szerokosc )
```

Funkcja pokazująca sumę wartości w kolumnie.

Parametry

in	*Szerokość*	Szerokość z której wartości są sumowane
out	*Wartość*	Suma wartości w kolumnie

Definicja w linii 224 pliku tablica.cpp.

```
225 {
226     double wynik = 0;
227     try
228     {
229         for (int i = 0; i < Arkusz::zwracanie_wysokosci(); i++)
230         {
231             wynik += stod(Arkusz::zwracanie_komorki(szerokosc, i)->zwracanie_wartosci());
232         }
233         return wynik;
234     }
235     catch(...)
236     {
237         std::cout « "W podanej kolumnie nie ma liczb" « "\n";
238     }
239 }
```

5.1.2.8 tworzenie_arkusza()

```
int Arkusz::tworzenie_arkusza (
            int szerokosc,
            int wysokosc )
```

Funkcja tworząca arkusz.

Parametry

in	*Szerokość*	Preferowana szerokość arkusza
in	*Wysokość*	Preferowana wysokość arkusza
out	*Kod_błędu*	Zwracany kod błędu

Definicja w linii 8 pliku tablica.cpp.

```
9 {
10     if (szerokosc < 1)
11     {
12         return 1;
13     }
14     if (wysokosc < 1)
15     {
16         return 2;
17     }
18
19     typy = new int[szerokosc];
20
21     for (int i = 0; i < szerokosc; i++)
22     {
23         typy[i] = 0;
24     }
25
26     Arkusz::szerokosc = szerokosc;
27     Arkusz::wysokosc = wysokosc;
28     Arkusz::tablica = new Komorka**[Arkusz::szerokosc];
29     for (int i = 0; i < Arkusz::szerokosc; i++)
30     {
31         Arkusz::tablica[i] = new Komorka*[Arkusz::wysokosc];
32     }
33
34     for (int i = 0; i < Arkusz::szerokosc; i++)
35     {
36         for (int z = 0; z < Arkusz::wysokosc; z++)
37         {
38             Arkusz::tablica[i][z] = new Komorka_double;
39         }
40     };
41
42     return 0;
43 }
```

5.1.2.9 zmiana_kolumny()

```
void Arkusz::zmiana_kolumny (
            int szerokosc,
            int typ )
```

Funkcja zmieniająca typ komórek w danej kolumnie.

Parametry

in	*Szerokość*	Wybrana kolumna
in	*Typ*	Preferowany typ komórek

Definicja w linii 95 pliku tablica.cpp.

```
96 {
97     switch (typ)
98     {
99         case 0:
100         {
101             if (typy[szerokosc] == 0);
102             else
103             {
104                 for (int i = 0; i < Arkusz::wysokosc; i++)
105                 {
106                     tablica[szerokosc][i] = new Komorka_double;
107                 }
108                 typy[szerokosc] = 0;
109             }
110             break;
111         }
112
113         case 1:
114         {
115             if (typy[szerokosc] == 1);
116             else
117             {
118                 for (int i = 0; i < Arkusz::wysokosc; i++)
119                 {
120                     tablica[szerokosc][i] = new Komorka_string;
121                 }
122                 typy[szerokosc] = 1;
123             }
124             break;
125         }
126
127         default:
128         {
129             break;
130         }
131     }
132 }
```

5.1.2.10 zmiana_rozmiaru()

```
void Arkusz::zmiana_rozmiaru (
            int szerokosc,
            int wysokosc )
```

Funkcja modyfikująca wielkość arkusza.

Parametry

in	*Szerokość*	Preferowana szerokość arkusza
in	*Wysokość*	Preferowana wysokość arkusza

Definicja w linii 45 pliku tablica.cpp.

```
46 {
47
48     Komorka*** tablica_po_zmianie = new Komorka ** [szerokosc];
49
50     for (int k = 0; k < szerokosc; k++)
51     {
52         tablica_po_zmianie[k] = new Komorka*[wysokosc];
53     }
54
55     int* typy_po_zmianie = new int[szerokosc];
56
57     for (int i = 0; i < szerokosc; i++)
58     {
59         if (this->szerokosc > i)
60         {
61             typy_po_zmianie[i] = typy[i];
62         }
63         else
64         {
65             typy_po_zmianie[i] = 0;
66         }
67     }
68
69     for (int i = 0; i < szerokosc; i++)
70     {
71         for (int z = 0; z < wysokosc; z++)
72         {
73             if (i < this->szerokosc && z < this->wysokosc)
74             {
75                 tablica_po_zmianie[i][z] = tablica[i][z];
76             }
77             else if (typy_po_zmianie[i] == 0)
78             {
79                 tablica_po_zmianie[i][z] = new Komorka_double;
80             }
81             else if(typy_po_zmianie[i] == 1)
82             {
83                 tablica_po_zmianie[i][z] = new Komorka_string;
84             }
85         }
86     };
87
88     Arkusz::tablica = tablica_po_zmianie;
89     this->typy = typy_po_zmianie;
90     Arkusz::szerokosc = szerokosc;
91     Arkusz::wysokosc = wysokosc;
92
93 }
```

5.1.2.11 zwracanie_komorki()

```
Komorka * Arkusz::zwracanie_komorki (
            int szerokosc,
            int wysokosc )
```

Funkcja zwracająca komórkę na podanych koordynatach.

Parametry

in	*Szerokość*	Podana szerokość szukanej komórki
in	*Wysokość*	Podana wysokość szukanej komórki
out	*Komórka*	Zwracana komórka

Definicja w linii 136 pliku tablica.cpp.

```
137 {
138     return tablica[szerokosc][wysokosc];
139 }
```

5.1.2.12 zwracanie_szerokosci()

```
int Arkusz::zwracanie_szerokosci ( )
```

Funkcja zwracająca szerokość

Parametry

out	*Szerokość*	Zwracana szerokość

Definicja w linii 143 pliku tablica.cpp.

```
144 {
145     return szerokosc;
146 }
```

5.1.2.13 zwracanie_typow()

```
int Arkusz::zwracanie_typow (
            int szerokosc )
```

Funkcja sprawdzająca typ komórek w kolumnie.

Parametry

in	*Szerokość*	Szerokość na której sprawdzany jest typ
out	*Typ*	Typ komórek w podanej kolumnie

Definicja w linii 153 pliku tablica.cpp.

```
154 {
155     return typy[szerokosc];
156 }
```

5.1.2.14 zwracanie_wysokosci()

```
int Arkusz::zwracanie_wysokosci ( )
```

Funkcja zwracająca szerokość

Parametry

out	*Szerokość*	Zwracana szerokość

Definicja w linii 148 pliku tablica.cpp.

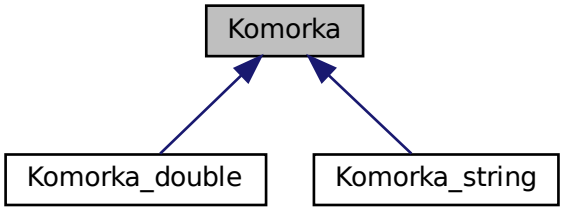
```
149 {
150     return wysokosc;
151 }
```

Dokumentacja dla tej klasy została wygenerowana z plików:

- tablica.h
- tablica.cpp

5.2 Dokumentacja klasy Komorka

Diagram dziedziczenia dla Komorka



Metody publiczne

- virtual std::string zwracanie_wartosci ()=0

  Funkcja odbierająca wartość z komórki.
- virtual void nadawanie_wartosci (std::string)=0

  Funkcja nadająca wartości danej komórce.

5.2.1 Opis szczegółowy

Definicja w linii 5 pliku tablica.h.

5.2.2 Dokumentacja funkcji składowych

5.2.2.1 nadawanie_wartosci()

```
virtual void Komorka::nadawanie_wartosci (
            std::string  )  [pure virtual]
```

Funkcja nadająca wartości danej komórce.

Parametry

in	*Wartość*	Wartość nadawana komórce

Implementowany w Komorka_string i Komorka_double.

5.2.2.2 zwracanie_wartosci()

```
virtual std::string Komorka::zwracanie_wartosci ( )  [pure virtual]
```

Funkcja odbierająca wartość z komórki.

Parametry

out	*Wartość*	Wartość pobierana z komórki

Implementowany w Komorka_string i Komorka_double.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- tablica.h

5.3 Dokumentacja klasy Komorka_double

Diagram dziedziczenia dla Komorka_double

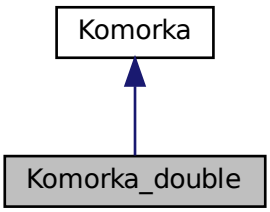
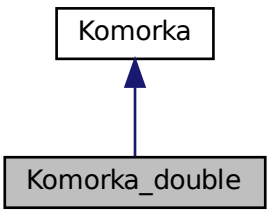


Diagram współpracy dla Komorka_double:



Metody publiczne

- std::string zwracanie_wartosci ()

  Funkcja zwracająca wartość komórki.
- void nadawanie_wartosci (std::string wartosc)

  Funkcja nadająca wartość komórki.

5.3.1 Opis szczegółowy

Definicja w linii 145 pliku tablica.h.

5.3.2 Dokumentacja funkcji składowych

5.3.2.1 nadawanie_wartosci()

```
void Komorka_double::nadawanie_wartosci (
            std::string wartosc )  [virtual]
```

Funkcja nadająca wartość komórki.

Parametry

in	*Wartość*	Wartość nadawana danej komórce

Implementuje Komorka.

Definicja w linii 268 pliku tablica.cpp.

```
269 {
270     try
271     {
272         this->wartosc = std::stod(wartosc);
273     }
274     catch (...)
275     {
276         this->wartosc = 0;
277     }
278 }
```

5.3.2.2 zwracanie_wartosci()

```
std::string Komorka_double::zwracanie_wartosci ( )  [virtual]
```

Funkcja zwracająca wartość komórki.

Parametry

out	*Wartość*	Wartość danej komórki

Implementuje Komorka.

Definicja w linii 263 pliku tablica.cpp.

```
264 {
265     return std::to_string(wartosc);
266 }
```

Dokumentacja dla tej klasy została wygenerowana z plików:

- tablica.h
- tablica.cpp

5.4 Dokumentacja klasy Komorka_string

Diagram dziedziczenia dla Komorka_string

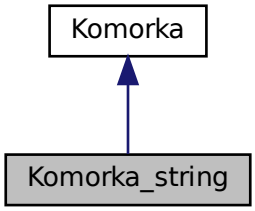
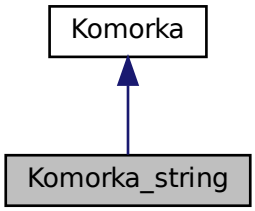


Diagram współpracy dla Komorka_string:



Metody publiczne

- std::string zwracanie_wartosci ()

  Funkcja zwracająca wartość komórki.
- void nadawanie_wartosci (std::string wartosc)

  Funkcja nadająca wartość komórki.

5.4.1 Opis szczegółowy

Definicja w linii 166 pliku tablica.h.

5.4.2 Dokumentacja funkcji składowych

5.4.2.1 nadawanie_wartosci()

```
void Komorka_string::nadawanie_wartosci (
            std::string wartosc )  [virtual]
```

Funkcja nadająca wartość komórki.

Parametry

in	*Wartość*	Wartość nadawana danej komórce

Implementuje Komorka.

Definicja w linii 287 pliku tablica.cpp.

```
288 {
289     this->wartosc = wartosc;
290 }
```

5.4.2.2 zwracanie_wartosci()

```
std::string Komorka_string::zwracanie_wartosci ( )  [virtual]
```

Funkcja zwracająca wartość komórki.

Parametry

out	*Wartość*	Wartość danej komórki

Implementuje Komorka.

Definicja w linii 282 pliku tablica.cpp.

```
283 {
284     return wartosc;
285 }
```

Dokumentacja dla tej klasy została wygenerowana z plików:

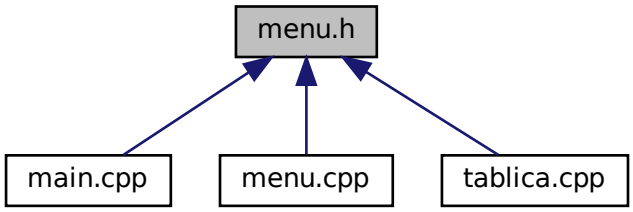
- tablica.h
- tablica.cpp

Chapter 6

Dokumentacja plików

6.1 Dokumentacja pliku menu.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



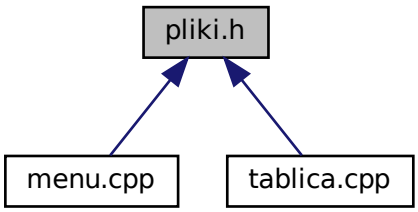
Funkcje

- void menu ()

  Funkcja włączająca menu.

6.2 Dokumentacja pliku pliki.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- void zapis (Arkusz arkusz)

  Funkcja zapisująca arkusz do pliku.
- Arkusz odczyt ()

  *Funkcja wczytująca warto*ci z pliku do arkusza.*

6.2.1 Dokumentacja funkcji

6.2.1.1 odczyt()

```
Arkusz odczyt ( )
```

Funkcja wczytująca warto*ci z pliku do arkusza.

Parametry

out	*Arkusz*	Arkusz nadpisany informacjami z pliku

Definicja w linii 34 pliku pliki.cpp.

```
35 {
36     Arkusz arkusz;
37     int szerokosc, wysokosc;
38     std::string wartosc;
39
40     std::ifstream plik("tablica.txt");
41     plik » szerokosc;
42     plik » wysokosc;
43     arkusz.nadawanie_szerokosci(szerokosc);
44     arkusz.nadawanie_wysokosci(wysokosc);
45
46     arkusz.tworzenie_arkusza(szerokosc, wysokosc);
47
```

```
48      int* typy = new int[arkusz.zwracanie_szerokosci()];
49
50      for (int i = 0; i < arkusz.zwracanie_szerokosci(); i++)
51      {
52          plik » typy[i];
53          if(typy[i]) arkusz.zmiana_kolumny(i, 1);
54      }
55      plik.ignore(1000, '\n');
56
57      for (int i = 0; i < arkusz.zwracanie_wysokosci(); i++)
58      {
59          for (int j = 0; j < arkusz.zwracanie_szerokosci(); j++)
60          {
61              plik » wartosc;
62              arkusz.zwracanie_komorki(j, i)->nadawanie_wartosci(wartosc);
63              plik.clear();
64              plik.ignore(1000, '\t');
65          }
66      }
67
68      arkusz.nadawanie_typow(typy);
69
70      return arkusz;
71  }
```

6.2.1.2 zapis()

```
void zapis (
              Arkusz arkusz )
```

Funkcja zapisująca arkusz do pliku.

Parametry

in	*Wartość*	Wartość nadawana danej komórce

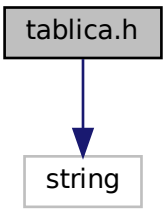
Definicja w linii 5 pliku pliki.cpp.

```
6  {
7      std::ofstream plik("tablica.txt");
8      std::string wartosc;
9      if (plik.good())
10      {
11          plik « arkusz.zwracanie_wysokosci() « std::endl;
12          plik « arkusz.zwracanie_szerokosci() « std::endl;
13
14          for (int i = 0; i < arkusz.zwracanie_szerokosci(); i++)
15          {
16              plik « arkusz.zwracanie_typow(i) « "\t";
17          }
18
19          plik « std::endl;
20
21          for (int i = 0; i < arkusz.zwracanie_wysokosci(); i++)
22          {
23              for (int j = 0; j < arkusz.zwracanie_szerokosci(); j++)
24              {
25                  wartosc = (arkusz.zwracanie_komorki(j, i)->zwracanie_wartosci());
26                  plik « wartosc « "\t";
27              }
28              plik « "\n";
29          }
30          plik.close();
31      }
32  }
```

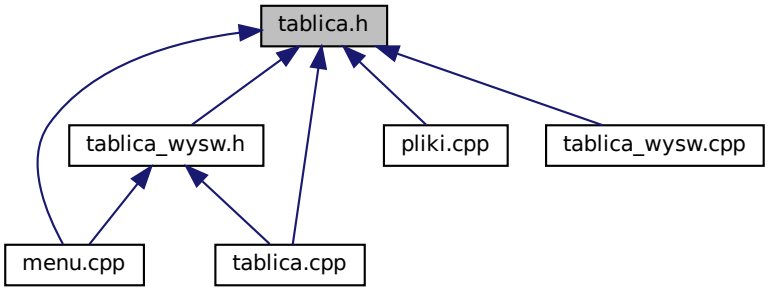
6.3 Dokumentacja pliku tablica.h

```
#include <string>
```

Wykres zależności załączania dla tablica.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



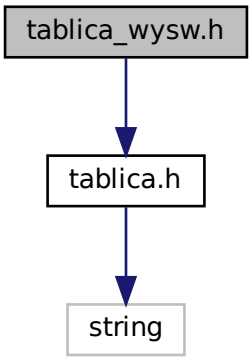
Komponenty

- class Komorka
- class Arkusz
- class Komorka_double
- class Komorka_string

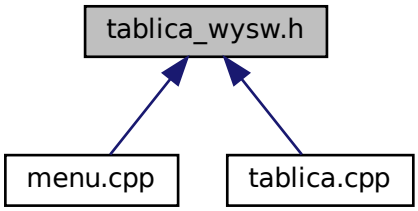
6.4 Dokumentacja pliku tablica_wysw.h

```
#include "tablica.h"
```

Wykres zależności załączania dla tablica_wysw.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- void wstep ()
  Funkcja wyświetlająca możliwe opcje.
- void wyswietlanie_arkusza (Arkusz arkusz)
  Funkcja wyświetlająca arkusz.

6.4.1 Dokumentacja funkcji

6.4.1.1 wyswietlanie_arkusza()

```
void wyswietlanie_arkusza (
            Arkusz arkusz )
```

Funkcja wyświetlająca arkusz.

Parametry

in	*Arkusz*	Arkusz przekazuje informacje

Definicja w linii 21 pliku tablica_wysw.cpp.

```
22 {
23     std::string wartosc;
24     for (int i = 0; i < arkusz.zwracanie_wysokosci(); i++)
25     {
26         for (int j = 0; j < arkusz.zwracanie_szerokosci(); j++)
27         {
28             wartosc = (arkusz.zwracanie_komorki(j, i)->zwracanie_wartosci());
29             std::cout « wartosc « "\t";
30         }
31         std::cout « std::endl « std::endl;
32     }
33 }
```