

Arkusz kalkulacyjny

Generated by Doxygen 1.9.1



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 Arkusz Class Reference	7
4.1.1 Detailed Description	8
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 Arkusz()	8
4.1.3 Member Function Documentation	8
4.1.3.1 convertColumn()	8
4.1.3.2 czyTekstowa()	9
4.1.3.3 getColumnType()	9
4.1.3.4 modyfikacjaWartosci() [1/2]	9
4.1.3.5 modyfikacjaWartosci() [2/2]	10
4.1.3.6 rozmiarX()	11
4.1.3.7 rozmiarY()	11
4.1.3.8 rozszerzArkusz()	12
4.1.3.9 tworzTablica()	13
4.1.3.10 zwrocWartosc()	14
4.2 Cell Class Reference	15
4.2.1 Detailed Description	15
4.2.2 Constructor & Destructor Documentation	15
4.2.2.1 Cell()	16
4.2.3 Member Function Documentation	16
4.2.3.1 getValue()	16
4.2.3.2 isCalcualteable()	16
4.2.3.3 operator+() [1/2]	17
4.2.3.4 operator+() [2/2]	17
4.2.3.5 toString()	18
4.3 Column Class Reference	19
4.3.1 Member Function Documentation	19
4.3.1.1 expandColumn()	19
4.3.1.2 modifyCell() [1/2]	19
4.3.1.3 modifyCell() [2/2]	20
4.4 intCell Class Reference	21
4.4.1 Detailed Description	22
4.4.2 Constructor & Destructor Documentation	22

4.4.2.1 intCell()	22
4.4.3 Member Function Documentation	23
4.4.3.1 getValue()	23
4.4.3.2 setValue()	23
4.4.3.3 toString()	23
4.5 stringCell Class Reference	24
4.5.1 Detailed Description	25
4.5.2 Constructor & Destructor Documentation	25
4.5.2.1 stringCell()	25
4.5.3 Member Function Documentation	26
4.5.3.1 setValue()	26
4.5.3.2 toString()	26
<b>5 File Documentation</b>	<b>29</b>
5.1 ProgramowanieObiektowe/arkusz/kolumna/column.hpp File Reference	29
5.2 ProgramowanieObiektowe/arkusz/komorka/cell.cpp File Reference	30
5.3 ProgramowanieObiektowe/arkusz/komorka/cell.hpp File Reference	30
5.4 ProgramowanieObiektowe/arkusz/komorka/cellConverter.cpp File Reference	31
5.4.1 Function Documentation	32
5.4.1.1 toIntCell()	32
5.4.1.2 toStringCell()	33
5.5 ProgramowanieObiektowe/arkusz/komorka/cellConverter.hpp File Reference	34
5.5.1 Function Documentation	35
5.5.1.1 toIntCell()	35
5.5.1.2 toStringCell()	35
5.6 ProgramowanieObiektowe/arkusz/komorka/cellTypes.hpp File Reference	36
5.7 ProgramowanieObiektowe/arkusz/komorka/intCell.cpp File Reference	37
5.8 ProgramowanieObiektowe/arkusz/komorka/intCell.hpp File Reference	38
5.9 ProgramowanieObiektowe/arkusz/komorka/stringCell.cpp File Reference	39
5.10 ProgramowanieObiektowe/arkusz/komorka/stringCell.hpp File Reference	39
5.11 ProgramowanieObiektowe/arkusz/tablica/tablica.cpp File Reference	40
5.12 ProgramowanieObiektowe/arkusz/tablica/tablica.hpp File Reference	41
5.13 ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.cpp File Reference	43
5.13.1 Function Documentation	43
5.13.1.1 wprowadzWartosc()	43
5.13.1.2 wyswietlTablica()	44
5.14 ProgramowanieObiektowe/error.hpp File Reference	46
5.14.1 Enumeration Type Documentation	46
5.14.1.1 Wyjatki	46
5.15 ProgramowanieObiektowe/io/zapisOdczyt.cpp File Reference	47
5.16 ProgramowanieObiektowe/io/zapisOdczyt.hpp File Reference	47
5.16.1 Function Documentation	48

---

5.16.1.1 wczytajPlik()	49
5.16.1.2 zapisPliku()	49
5.17 ProgramowanieObiektowe/main.cpp File Reference	50
5.18 ProgramowanieObiektowe/menu/menu.cpp File Reference	51
5.18.1 Function Documentation	52
5.18.1.1 generujMenu()	52
5.18.1.2 obslugaBledow()	52
5.18.1.3 obslugaMenu()	53
5.18.1.4 parametry()	53
5.18.1.5 parametryKolumny()	54
5.18.1.6 parametryWiersza()	55
5.18.1.7 rozszerzArkusz()	56
5.18.1.8 tworzArkusz()	56
5.18.1.9 wczytanie()	57
5.18.1.10 zapis()	58
5.18.1.11 zmienKolumne()	59
5.19 ProgramowanieObiektowe/menu/menu.hpp File Reference	59
5.19.1 Function Documentation	61
5.19.1.1 generujMenu()	61
5.19.1.2 obslugaBledow()	61
5.19.1.3 obslugaMenu()	62
5.19.1.4 parametry()	62
5.19.1.5 parametryKolumny()	63
5.19.1.6 parametryWiersza()	64
5.19.1.7 rozszerzArkusz()	65
5.19.1.8 tworzArkusz()	65
5.19.1.9 wczytanie()	66
5.19.1.10 zapis()	67
5.19.1.11 zmienKolumne()	68
<b>Index</b>	<b>69</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

Arkusz . . . . .	7
Cell . . . . .	15
intCell . . . . .	21
stringCell . . . . .	24
Column . . . . .	19





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Arkusz</a>	Struktura <a href="#">Arkusz</a> Struktura <a href="#">Arkusz</a> przechowująca tablicę i jej rozmiar . . . . .	7
<a href="#">Cell</a>	Prototyp klasy komórki . . . . .	15
<a href="#">Column</a>	. . . . .	19
<a href="#">intCell</a>	Klasa komórki wartości całkowitych . . . . .	21
<a href="#">stringCell</a>	Klasa komórki tekstowej . . . . .	24



## Chapter 3

# File Index

### 3.1 File List

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

ProgramowanieObiektowe/ <a href="#">error.hpp</a>	46
ProgramowanieObiektowe/ <a href="#">main.cpp</a>	50
ProgramowanieObiektowe/arkusz/kolumna/ <a href="#">column.hpp</a>	29
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">cell.cpp</a>	30
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">cell.hpp</a>	30
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">cellConverter.cpp</a>	31
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">cellConverter.hpp</a>	34
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">cellTypes.hpp</a>	36
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">intCell.cpp</a>	37
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">intCell.hpp</a>	38
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">stringCell.cpp</a>	39
ProgramowanieObiektowe/arkusz/komorka/ <a href="#">stringCell.hpp</a>	39
ProgramowanieObiektowe/arkusz/tablica/ <a href="#">tablica.cpp</a>	40
ProgramowanieObiektowe/arkusz/tablica/ <a href="#">tablica.hpp</a>	41
ProgramowanieObiektowe/arkusz/tablica/ <a href="#">tablica_wysw.cpp</a>	43
ProgramowanieObiektowe/arkusz/tablica/ <a href="#">tablica_wysw.hpp</a>	??
ProgramowanieObiektowe/io/ <a href="#">zapisOdczyt.cpp</a>	47
ProgramowanieObiektowe/io/ <a href="#">zapisOdczyt.hpp</a>	47
ProgramowanieObiektowe/menu/ <a href="#">menu.cpp</a>	51
ProgramowanieObiektowe/menu/ <a href="#">menu.hpp</a>	59
ProgramowanieObiektowe/operacje/ <a href="#">operacje.hpp</a>	??
ProgramowanieObiektowe/utility/ <a href="#">utility.hpp</a>	??



## Chapter 4

# Class Documentation

### 4.1 Arkusz Class Reference

Struktura [Arkusz](#) Struktura [Arkusz](#) przechowująca tablicę i jej rozmiar.

```
#include <tablica.hpp>
```

#### Public Member Functions

- [Arkusz](#) (size\_t x, size\_t y, bool [czyTekstowa](#)=false)  
*Konstruktor tworzący arkusz z tablicą o wyznaczonym rozmiarze i wybranym typie Konstruktor tworzący arkusz z tablicą o wyznaczonej ilości kolumn i wierszy.*
- [Wyjątki modyfikacjaWartosci](#) (size\_t x, size\_t y, std::string wartosc)  
*Wstawia określoną wartość do komórki tekstowej tablicy w arkuszu.*
- [Wyjątki modyfikacjaWartosci](#) (size\_t x, size\_t y, int wartosc)  
*Wstawia określoną wartość do komórki liczbowej tablicy w arkuszu.*
- [Wyjątki rozszerzArkusz](#) (size\_t x, size\_t y)  
*Modyfikacja rozmiaru tablicy.*
- [Komorka](#) & [zwrocWartosc](#) (size\_t x, size\_t y)  
*Zwraca wybraną komórkę*
- size\_t [rozmiarX](#) ()  
*Getter ilości kolumn.*
- size\_t [rozmiarY](#) ()  
*Getter ilości wierszy.*
- bool [czyTekstowa](#) ()  
*zwraca czy tablica jest typu tekstowego Funkcja zwraca czy tablica jest typu tekstowego*
- void [convertColumn](#) ([cellType](#) type, size\_t column)  
*Umożliwia zmianę typu kolumny.*
- [cellType](#) [getColumnType](#) (size\_t column)  
*Getter typu wybranej kolumny Metoda zwraca typ wybranej kolumny.*
- bool [calculateableColumn](#) (size\_t column)
- bool [hasCalculateableColumn](#) ()
- size\_t [calculateableColumnsCount](#) ()

## Static Public Member Functions

- static [Tablica](#) [tworzTablica](#) (size\_t kolumny, size\_t wiersze, [cellType](#) typ)  
*Tablica która określa typy kolumn.*

### 4.1.1 Detailed Description

Struktura [Arkusz](#) Struktura [Arkusz](#) przechowująca tablicę i jej rozmiar.

### 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 Arkusz()

```
Arkusz::Arkusz (
    size_t x,
    size_t y,
    bool czyTekstowa = false )
```

Konstruktor tworzący arkusz z tablicą o wyznaczonym rozmiarze i wybranym typie Konstruktor tworzący arkusz z tablicą o wyznaczonej ilości kolumn i wierszy.

#### Parameters

in	<i>x</i>	Szerokość tablicy nowego arkusza
in	<i>y</i>	Wysokość tablicy nowego arkusza
in	<i>czyTekstowa</i>	Określa czy tworzony arkusz jest tekstowy czy liczbowy

### 4.1.3 Member Function Documentation

#### 4.1.3.1 convertColumn()

```
void Arkusz::convertColumn (
    cellType type,
    size_t column )
```

Umożliwia zmianę typu kolumny.

#### Parameters

<i>type</i>	typ na którą zmieniamy kolumnę
<i>column</i>	określona kolumna

#### 4.1.3.2 czyTekstowa()

```
bool Arkusz::czyTekstowa ( )
```

zwraca czy tablica jest typu tekstowego Funkcja zwraca czy tablica jest typu tekstowego

##### Returns

true komórki tablicy są typu std::string

false komórki są typu int

#### 4.1.3.3 getColumnType()

```
cellType Arkusz::getColumnType (
    size_t column )
```

Getter typu wybranej kolumny Metoda zwraca typ wybranej kolumny.

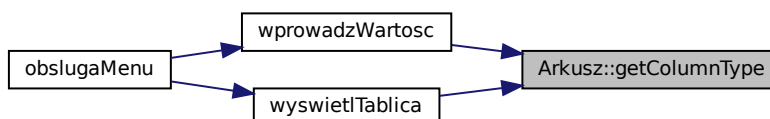
##### Parameters

<i>column</i>	wybrana kolumna
---------------	-----------------

##### Returns

cellType typ komórek znajdujących się w kolumnie

Here is the caller graph for this function:



#### 4.1.3.4 modyfikacjaWartosci() [1/2]

```
Wyjatk Arkusz::modyfikacjaWartosci (
    size_t x,
```

```
size_t y,
int wartosc )
```

Wstawia określoną wartość do komórki liczbowej tablicy w arkuszu.

Funkcja wstawia wartość do komórki o określonym adresie

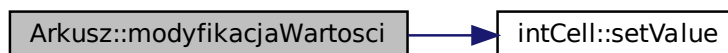
#### Parameters

in	<i>x</i>	określona kolumna
in	<i>y</i>	określony wiersz
in	<i>wart</i>	wprowadzana wartość

#### Returns

Kod błędu, BRAK - Brak błędu, [TABLICA\\_ZAKR](#) - indeks spoza zakresu tablicy

Here is the call graph for this function:



#### 4.1.3.5 modyfikacjaWartosci() [2/2]

```
Wyjatkii Arkusz::modyfikacjaWartosci (
    size_t x,
    size_t y,
    std::string wartosc )
```

Wstawia określoną wartość do komórki tekstowej tablicy w arkuszu.

Funkcja wstawia wartość do komórki o określonym adresie

#### Parameters

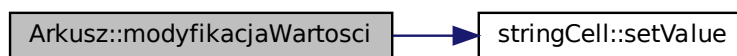
in	<i>x</i>	określona kolumna
in	<i>y</i>	określony wiersz
in	<i>wart</i>	wprowadzana wartość

#### Returns

Kod błędu, BRAK - Brak błędu, [TABLICA\\_ZAKR](#) - indeks spoza zakresu tablicy



Here is the call graph for this function:



#### 4.1.3.6 rozmiarX()

```
size_t Arkusz::rozmiarX ( )
```

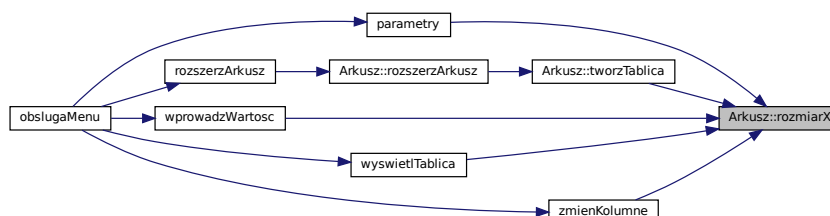
Getter ilości kolumn.

Funkcja zwraca ilość kolumn danego arkusza

##### Returns

Ilość kolumn arkusza

Here is the caller graph for this function:



#### 4.1.3.7 rozmiarY()

```
size_t Arkusz::rozmiarY ( )
```

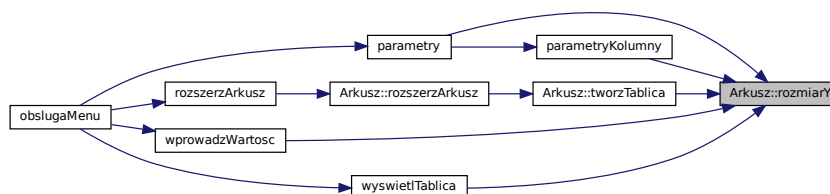
Getter ilości wierszy.

Funkcja zwraca ilość wierszy danego arkusza

**Returns**

Ilość wierszy arkusza

Here is the caller graph for this function:

**4.1.3.8 rozszerzArkusz()**

**Wyjatk** Arkusz::rozszerzArkusz (   
     size\_t x,   
     size\_t y )

Modyfikacja rozmiaru tablicy.

Funkcja modyfikująca rozmiar arkusza, wartości modyfikowanego arkusza zostają zachowane wraz z rozszerzaniem bądź redukcją rozmiaru (przy redukcji możliwa utrata kolumn/wierszy)

**Parameters**

in	<i>nowyX</i>	nowa szerokość tablicy
in	<i>nowyY</i>	nowa wysokość tablicy

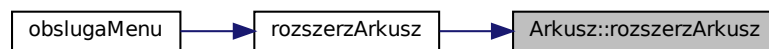
**Returns**

Kod błędu, BRAK - Brak błędu, [TABLICA\\_SIZE](#) - Niepoprawny rozmiar tablicy

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.1.3.9 tworzTablica()

```

Tablica Arkusz::tworzTablica (
    size_t kolumny,
    size_t wiersze,
    cellType typ ) [static]
  
```

Tablica która określa typy kolumn.

Tworzy nową dwuwymiarową tablicę.

Funkcja generująca tablicę pustą tablicę komórek o określonym rozmiarze

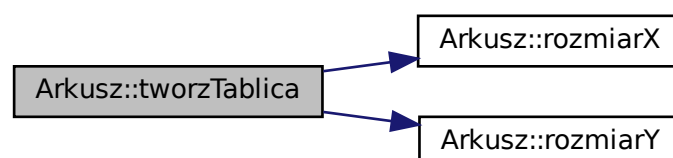
##### Parameters

in	<i>rozmiarX</i>	Szerokość nowej tablicy
in	<i>rozmiarY</i>	Wysokość nowej tablicy

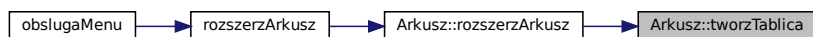
##### Returns

Nową tablicę dwuwymiarową z komórkami liczbowymi o wyznaczonych rozmiarach

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.1.3.10 zwrocWartosc()

```
Komorka & Arkusz::zwrocWartosc (
    size_t x,
    size_t y )
```

Zwraca wybraną komórkę

Funkcja zwraca referencje wybranej komórki

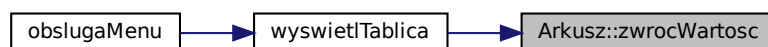
##### Parameters

in	x	Wybrana kolumna
in	y	Wybrany wiersz

##### Returns

Wybrana komórka

Here is the caller graph for this function:



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

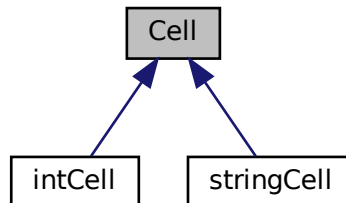
- ProgramowanieObiektowe/arkusz/tablica/[tablica.hpp](#)
- ProgramowanieObiektowe/arkusz/tablica/[tablica.cpp](#)

## 4.2 Cell Class Reference

Prototyp klasy komórki.

```
#include <cell.hpp>
```

Inheritance diagram for Cell:



### Public Member Functions

- [Cell](#) ()  
*Konstruktor domyślny komórki konstruktor tworzy nową "nieobliczalną" komórkę*
- [Cell](#) (bool isCalc)  
*Konstruktor komórki Konstruktor komórki określający czy komórka może być "obliczalna".*
- virtual int [getValue](#) ()  
*Metoda od pobierania wartości matematycznej Zwraca wartość typu Int.*
- virtual std::string [toString](#) ()  
*Metoda od pobierania wartości ciągu znaków Zwraca wartość typu String - ciąg znaków do wykorzystywania przykładowo w wypisywaniu.*
- bool [isCalculable](#) ()  
*Sprawdź czy komórka może posłużyć w obliczeniach Metoda służy do sprawdzania czy typ komórki umożliwia jej branie udziału w funkcjach arytmetycznych.*
- int [operator+](#) ([Cell](#) c1)  
*Zwraca arytmetyczną sumę komórek Zwraca sumę komórek uzyskanych za pomocą wartości zwracanych przez [getValue\(\)](#) obu sumowanych komórek.*
- int [operator+](#) (int wart)  
*Zwraca sumę wartości Int z komórką Zwraca sumę komórki i zmiennej całkowitej.*

### 4.2.1 Detailed Description

Prototyp klasy komórki.

Klasa prototypowa komórki, posiadająca ogólny szkielet tego co komórka powinna zawierać

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 Cell()

```
Cell::Cell (
    bool isCalc ) [inline]
```

Konstruktor komórki Konstruktor komórki określający czy komórka może być "obliczalna".

##### Parameters

in	<i>isCalc</i>	czy komórka jest obliczalna
----	---------------	-----------------------------

### 4.2.3 Member Function Documentation

#### 4.2.3.1 getValue()

```
int Cell::getValue ( ) [virtual]
```

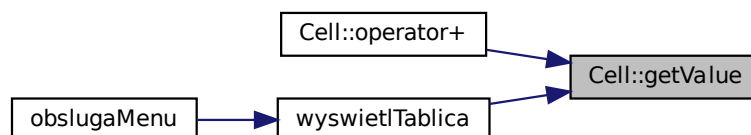
Metoda od pobierania wartości matematycznej Zwraca wartość typu Int.

##### Returns

Zwraca wartość typu int, jeśli komórka jest innego typu niż int zwraca 0.

Reimplemented in [intCell](#).

Here is the caller graph for this function:



#### 4.2.3.2 isCalcualteable()

```
bool Cell::isCalcualteable ( )
```

Sprawdź czy komórka może posłużyć w obliczeniach Metoda służy do sprawdzania czy typ komórki umożliwia jej branie udziału w funkcjach arytmetycznych.

##### Returns

Zwraca wartość logiczną 'true' dla elementów które można obliczać a 'false' dla tych które nie mogą być obliczane.

#### 4.2.3.3 operator+() [1/2]

```
int Cell::operator+ (  
    Cell c1 )
```

Zwraca arytmetyczną sumę komórek Zwraca sumę komórek uzyskanych za pomocą wartości zwracanych przez `getValue()` obu sumowanych komórek.

##### Parameters

in	<i>c1</i>	druga sumowana komórka
----	-----------	------------------------

##### Returns

int Suma wartości komórek

Here is the call graph for this function:



#### 4.2.3.4 operator+() [2/2]

```
int Cell::operator+ (  
    int wart )
```

Zwraca sumę wartości Int z komórką Zwraca sumę komórki i zmiennej całkowitej.

##### Parameters

in	<i>wart</i>	wartość całkowita
----	-------------	-------------------

**Returns**

int Suma wartości komórek

Here is the call graph for this function:

**4.2.3.5 toString()**

```
std::string Cell::toString ( ) [virtual]
```

Metoda od pobierania wartości ciągu znaków Zwraca wartość typu String - ciąg znaków do wykorzystywania przykładowo w wypisywaniu.

**Returns**

Zwraca wartość typu string

Reimplemented in [stringCell](#), and [intCell](#).

Here is the caller graph for this function:



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

- ProgramowanieObiektowe/arkusz/komorka/[cell.hpp](#)
- ProgramowanieObiektowe/arkusz/komorka/[cell.cpp](#)



## 4.3 Column Class Reference

### Public Member Functions

- **Column** (size\_t y, [cellType](#) type)
- [cellType](#) **getType** ()
- [Cell](#) & **getElement** (size\_t y)
- size\_t **getSizeY** ()
- [Wyjatk](#) **modifyCell** (size\_t y, std::string wartosc)  
*Wstawia określoną wartość do komórki tekstowej tablicy w arkuszu.*
- [Wyjatk](#) **modifyCell** (size\_t y, int wartosc)  
*Wstawia określoną wartość do komórki liczbowej tablicy w arkuszu.*
- [Wyjatk](#) **expandColumn** (size\_t newY)  
*Modyfikacja rozmiaru tablicy.*
- [Cell](#) \*\* **createColumnArray** (size\_t newY, [cellType](#) type)

### 4.3.1 Member Function Documentation

#### 4.3.1.1 **expandColumn()**

[Wyjatk](#) Column::expandColumn (   
 size\_t newY )

Modyfikacja rozmiaru tablicy.

Funkcja modyfikująca rozmiar arkusza, wartości modyfikowanego arkusza zostają zachowane wraz z rozszerzaniem bądź redukcją rozmiaru (przy redukcji możliwa utrata kolumn/wierszy)

#### Parameters

in	nowyY	nowa wysokość kolumny
----	-------	-----------------------

#### Returns

Kod błędu, BRAK - Brak błędu, [TABLICA\\_SIZE](#) - Niepoprawny rozmiar tablicy

#### 4.3.1.2 **modifyCell()** [1/2]

[Wyjatk](#) Column::modifyCell (   
 size\_t y,   
 int wartosc )

Wstawia określoną wartość do komórki liczbowej tablicy w arkuszu.

Funkcja wstawia wartość do komórki o określonym adresie

**Parameters**

in	<i>y</i>	określony wiersz
in	<i>wart</i>	wprowadzana wartość

**Returns**

Kod błędu, BRAK - Brak błędu, [TABLICA\\_ZAKR](#) - indeks spoza zakresu tablicy

Here is the call graph for this function:

**4.3.1.3 modifyCell() [2/2]**

**Wyjatk**i Column::modifyCell (   
size\_t y,   
std::string wartosc )

Wstawia określoną wartość do komórki tekstowej tablicy w arkuszu.

Funkcja wstawia wartość do komórki o określonym adresie

**Parameters**

in	<i>y</i>	określony wiersz
in	<i>wart</i>	wprowadzana wartość

## Returns

Kod błędu, BRAK - Brak błędu, [TABLICA\\_ZAKR](#) - indeks spoza zakresu tablicy

Here is the call graph for this function:



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

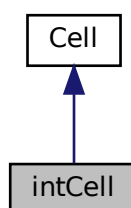
- [ProgramowanieObiektowe/arkusz/kolumna/column.hpp](#)
- [ProgramowanieObiektowe/arkusz/kolumna/column.cpp](#)

## 4.4 intCell Class Reference

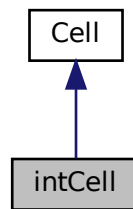
Klasa komórki wartości całkowitych.

```
#include <intCell.hpp>
```

Inheritance diagram for intCell:



Collaboration diagram for intCell:



## Public Member Functions

- `intCell` (int Val)  
*Konstruktor komórki całkowitej Konstruktor komórki z wartością początkową*
- int `getValue` ()  
*Metoda od pobierania wartości matematycznej Zwraca wartość typu Int.*
- std::string `toString` ()  
*Metoda od pobierania wartości ciągu znaków Zwraca wartość typu String - ciąg znaków do wykorzystywania przykładowo w wypisywaniu.*
- void `setValue` (int Val)  
*setter wartości komórki Setter umożliwiający ustawienie wartości całkowitej w komórce*

### 4.4.1 Detailed Description

Klasa komórki wartości całkowitych.

Klasa komórki posiadającej wartość typu Int

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 intCell()

```
intCell::intCell (
    int Val )
```

Konstruktor komórki całkowitej Konstruktor komórki z wartością początkową

#### Parameters

in	Val	Wartość początkowa komórki
----	-----	----------------------------

### 4.4.3 Member Function Documentation

#### 4.4.3.1 getValue()

```
int intCell::getValue ( ) [virtual]
```

Metoda od pobierania wartości matematycznej Zwraca wartość typu Int.

##### Returns

Zwraca wartość typu int, jeśli komórka jest innego typu niż int zwraca 0.

Reimplemented from [Cell](#).

#### 4.4.3.2 setValue()

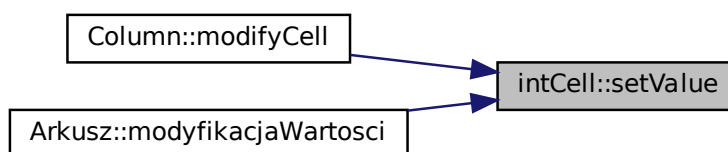
```
void intCell::setValue (
    int Val )
```

setter wartości komórki Setter umożliwiający ustawienie wartości całkowitej w komórce

##### Parameters

in	Val	umożliwia modyfikację wartości komórki
----	-----	--

Here is the caller graph for this function:



#### 4.4.3.3 toString()

```
std::string intCell::toString ( ) [virtual]
```

Metoda od pobierania wartości ciągu znaków Zwraca wartość typu String - ciąg znaków do wykorzystywania przykładowo w wypisywaniu.

**Returns**

Zwraca wartość typu string

Reimplemented from [Cell](#).

Here is the caller graph for this function:



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

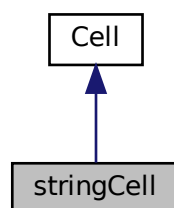
- [ProgramowanieObiektowe/arkusz/komorka/intCell.hpp](#)
- [ProgramowanieObiektowe/arkusz/komorka/intCell.cpp](#)

## 4.5 stringCell Class Reference

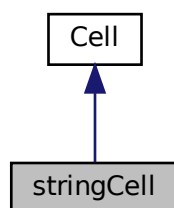
Klasa komórki tekstowej.

```
#include <stringCell.hpp>
```

Inheritance diagram for stringCell:



Collaboration diagram for stringCell:



## Public Member Functions

- [stringCell](#) (std::string Val)  
*Konstruktor komórki tekstowej Konstruktor komórki z wartością początkową*
- std::string [toString](#) ()  
*Metoda od pobierania wartości ciągu znaków Zwraca wartość typu String - ciąg znaków do wykorzystywania przykładowo w wypisywaniu.*
- void [setValue](#) (std::string Val)  
*setter wartości komórki Setter umożliwiający ustawienie wartości tekstowej w komórce*

### 4.5.1 Detailed Description

Klasa komórki tekstowej.

Klasa komórki posiadającej wartość typu string

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 stringCell()

```
stringCell::stringCell (
    std::string Val )
```

Konstruktor komórki tekstowej Konstruktor komórki z wartością początkową

#### Parameters

in	Val	Wartość początkowa komórki
----	-----	----------------------------

### 4.5.3 Member Function Documentation

#### 4.5.3.1 setValue()

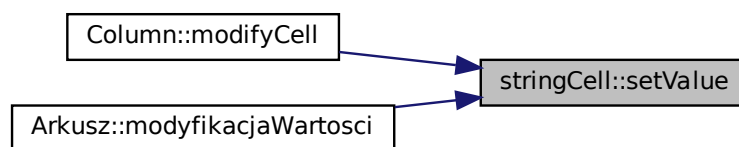
```
void stringCell::setValue (
    std::string Val )
```

setter wartości komórki Setter umożliwiający ustawienie wartości tekstowej w komórce

##### Parameters

in	Val	umożliwia modyfikację wartości komórki
----	-----	--

Here is the caller graph for this function:



#### 4.5.3.2 toString()

```
std::string stringCell::toString ( ) [virtual]
```

Metoda od pobierania wartości ciągu znaków Zwraca wartość typu String - ciąg znaków do wykorzystywania przykładowo w wypisywaniu.

##### Returns

Zwraca wartość typu string

Reimplemented from [Cell](#).

Here is the caller graph for this function:





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

- ProgramowanieObiektowe/arkusz/komorka/[stringCell.hpp](#)
- ProgramowanieObiektowe/arkusz/komorka/[stringCell.cpp](#)

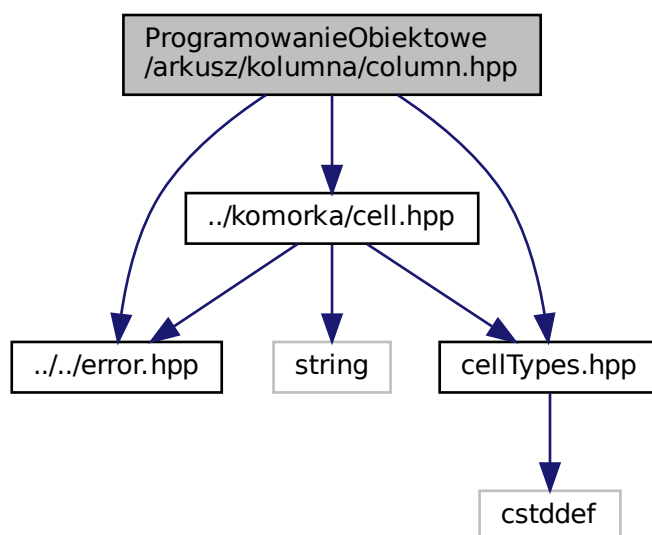


## Chapter 5

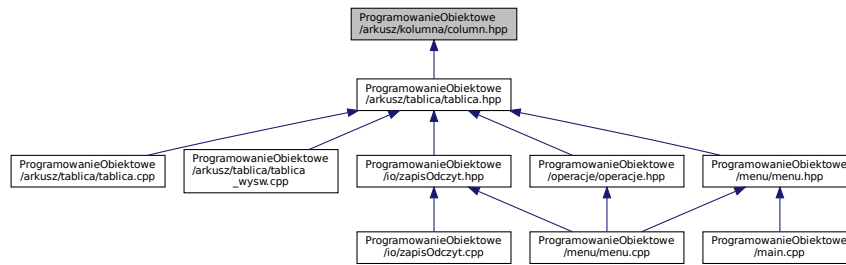
# File Documentation

### 5.1 ProgramowanieObiektowe/arkusz/kolumna/column.hpp File Reference

```
#include "../error.hpp"
#include "../komorka/cell.hpp"
#include "../komorka/cellTypes.hpp"
Include dependency graph for column.hpp:
```



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



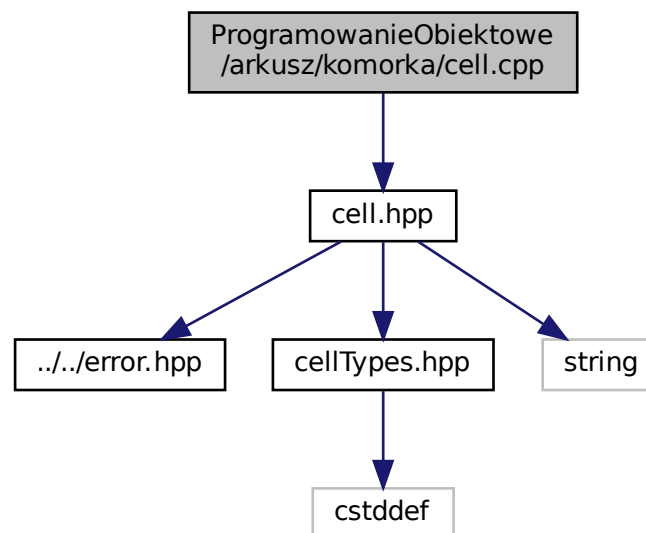
## Classes

- class [Column](#)

## 5.2 ProgramowanieObiektowe/arkusz/komorka/cell.cpp File Reference

```
#include "cell.hpp"
```

Include dependency graph for cell.cpp:



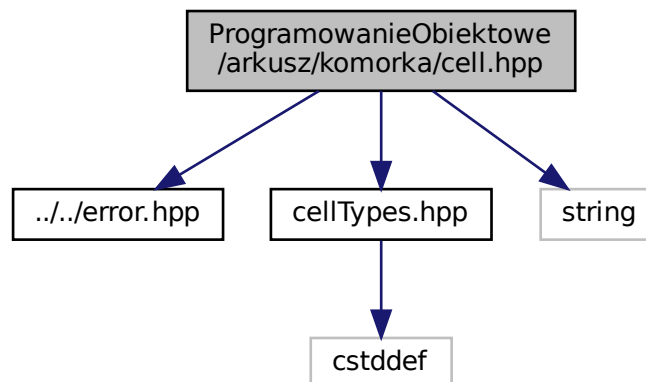
## 5.3 ProgramowanieObiektowe/arkusz/komorka/cell.hpp File Reference

```
#include "../error.hpp"
```

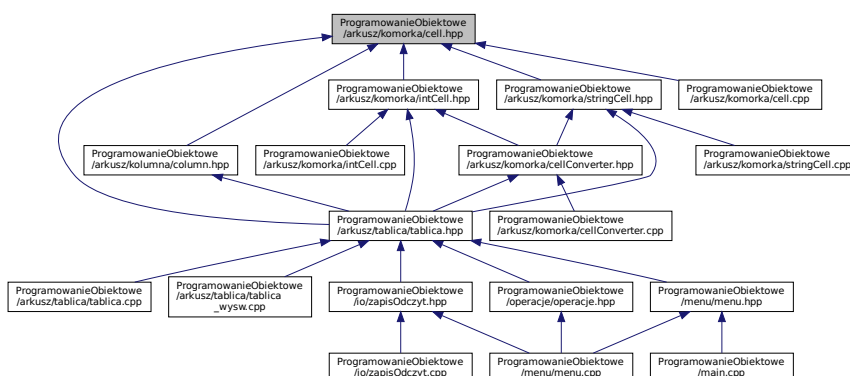
```
#include "cellTypes.hpp"
```

```
#include <string>
```

Include dependency graph for cell.hpp:



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



## Classes

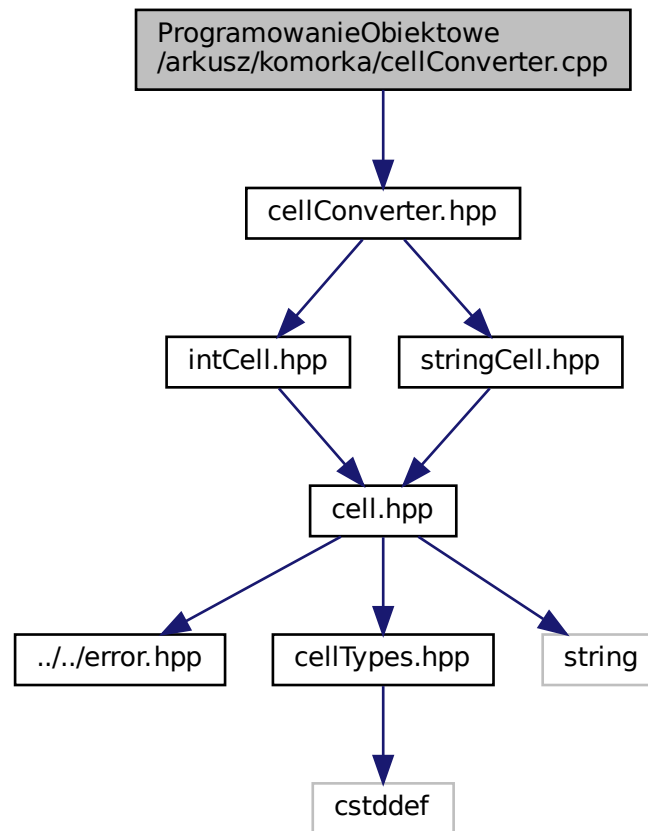
- class [Cell](#)

*Prototyp klasy komórki.*

## 5.4 ProgramowanieObiektowe/arkusz/komorka/cellConverter.cpp File Reference

```
#include "cellConverter.hpp"
```

Include dependency graph for cellConverter.cpp:



## Functions

- `intCell * toIntCell (stringCell convertedCell)`  
*Funkcja konwertująca komórkę string na komórkę int.*
- `stringCell * toStringCell (intCell convertedCell)`  
*Funkcja konwertująca komórkę int na komórkę string.*

### 5.4.1 Function Documentation

#### 5.4.1.1 toIntCell()

```
intCell* toIntCell (  
    stringCell convertedCell )
```

Funkcja konwertująca komórkę string na komórkę int.

## Parameters

<i>convertedCell</i>	komórka do przetworzenia
----------------------	--------------------------

## Returns

intCell\* nowa komórka typu int

Here is the call graph for this function:



### 5.4.1.2 toStringCell()

```
stringCell* toStringCell (  
    intCell convertedCell )
```

Funkcja konwertująca komórkę int na komórkę string.

## Parameters

<i>convertedCell</i>	komórka do przetworzenia
----------------------	--------------------------

## Returns

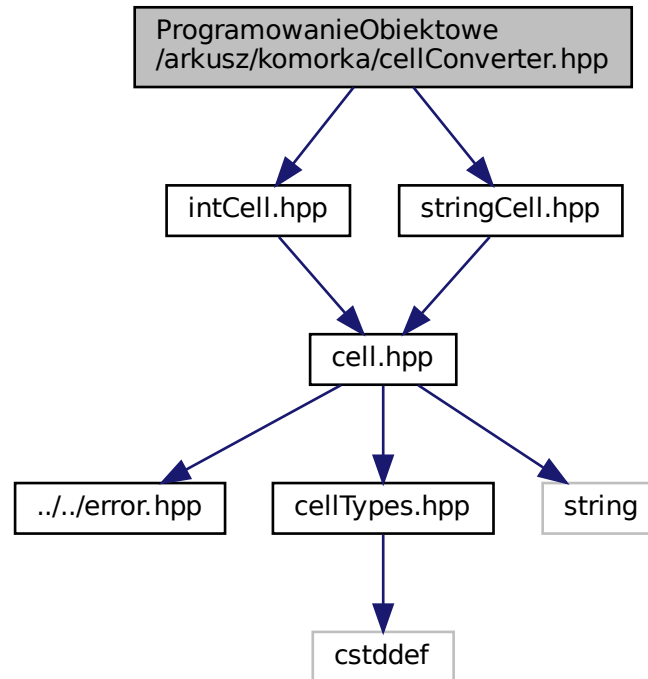
stringCell\* nowa komórka typu string

Here is the call graph for this function:

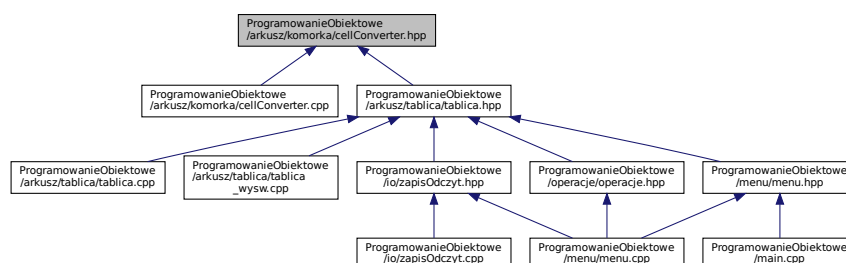


## 5.5 ProgramowanieObiektowe/arkusz/komorka/cellConverter.hpp File Reference

```
#include "intCell.hpp"
#include "stringCell.hpp"
Include dependency graph for cellConverter.hpp:
```



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



## Functions

- `intCell` \* `toIntCell` (`stringCell` convertedCell)



*Funkcja konwertująca komórkę string na komórkę int.*

- `stringCell * toStringCell (intCell convertedCell)`

*Funkcja konwertująca komórkę int na komórkę string.*

## 5.5.1 Function Documentation

### 5.5.1.1 toIntCell()

```
intCell* toIntCell (  
    stringCell convertedCell )
```

Funkcja konwertująca komórkę string na komórkę int.

#### Parameters

<code>convertedCell</code>	komórka do przetworzenia
----------------------------	--------------------------

#### Returns

`intCell*` nowa komórka typu int

Here is the call graph for this function:



### 5.5.1.2 toStringCell()

```
stringCell* toStringCell (  
    intCell convertedCell )
```

Funkcja konwertująca komórkę int na komórkę string.

#### Parameters

<code>convertedCell</code>	komórka do przetworzenia
----------------------------	--------------------------

**Returns**

stringCell\* nowa komórka typu string

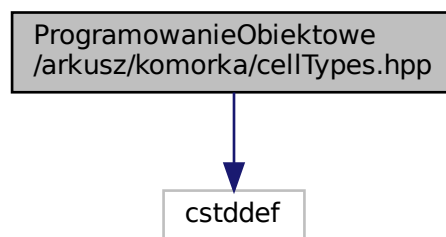
Here is the call graph for this function:



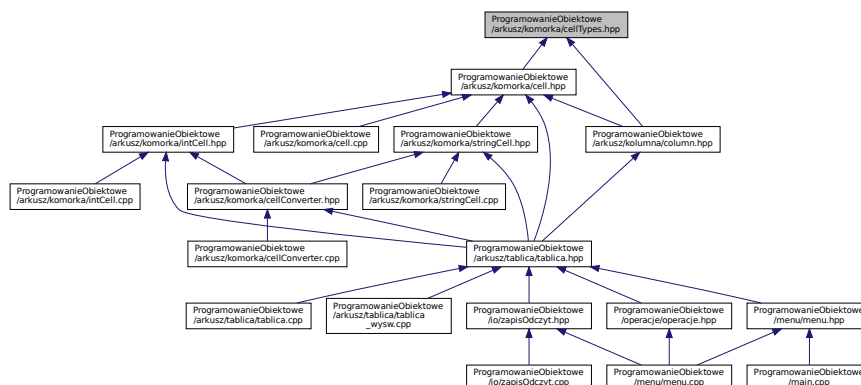
## 5.6 ProgramowanieObiektowe/arkusz/komorka/cellTypes.hpp File Reference

```
#include <cstdint>
```

Include dependency graph for cellTypes.hpp:



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



## Enumerations

- enum class `cellType` { `typeGeneric` = 'G', `typeInt` = 'I', `typeString` = 'S' }

Enum do monitorowania dostępnych typów komórek. Umożliwia bezproblemowe określanie jakiego typu jest kolumna czy komórka.

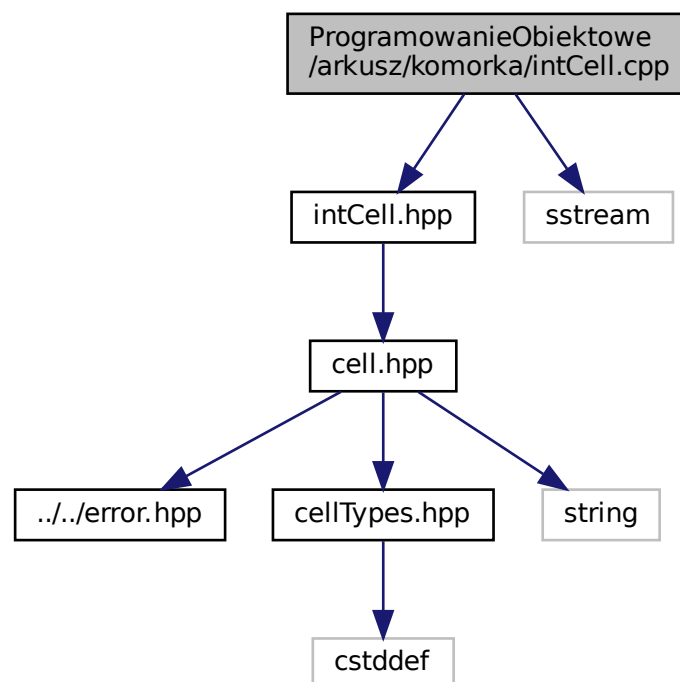
## Variables

- constexpr `cellType` `calculateableTypes` [] = {`cellType::typeInt`}
- constexpr size\_t `calculateableTypesCount` = 1

Dostępne typy komórek obliczeniowych.

## 5.7 ProgramowanieObiektowe/arkusz/komorka/intCell.cpp File Reference

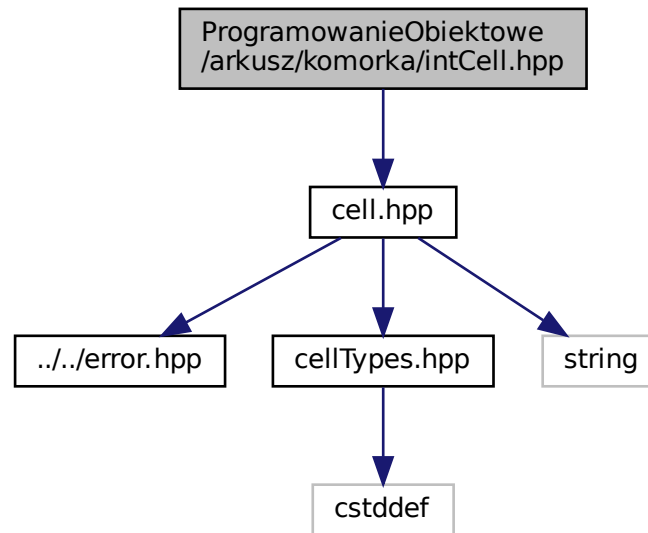
```
#include "intCell.hpp"
#include <sstream>
Include dependency graph for intCell.cpp:
```



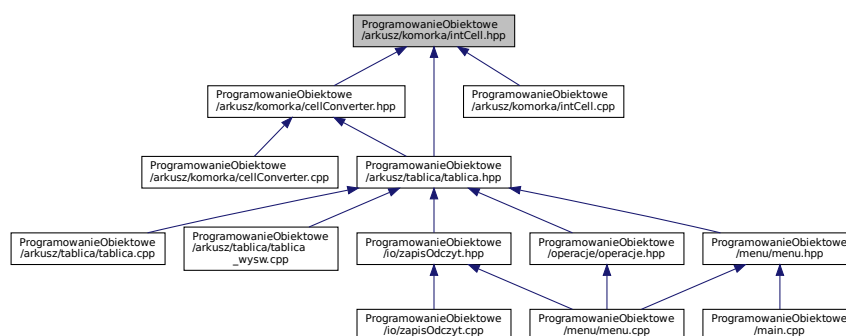
## 5.8 ProgramowanieObiektowe/arkusz/komorka/intCell.hpp File Reference

```
#include "cell.hpp"
```

Include dependency graph for intCell.hpp:



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



## Classes

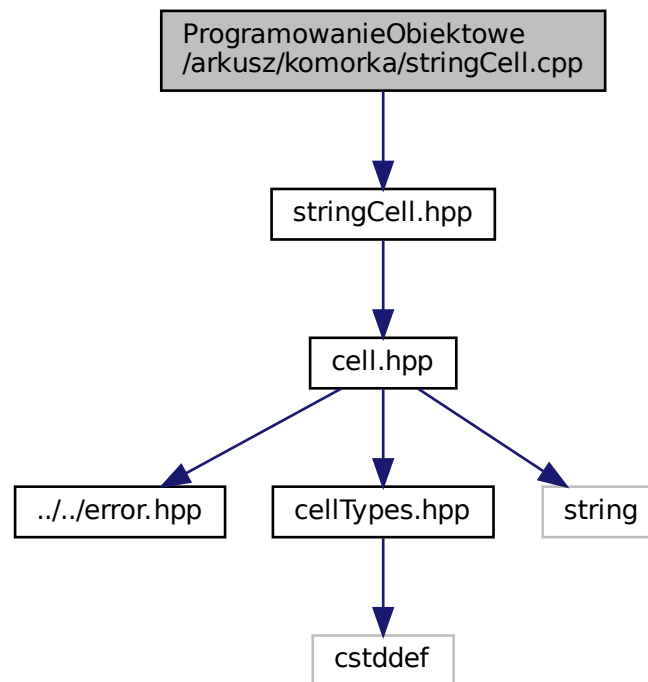
- class `intCell`

*Klasa komórki wartości całkowitych.*

## 5.9 ProgramowanieObiektowe/arkusz/komorka/stringCell.cpp File Reference

```
#include "stringCell.hpp"
```

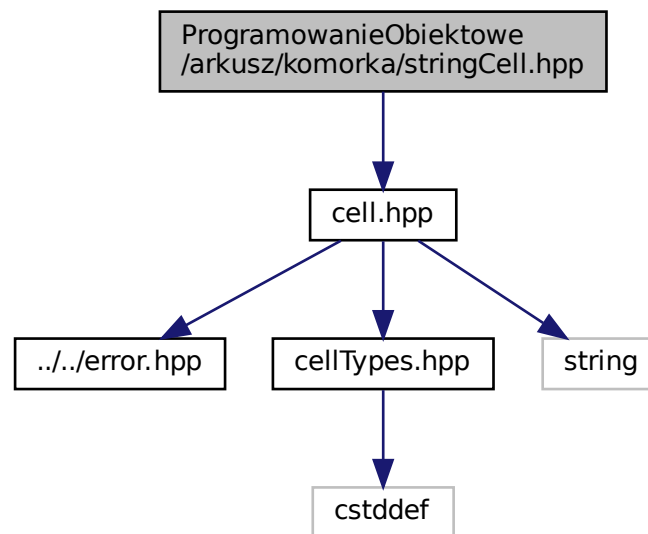
Include dependency graph for stringCell.cpp:



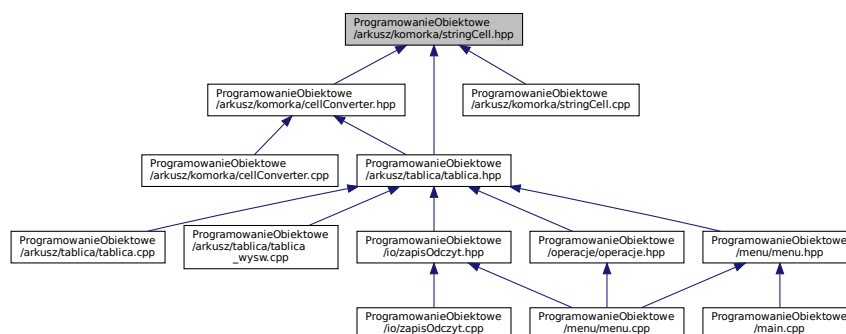
## 5.10 ProgramowanieObiektowe/arkusz/komorka/stringCell.hpp File Reference

```
#include "cell.hpp"
```

Include dependency graph for stringCell.hpp:



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



## Classes

- class `stringCell`

*Klasa komórki tekstowej.*

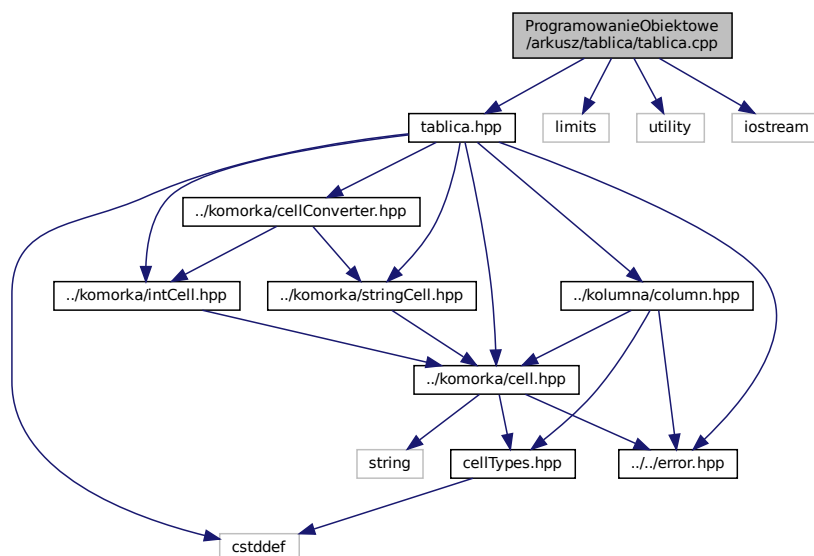
## 5.11 ProgramowanieObiektowe/arkusz/tablica/tablica.cpp File Reference

```
#include "tablica.hpp"
#include <limits>
```

```
#include <utility>
```

```
#include <iostream>
```

Include dependency graph for tablica.cpp:



## 5.12 ProgramowanieObiektowe/arkusz/tablica/tablica.hpp File Reference

```
#include <cstdddef>
```

```
#include "../error.hpp"
```

```
#include "../komorka/cell.hpp"
```

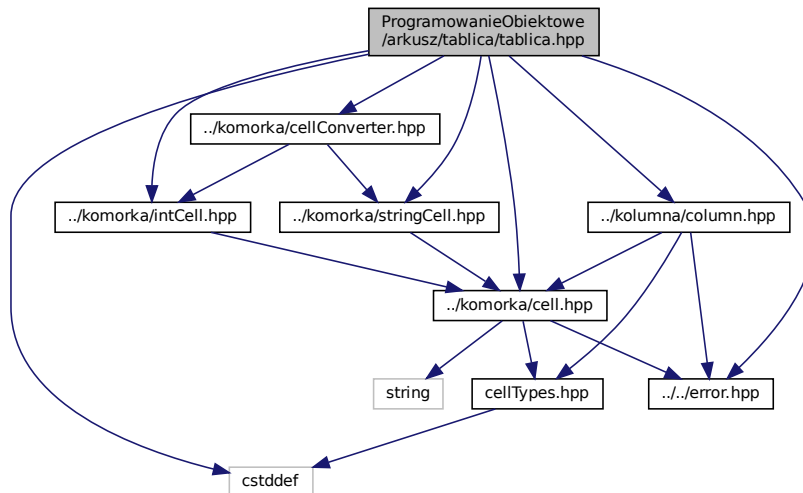
```
#include "../kolumna/column.hpp"
```

```
#include "../komorka/intCell.hpp"
```

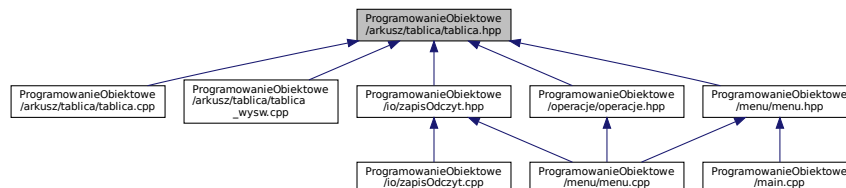
```
#include "../komorka/stringCell.hpp"
```

```
#include "../komorka/cellConverter.hpp"
```

Include dependency graph for `tablica.hpp`:



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



## Classes

- class [Arkusz](#)

Struktura [Arkusz](#) Struktura [Arkusz](#) przechowująca tablicę i jej rozmiar.

## Typedefs

- typedef [Cell](#) [Komorka](#)

Definicja/alias typu `int` jako typ określający komórkę

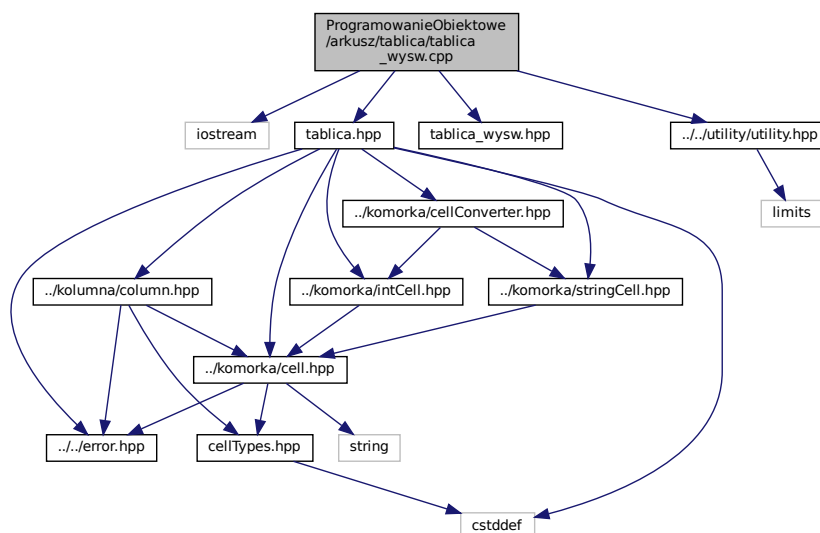
- typedef [Column](#) \*\* [Tablica](#)

Definicja/alias typu `Komorka**` jako `Tablica` komórek.



## 5.13 ProgramowanieObiektowe/arkusz/tablica/tablica\_wysw.cpp File Reference

```
#include <iostream>
#include "tablica.hpp"
#include "tablica_wysw.hpp"
#include "../utility/utility.hpp"
Include dependency graph for tablica_wysw.cpp:
```



### Functions

- void [wyswietlTablica](#) ([Arkusz](#) arkusz)  
*Generuje podgląd tablicy.*
- void [wprowadzWartosc](#) ([Arkusz](#) \*arkusz)  
*Interfejs użytkownika wprowadzania nowej wartości.*

### 5.13.1 Function Documentation

#### 5.13.1.1 wprowadzWartosc()

```
void wprowadzWartosc (
    Arkusz * arkusz )
```

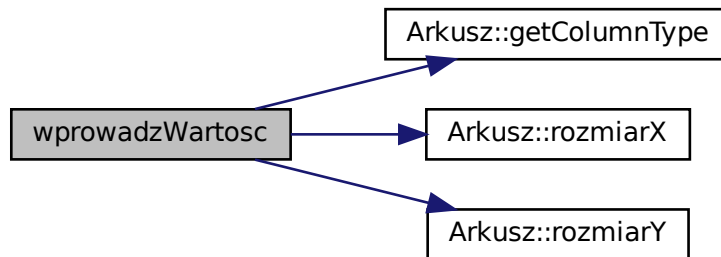
Interfejs użytkownika wprowadzania nowej wartości.

Funkcja interfejsu wprowadzenia wartości do tablicy. Wprowadzana wartość spoza przedziału zostanie zablokowana. W zależności od typu arkusza umożliwia wprowadzenie wartości tekstowej bądź liczbowej.

**Parameters**

in, out	<i>arkusz</i>	arkusz która zostaje zmodyfikowany
---------	---------------	------------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:

**5.13.1.2 wyswietlTablica()**

```
void wyswietlTablica (  
    Arkusz arkusz )
```

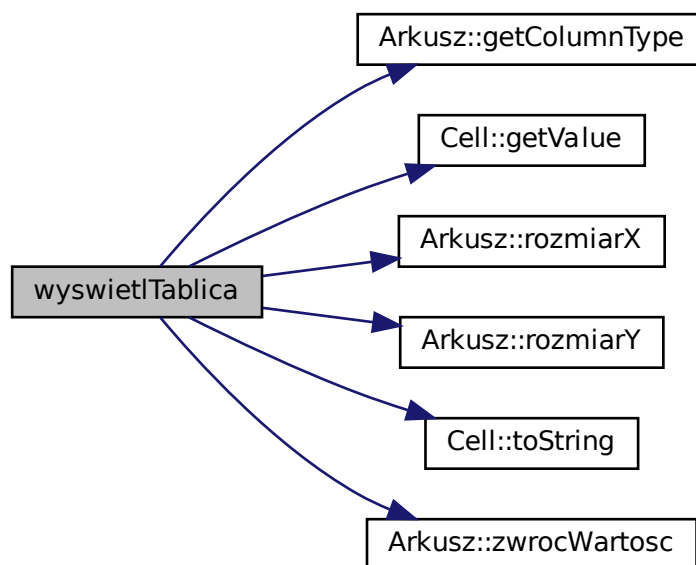
Generuje podgląd tablicy.

Generuje tablicę na konsoli wyświetlając wszystkie zawarte w niej komórki

**Parameters**

in	<i>arkusz</i>	pobiera arkusz celem jego wyświetlenia
----	---------------	--

Here is the call graph for this function:

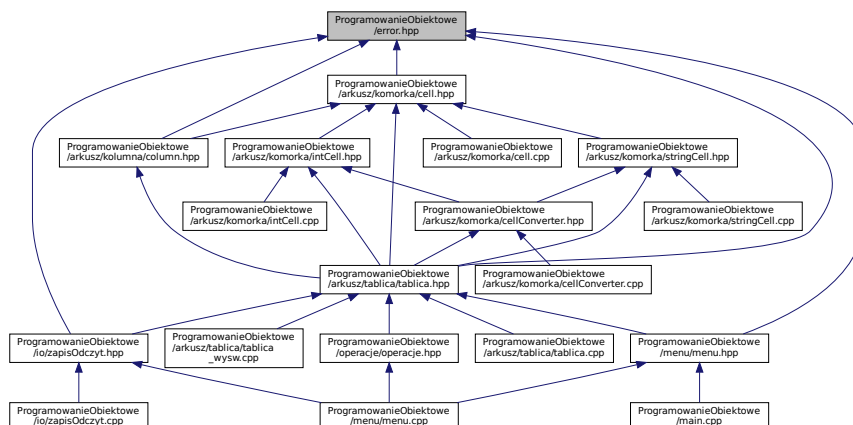


Here is the caller graph for this function:



## 5.14 ProgramowanieObiektowe/error.hpp File Reference

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



### Enumerations

- enum class [Wyjatk](#)i : unsigned int {  
**BRAK** = 0 , **TABLICA\_SIZE** = 1 , **TABLICA\_ZAKR** = 2 , **PLIK\_ACCESS** = 10 ,  
**PLIK\_FORMAT** = 11 , **PLIK\_ROZMIAR** = 12 }

*Wyjątki występujące w programie Typ wyliczeniowy który zawiera wszystkie występujące wyjątki.*

### 5.14.1 Enumeration Type Documentation

#### 5.14.1.1 Wyjątki

```
enum Wyjatki : unsigned int [strong]
```

Wyjątki występujące w programie Typ wyliczeniowy który zawiera wszystkie występujące wyjątki.

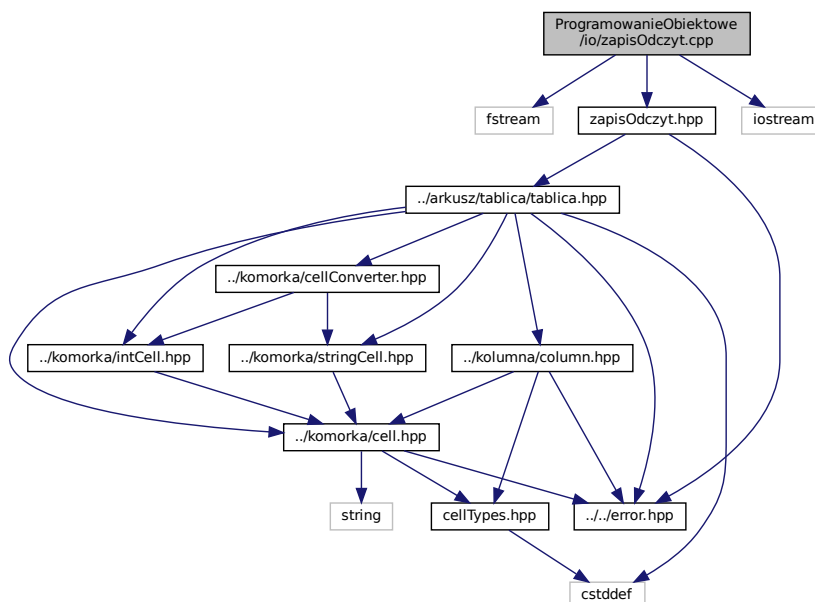
#### Enumerator

TABLICA_SIZE	Brak błędów.
TABLICA_ZAKR	Próba dostępu do elementu poza zakresem tablicy.
PLIK_ACCESS	Próba utworzenia tablicy o niepoprawnym rozmiarze.
PLIK_FORMAT	Niepoprawna nazwa lub brak dostępu do pliku.
PLIK_ROZMIAR	Niepoprawny format wczytywanego pliku.

## 5.15 ProgramowanieObiektowe/io/zapisOdczyt.cpp File Reference

```
#include <fstream>
#include "zapisOdczyt.hpp"
#include <iostream>
```

Include dependency graph for zapisOdczyt.cpp:



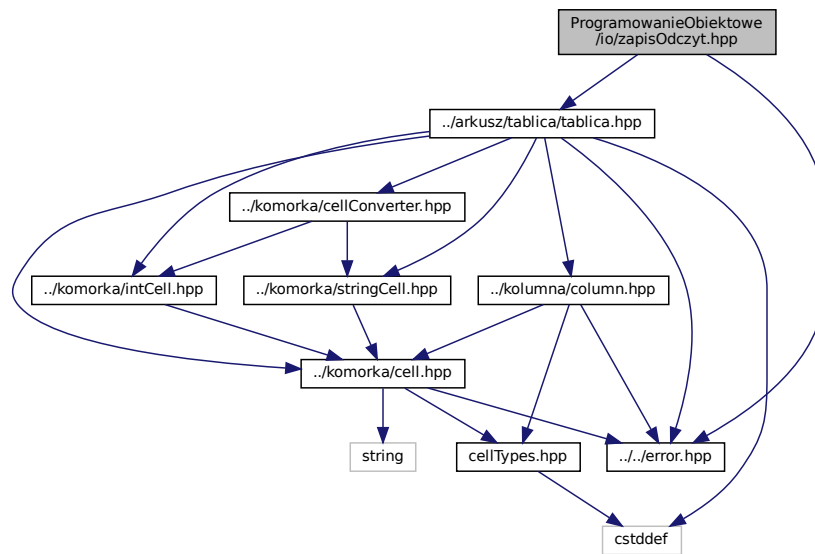
### Functions

- Wyjatk `zapisPliku` (`Arkusz` arkusz, `string` nazwa)
- Wyjatk `wczytajPlik` (`Arkusz` \*arkusz, `string` nazwa)

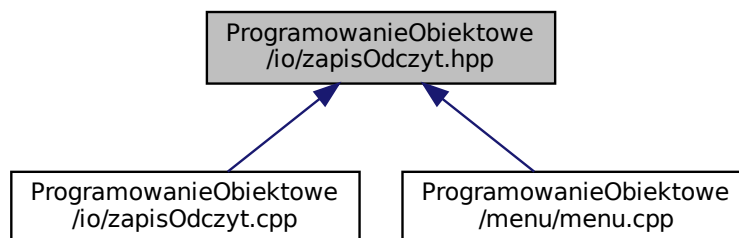
## 5.16 ProgramowanieObiektowe/io/zapisOdczyt.hpp File Reference

```
#include "../arkusz/tablica/tablica.hpp"
#include "../error.hpp"
```

Include dependency graph for zapisOdczyt.hpp:



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



## Functions

- [Wyjątki zapisPliku](#) (Arkusz arkusz, std::string nazwa)  
*Funkcja zapisu do pliku.*
- [Wyjątki wczytajPlik](#) (Arkusz \*arkusz, std::string nazwa)  
*Funkcja wczytywania tablicy z pliku.*

### 5.16.1 Function Documentation

### 5.16.1.1 wczytajPlik()

```
WyjatkI wczytajPlik (
    Arkusz * arkusz,
    std::string nazwa )
```

Funkcja wczytywania tablicy z pliku.

Funkcja wykonuje wczytanie arkusza z wybranego pliku, w przypadku niepowodzenia zwraca wartość różną od 0.

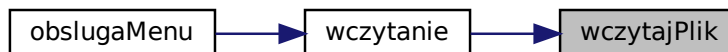
#### Parameters

in, out	arkusz	Arkusz do nadpisanie wczytywaną tablicą
in	nazwa	Nazwa wczytywanego pliku

#### Returns

Zwraca stan funkcji "::BRAK - Funkcja wykonana bez błędnie", "1::PLIK\_ACCESS - Niepoprawna nazwa bądź brak dostępu do pliku", "::PLIK\_FORMAT - Niepoprawny format", "::PLIK\_ROZMIAR - Niepoprawny rozmiar wczytywanej tablicy"

Here is the caller graph for this function:



### 5.16.1.2 zapisPliku()

```
WyjatkI zapisPliku (
    Arkusz arkusz,
    std::string nazwa )
```

Funkcja zapisu do pliku.

Funkcja wykonuje zapis do wybranego przez nas pliku, w przypadku błędu zwraca wartość różną od 0. Początkowo wczytana tablica jest typu tekstowego, dopiero po wczytaniu wszystkich elementów jest sprawdzana możliwość tego czy tablica jest typu liczbowego (preferowana opcja).

#### Parameters

in	arkusz	Arkusz przeznaczony do zapisu
in	nazwa	Nazwa zapisywanego pliku

## Returns

Zwraca stan funkcji "::BRAK - Funkcja wykonana bez błędnie", "::PLIK\_ACCESS - Niepoprawna nazwa bądź brak dostępu do pliku"

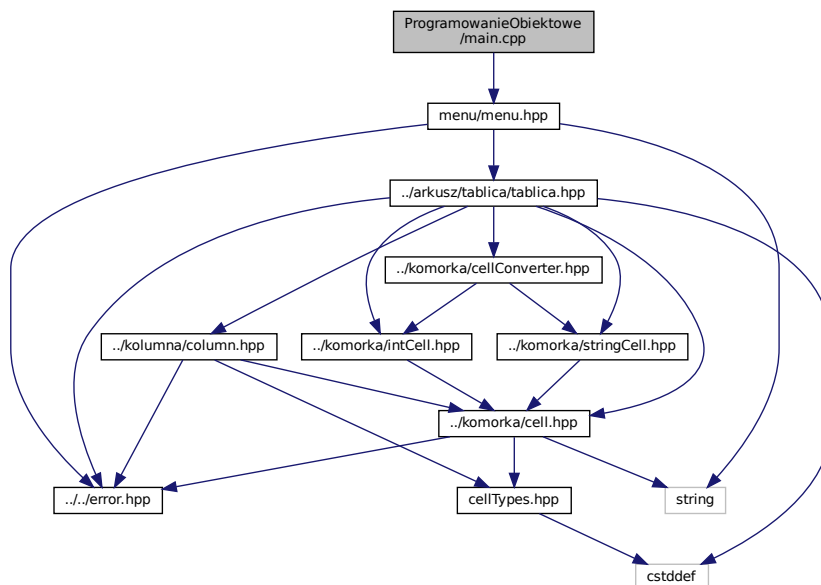
Here is the caller graph for this function:



## 5.17 ProgramowanieObiektowe/main.cpp File Reference

```
#include "menu/menu.hpp"
```

Include dependency graph for main.cpp:



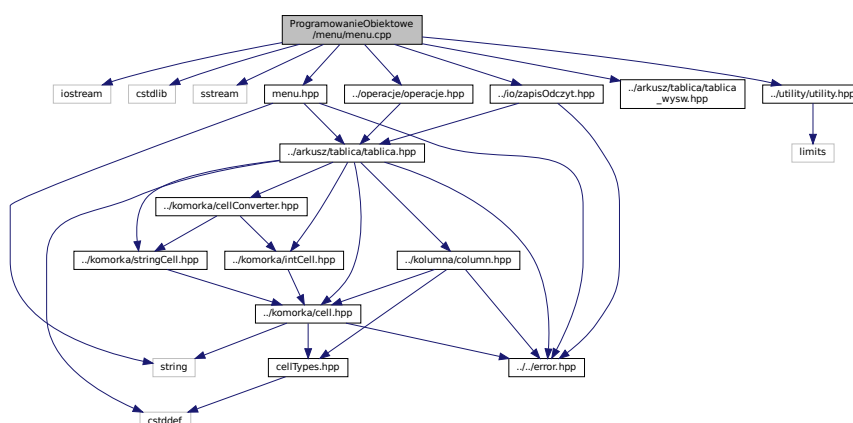
## Functions

- `int main ()`



## 5.18 ProgramowanieObiektowe/menu/menu.cpp File Reference

```
#include <iostream>
#include <cstdlib>
#include <sstream>
#include "menu.hpp"
#include "../io/zapisOdczyt.hpp"
#include "../arkusz/tablica/tablica_wysw.hpp"
#include "../utility/utility.hpp"
#include "../operacje/operacje.hpp"
Include dependency graph for menu.cpp:
```



### Functions

- void [generujMenu](#) ()  
*Funkcja tworząca menu.*
- void [obsługaMenu](#) ()  
*Funkcja kontrolująca działanie programu.*
- void [wczytanie](#) ([Arkusz](#) \*arkusz)  
*Funkcja wczytywania arkusza Funkcja menu od wczytywania arkusza, ma za zadanie opakowanie funkcji IO wczytajPlik.*
- void [zapis](#) ([Arkusz](#) arkusz)  
*Funkcja menu od zapisu.*
- [Arkusz](#) [tworzArkusz](#) ()  
*Funkcja tworząca nową tablicę.*
- void [rozszerzArkusz](#) ([Arkusz](#) \*arkusz)  
*Funkcja modyfikująca rozmiar arkusza.*
- void [parametry](#) ([Arkusz](#) arkusz)  
*Funkcja menu od wyboru względem czego wyznacza parametry.*
- string [parametryWiersza](#) ([Arkusz](#) arkusz, int wiersz)  
*Funkcja od wyznaczania parametrów wiersza arkusza.*
- string [parametryKolumny](#) ([Arkusz](#) arkusz, int kolumna)  
*Funkcja od wyznaczania parametrów kolumny arkusza.*
- void [obsługaBledow](#) ([Wyjatk](#) wyjatek)  
*Funkcja zajmująca się "Przechwytywaniem wyjątków" i ich wypisywaniem Wypisuje napotkany "Wyjątek", Nie wypisze nic jeśli wyjątek ma wartość ::BRAK.*
- void [zmienKolumny](#) ([Arkusz](#) \*arkusz)  
*Funkcja od zmiany typu kolumny w arkuszu.*

## 5.18.1 Function Documentation

### 5.18.1.1 generujMenu()

```
void generujMenu ( )
```

Funkcja tworząca menu.

Funkcja od tworzenia listy dostępnych pozycji menu. Here is the caller graph for this function:



### 5.18.1.2 obslugaBledow()

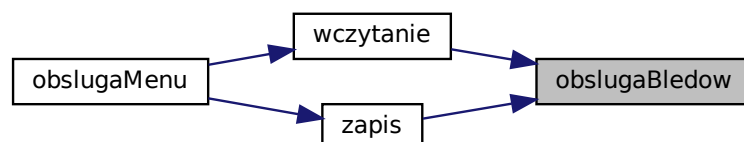
```
void obslugaBledow (
    Wyjatki wyjatek )
```

Funkcja zajmująca się "Przechwytywaniem wyjątków" i ich wypisywaniem Wypisuje napotkany "Wyjątek", Nie wypisze nic jeśli wyjątek ma wartość ::BRAK.

#### Parameters

in	<i>wyjatek</i>	Wyjątek który zostanie przechwycony
----	----------------	-------------------------------------

Here is the caller graph for this function:

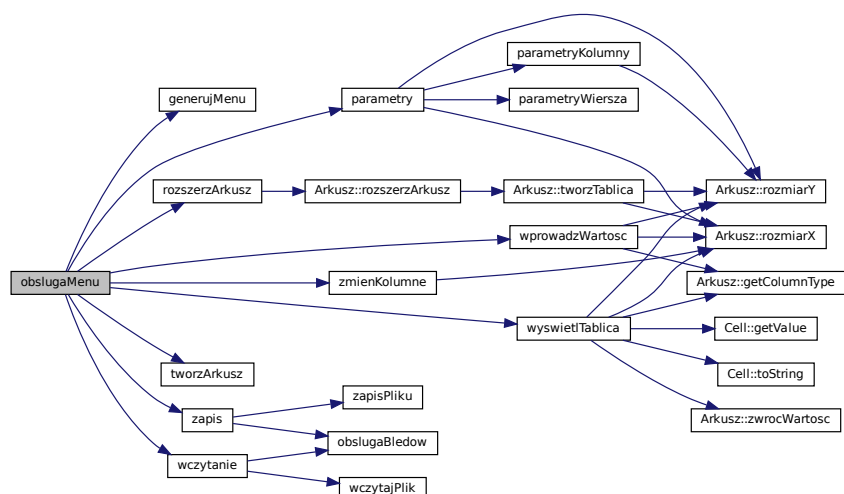


### 5.18.1.3 obslugaMenu()

```
void obslugaMenu ( )
```

Funkcja kontrolująca działanie programu.

Funkcja zajmująca się obsługą menu programu zarządza tym co będzie wywoływane Here is the call graph for this function:



### 5.18.1.4 parametry()

```
void parametry (
    Arkusz arkusz )
```

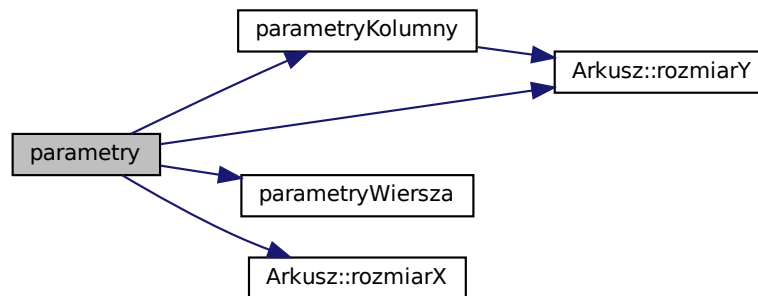
Funkcja menu od wyboru względem czego wyznacza parametry.

Funkcja menu od wyboru atrybutu tablicy (kolumny lub wiersza) która ma za wyświetlenie parametrów wybranego atrybutu. Funkcja nie wykona się dla tablic czysto tekstowych.

#### Parameters

in	arkusz	Przekazywany arkusz do wykorzystania w funkcji wyboru parametrów
----	--------	--

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.18.1.5 parametryKolumny()

```

string parametryKolumny (
    Arkusz arkusz,
    int kolumna )
  
```

Funkcja od wyznaczania parametrów kolumny arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranej kolumny

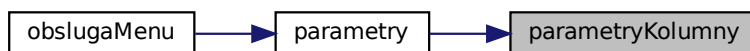
##### Parameters

in	<i>arkusz</i>	Przekazywany arkusz do wykorzystania w funkcji obliczania parametrów
in	<i>kolumna</i>	Kolumna względem której zostaną obliczone parametry

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.18.1.6 parametryWiersza()

```

string parametryWiersza (
    Arkusz arkusz,
    int wiersz )
  
```

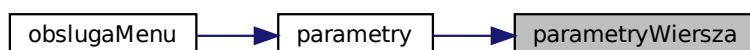
Funkcja od wyznaczania parametrów wiersza arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranego wiersza

##### Parameters

in	<i>arkusz</i>	Przekazywany arkusz do wykorzystania w funkcji obliczania parametrów
in	<i>wiersz</i>	Wiersz względem której zostaną obliczone parametry

Here is the caller graph for this function:



### 5.18.1.7 rozszerzArkusz()

```
void rozszerzArkusz (
    Arkusz * arkusz )
```

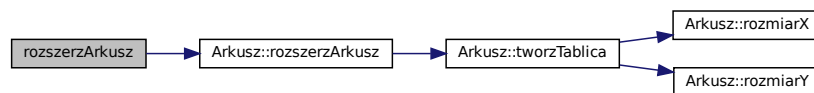
Funkcja modyfikująca rozmiar arkusza.

Interfejs umożliwiający modyfikację rozmiaru istniejącego arkusza.

#### Parameters

in, out	arkusz	Arkusz przeznaczony do modyfikacji rozmiaru
---------	--------	---

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.18.1.8 tworzArkusz()

```
Arkusz tworzArkusz ( )
```

Funkcja tworząca nową tablicę.

Funkcja zawierająca interfejs umożliwiający tworzenie nowego Arkusza z tablicą dwuwymiarową.

### Returns

Nowy [Arkusz](#) do wykorzystywania w programie

Here is the caller graph for this function:



### 5.18.1.9 wczytanie()

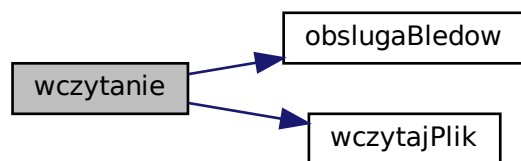
```
void wczytanie (
    Arkusz * arkusz )
```

Funkcja wczytywania arkusza Funkcja menu od wczytywania arkusza, ma za zadanie opakowanie funkcji IO wczytajPlik.

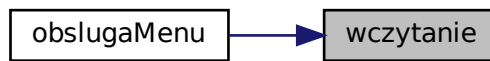
### Parameters

<code>in, out</code>	<code>arkusz</code>	<a href="#">Arkusz</a> do którego mogą być wczytane elementy
----------------------	---------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.18.1.10 zapis()

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

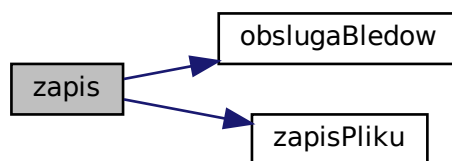
Funkcja menu od zapisu.

Funkcja menu od zapisu która ma za zadanie przetworzenie i opakowanie funkcji IO zapisPliku

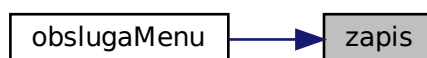
##### Parameters

in	arkusz	Przekazywany arkusz do wykorzystania w funkcji zapisującej do pliku
----	--------	---

Here is the call graph for this function:



Here is the caller graph for this function:





### 5.18.1.11 zmienKolumne()

```
void zmienKolumne (
    Arkusz * arkusz )
```

Funkcja od zmiany typu kolumny w arkuszu.

#### Parameters

in	arkusz	Arkusz w którym jest modyfikowana kolumna
----	--------	---

Here is the call graph for this function:



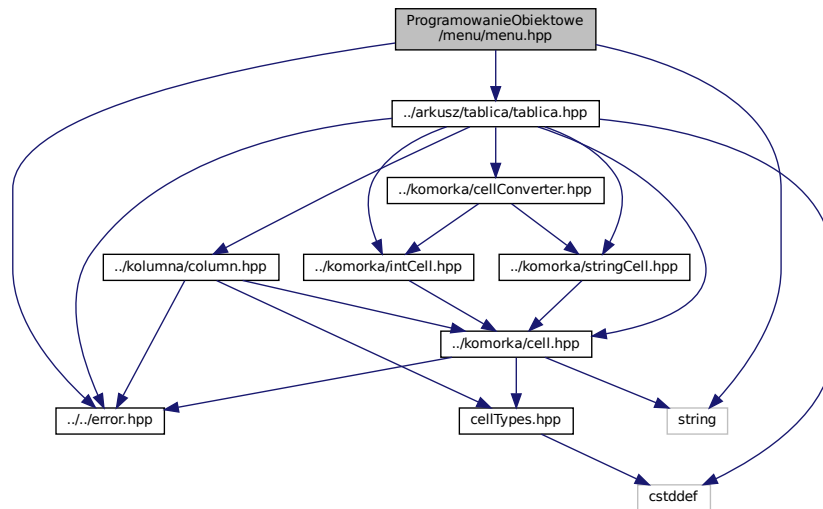
Here is the caller graph for this function:



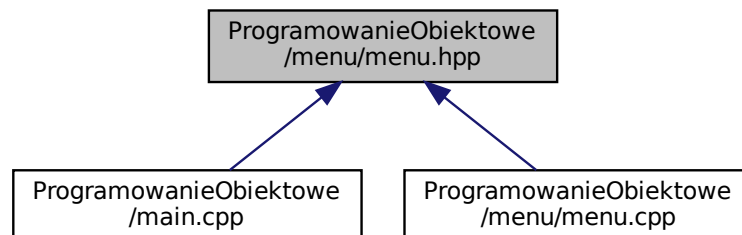
## 5.19 ProgramowanieObiektowe/menu/menu.hpp File Reference

```
#include "../arkusz/tablica/tablica.hpp"
#include <string>
#include "../error.hpp"
```

Include dependency graph for menu.hpp:



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



## Functions

- void `obsługaMenu` ()  
*Funkcja kontrolująca działanie programu.*
- void `generujMenu` ()  
*Funkcja tworząca menu.*
- `Arkusz` `tworzArkusz` ()  
*Funkcja tworząca nową tablicę.*
- void `rozszerzArkusz` (`Arkusz` \*arkusz)  
*Funkcja modyfikująca rozmiar arkusza.*
- void `wczytanie` (`Arkusz` \*arkusz)  
*Funkcja wczytywania arkusza Funkcja menu od wczytywania arkusza, ma za zadanie opakowanie funkcji IO wczytajPlik.*
- void `zapis` (`Arkusz` arkusz)

- Funkcja menu od zapisu.*
- void [parametry](#) ([Arkusz](#) arkusz)
- Funkcja menu od wyboru względem czego wyznacza parametry.*
- std::string [parametryWiersza](#) ([Arkusz](#) arkusz, int wiersz)
- Funkcja od wyznaczania parametrów wiersza arkusza.*
- std::string [parametryKolumny](#) ([Arkusz](#) arkusz, int kolumna)
- Funkcja od wyznaczania parametrów kolumny arkusza.*
- void [obsługaBledow](#) ([Wyjatk](#) wyjatek)
- Funkcja zajmująca się "Przechwytywaniem wyjątków" i ich wypisywaniem Wypisuje napotkany "Wyjątek", Nie wypisze nic jeśli wyjątek ma wartość ::BRAK.*
- void [zmienKolumne](#) ([Arkusz](#) \*arkusz)
- Funkcja od zmiany typu kolumny w arkuszu.*

## 5.19.1 Function Documentation

### 5.19.1.1 generujMenu()

```
void generujMenu ( )
```

Funkcja tworząca menu.

Funkcja od tworzenia listy dostępnych pozycji menu. Here is the caller graph for this function:



### 5.19.1.2 obsługaBledow()

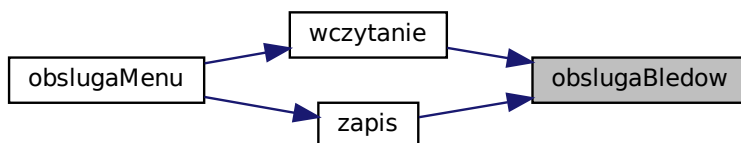
```
void obsługaBledow (
    Wyjatk wyjatek )
```

Funkcja zajmująca się "Przechwytywaniem wyjątków" i ich wypisywaniem Wypisuje napotkany "Wyjątek", Nie wypisze nic jeśli wyjątek ma wartość ::BRAK.

#### Parameters

in	<a href="#">wyjatek</a>	Wyjątek który zostanie przechwycony
----	-------------------------	-------------------------------------

Here is the caller graph for this function:

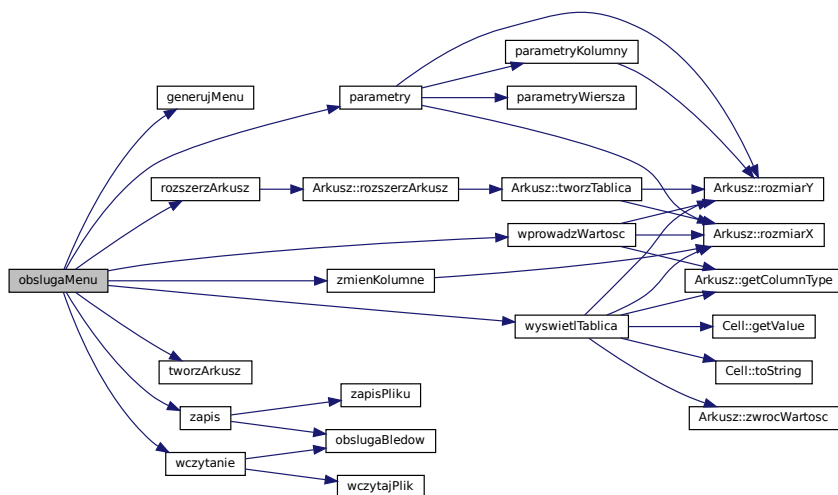


### 5.19.1.3 obsługaMenu()

```
void obsługaMenu ( )
```

Funkcja kontrolująca działanie programu.

Funkcja zajmująca się obsługą menu programu zarządza tym co będzie wywoływane Here is the call graph for this function:



### 5.19.1.4 parametry()

```
void parametry (
    Arkusz arkusz )
```

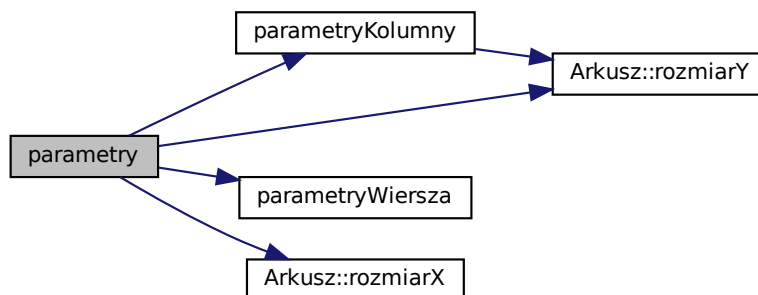
Funkcja menu od wyboru względem czego wyznacza parametry.

Funkcja menu od wyboru atrybutu tablicy (kolumny lub wiersza) która ma za wyświetlenie parametrów wybranego atrybutu. Funkcja nie wykona się dla tablic czysto tekstowych.

## Parameters

in	<i>arkusz</i>	Przekazywany arkusz do wykorzystania w funkcji wyboru parametrów
----	---------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



## 5.19.1.5 parametryKolumny()

```

std::string parametryKolumny (
    Arkusz arkusz,
    int kolumna )

```

Funkcja od wyznaczania parametrów kolumny arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranej kolumny

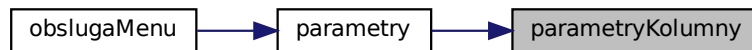
## Parameters

in	<i>arkusz</i>	Przekazywany arkusz do wykorzystania w funkcji obliczania parametrów
in	<i>kolumna</i>	Kolumna względem której zostaną obliczone parametry

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.19.1.6 parametryWiersza()

```
std::string parametryWiersza (
    Arkusz arkusz,
    int wiersz )
```

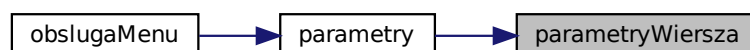
Funkcja od wyznaczania parametrów wiersza arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranego wiersza

##### Parameters

in	<i>arkusz</i>	Przekazywany arkusz do wykorzystania w funkcji obliczania parametrów
in	<i>wiersz</i>	Wiersz względem której zostaną obliczone parametry

Here is the caller graph for this function:



### 5.19.1.7 rozszerzArkusz()

```
void rozszerzArkusz (
    Arkusz * arkusz )
```

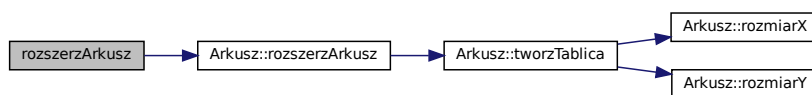
Funkcja modyfikująca rozmiar arkusza.

Interfejs umożliwiający modyfikację rozmiaru istniejącego arkusza.

#### Parameters

in, out	arkusz	Arkusz przeznaczony do modyfikacji rozmiaru
---------	--------	---

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.19.1.8 tworzArkusz()

```
Arkusz tworzArkusz ( )
```

Funkcja tworząca nową tablicę.

Funkcja zawierająca interfejs umożliwiający tworzenie nowego Arkusza z tablicą dwuwymiarową.

### Returns

Nowy [Arkusz](#) do wykorzystywania w programie

Here is the caller graph for this function:



### 5.19.1.9 wczytanie()

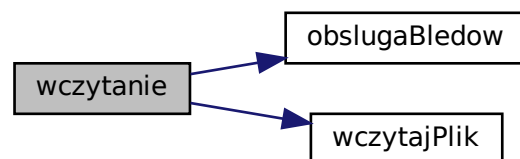
```
void wczytanie (
    Arkusz * arkusz )
```

Funkcja wczytywania arkusza Funkcja menu od wczytywania arkusza, ma za zadanie opakowanie funkcji IO wczytajPlik.

### Parameters

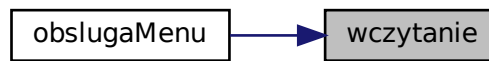
<code>in, out</code>	<code>arkusz</code>	<a href="#">Arkusz</a> do którego mogą być wczytane elementy
----------------------	---------------------	--

Here is the call graph for this function:





Here is the caller graph for this function:



#### 5.19.1.10 zapis()

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

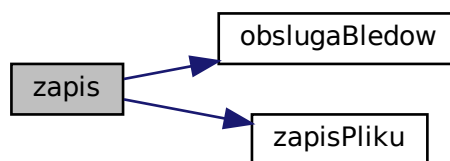
Funkcja menu od zapisu.

Funkcja menu od zapisu która ma za zadanie przetworzenie i opakowanie funkcji IO zapisPliku

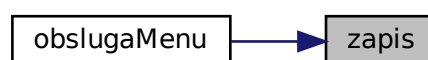
##### Parameters

in	arkusz	Przekazywany arkusz do wykorzystania w funkcji zapisującej do pliku
----	--------	---

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.19.1.11 zmienKolumne()

```
void zmienKolumne (
    Arkusz * arkusz )
```

Funkcja od zmiany typu kolumny w arkuszu.

#### Parameters

in	arkusz	Arkusz w którym jest modyfikowana kolumna
----	--------	---

Here is the call graph for this function:



Here is the caller graph for this function:



# Index

Arkusz, [7](#)  
    Arkusz, [8](#)  
    convertColumn, [8](#)  
    czyTekstowa, [9](#)  
    getColumnType, [9](#)  
    modyfikacjaWartosci, [9](#), [10](#)  
    rozmiarX, [11](#)  
    rozmiarY, [11](#)  
    rozszerzArkusz, [12](#)  
    tworzTablica, [13](#)  
    zwrocWartosc, [14](#)

Cell, [15](#)  
    Cell, [15](#)  
    getValue, [16](#)  
    isCalcualteable, [16](#)  
    operator+, [16](#), [17](#)  
    toString, [18](#)

cellConverter.cpp  
    toIntCell, [32](#)  
    toStringCell, [33](#)

cellConverter.hpp  
    toIntCell, [35](#)  
    toStringCell, [35](#)

Column, [19](#)  
    expandColumn, [19](#)  
    modifyCell, [19](#), [20](#)

convertColumn  
    Arkusz, [8](#)

czyTekstowa  
    Arkusz, [9](#)

error.hpp  
    PLIK\_ACCESS, [46](#)  
    PLIK\_FORMAT, [46](#)  
    PLIK\_ROZMIAR, [46](#)  
    TABLICA\_SIZE, [46](#)  
    TABLICA\_ZAKR, [46](#)  
    Wyjatki, [46](#)

expandColumn  
    Column, [19](#)

generujMenu  
    menu.cpp, [52](#)  
    menu.hpp, [61](#)

getColumnType  
    Arkusz, [9](#)

getValue  
    Cell, [16](#)  
    intCell, [23](#)

intCell, [21](#)  
    getValue, [23](#)  
    intCell, [22](#)  
    setValue, [23](#)  
    toString, [23](#)

isCalcualteable  
    Cell, [16](#)

menu.cpp  
    generujMenu, [52](#)  
    obsługaBledow, [52](#)  
    obsługaMenu, [52](#)  
    parametry, [53](#)  
    parametryKolumny, [54](#)  
    parametryWiersza, [55](#)  
    rozszerzArkusz, [55](#)  
    tworzArkusz, [56](#)  
    wczytanie, [57](#)  
    zapis, [58](#)  
    zmienKolumne, [59](#)

menu.hpp  
    generujMenu, [61](#)  
    obsługaBledow, [61](#)  
    obsługaMenu, [62](#)  
    parametry, [62](#)  
    parametryKolumny, [63](#)  
    parametryWiersza, [64](#)  
    rozszerzArkusz, [64](#)  
    tworzArkusz, [65](#)  
    wczytanie, [66](#)  
    zapis, [67](#)  
    zmienKolumne, [68](#)

modifyCell  
    Column, [19](#), [20](#)

modyfikacjaWartosci  
    Arkusz, [9](#), [10](#)

obsługaBledow  
    menu.cpp, [52](#)  
    menu.hpp, [61](#)

obsługaMenu  
    menu.cpp, [52](#)  
    menu.hpp, [62](#)

operator+  
    Cell, [16](#), [17](#)

parametry  
    menu.cpp, [53](#)  
    menu.hpp, [62](#)

parametryKolumny

- menu.cpp, 54
- menu.hpp, 63
- parametryWiersza
  - menu.cpp, 55
  - menu.hpp, 64
- PLIK\_ACCESS
  - error.hpp, 46
- PLIK\_FORMAT
  - error.hpp, 46
- PLIK\_ROZMIAR
  - error.hpp, 46
- ProgramowanieObiektowe/arkusz/kolumna/column.hpp, 29
- ProgramowanieObiektowe/arkusz/komorka/cell.cpp, 30
- ProgramowanieObiektowe/arkusz/komorka/cell.hpp, 30
- ProgramowanieObiektowe/arkusz/komorka/cellConverter.cpp, 31
- ProgramowanieObiektowe/arkusz/komorka/cellConverter.hpp, 34
- ProgramowanieObiektowe/arkusz/komorka/cellTypes.hpp, 36
- ProgramowanieObiektowe/arkusz/komorka/intCell.cpp, 37
- ProgramowanieObiektowe/arkusz/komorka/intCell.hpp, 38
- ProgramowanieObiektowe/arkusz/komorka/stringCell.cpp, 39
- ProgramowanieObiektowe/arkusz/komorka/stringCell.hpp, 39
- ProgramowanieObiektowe/arkusz/tablica/tablica.cpp, 40
- ProgramowanieObiektowe/arkusz/tablica/tablica.hpp, 41
- ProgramowanieObiektowe/arkusz/tablica/tablica\_wysw.cpp, 43
- ProgramowanieObiektowe/error.hpp, 46
- ProgramowanieObiektowe/io/zapisOdczyt.cpp, 47
- ProgramowanieObiektowe/io/zapisOdczyt.hpp, 47
- ProgramowanieObiektowe/main.cpp, 50
- ProgramowanieObiektowe/menu/menu.cpp, 51
- ProgramowanieObiektowe/menu/menu.hpp, 59
- rozmiarX
  - Arkusz, 11
- rozmiarY
  - Arkusz, 11
- rozszerzArkusz
  - Arkusz, 12
  - menu.cpp, 55
  - menu.hpp, 64
- setValue
  - intCell, 23
  - stringCell, 26
- stringCell, 24
  - setValue, 26
  - stringCell, 25
  - toString, 26
- TABLICA\_SIZE
  - error.hpp, 46
- tablica\_wysw.cpp
  - wprowadzWartosc, 43
  - wyswietlTablica, 44
- TABLICA\_ZAKR
  - error.hpp, 46
- toIntCell
  - cellConverter.cpp, 32
  - cellConverter.hpp, 35
- toString
  - Cell, 18
  - intCell, 23
  - stringCell, 26
- toStringCell
  - cellConverter.cpp, 33
  - cellConverter.hpp, 35
- tworzArkusz
  - menu.cpp, 56
  - menu.hpp, 65
- tworzTablica
  - Arkusz, 13
- wczytajPlik
  - zapisOdczyt.hpp, 48
- wczytanie
  - menu.cpp, 57
  - menu.hpp, 66
- wprowadzWartosc
  - tablica\_wysw.cpp, 43
- Wyjatk
  - error.hpp, 46
- wyswietlTablica
  - tablica\_wysw.cpp, 44
- zapis
  - menu.cpp, 58
  - menu.hpp, 67
- zapisOdczyt.hpp
  - wczytajPlik, 48
  - zapisPliku, 49
- zapisPliku
  - zapisOdczyt.hpp, 49
- zmienKolumny
  - menu.cpp, 59
  - menu.hpp, 68
- zwrocWartosc
  - Arkusz, 14