

Arkusz Kalkulacyjny

Kamil Czop
Projekt Programowanie Obiektowe

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Indeks plików	5
3.1 Lista plików	5
4 Dokumentacja klas	7
4.1 Dokumentacja struktury <code>BadFileException</code>	7
4.1.1 Opis szczegółowy	8
4.2 Dokumentacja klasy <code>Cell</code>	8
4.2.1 Opis szczegółowy	9
4.2.2 Dokumentacja funkcji składowych	10
4.2.2.1 <code>getValue()</code>	10
4.2.2.2 <code>setValue()</code>	10
4.3 Dokumentacja klasy <code>Column</code>	10
4.3.1 Opis szczegółowy	12
4.3.2 Dokumentacja konstruktora i destruktora	12
4.3.2.1 <code>Column()</code>	12
4.3.3 Dokumentacja funkcji składowych	12
4.3.3.1 <code>begin()</code>	12
4.3.3.2 <code>end()</code>	13
4.3.3.3 <code>generateCellArray()</code>	13
4.3.3.4 <code>getCell()</code>	14
4.3.3.5 <code>getHeight()</code>	14
4.3.3.6 <code>getType()</code>	15
4.3.3.7 <code>operator[]()</code>	15
4.3.3.8 <code>resize()</code>	16
4.4 Dokumentacja klasy <code>IntCell</code>	18
4.4.1 Dokumentacja konstruktora i destruktora	19
4.4.1.1 <code>IntCell()</code> [1/2]	19
4.4.1.2 <code>IntCell()</code> [2/2]	19
4.4.2 Dokumentacja funkcji składowych	20
4.4.2.1 <code>getIntValue()</code>	20
4.4.2.2 <code>getValue()</code>	20
4.4.2.3 <code>setValue()</code>	20
4.5 Dokumentacja struktury <code>NotNumericValue</code>	21
4.5.1 Opis szczegółowy	22
4.6 Dokumentacja klasy <code>Sheet</code>	22
4.6.1 Opis szczegółowy	23
4.6.2 Dokumentacja konstruktora i destruktora	23

4.6.2.1 Sheet()	24
4.6.3 Dokumentacja funkcji składowych	24
4.6.3.1 createColumnArray()	24
4.6.3.2 getColumn()	25
4.6.3.3 getHeight()	26
4.6.3.4 getWidth()	27
4.6.3.5 operator[]()	27
4.6.3.6 resize()	28
4.7 Dokumentacja klasy StringCell	29
4.7.1 Dokumentacja konstruktora i destruktor	30
4.7.1.1 StringCell()	30
4.7.2 Dokumentacja funkcji składowych	31
4.7.2.1 getValue()	31
4.7.2.2 setValue()	31
5 Dokumentacja plików	33
5.1 Dokumentacja pliku ProgramowanieObiektowe/arkusz/kolumna/column.cpp	33
5.2 Dokumentacja pliku ProgramowanieObiektowe/arkusz/kolumna/column.hpp	33
5.3 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/adapter.cpp	34
5.4 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/adapter.hpp	35
5.5 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/cell.cpp	36
5.6 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/cell.hpp	36
5.7 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/cellType.hpp	37
5.7.1 Dokumentacja typów wyliczanych	38
5.7.1.1 CellType	38
5.8 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/intCell.cpp	38
5.9 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/intCell.hpp	39
5.10 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/stringCell.cpp	39
5.11 Dokumentacja pliku ProgramowanieObiektowe/arkusz/komorka/stringCell.hpp	40
5.12 Dokumentacja pliku ProgramowanieObiektowe/arkusz/tablica/tablica.cpp	41
5.13 Dokumentacja pliku ProgramowanieObiektowe/arkusz/tablica/tablica.hpp	41
5.14 Dokumentacja pliku ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.cpp	42
5.14.1 Dokumentacja funkcji	43
5.14.1.1 DisplaySheet()	43
5.15 Dokumentacja pliku ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.hpp	44
5.15.1 Dokumentacja funkcji	45
5.15.1.1 DisplaySheet()	45
5.16 Dokumentacja pliku ProgramowanieObiektowe/error.hpp	46
5.16.1 Dokumentacja typów wyliczanych	47
5.16.1.1 Wyjatk	47
5.17 Dokumentacja pliku ProgramowanieObiektowe/io/zapisOdczyt.cpp	48
5.17.1 Dokumentacja funkcji	48

5.17.1.1 loadFile()	48
5.17.1.2 saveFile()	49
5.18 Dokumentacja pliku ProgramowanieObiektowe/io/zapisOdczyt.hpp	50
5.18.1 Dokumentacja funkcji	51
5.18.1.1 loadFile()	51
5.18.1.2 saveFile()	52
5.19 Dokumentacja pliku ProgramowanieObiektowe/main.cpp	53
5.20 Dokumentacja pliku ProgramowanieObiektowe/menu/menu.cpp	53
5.20.1 Dokumentacja funkcji	54
5.20.1.1 generujMenu()	55
5.20.1.2 obslugaMenu()	55
5.20.1.3 parametry()	56
5.20.1.4 parametryKolumny()	57
5.20.1.5 parametryWiersza()	58
5.20.1.6 rozszerzArkusz()	59
5.20.1.7 sortowanie()	60
5.20.1.8 tworzArkusz()	61
5.20.1.9 wczytanie()	61
5.20.1.10 wprowadzWartosc()	62
5.20.1.11 zapis()	63
5.20.1.12 zmienTyp()	64
5.21 Dokumentacja pliku ProgramowanieObiektowe/menu/menu.hpp	65
5.21.1 Dokumentacja funkcji	66
5.21.1.1 generujMenu()	66
5.21.1.2 obslugaMenu()	67
5.21.1.3 parametry()	67
5.21.1.4 parametryKolumny()	68
5.21.1.5 parametryWiersza()	69
5.21.1.6 rozszerzArkusz()	70
5.21.1.7 sortowanie()	71
5.21.1.8 tworzArkusz()	72
5.21.1.9 wczytanie()	72
5.21.1.10 wprowadzWartosc()	73
5.21.1.11 zapis()	74
5.21.1.12 zmienTyp()	75
5.22 Dokumentacja pliku ProgramowanieObiektowe/operacje/operacje.cpp	76
5.22.1 Dokumentacja funkcji	77
5.22.1.1 countCalculateableColumns()	77
5.22.1.2 maxKolumna()	78
5.22.1.3 maxWiersz()	78
5.22.1.4 minKolumna()	79
5.22.1.5 minWiersz()	80

5.22.1.6 sortKolumna()	81
5.22.1.7 sumaKolumna()	82
5.22.1.8 sumaWiersz()	83
5.23 Dokumentacja pliku ProgramowanieObiektowe/operacje/operacje.hpp	83
5.23.1 Dokumentacja funkcji	85
5.23.1.1 countCalculateableColumns()	85
5.23.1.2 maxKolumna()	86
5.23.1.3 maxWiersz()	86
5.23.1.4 minKolumna()	87
5.23.1.5 minWiersz()	88
5.23.1.6 sortKolumna()	89
5.23.1.7 sumaKolumna()	90
5.23.1.8 sumaWiersz()	91
5.24 Dokumentacja pliku ProgramowanieObiektowe/utility/utility.cpp	91
5.24.1 Dokumentacja funkcji	92
5.24.1.1 czyscBufor()	92
5.24.1.2 wprowadzZakres()	93
5.25 Dokumentacja pliku ProgramowanieObiektowe/utility/utility.hpp	94
5.25.1 Dokumentacja funkcji	95
5.25.1.1 czyscBufor()	95
5.25.1.2 wprowadzZakres()	96
Index	97

Chapter 1

Indeks hierarchiczny

1.1 Hierarchia klas

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

Cell	8
IntCell	18
StringCell	29
Column	10
std::exception	
BadFileException	7
NotNumericValue	21
Sheet	22

Chapter 2

Indeks klas

2.1 Lista klas

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

BadFileException	Wyjątek wyrzucany w przypadku braku dostępu. Wyjątek jest wyrzucany gdy użytkownik nie ma dostępu do pliku lub nieistnieje (odczyt)	7
Cell	Interfejs/ klasa czysto wirtualna komórka Interfejs komórka posiadający metody ogólne klas dziedziczących	8
Column	Klasa określająca kolumnę Klasa kolumna obsługująca podstawowe funkcjonalności kolekcji posiada wskaźnik początku i końca zakresu celem wykorzystania w funkcjach z biblioteki algorithm czy pętlach zakresowych	10
IntCell	18
NotNumericValue	Wyjątek wyrzucany w przypadku wprowadzenia wartości nie numerycznej do komórki. Wyjątek powinien być wyrzucany gdy wprowadzony element nie może być przetworzony na wartość liczbowa	21
Sheet	Klasa opisująca Arkusz Klasa Arkusz przechowująca tablicę kolumn i jej rozmiar	22
StringCell	29

Chapter 3

Indeks plików

3.1 Lista plików

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

ProgramowanieObiektowe/error.hpp	46
ProgramowanieObiektowe/main.cpp	53
ProgramowanieObiektowe/arkusz/kolumna/column.cpp	33
ProgramowanieObiektowe/arkusz/kolumna/column.hpp	33
ProgramowanieObiektowe/arkusz/komorka/adapter.cpp	34
ProgramowanieObiektowe/arkusz/komorka/adapter.hpp	35
ProgramowanieObiektowe/arkusz/komorka/cell.cpp	36
ProgramowanieObiektowe/arkusz/komorka/cell.hpp	36
ProgramowanieObiektowe/arkusz/komorka/cellType.hpp	37
ProgramowanieObiektowe/arkusz/komorka/intCell.cpp	38
ProgramowanieObiektowe/arkusz/komorka/intCell.hpp	39
ProgramowanieObiektowe/arkusz/komorka/stringCell.cpp	39
ProgramowanieObiektowe/arkusz/komorka/stringCell.hpp	40
ProgramowanieObiektowe/arkusz/tablica/tablica.cpp	41
ProgramowanieObiektowe/arkusz/tablica/tablica.hpp	41
ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.cpp	42
ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.hpp	44
ProgramowanieObiektowe/io/zapisOdczyt.cpp	48
ProgramowanieObiektowe/io/zapisOdczyt.hpp	50
ProgramowanieObiektowe/menu/menu.cpp	53
ProgramowanieObiektowe/menu/menu.hpp	65
ProgramowanieObiektowe/operacje/operacje.cpp	76
ProgramowanieObiektowe/operacje/operacje.hpp	83
ProgramowanieObiektowe/utility/utility.cpp	91
ProgramowanieObiektowe/utility/utility.hpp	94

Chapter 4

Dokumentacja klas

4.1 Dokumentacja struktury BadFileException

Wyjątek wyrzucany w przypadku braku dostępu. Wyjątek jest wyrzucany gdy użytkownik nie ma dostępu do pliku lub nie istnieje (odczyt).

```
#include <error.hpp>
```

Diagram dziedziczenia dla BadFileException

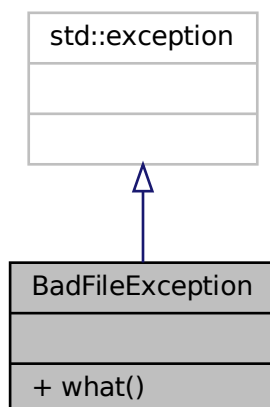
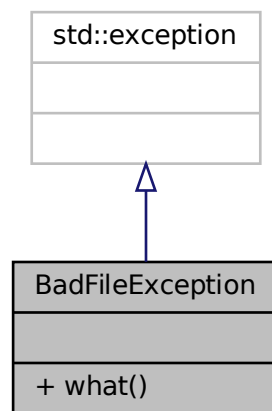


Diagram współpracy dla BadFileException:



Metody publiczne

- `const char * what () const throw ()`

4.1.1 Opis szczegółowy

Wyjątek wyrzucany w przypadku braku dostępu. Wyjątek jest wyrzucany gdy użytkownik nie ma dostępu do pliku lub nie istnieje (odczyt).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- ProgramowanieObiektowe/[error.hpp](#)

4.2 Dokumentacja klasy Cell

Interfejs/ klasa czysto wirtualna komórka Interfejs komórka posiadający metody ogólne klas dziedziczących.

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

Diagram dziedziczenia dla Cell

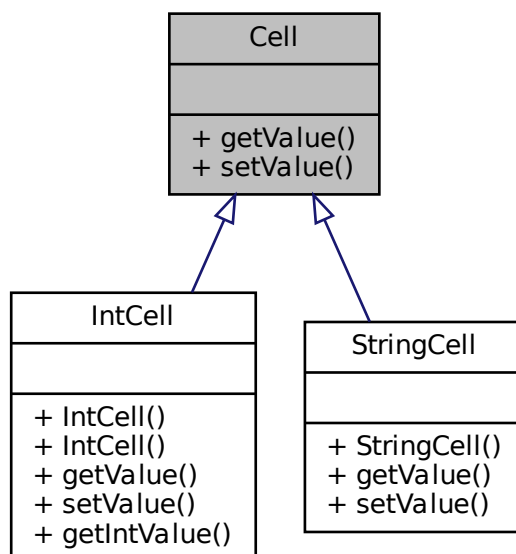
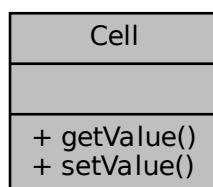


Diagram współpracy dla Cell:



Metody publiczne

- virtual std::string `getValue()`=0
getValue Metoda ogólna od uzyskiwania wartości komórki Metoda zwraca wartość w postaci wartości tekstowej komórki
- virtual void `setValue(std::string value)`=0
setValue Metoda ogólna od ustawiania wartości Ustawia wartość komórki za pomocą wartości string

4.2.1 Opis szczegółowy

Interfejs/ klasa czysto wirtualna komórka Interfejs komórka posiadający metody ogólne klas dziedziczących.

4.2.2 Dokumentacja funkcji składowych

4.2.2.1 getValue()

```
virtual std::string Cell::getValue ( ) [pure virtual]
```

getValue Metoda ogólna od uzyskiwania wartości komórki Metoda zwraca wartość w postaci wartości tekstowej komórki

Zwraca

zawartość tekstowa komórki

Implementowany w [StringCell](#) i [IntCell](#).

4.2.2.2 setValue()

```
virtual void Cell::setValue (
    std::string value ) [pure virtual]
```

setValue Metoda ogólna od ustawiania wartości Ustawia wartość komórki za pomocą wartości string

Parametry

in	value	ustawiana wartość
----	-------	-------------------

Implementowany w [StringCell](#) i [IntCell](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

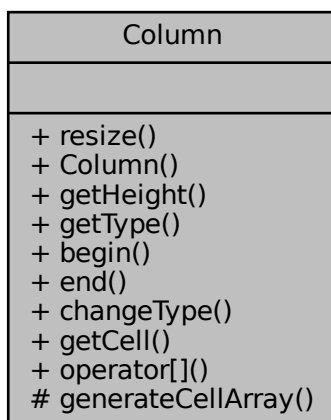
- ProgramowanieObiektowe/arkusz/komorka/[cell.hpp](#)

4.3 Dokumentacja klasy Column

Klasa określająca kolumnę Klasa kolumna obsługująca podstawowe funkcjonalności kolekcji posiada wskaźnik początku i końca zakresu celem wykorzystania w funkcjach z biblioteki algorithm czy pętlach zakresowych.

```
#include <column.hpp>
```

Diagram współpracy dla Column:



Metody publiczne

- void **resize** (size_t newHeight)
resize metoda rozszerzająca kolumnę Metoda zajmuje się rozszerzaniem i zmniejszaniem obecnie przechowywanej tablicy Możliwa utrata danych przy zmienianiu rozmiaru na mniejszy
- **Column** (size_t height, **CellType** type)
Column Konstruktor kolumny o określonym rozmiarze i typie Tworzy nową kolumnę z tablicą o określonym rozmiarze na wskaźniki komórek określonego typu.
- std::size_t **getHeight** ()
Getter wysokości kolumny Zwraca rozmiar tablicy w kolumnie.
- **CellType** **getType** ()
getType Getter typu kolumny Zwraca typ komórek jaką kolumna przechowywuje
- **Cell** ** **begin** ()
begin Zwraca początek tablicy Metoda zwracają wskaźnik na początek tablicy komórek
- **Cell** ** **end** ()
end Zwraca koniec tablicy Metoda zwracająca wskaźnik na koniec tablicy komórek
- void **changeType** (**CellType** newType)
- **Cell** & **getCell** (size_t y)
getCell metoda zwraca referencje do komórki Metoda zwraca referencje do wybranej komórki jeśli jest ona w zakresie kolumny
- **Cell** & **operator[]** (size_t y)
getCell operator zwracający referencje do komórki Operator zwraca referencje do wybranej komórki jeśli jest ona w zakresie kolumny

Statyczne metody chronione

- static **Cell** ** **generateCellArray** (size_t height, **CellType** type)
Typ komórek w kolumnie.

4.3.1 Opis szczegółowy

Klasa określająca kolumnę Klasa kolumna obsługująca podstawowe funkcjonalności kolekcji posiada wskaźnik początku i końca zakresu celem wykorzystania w funkcjach z biblioteki `algorithm` czy pętlach zakresowych.

4.3.2 Dokumentacja konstruktora i destruktora

4.3.2.1 Column()

```
Column::Column (
    size_t height,
    CellType type )
```

Column Konstruktor kolumny o określonym rozmiarze i typie Tworzy nową kolumnę z tablicą o określonym rozmiarze na wskaźniki komórek określonego typu.

Parametry

in	<i>height</i>	Rozmiar tablicy komórek w kolumnie
in	<i>type</i>	Typ tworzonych komórek w kolumnie

Wyjątki

<code>std::bad_array_new_length</code>	w przypadku zerowego rozmiaru
--	-------------------------------

Oto graf wywołań dla tej funkcji:



4.3.3 Dokumentacja funkcji składowych

4.3.3.1 begin()

```
Cell ** Column::begin ( )
```

begin Zwraca początek tablicy Metoda zwracająca wskaźnik na początek tablicy komórek

Zwraca

początek wewnętrznej tablicy

4.3.3.2 end()

```
Cell ** Column::end ( )
```

end Zwraca koniec tablicy Metoda zwracająca wskaźnik na koniec tablicy komórek

Zwraca

koniec tablicy komórek

4.3.3.3 generateCellArray()

```
Cell ** Column::generateCellArray (
    size_t height,
    CellType type ) [static], [protected]
```

Typ komórek w kolumnie.

generateCellArray metoda tworząca nową tablicę komórek Statyczna metoda zajmująca się tworzeniem jednowymiarowej tablicy komórek określonego typu

Parametry

in	<i>height</i>	Wysokość nowej tablicy
in	<i>type</i>	Typ tworzonych komórek

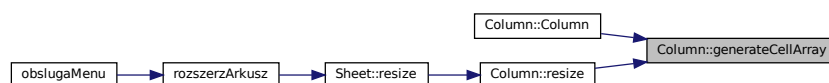
Wyjątki

<code>std::bad_array_new_length</code>	w przypadku zerowego rozmiaru
--	-------------------------------

Zwraca

Tablica jednowymiarowa wskaźników na komórki określonego typu

Oto graf wywoływań tej funkcji:



4.3.3.4 getCell()

```
Cell & Column::getCell (
    size_t y )
```

getCell metoda zwraca referencje do komórki Metoda zwraca referencje do wybranej komórki jeśli jest ona w zakresie kolumny

Parametry

in	y	współzędna do komórki w tablicy
----	---	---------------------------------

Wyjątki

<i>std::out_of_range</i>	Gdy y jest poza zakresem kolumny (y > height)
--------------------------	---

Zwraca

referencja komórki z tablicy

Oto graf wywoływań tej funkcji:



4.3.3.5 getHeight()

```
size_t Column::getHeight ( )
```

Getter wysokości kolumny Zwraca rozmiar tablicy w kolumnie.

Zwraca

Rozmiar kolumny

4.3.3.6 getType()

```
CellType Column::getType ( )
```

getType Getter typu kolumny Zwraca typ komórek jaką kolumna przechowuje

Zwraca

Typ komórek w kolumnie

4.3.3.7 operator[]()

```
Cell & Column::operator[] (
    size_t y )
```

getCell operator zwracający referencje do komórki Operator zwraca referencje do wybranej komórki jeśli jest ona w zakresie kolumny

Parametry

in	y	współzędna do komórki w tablicy
----	---	---------------------------------

Wyjątki

<i>std::out_of_range</i>	Gdy y jest poza zakresem kolumny ($y > \text{height}$)
--------------------------	--

Zwraca

referencja komórki z tablicy

Oto graf wywołań dla tej funkcji:



4.3.3.8 **resize()**

```
void Column::resize (
    size_t newHeight )
```

resize metoda rozszerzająca kolumnę Metoda zajmuje się rozszerzaniem i zmniejszaniem obecnie przechowywanej tablicy Możliwa utrata danych przy zmienianiu rozmiaru na mniejszy

Wyjątki

<code>std::bad_array_new_length</code>	w przypadku zerowego rozmiaru
--	-------------------------------

Parametry

in	<code>newHeight</code>	Nowy rozmiar kolumny
----	------------------------	----------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.4 Dokumentacja klasy IntCell

Diagram dziedziczenia dla IntCell

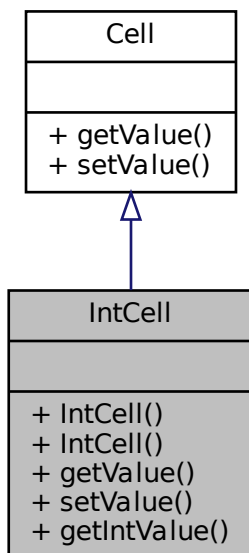
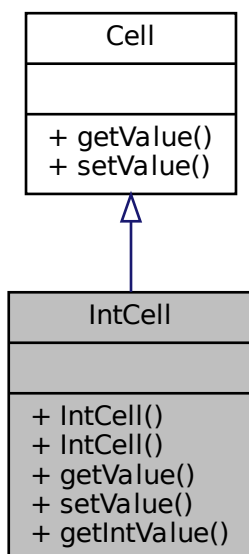


Diagram współpracy dla IntCell:



Metody publiczne

- `IntCell` (int value=0)
IntCell Konstruktor domyślny z opcjonalnym parametrem Konstruktor umożliwiający określenie początkowej wartości komórki.
- `IntCell` (std::string value)
IntCell Konstruktor z parametrem string Konstruktor umożliwiający określenie początkowej wartości komórki Wywołuje funkcję `setValue(std::string value)`
- std::string `getValue` ()
getValue Metoda ogólna od uzyskiwania wartości komórki Metoda zwraca wartość w postaci wartości tekstowej komórki
- void `setValue` (std::string value)
setValue Metoda ogólna od ustawiania wartości Ustawia wartość komórki za pomocą wartości string Wartość ta jest później parsowana W przypadku braku możliwości jej ustawienia wyrzucany jest wyjątek
- int `getIntValue` ()
getIntValue Metoda klasy umożliwiająca bezpośrednie uzyskanie wartości int Metoda umożliwia uzyskanie wartości bezpośrednie bez potrzeby parsowania wartości

4.4.1 Dokumentacja konstruktora i destruktor

4.4.1.1 IntCell() [1/2]

```
IntCell::IntCell (
    int value = 0 )
```

IntCell Konstruktor domyślny z opcjonalnym parametrem Konstruktor umożliwiający określenie początkowej wartości komórki.

Parametry

in	value	Wartość początkowa komórki
----	-------	----------------------------

4.4.1.2 IntCell() [2/2]

```
IntCell::IntCell (
    std::string value )
```

IntCell Konstruktor z parametrem string Konstruktor umożliwiający określenie początkowej wartości komórki Wywołuje funkcję `setValue(std::string value)`

Parametry

in	value	Wartość początkowa komórki
----	-------	----------------------------

4.4.2 Dokumentacja funkcji składowych

4.4.2.1 getIntValue()

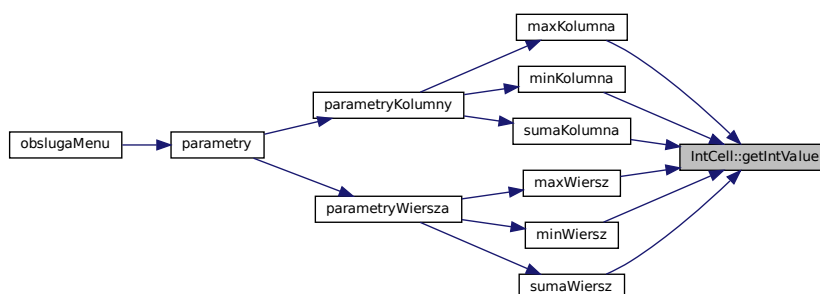
```
int IntCell::getIntValue ( )
```

getIntValue Metoda klasy umożliwiająca bezpośrednie uzyskanie wartości int Metoda umożliwia uzyskanie wartości bezpośrednie bez potrzeby parsowania wartości

Zwraca

wartość komórki

Oto graf wywołań tej funkcji:



4.4.2.2 getValue()

```
std::string IntCell::getValue ( ) [virtual]
```

getValue Metoda ogólna od uzyskiwania wartości komórki Metoda zwraca wartość w postaci wartości tekstowej komórki

Zwraca

zawartość tekstowa komórki

Implementuje [Cell](#).

4.4.2.3 setValue()

```
void IntCell::setValue (
    std::string value ) [virtual]
```

setValue Metoda ogólna od ustawiania wartości Ustawia wartość komórki za pomocą wartości string Wartość ta jest później parsowana W przypadku braku możliwości jej ustawienia wyrzucany jest wyjątek

Wyjątki

NotNumericValue	Brak możliwości przetworzenia wartości tekstowej na liczbową
---------------------------------	--

Parametry

in	value	ustawiana wartość
----	-------	-------------------

Implementuje [Cell](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.5 Dokumentacja struktury NotNumericValue

Wyjątek wyrzucany w przypadku wprowadzenia wartości nie numerycznej do komórki. Wyjątek powinien być wyrzucany gdy wprowadzony element nie może być przetworzony na wartość liczbową.

```
#include <error.hpp>
```

Diagram dziedziczenia dla NotNumericValue

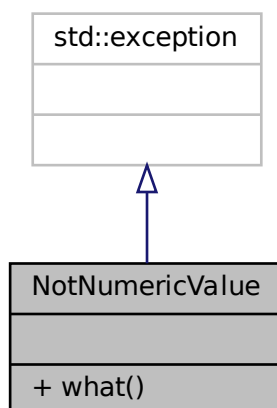
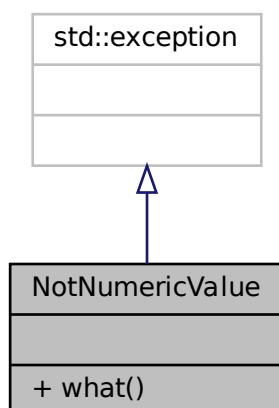


Diagram współpracy dla NotNumericValue:



Metody publiczne

- `const char * what () const throw ()`

4.5.1 Opis szczegółowy

Wyjątek wyrzucany w przypadku wprowadzenia wartości nie numerycznej do komórki. Wyjątek powinien być wyrzucany gdy wprowadzony element nie może być przetworzony na wartość liczbową.

Dokumentacja dla tej struktury została wygenerowana z pliku:

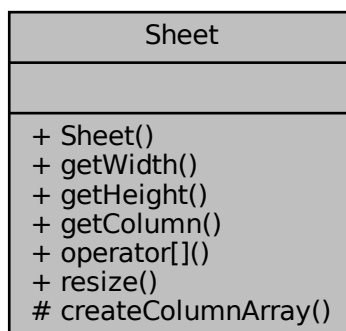
- ProgramowanieObiektowe/[error.hpp](#)

4.6 Dokumentacja klasy Sheet

Klasa opisująca Arkusz Klasa Arkusz przechowujący tablicę kolumn i jej rozmiar.

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

Diagram współpracy dla Sheet:



Metody publiczne

- [Sheet](#) (size_t width, size_t height, [CellType](#) *types)
[Sheet](#) Konstruktor tworzący arkusz z tablicą o wyznaczonym rozmiarze Konstruktor tworzący arkusz z tablicą o wyznaczonej ilości kolumn określonego typu i wierszy.
- size_t [getWidth](#) ()
getWidth getter szerokości Zwraca ilość kolumn w arkuszu
- size_t [getHeight](#) ()
getHeight getter wysokości Zwraca ilość komórek w kolumnie kiedy arkusz był tworzony/rozszerzany
- [Column](#) & [getColumn](#) (size_t x)
getColumn Metoda zwracająca referencję na kolumnę Zwraca referencję na wybraną kolumnę z arkusza
- [Column](#) & [operator\[\]](#) (size_t x)
operator [] przeciążenie operatora[] celem uzyskiwania odrębnej kolumny
- void [resize](#) (size_t x, size_t y)
resize Metoda rozszerzania arkusza

Statyczne metody chronione

- static [Column](#) ** [createColumnArray](#) (size_t width, size_t height, [CellType](#) *types)
Wysokość tablicy - ilość komórek w utworzonych kolumnach.

4.6.1 Opis szczegółowy

Klasa opisująca Arkusz Klasa Arkusz przechowująca tablicę kolumn i jej rozmiar.

4.6.2 Dokumentacja konstruktora i destruktor

4.6.2.1 Sheet()

```
Sheet::Sheet (
    size_t width,
    size_t height,
    CellType * types )
```

Sheet Konstruktor tworzący arkusz z tablicą o wyznaczonym rozmiarze Konstruktor tworzący arkusz z tablicą o wyznaczonej ilości kolumn określonego typu i wierszy.

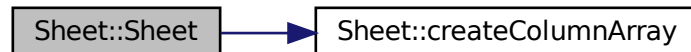
Parametry

in	<i>width</i>	Szerokość tablicy nowego arkusza
in	<i>height</i>	Wysokość nowej tablicy
in	<i>types</i>	Typy tworzonych kolumn

Wyjątki

<i>bad_array_new_length</i>	Zły rozmiar tworzonego arkusza
-----------------------------	--------------------------------

Oto graf wywołań dla tej funkcji:



4.6.3 Dokumentacja funkcji składowych

4.6.3.1 createColumnArray()

```
Column ** Sheet::createColumnArray (
    size_t width,
    size_t height,
    CellType * types ) [static], [protected]
```

Wysokość tablicy - ilość komórek w utworzonych kolumnach.

createColumnArray Tworzy nową dwuwymiarową tablicę. Funkcja generująca tablicę o określonym rozmiarze

Parametry

in	<i>width</i>	Szerokość nowej tablicy - ilość kolumn
in	<i>height</i>	Wysokość nowej tablicy - ilość komórek w kolumnach
in	<i>types</i>	Typy tworzonych kolumn

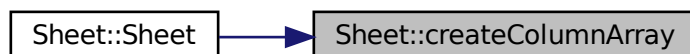
Wyjątki

<code>bad_array_new_length</code>	Zły rozmiar tworzonego arkusza
-----------------------------------	--------------------------------

Zwraca

Tworzy nową tablicę wskaźników kolumn o wyznaczonych rozmiarach

Oto graf wywoływań tej funkcji:



4.6.3.2 getColumn()

```
Column & Sheet::getColumn (
    size_t x )
```

getColumn Metoda zwracająca referencję na kolumnę Zwraca referencję na wybraną kolumnę z arkusza

Parametry

in	x	Indeks kolumny
----	---	----------------

Wyjątki

<code>std::out_of_range</code>	Gdy x jest poza zakresem arkusza ($x > \text{width}$)
--------------------------------	---

Zwraca

Referencja na kolumnę

Oto graf wywoływań tej funkcji:



4.6.3.3 getHeight()

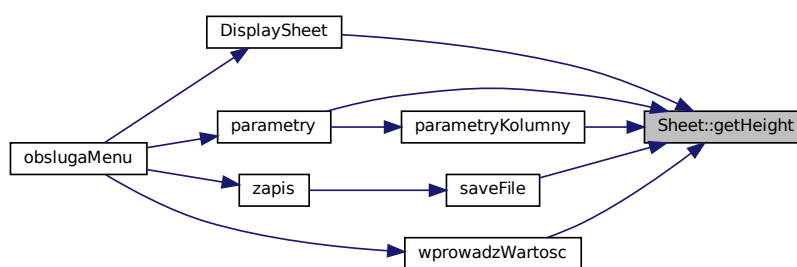
```
size_t Sheet::getHeight ( )
```

getHeight getter wysokości Zwraca ilość komórek w kolumnie kiedy arkusz był tworzony/rozszerzany

Zwraca

wysokość arkusza / ilość wierszy

Oto graf wywoływań tej funkcji:



4.6.3.4 getWidth()

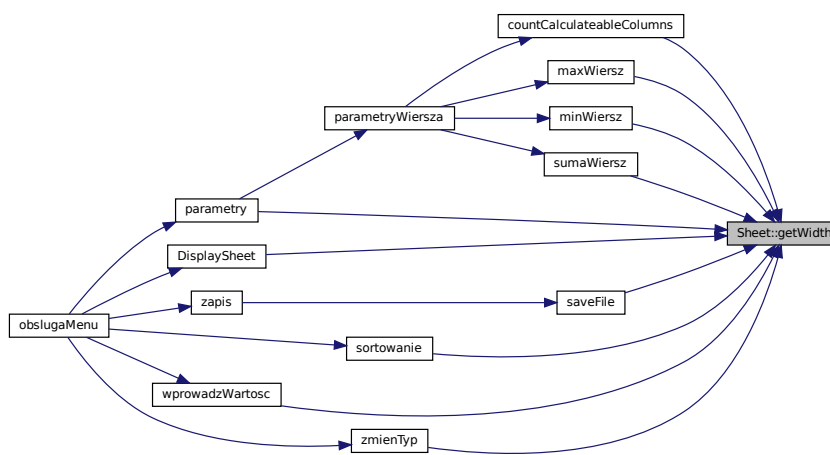
```
size_t Sheet::getWidth ( )
```

getWidth getter szerokości Zwraca ilość kolumn w arkuszu

Zwraca

ilość kolumn

Oto graf wywoływań tej funkcji:



4.6.3.5 operator[]()

```
Column & Sheet::operator[] (
    size_t x )
```

operator [] przeciążenie operatora[] celem uzyskiwania odrębnej kolumny

Zwraca referencję na wybraną kolumnę

Parametry

in	x	Indeks kolumny
----	---	----------------

Wyjątki

<code>std::out_of_range</code>	Gdy x jest poza zakresem arkusza (x > width)
--------------------------------	--

Zwraca

Referencja na wybraną kolumnę

Oto graf wywołań dla tej funkcji:

**4.6.3.6 resize()**

```
void Sheet::resize (
    size_t x,
    size_t y )
```

resize Metoda rozszerzania arkusza

Metoda zmienia rozmiar arkusza kopiując kolumny które także przechodzą zmianę rozmiaru. Nowe kolumny są automatycznie przeznaczone pod komórki typu [IntCell](#). Utrata danych w przypadku zmniejszania rozmiaru arkusza.

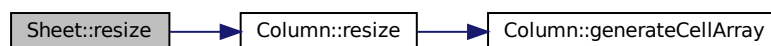
Parametry

in	x	Nowa szerokość arkusza
in	y	Nowa wysokość kolumn arkusza

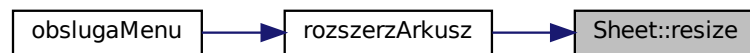
Wyjątki

<i>bad_array_new_length</i>	Zły rozmiar tworzonego arkusza
-----------------------------	--------------------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.7 Dokumentacja klasy StringCell

Diagram dziedziczenia dla StringCell

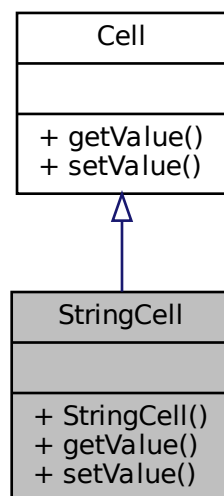
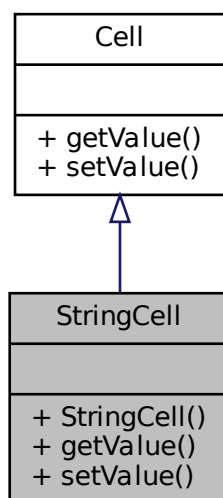


Diagram współpracy dla StringCell:



Metody publiczne

- [StringCell](#) (std::string value="?")
StringCell Konstruktor domyślny z opcjonalnym parametrem Konstruktor umożliwiający określenie początkowej wartości komórki.
- std::string [getValue](#) ()
getValue Metoda ogólna od uzyskiwania wartości komórki Metoda zwraca wartość w postaci wartości tekstowej komórki
- void [setValue](#) (std::string value)
setValue Metoda ogólna od ustawiania wartości Ustawia wartość komórki za pomocą wartości string

4.7.1 Dokumentacja konstruktora i destruktora

4.7.1.1 StringCell()

```
StringCell::StringCell (
    std::string value = "?" )
```

[StringCell](#) Konstruktor domyślny z opcjonalnym parametrem Konstruktor umożliwiający określenie początkowej wartości komórki.

Parametry

in	value	Wartość początkowa komórki
----	-------	----------------------------

4.7.2 Dokumentacja funkcji składowych

4.7.2.1 getValue()

```
std::string StringCell::getValue ( ) [virtual]
```

getValue Metoda ogólna od uzyskiwania wartości komórki Metoda zwraca wartość w postaci wartości tekstowej komórki

Zwraca

zawartość tekstowa komórki

Implementuje [Cell](#).

4.7.2.2 setValue()

```
void StringCell::setValue (
    std::string value ) [virtual]
```

setValue Metoda ogólna od ustawiania wartości Ustawia wartość komórki za pomocą wartości string

Parametry

in	value	ustawiana wartość
----	-------	-------------------

Implementuje [Cell](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

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

Chapter 5

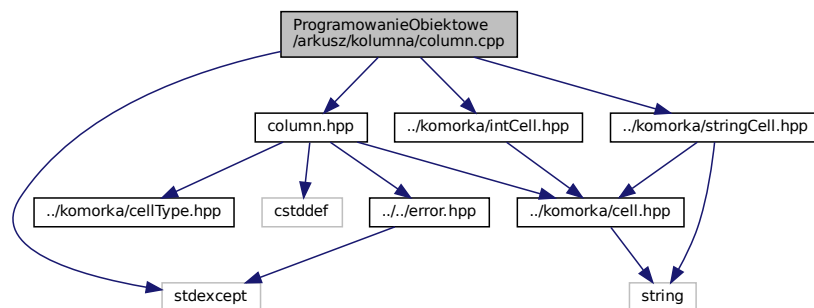
Dokumentacja plików

5.1 Dokumentacja pliku

ProgramowanieObiektowe/arkusz/kolumna/column.cpp

```
#include "column.hpp"  
#include "../komorka/intCell.hpp"  
#include "../komorka/stringCell.hpp"  
#include <stdexcept>
```

Wykres zależności załączania dla column.cpp:



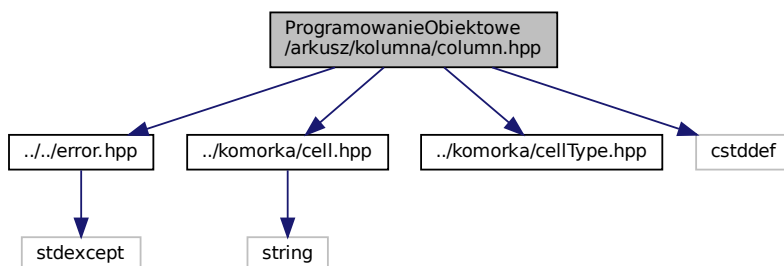
5.2 Dokumentacja pliku

ProgramowanieObiektowe/arkusz/kolumna/column.hpp

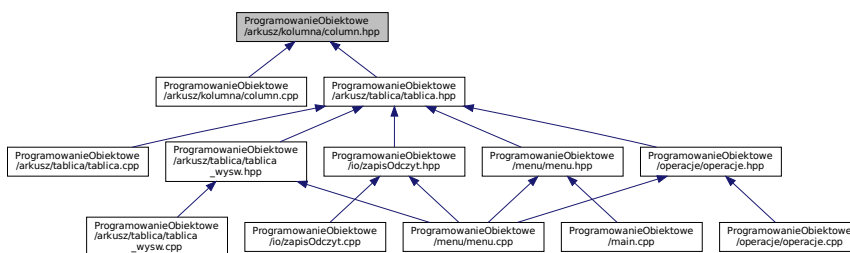
```
#include ".././error.hpp"  
#include "../komorka/cell.hpp"  
#include "../komorka/cellType.hpp"
```

```
#include <cstdint>
```

Wykres zależności załączania dla column.hpp:



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



Komponenty

- class [Column](#)

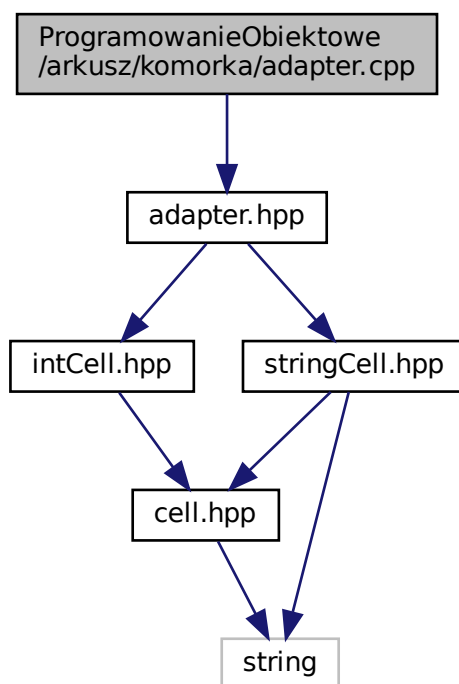
Klasa określająca kolumnę. Klasa kolumna obsługująca podstawowe funkcjonalności kolekcji posiada wskaźnik początku i końca zakresu celem wykorzystania w funkcjach z biblioteki algorithm czy pętla zakresowych.

5.3 Dokumentacja pliku

ProgramowanieObiektowe/arkusz/komorka/adapter.cpp

```
#include "adapter.hpp"
```


Wykres zależności załączania dla adapter.cpp:

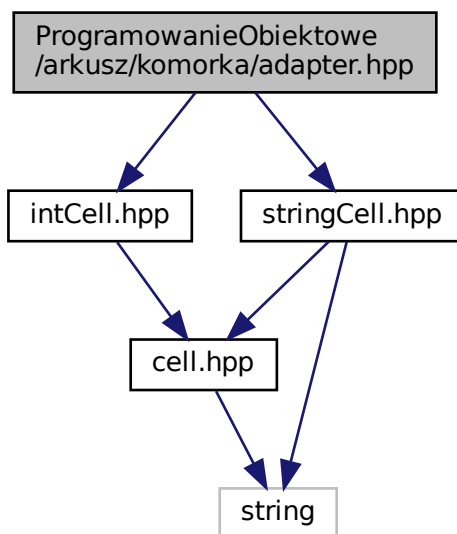


5.4 Dokumentacja pliku

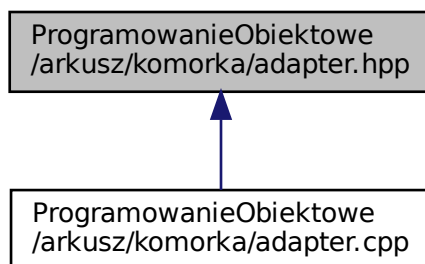
ProgramowanieObiektowe/arkusz/komorka/adapter.hpp

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

Wykres zależności załączania dla adapter.hpp:



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



5.5 Dokumentacja pliku

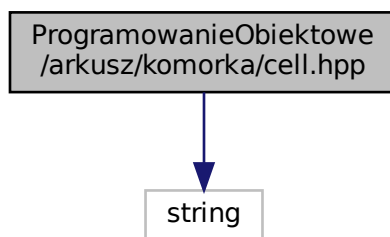
ProgramowanieObiektowe/arkusz/komorka/cell.cpp

5.6 Dokumentacja pliku

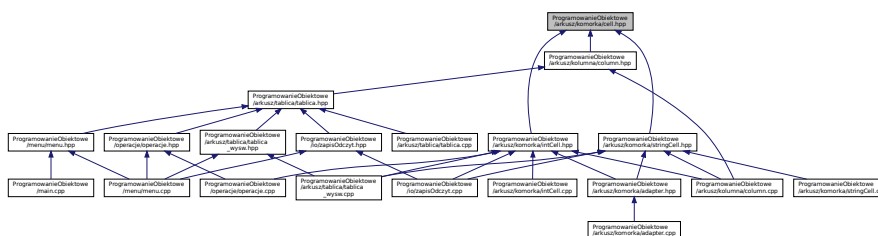
ProgramowanieObiektowe/arkusz/komorka/cell.hpp

```
#include <string>
```

Wykres zależności załączania dla cell.hpp:



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



Komponenty

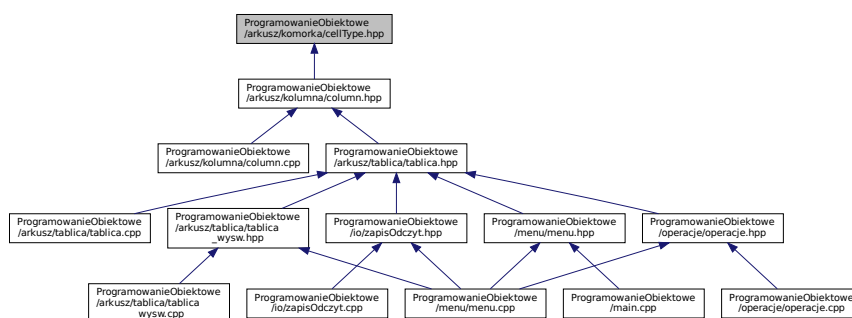
- class **Cell**

Interfejs/ klasa czysto wirtualna komórka Interfejs komórka posiadający metody ogólne klas dziedziczących.

5.7 Dokumentacja pliku

ProgramowanieObiektowe/arkusz/komorka/cellType.hpp

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



Wyliczenia

- enum class `CellType` { `StringCell` = 0 , `IntCell` = 1 }

Typ wyliczeniowy typów komórek.

5.7.1 Dokumentacja typów wyliczanych

5.7.1.1 CellType

```
enum CellType [strong]
```

Typ wyliczeniowy typów komórek.

Wartości wyliczeń

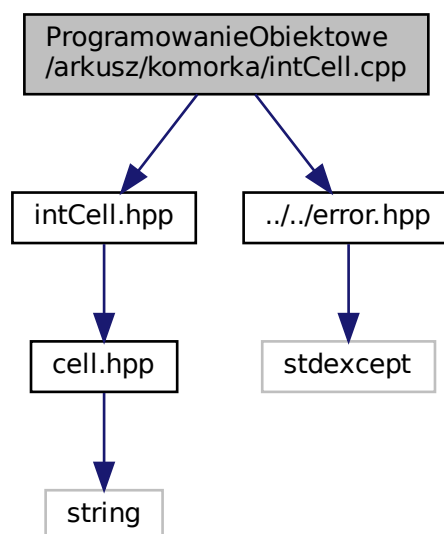
IntCell	Komórka tekstowa.
---------	-------------------

5.8 Dokumentacja pliku

ProgramowanieObiektowe/arkusz/komorka/intCell.cpp

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

Wykres zależności załączania dla intCell.cpp:

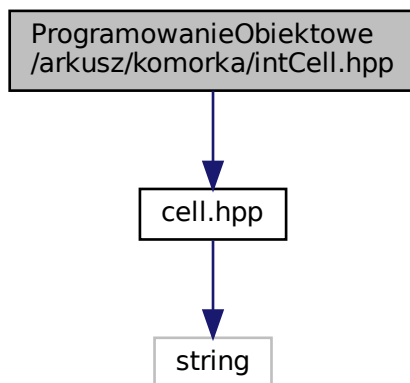


5.9 Dokumentacja pliku

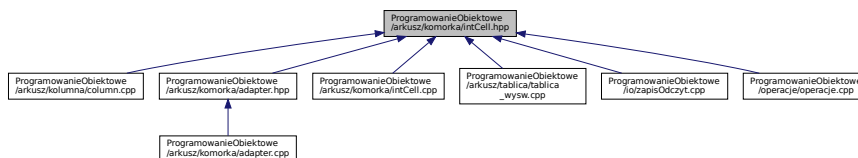
ProgramowanieObiektowe/arkusz/komorka/intCell.hpp

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

Wykres zależności załączania dla intCell.hpp:



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



Komponenty

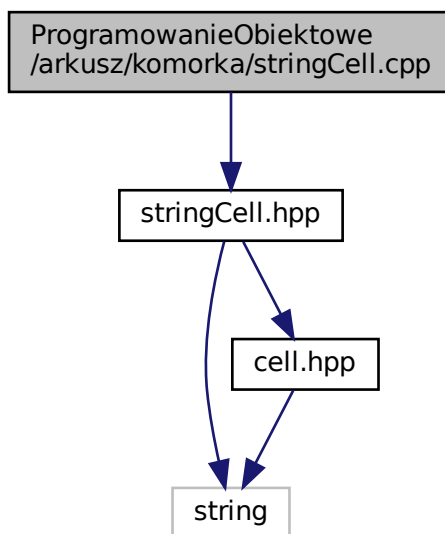
- class `IntCell`

5.10 Dokumentacja pliku

ProgramowanieObiektowe/arkusz/komorka/stringCell.cpp

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

Wykres zależności załączania dla stringCell.cpp:

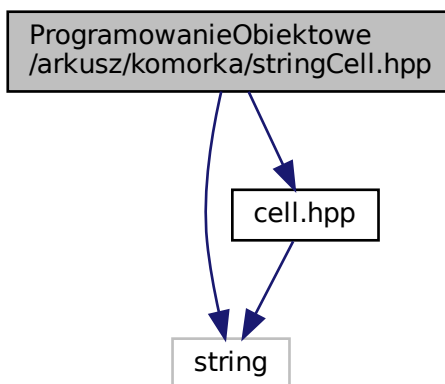


5.11 Dokumentacja pliku

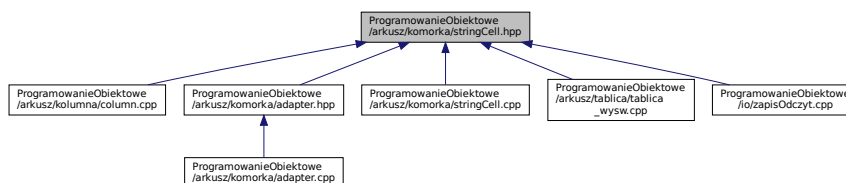
ProgramowanieObiektowe/arkusz/komorka/stringCell.hpp

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

Wykres zależności załączania dla stringCell.hpp:



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



Komponenty

- class [StringCell](#)

5.12 Dokumentacja pliku

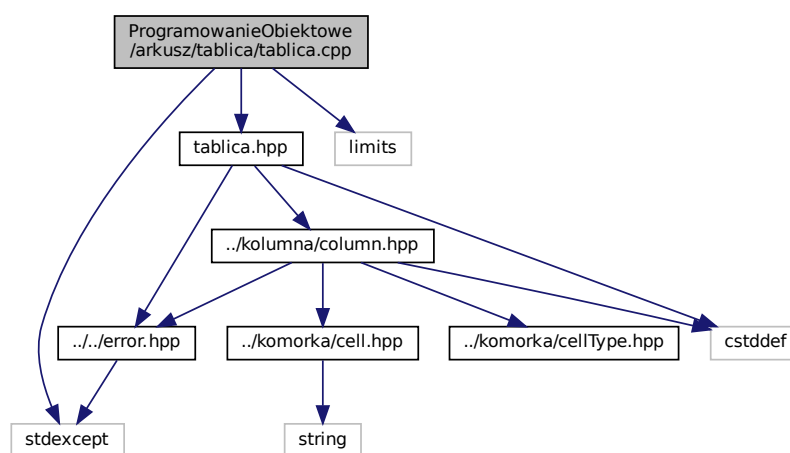
ProgramowanieObiektowe/arkusz/tablica/tablica.cpp

```
#include "tablica.hpp"
```

```
#include <limits>
```

```
#include <stdexcept>
```

Wykres zależności załączania dla tablica.cpp:



5.13 Dokumentacja pliku

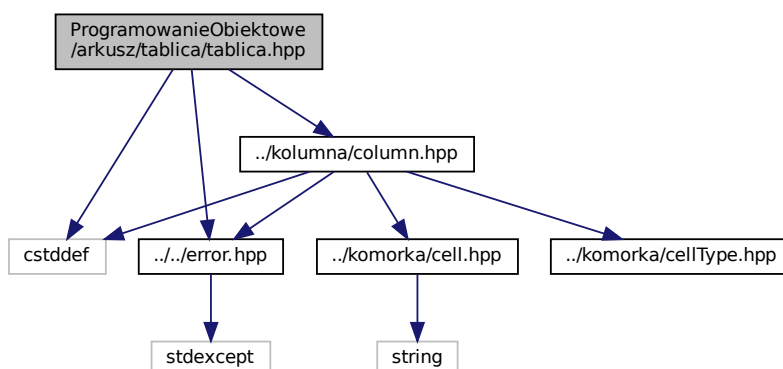
ProgramowanieObiektowe/arkusz/tablica/tablica.hpp

```
#include <cstdint>
```

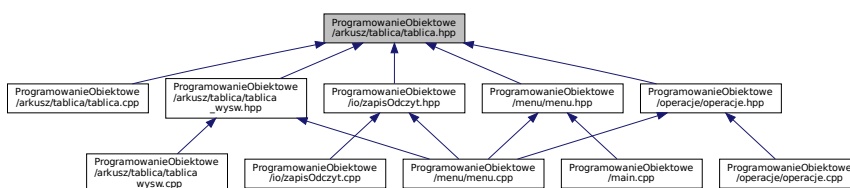
```
#include "../error.hpp"
```

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

Wykres zależności załączania dla tablica.hpp:



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



Komponenty

- class [Sheet](#)

Klasa opisująca Arkusz Klasa Arkusz przechowująca tablicę kolumn i jej rozmiar.

5.14 Dokumentacja pliku

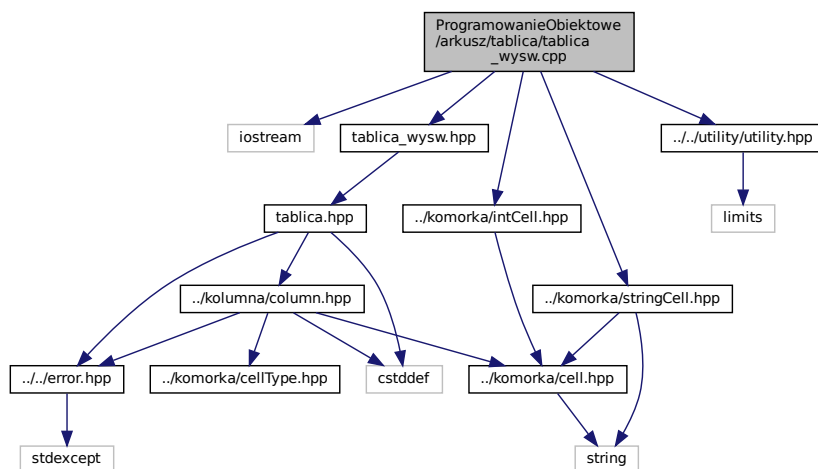
ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.cpp

```
#include <iostream>
#include "tablica_wysw.hpp"
#include "../komorka/intCell.hpp"
#include "../komorka/stringCell.hpp"
```



```
#include "../utility/utility.hpp"
```

Wykres zależności załączania dla tablica_wysw.cpp:



Funkcje

- void [DisplaySheet](#) ([Sheet](#) sheet)

Metoda od wyświetlania arkusza Metoda wyświetla wszystkie elementy znajdujące się w arkuszu.

5.14.1 Dokumentacja funkcji

5.14.1.1 DisplaySheet()

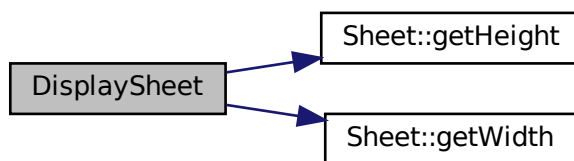
```
void DisplaySheet (
    Sheet sheet )
```

Metoda od wyświetlania arkusza Metoda wyświetla wszystkie elementy znajdujące się w arkuszu.

Parametry

in	sheet	Arkusz przeznaczony do wyświetlenia
----	-------	-------------------------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

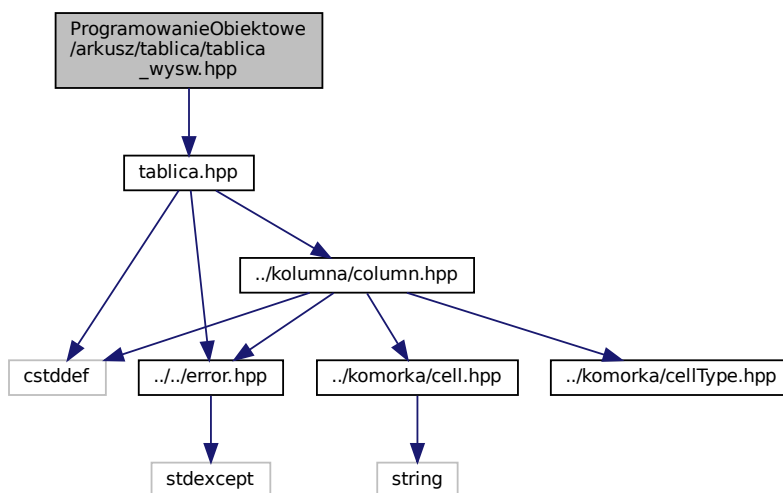


5.15 Dokumentacja pliku

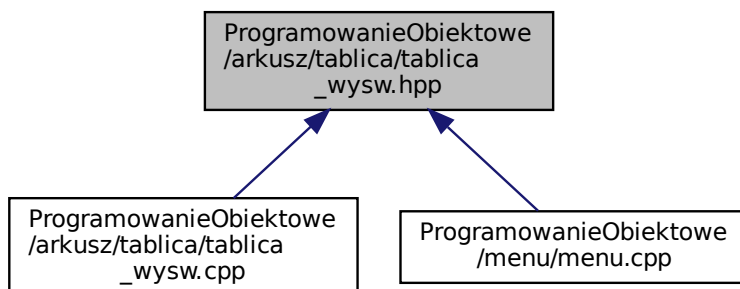
ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.hpp

```
#include "tablica.hpp"
```

Wykres zależności załączania dla `tablica_wysw.hpp`:



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



Funkcje

- void `DisplaySheet` (`Sheet` sheet)

Metoda od wyświetlania arkusza Metoda wyświetla wszystkie elementy znajdujące się w arkuszu.

5.15.1 Dokumentacja funkcji

5.15.1.1 DisplaySheet()

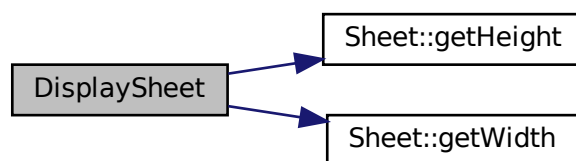
```
void DisplaySheet (  
    Sheet sheet )
```

Metoda od wyświetlania arkusza Metoda wyświetla wszystkie elementy znajdujące się w arkuszu.

Parametry

in	sheet	Arkusz przeznaczony do wyświetlenia
----	-------	-------------------------------------

Oto graf wywołań dla tej funkcji:



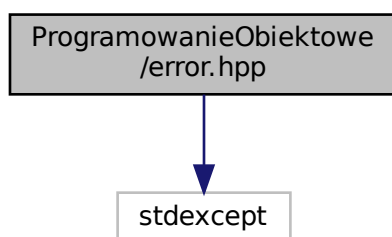
Oto graf wywoływań tej funkcji:



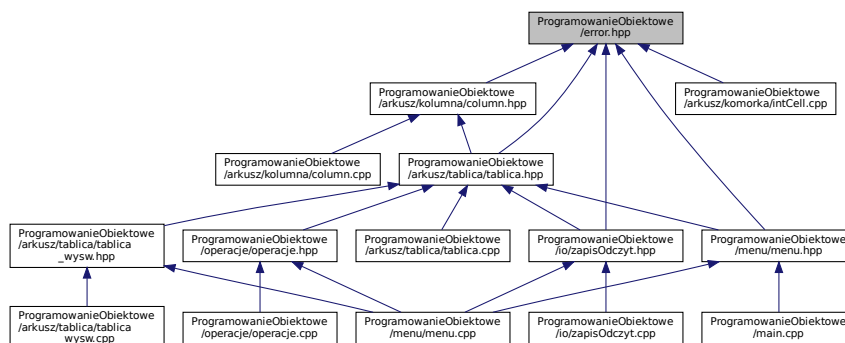
5.16 Dokumentacja pliku ProgramowanieObiektowe/error.hpp

```
#include <stdexcept>
```

Wykres zależności załączania dla `error.hpp`:



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



Komponenty

- struct [BadFileException](#)

Wyjątek wyrzucany w przypadku braku dostępu. Wyjątek jest wyrzucany gdy użytkownik nie ma dostępu do pliku lub nie istnieje (odczyt).

- struct [NotNumericValue](#)

Wyjątek wyrzucany w przypadku wprowadzenia wartości nie numerycznej do komórki. Wyjątek powinien być wyrzucany gdy wprowadzony element nie może być przetworzony na wartość liczbową.

Wyliczenia

- enum class [Wyjatki](#) : 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.

5.16.1 Dokumentacja typów wyliczanych

5.16.1.1 Wyjątki

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

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

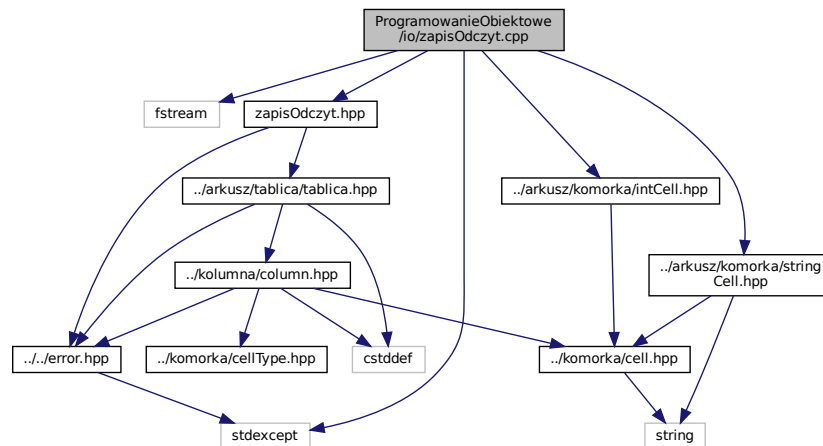
Wartości wyliczeń

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.

5.17 Dokumentacja pliku ProgramowanieObiektowe/io/zapisOdczyt.cpp

```
#include <fstream>
#include "zapisOdczyt.hpp"
#include <stdexcept>
#include "../arkusz/komorka/intCell.hpp"
#include "../arkusz/komorka/stringCell.hpp"
```

Wykres zależności załączania dla zapisOdczyt.cpp:



Funkcje

- void [saveFile](#) ([Sheet](#) sheet, std::string fileName)
Funkcja zapisu do pliku.
- void [loadFile](#) ([Sheet](#) *sheet, std::string fileName)
Funkcja wczytywania tablicy z pliku.

5.17.1 Dokumentacja funkcji

5.17.1.1 loadFile()

```
void loadFile (
    Sheet * sheet,
    std::string fileName )
```

Funkcja wczytywania tablicy z pliku.

Funkcja wykonuje wczytanie arkusza z wybranego pliku.

Parametry

in, out	<i>sheet</i>	Arkusz do nadpisania wczytywaną tablicą
in	<i>fileName</i>	Nazwa wczytywanego pliku

Wyjątki

<i>BadFileException</i>	W przypadku braku pliku lub braku dostępu
-------------------------	---

Oto graf wywoływań tej funkcji:



5.17.1.2 saveFile()

```
void saveFile (
    Sheet sheet,
    std::string fileName = "Arkusz.csv" )
```

Funkcja zapisu do pliku.

Funkcja wykonuje zapis do wybranego przez nas pliku

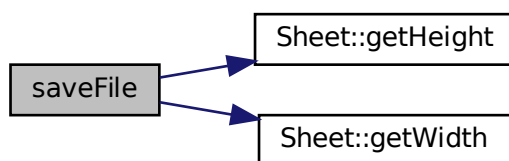
Parametry

in	<i>sheet</i>	Arkusz przeznaczony do zapisu
in	<i>fileName</i>	Nazwa zapisywanego pliku

Wyjątki

<i>BadFileException</i>	W przypadku braku dostępu do zapisu
-------------------------	-------------------------------------

Oto graf wywołań dla tej funkcji:



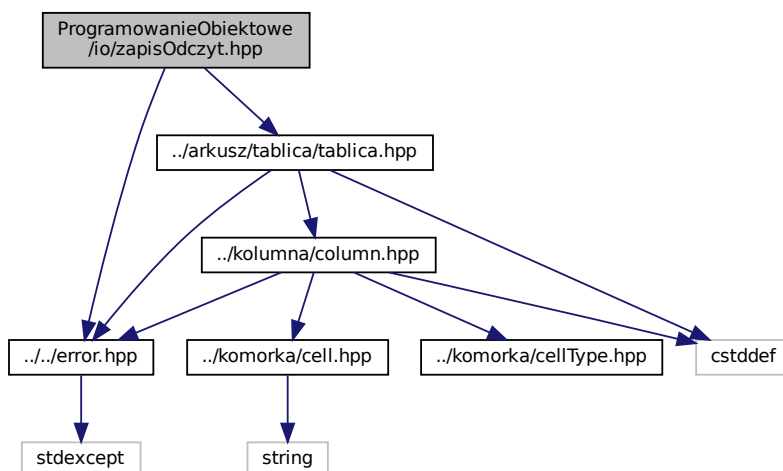
Oto graf wywoływań tej funkcji:



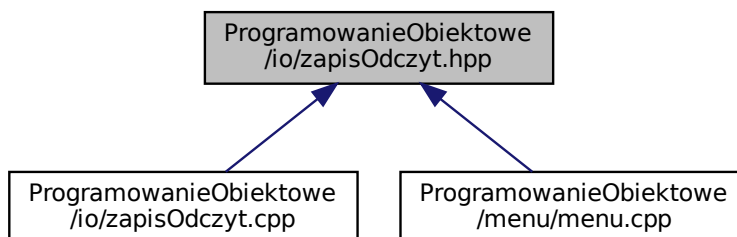
5.18 Dokumentacja pliku ProgramowanieObiektowe/io/zapisOdczyt.hpp

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

Wykres zależności załączania dla zapisOdczyt.hpp:



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



Funkcje

- void `saveFile` (`Sheet` sheet, `std::string` fileName="Arkusz.csv")
Funkcja zapisu do pliku.
- void `loadFile` (`Sheet` *sheet, `std::string` fileName)
Funkcja wczytywania tablicy z pliku.

5.18.1 Dokumentacja funkcji

5.18.1.1 loadFile()

```
void loadFile (  
    Sheet * sheet,  
    std::string fileName )
```

Funkcja wczytywania tablicy z pliku.

Funkcja wykonuje wczytanie arkusza z wybranego pliku.

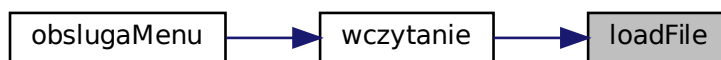
Parametry

in, out	<code>sheet</code>	Arkusz do nadpisania wczytywaną tablicą
in	<code>fileName</code>	Nazwa wczytywanego pliku

Wyjątki

<code>BadFileException</code>	W przypadku braku pliku lub braku dostępu
-------------------------------	---

Oto graf wywoływań tej funkcji:



5.18.1.2 saveFile()

```
void saveFile (
    Sheet sheet,
    std::string fileName = "Arkusz.csv" )
```

Funkcja zapisu do pliku.

Funkcja wykonuje zapis do wybranego przez nas pliku

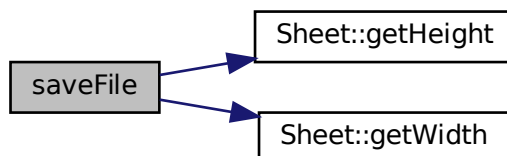
Parametry

in	<i>sheet</i>	Arkusz przeznaczony do zapisu
in	<i>fileName</i>	Nazwa zapisywanego pliku

Wyjątki

<i>BadFileException</i>	W przypadku braku dostępu do zapisu
---	-------------------------------------

Oto graf wywołań dla tej funkcji:



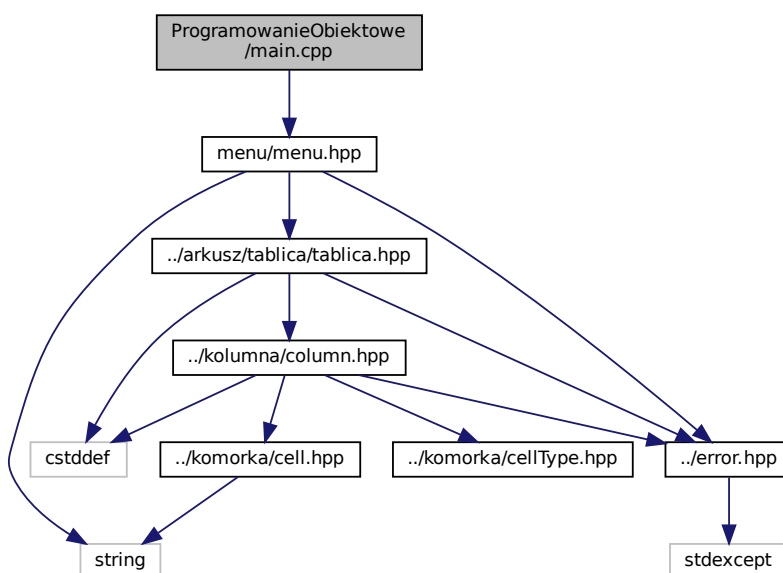
Oto graf wywoływań tej funkcji:



5.19 Dokumentacja pliku ProgramowanieObiektowe/main.cpp

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

Wykres zależności załączania dla main.cpp:



Funkcje

- `int main ()`

5.20 Dokumentacja pliku ProgramowanieObiektowe/menu/menu.cpp

```
#include <iostream>
#include <cstdlib>
#include <sstream>
#include "menu.hpp"
```


5.20.1.1 generujMenu()

```
void generujMenu ( )
```

Funkcja tworząca menu.

Funkcja od tworzenia listy dostępnych pozycji menu. Oto graf wywołań tej funkcji:



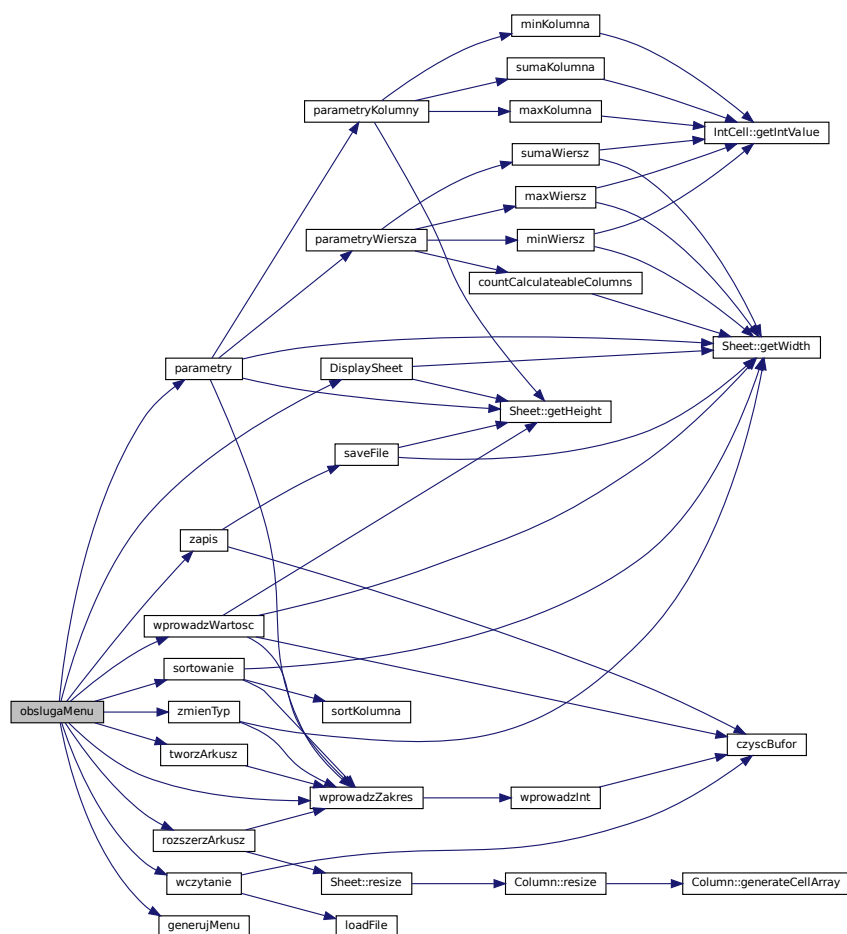
5.20.1.2 obslugaMenu()

```
void obslugaMenu ( )
```

Funkcja kontrolująca działanie programu.

Funkcja zajmująca się obsługą menu programu zarządza tym co będzie wywoływane. Oto graf wywołań dla tej

funkcji:



5.20.1.3 parametry()

```
void parametry (
    Sheet arkusz )
```

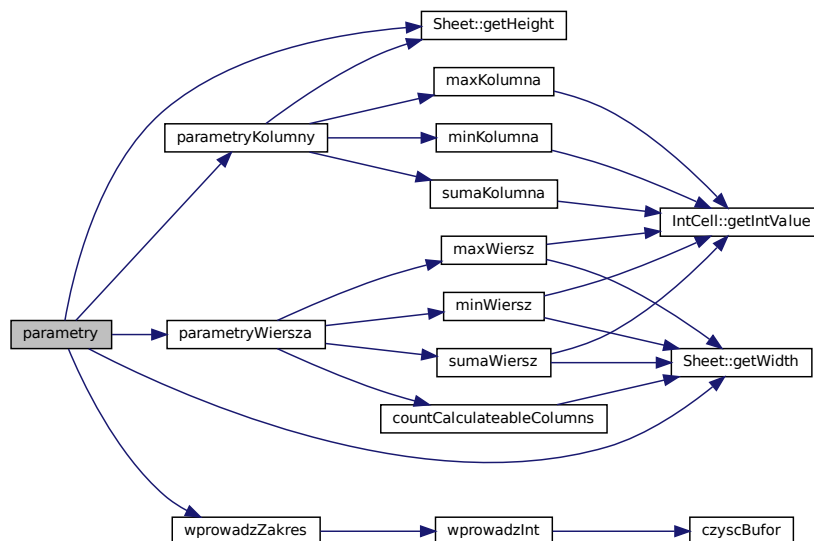
Funkcja menu od wyboru względem czego wyznacza parametry.

Funkcja menu od wyboru atrybutu tablicy (kolumny lub wiersza) która ma za wyświetlenie parametrów wybranego atrybutu.

Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.4 parametryKolumny()

```

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

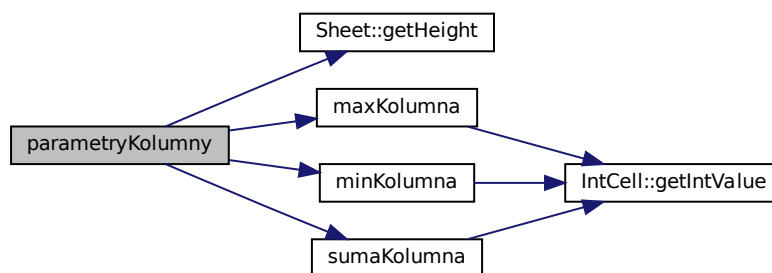
Funkcja od wyznaczania parametrów kolumny arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranej kolumny

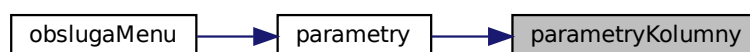
Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.5 parametryWiersza()

```

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

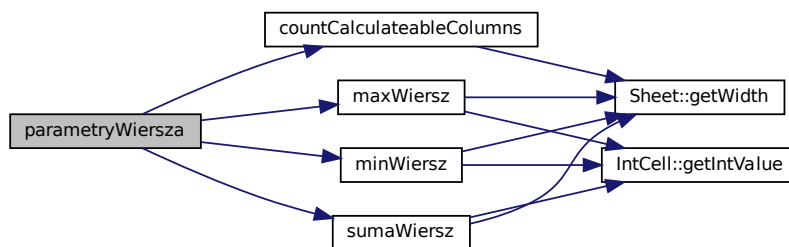
Funkcja od wyznaczania parametrów wiersza arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranego wiersza

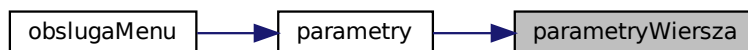
Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.6 rozszerzArkusz()

```
void rozszerzArkusz (
    Sheet * arkusz )
```

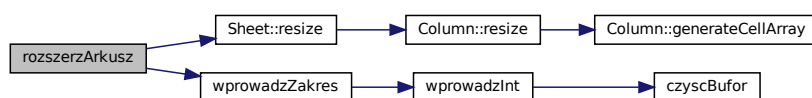
Funkcja modyfikująca rozmiar arkusza.

Interfejs umożliwiający modyfikację rozmiaru istniejącego arkusza.

Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.7 sortowanie()

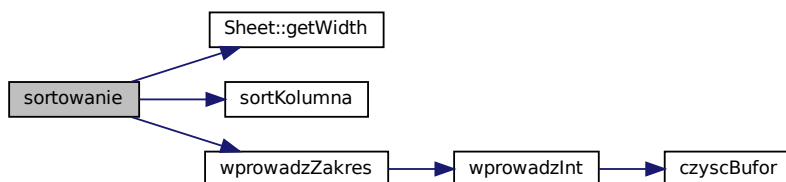
```
void sortowanie (
    Sheet * arkusz )
```

Funkcja interfejsu od sortowania kolumny Metoda zawiera interfejs do sortowania kolumny z arkusza.

Parametry

<code>in, out</code>	<i>Arkusz</i>	którego kolumna będzie sortowana
----------------------	---------------	----------------------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.8 tworzArkusz()

```
Sheet tworzArkusz ( )
```

Funkcja tworząca nową tablicę.

Funkcja zawierająca interfejs umożliwiający tworzenie nowego Arkusza z tablicą dwuwymiarową.

Zwraca

Nowy Arkusz do wykorzystywania w programie

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.9 wczytanie()

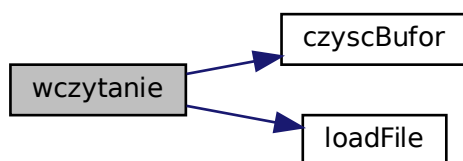
```
void wczytanie (  
    Sheet * arkusz )
```

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

Parametry

in, out	arkusz	Arkusz do którego mogą być wczytane elementy
---------	--------	--

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.10 wprowadzWartosc()

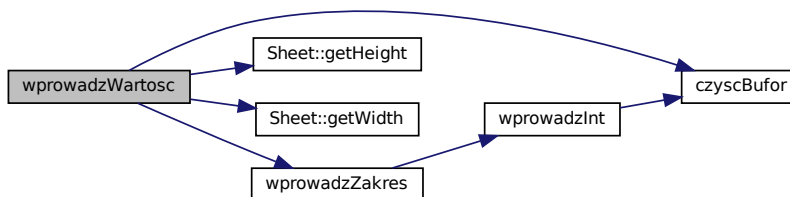
```
void wprowadzWartosc (  
    Sheet * arkusz )
```

wprowadzWartosc modyfikuje wartość komórki Frontend od modyfikacji wartości który w zależności od typu komórki zmieni jej wartość

Parametry

arkusz	Arkusz którego element będzie modyfikowany
--------	--

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.11 zapis()

```
void zapis (
    Sheet arkusz )
```

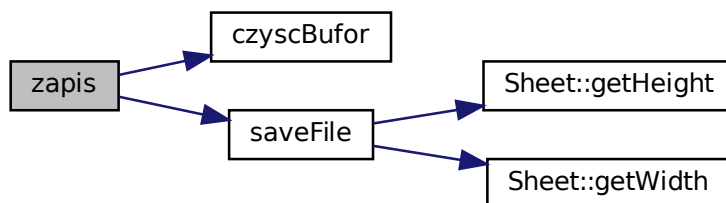
Funkcja menu od zapisu.

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

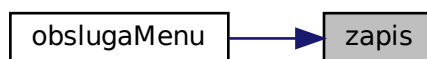
Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.20.1.12 zmienTyp()

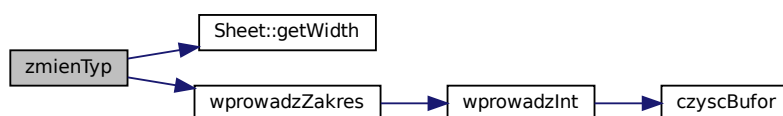
```
void zmienTyp (
    Sheet * arkusz )
```

Funkcja od zmiany typu kolumny Interfejs od metody zmieniającej typ kolumny arkusza.

Parametry

in	arkusz	Arkusz którego kolumna zostaje zmieniona
----	--------	--

Oto graf wywołań dla tej funkcji:



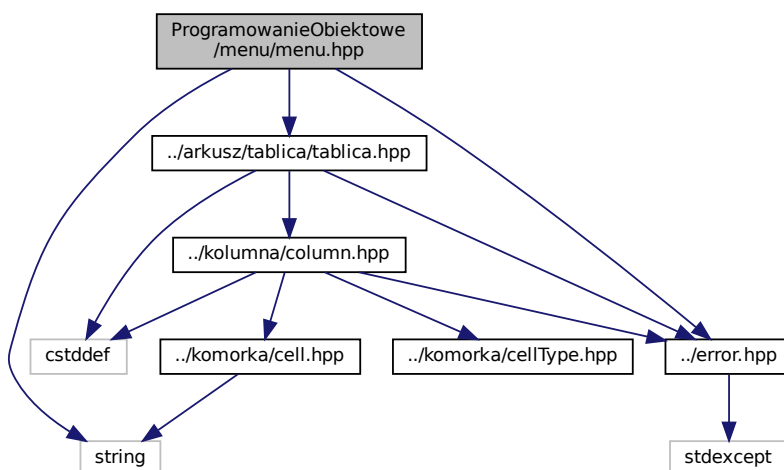
Oto graf wywoływań tej funkcji:



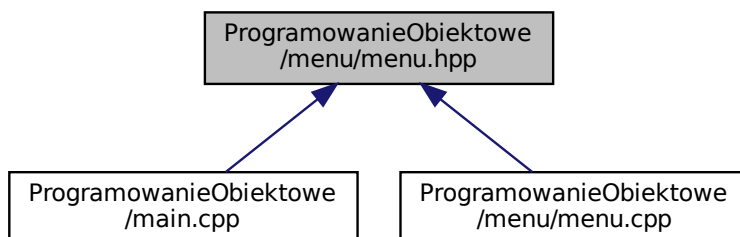
5.21 Dokumentacja pliku ProgramowanieObiektowe/menu/menu.hpp

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

Wykres zależności załączania dla menu.hpp:



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



Funkcje

- void `obslugaMenu` ()
Funkcja kontrolująca działanie programu.
- void `generujMenu` ()
Funkcja tworząca menu.
- `Sheet` `tworzArkusz` ()
Funkcja tworząca nową tablicę.
- void `rozszerzArkusz` (`Sheet` *arkusz)
Funkcja modyfikująca rozmiar arkusza.
- void `wczytanie` (`Sheet` *arkusz)
Funkcja wczytywania arkusza Funkcja menu od wczytywania arkusza, ma za zadanie opakowanie funkcji IO wczytajPlik.
- void `zapis` (`Sheet` arkusz)
Funkcja menu od zapisu.
- void `parametry` (`Sheet` arkusz)
Funkcja menu od wyboru względem czego wyznacza parametry.
- std::string `parametryWiersza` (`Sheet` arkusz, int wiersz)
Funkcja od wyznaczania parametrów wiersza arkusza.
- std::string `parametryKolumny` (`Sheet` arkusz, int kolumna)
Funkcja od wyznaczania parametrów kolumny arkusza.
- void `wprowadzWartosc` (`Sheet` *arkusz)
wprowadzWartosc modyfikuje wartość komórki Frontend od modyfikacji wartości który w zależności od typu komórki zmieni jej wartość
- void `zmienTyp` (`Sheet` *arkusz)
Funkcja od zmiany typu kolumny Interfejs od metody zmieniającej typ kolumny arkusza.
- void `sortowanie` (`Sheet` *arkusz)
Funkcja interfejsu od sortowania kolumny Metoda zawiera interfejs do sortowania kolumny z arkusza.

5.21.1 Dokumentacja funkcji

5.21.1.1 generujMenu()

```
void generujMenu ( )
```

Funkcja tworząca menu.

Funkcja od tworzenia listy dostępnych pozycji menu. Oto graf wywoływań tej funkcji:

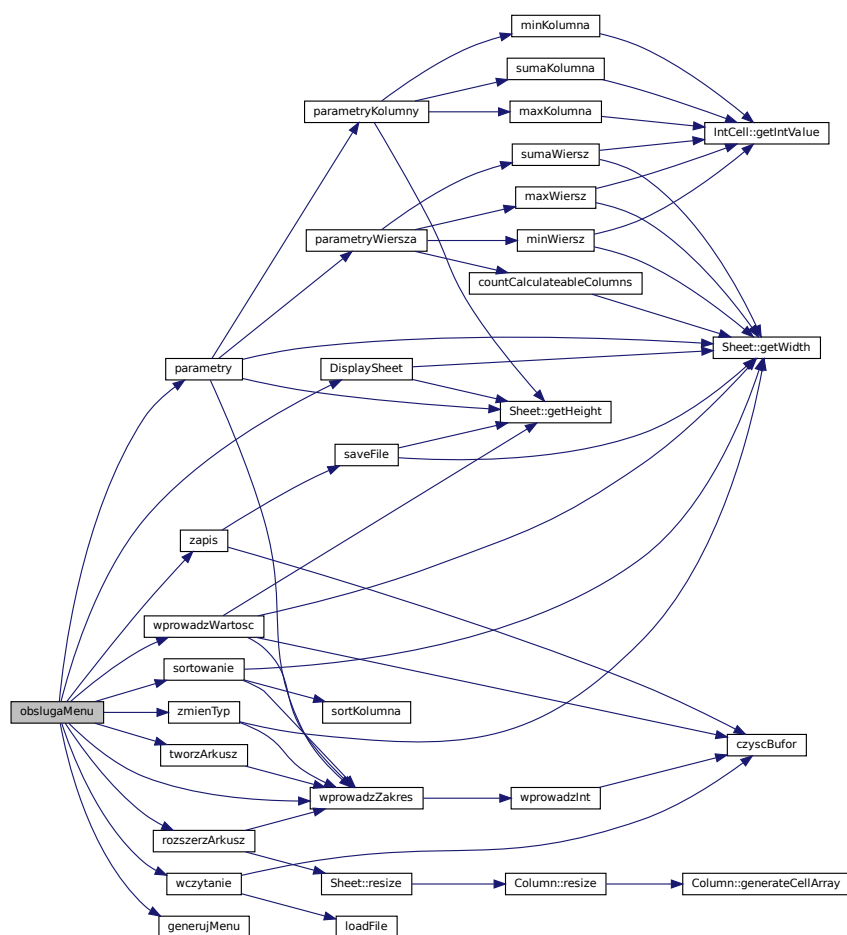


5.21.1.2 obslugaMenu()

```
void obslugaMenu ( )
```

Funkcja kontrolująca działanie programu.

Funkcja zajmująca się obsługą menu programu zarządza tym co będzie wywoływane. Oto graf wywołań dla tej funkcji:



5.21.1.3 parametry()

```
void parametry (
    Sheet arkusz )
```

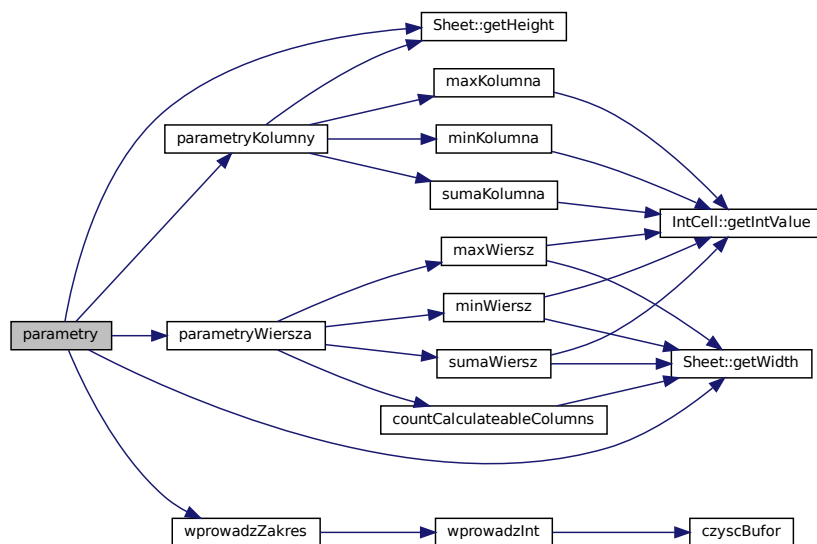
Funkcja menu od wyboru względem czego wyznacza parametry.

Funkcja menu od wyboru atrybutu tablicy (kolumny lub wiersza) która ma za wyświetlenie parametrów wybranego atrybutu.

Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.4 parametryKolumny()

```
std::string parametryKolumny (
    Sheet arkusz,
    int kolumna )
```

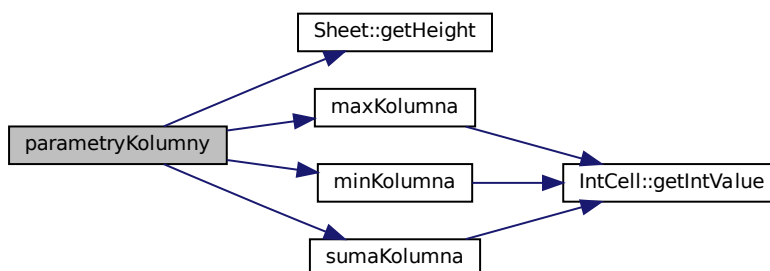
Funkcja od wyznaczania parametrów kolumny arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranej kolumny

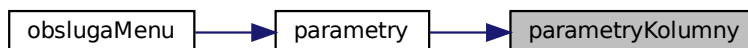
Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**5.21.1.5 parametryWiersza()**

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

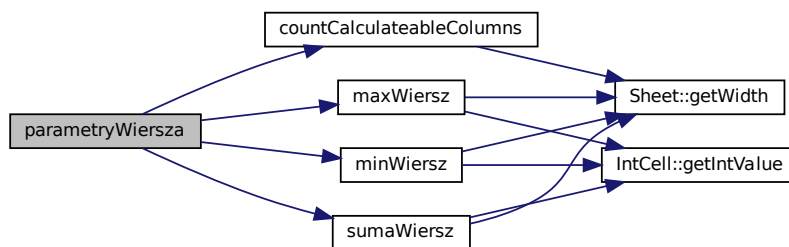
Funkcja od wyznaczania parametrów wiersza arkusza.

Funkcja zwraca w postaci tekstowej wszystkie parametry wybranego wiersza

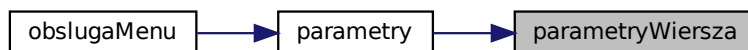
Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.6 rozszerzArkusz()

```
void rozszerzArkusz (
    Sheet * arkusz )
```

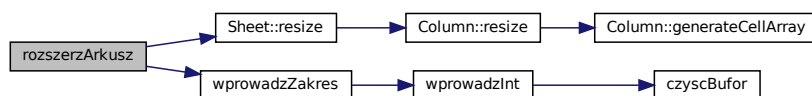
Funkcja modyfikująca rozmiar arkusza.

Interfejs umożliwiający modyfikację rozmiaru istniejącego arkusza.

Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.7 sortowanie()

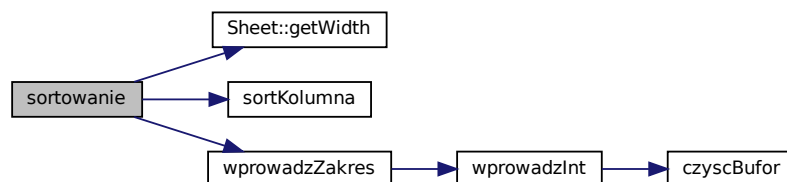
```
void sortowanie (
    Sheet * arkusz )
```

Funkcja interfejsu od sortowania kolumny Metoda zawiera interfejs do sortowania kolumny z arkusza.

Parametry

<code>in, out</code>	<i>Arkusz</i>	którego kolumna będzie sortowana
----------------------	---------------	----------------------------------

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.8 tworzArkusz()

```
Sheet tworzArkusz ( )
```

Funkcja tworząca nową tablicę.

Funkcja zawierająca interfejs umożliwiający tworzenie nowego Arkusza z tablicą dwuwymiarową.

Zwraca

Nowy Arkusz do wykorzystywania w programie

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.9 wczytanie()

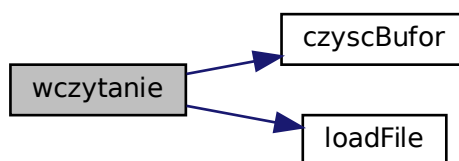
```
void wczytanie (
    Sheet * arkusz )
```

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

Parametry

in, out	arkusz	Arkusz do którego mogą być wczytane elementy
---------	--------	--

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.10 wprowadzWartosc()

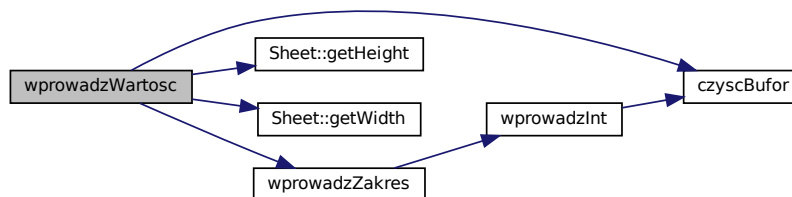
```
void wprowadzWartosc (  
    Sheet * arkusz )
```

wprowadzWartosc modyfikuje wartość komórki Frontend od modyfikacji wartości który w zależności od typu komórki zmieni jej wartość

Parametry

arkusz	Arkusz którego element będzie modyfikowany
--------	--

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.11 zapis()

```
void zapis (
    Sheet arkusz )
```

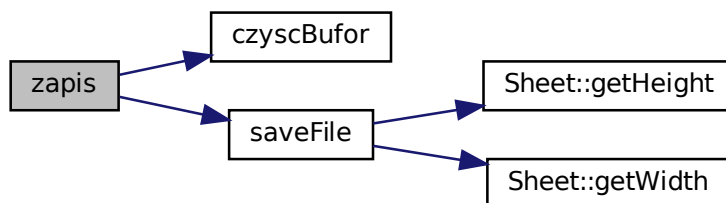
Funkcja menu od zapisu.

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

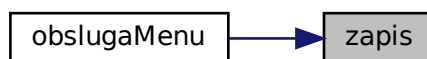
Parametry

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

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.21.1.12 zmienTyp()

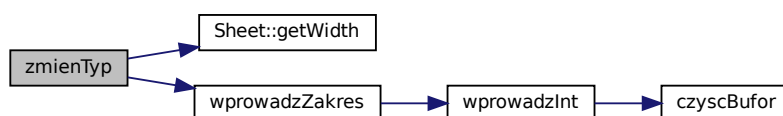
```
void zmienTyp (
    Sheet * arkusz )
```

Funkcja od zmiany typu kolumny Interfejs od metody zmieniającej typ kolumny arkusza.

Parametry

in	arkusz	Arkusz którego kolumna zostaje zmieniona
----	--------	--

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:

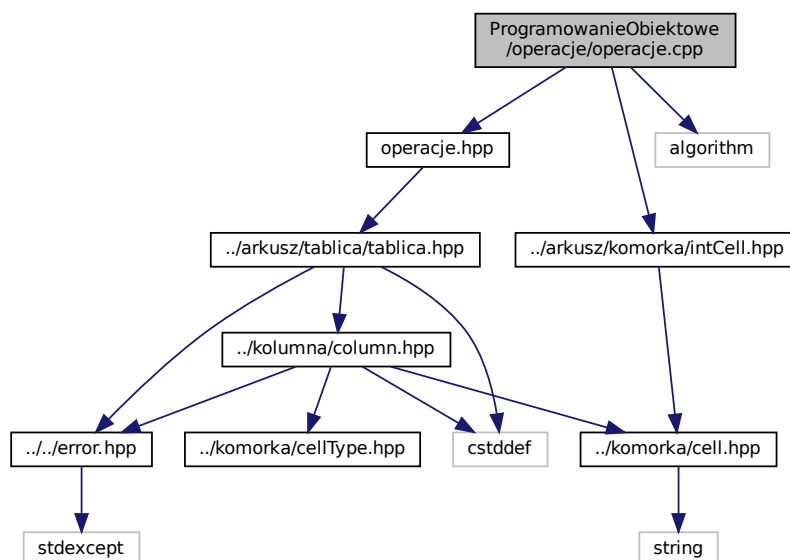


5.22 Dokumentacja pliku

ProgramowanieObiektowe/operacje/operacje.cpp

```
#include "operacje.hpp"
#include "../arkusz/komorka/intCell.hpp"
#include <algorithm>
```

Wykres zależności załączania dla operacje.cpp:



Funkcje

- int [maxWiersz](#) ([Sheet](#) sheet, size_t row)
Funkcja szukania maksymalnej wartości wiersza.
- int [minWiersz](#) ([Sheet](#) sheet, size_t row)
Funkcja szukania minimalnej wartości wiersza.
- int [sumaWiersz](#) ([Sheet](#) sheet, size_t row)
Funkcja licząca sumę elementów wiersza.
- int [maxKolumna](#) ([Column](#) column)

- Funkcja szukania maksymalnej wartości kolumny.*

 - int `minKolumna` (`Column` column)

Funkcja szukania minimalnej wartości kolumny.
- int `sumaKolumna` (`Column` column)

Funkcja licząca sumę elementów kolumny.
- void `sortKolumna` (`Column` *column, bool descending)

sortKolumna Sortuje komórki w kolumnie rosnąco lub malejąco Sortuje kolumnę w zależności od podanego parametru descending
- int `countCalculateableColumns` (`Sheet` sheet)

countCalculateableRow Liczy ilość kolumn obliczalnych Liczy ile jest kolumn które mogą być wykorzystywane przy obliczeniach

5.22.1 Dokumentacja funkcji

5.22.1.1 countCalculateableColumns()

```
int countCalculateableColumns (
    Sheet sheet )
```

`countCalculateableRow` Liczy ilość kolumn obliczalnych Liczy ile jest kolumn które mogą być wykorzystywane przy obliczeniach

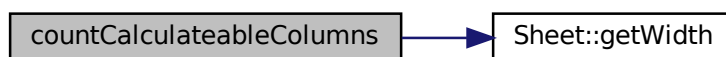
Parametry

<code>sheet</code>	Arkusz którego elementy będą liczone
--------------------	--------------------------------------

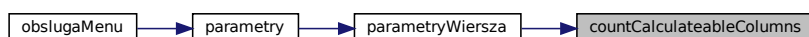
Zwraca

liczba kolumn typów obliczalnych

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.22.1.2 maxKolumna()

```
int maxKolumna (  
    Column column )
```

Funkcja szukania maksymalnej wartości kolumny.

Funkcja przeszukuje całą kolumnę celem określenia największej wprowadzonej wartości

Parametry

in	column	Kolumna względem którego obliczamy parametr
----	--------	---

Zwraca

Zwraca wartość największą kolumny

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.22.1.3 maxWiersz()

```
int maxWiersz (  
    Sheet sheet,  
    size_t row )
```

Funkcja szukania maksymalnej wartości wiersza.

Funkcja przeszukuje cały wiersz celem określenia największej wprowadzonej wartości

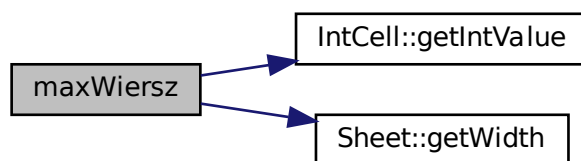
Parametry

in	sheet	Arkusz przeznaczony do obliczania parametru
in	row	Wiersz względem którego obliczamy parametr

Zwraca

Zwraca wartość maksymalną wiersza

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**5.22.1.4 minKolumna()**

```
int minKolumna (
    Column column )
```

Funkcja szukania minimalnej wartości kolumny.

Funkcja przeszukuje całą kolumnę celem określenia najmniejszej wprowadzonej wartości

Parametry

in	column	Kolumna względem którego obliczamy parametr
----	--------	---

Zwraca

Zwraca wartość najmniejszą kolumny

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**5.22.1.5 minWiersz()**

```
int minWiersz (
    Sheet sheet,
    size_t row )
```

Funkcja szukania minimalnej wartości wiersza.

Funkcja przeszukuje cały wiersz celem określenia najmniejszej wprowadzonej wartości

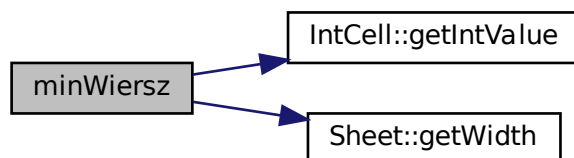
Parametry

in	<i>sheet</i>	Arkusz przeznaczony do obliczania parametru
in	<i>row</i>	Wiersz względem którego obliczamy parametr

Zwraca

Zwraca wartość najmniejszą wiersza

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**5.22.1.6 sortKolumna()**

```

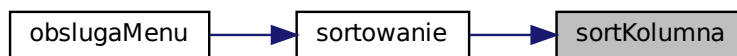
void sortKolumna (
    Column * column,
    bool descending = false )
  
```

`sortKolumna` Sortuje komórki w kolumnie rosnąco lub malejąco Sortuje kolumnę w zależności od podanego parametru `descending`

Parametry

in, out	<i>column</i>	Kolumna przeznaczona do sortowania
in	<i>descending</i>	Definiuje czy kolumna będzie sortowana rosnąco lub malejąco

Oto graf wywoływań tej funkcji:



5.22.1.7 sumaKolumna()

```
int sumaKolumna (
    Column column )
```

Funkcja licząca sumę elementów kolumny.

Funkcja zwraca sumę całej kolumny

Parametry

in	column	Kolumna względem którego obliczamy parametr
----	--------	---

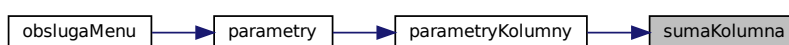
Zwraca

Zwraca sumę wszystkich elementów kolumny

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.22.1.8 sumaWiersz()

```
int sumaWiersz (
    Sheet sheet,
    size_t row )
```

Funkcja licząca sumę elementów wiersza.

Funkcja zwraca sumę całego wiersza

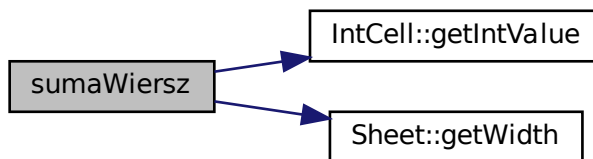
Parametry

in	sheet	Arkusz przeznaczony do obliczania parametru
in	row	Wiersz względem którego obliczamy parametr

Zwraca

Zwraca sumę wszystkich elementów wiersza

Oto graf wywołań dla tej funkcji:



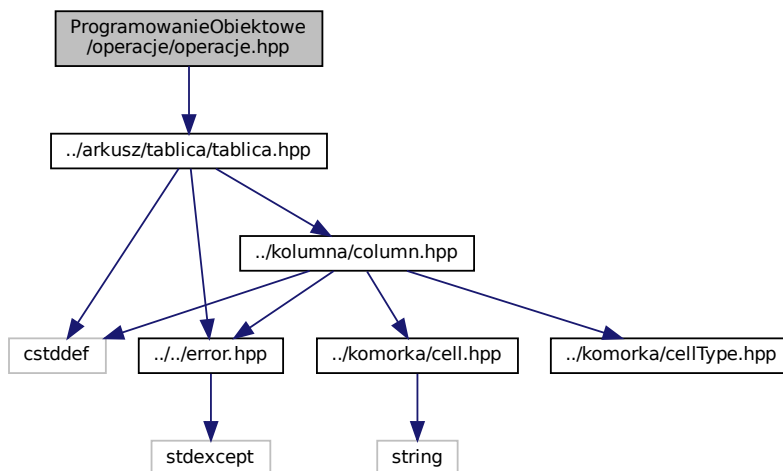
Oto graf wywoływań tej funkcji:



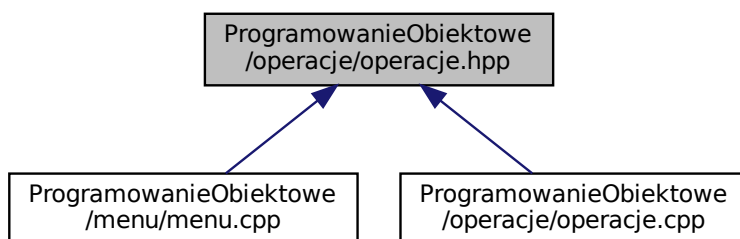
5.23 Dokumentacja pliku ProgramowanieObiektowe/operacje/operacje.hpp

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

Wykres zależności załączania dla operacje.hpp:



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



Funkcje

- int [maxWiersz](#) ([Sheet](#) sheet, size_t row)
Funkcja szukania maksymalnej wartości wiersza.
- int [minWiersz](#) ([Sheet](#) sheet, size_t row)
Funkcja szukania minimalnej wartości wiersza.
- int [sumaWiersz](#) ([Sheet](#) sheet, size_t row)
Funkcja licząca sumę elementów wiersza.
- int [countCalculateableColumns](#) ([Sheet](#) sheet)
countCalculateableRow Liczy ilość kolumn obliczalnych Liczy ile jest kolumn które mogą być wykorzystywane przy obliczeniach
- int [maxKolumna](#) ([Column](#) column)
Funkcja szukania maksymalnej wartości kolumny.
- int [minKolumna](#) ([Column](#) column)

Funkcja szukania minimalnej wartości kolumny.

- int `sumaKolumna` (`Column` column)

Funkcja licząca sumę elementów kolumny.

- void `sortKolumna` (`Column` *column, bool descending=false)

sortKolumna Sortuje komórki w kolumnie rosnąco lub malejąco Sortuje kolumny w zależności od podanego parametru descending

5.23.1 Dokumentacja funkcji

5.23.1.1 countCalculateableColumns()

```
int countCalculateableColumns (
    Sheet sheet )
```

`countCalculateableRow` Liczy ilość kolumn obliczalnych Liczy ile jest kolumn które mogą być wykorzystywane przy obliczeniach

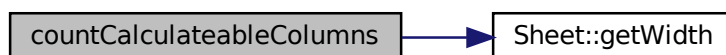
Parametry

<code>sheet</code>	Arkusz którego elementy będą liczone
--------------------	--------------------------------------

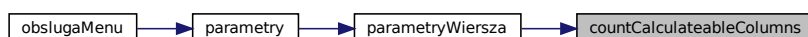
Zwraca

liczba kolumn typów obliczalnych

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.23.1.2 maxKolumna()

```
int maxKolumna (
    Column column )
```

Funkcja szukania maksymalnej wartości kolumny.

Funkcja przeszukuje całą kolumnę celem określenia największej wprowadzonej wartości

Parametry

in	column	Kolumna względem którego obliczamy parametr
----	--------	---

Zwraca

Zwraca wartość największą kolumny

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.23.1.3 maxWiersz()

```
int maxWiersz (
    Sheet sheet,
    size_t row )
```

Funkcja szukania maksymalnej wartości wiersza.

Funkcja przeszukuje cały wiersz celem określenia największej wprowadzonej wartości

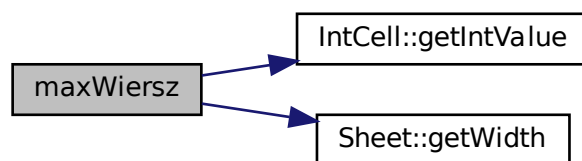
Parametry

in	<i>sheet</i>	Arkusz przeznaczony do obliczania parametru
in	<i>row</i>	Wiersz względem którego obliczamy parametr

Zwraca

Zwraca wartość maksymalną wiersza

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**5.23.1.4 minKolumna()**

```
int minKolumna (
    Column column )
```

Funkcja szukania minimalnej wartości kolumny.

Funkcja przeszukuje całą kolumnę celem określenia najmniejszej wprowadzonej wartości

Parametry

in	<i>column</i>	Kolumna względem którego obliczamy parametr
----	---------------	---

Zwraca

Zwraca wartość najmniejszą kolumny

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**5.23.1.5 minWiersz()**

```
int minWiersz (  
    Sheet sheet,  
    size_t row )
```

Funkcja szukania minimalnej wartości wiersza.

Funkcja przeszukuje cały wiersz celem określenia najmniejszej wprowadzonej wartości

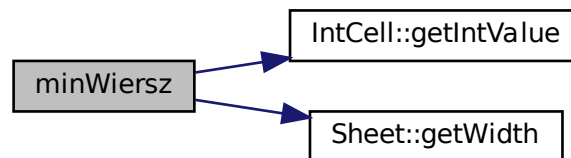
Parametry

in	sheet	Arkusz przeznaczony do obliczania parametru
in	row	Wiersz względem którego obliczamy parametr

Zwraca

Zwraca wartość najmniejszą wiersza

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**5.23.1.6 sortKolumna()**

```

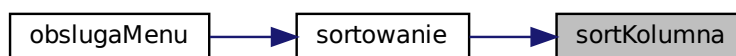
void sortKolumna (
    Column * column,
    bool descending = false )
  
```

`sortKolumna` Sortuje komórki w kolumnie rosnąco lub malejąco Sortuje kolumnę w zależności od podanego parametru `descending`

Parametry

in, out	<i>column</i>	Kolumna przeznaczona do sortowania
in	<i>descending</i>	Definiuje czy kolumna będzie sortowana rosnąco lub malejąco

Oto graf wywoływań tej funkcji:



5.23.1.7 sumaKolumna()

```
int sumaKolumna (
    Column column )
```

Funkcja licząca sumę elementów kolumny.

Funkcja zwraca sumę całej kolumny

Parametry

in	column	Kolumna względem którego obliczamy parametr
----	--------	---

Zwraca

Zwraca sumę wszystkich elementów kolumny

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.23.1.8 sumaWiersz()

```
int sumaWiersz (
    Sheet sheet,
    size_t row )
```

Funkcja licząca sumę elementów wiersza.

Funkcja zwraca sumę całego wiersza

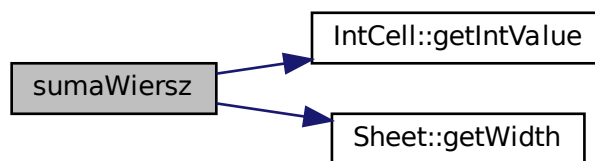
Parametry

in	sheet	Arkusz przeznaczony do obliczania parametru
in	row	Wiersz względem którego obliczamy parametr

Zwraca

Zwraca sumę wszystkich elementów wiersza

Oto graf wywołań dla tej funkcji:



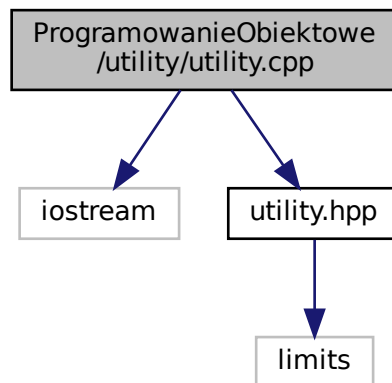
Oto graf wywoływań tej funkcji:



5.24 Dokumentacja pliku ProgramowanieObiektowe/utility/utility.cpp

```
#include <iostream>
#include "utility.hpp"
```

Wykres zależności załączania dla utility.cpp:



Funkcje

- int `wprowadzInt()`
funkcja od wprowadzania wartości typu int
- int `wprowadzZakres(int min, int max)`
Wprowadź wartość z wyznaczonego zakresu w konsoli. Funkcja służy do zwracania wartości wprowadzonej konsolowo, która mieści się wyznaczonym zakresie.
- void `czyszcBufor()`
Funkcja od czyszczenia buforu strumienia CIN.

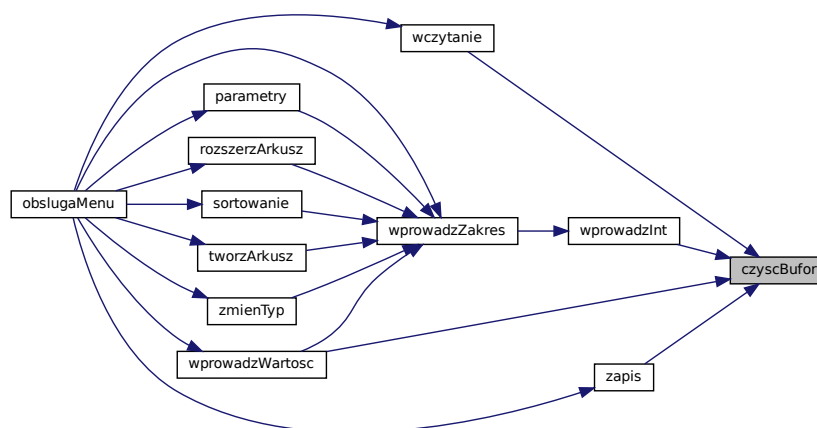
5.24.1 Dokumentacja funkcji

5.24.1.1 czyszcBufor()

```
void czyszcBufor ( )
```

Funkcja od czyszczenia buforu strumienia CIN.

Funkcja ma za zadanie wyczyścić bufor strumienia wejściowego CIN celem wprowadzenia np. string'a Oto graf wywołań tej funkcji:



5.24.1.2 wprowadzZakres()

```

int wprowadzZakres (
    int min = 1,
    int max = std::numeric_limits< int >::max() )

```

Wprowadź wartość z wyznaczonego zakresu w konsoli Funkcja służy do zwracania wartości wprowadzonej konsolowo, która mieści się wyznaczonym zakresie.

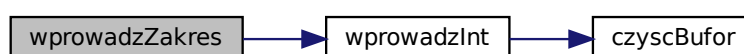
Parametry

in	<i>min</i>	Minimalna wartość jaką można wprowadzić
in	<i>max</i>	Maksymalna wartość jaką można wprowadzić

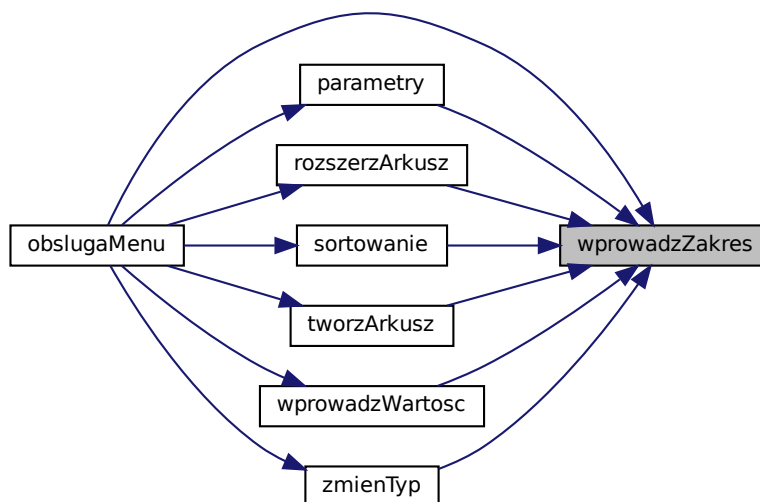
Zwraca

Wartość z zakresu <min; max>

Oto graf wywołań dla tej funkcji:



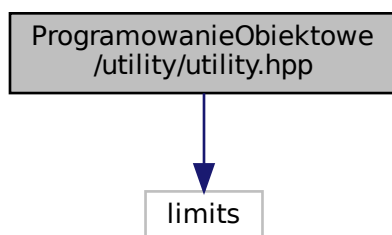
Oto graf wywoływań tej funkcji:



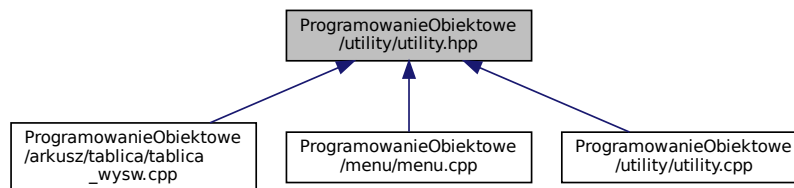
5.25 Dokumentacja pliku ProgramowanieObiektowe/utility/utility.hpp

```
#include <limits>
```

Wykres zależności załączania dla `utility.hpp`:



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



Funkcje

- int `wprowadzInt` ()
funkcja od wprowadzania wartości typu int
- int `wprowadzZakres` (int min=1, int max=std::numeric_limits< int >::max())
Wprowadź wartość z wyznaczonego zakresu w konsoli Funkcja służy do zwracania wartości wprowadzonej konsolowo, która mieści się wyznaczonym zakresie.
- void `czyszcBufor` ()
Funkcja od czyszczenia buforu strumienia CIN.

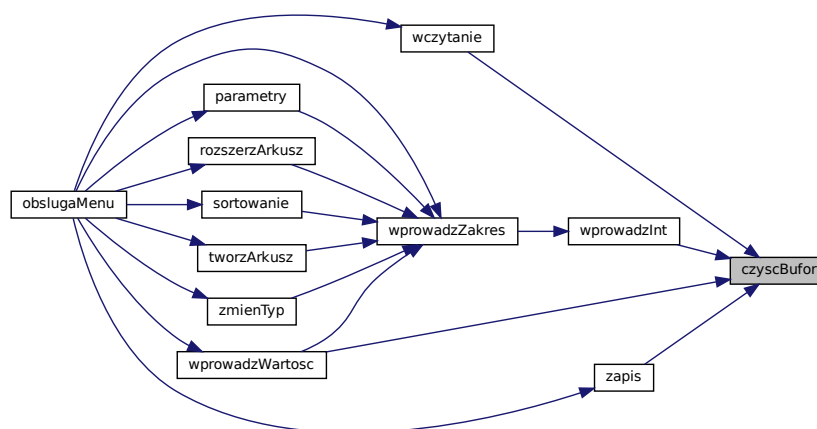
5.25.1 Dokumentacja funkcji

5.25.1.1 czyszcBufor()

```
void czyszcBufor ( )
```

Funkcja od czyszczenia buforu strumienia CIN.

Funkcja ma za zadanie wyczyścić bufor strumienia wejściowego CIN celem wprowadzenia np. string'a Oto graf wywołań tej funkcji:



5.25.1.2 wprowadzZakres()

```
int wprowadzZakres (
    int min = 1,
    int max = std::numeric_limits< int >::max() )
```

Wprowadź wartość z wyznaczonego zakresu w konsoli Funkcja służy do zwracania wartości wprowadzonej konsolowo, która mieści się wyznaczonym zakresie.

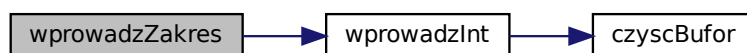
Parametry

in	<i>min</i>	Minimalna wartość jaką można wprowadzić
in	<i>max</i>	Maksymalna wartość jaką można wprowadzić

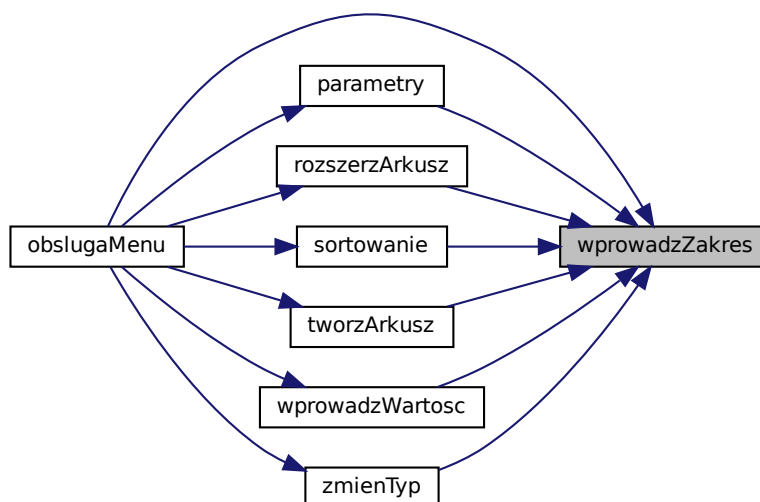
Zwraca

Wartość z zakresu <min; max>

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



Index

BadFileException, 7

begin

Column, 12

Cell, 8

getValue, 10

setValue, 10

CellType

cellType.hpp, 38

cellType.hpp

CellType, 38

IntCell, 38

Column, 10

begin, 12

Column, 12

end, 13

generateCellArray, 13

getCell, 14

getHeight, 14

getType, 14

operator[], 15

resize, 15

countCalculateableColumns

operacje.cpp, 77

operacje.hpp, 85

createColumnArray

Sheet, 24

czyszcBufor

utility.cpp, 92

utility.hpp, 95

DisplaySheet

tablica_wysw.cpp, 43

tablica_wysw.hpp, 45

end

Column, 13

error.hpp

PLIK_ACCESS, 47

PLIK_FORMAT, 47

PLIK_ROZMIAR, 47

TABLICA_SIZE, 47

TABLICA_ZAKR, 47

Wyjatki, 47

generateCellArray

Column, 13

generujMenu

menu.cpp, 54

menu.hpp, 66

getCell

Column, 14

getColumn

Sheet, 25

getHeight

Column, 14

Sheet, 26

getIntValue

IntCell, 20

getType

Column, 14

getValue

Cell, 10

IntCell, 20

StringCell, 31

getWidth

Sheet, 26

IntCell, 18

cellType.hpp, 38

getIntValue, 20

getValue, 20

IntCell, 19

setValue, 20

loadFile

zapisOdczyt.cpp, 48

zapisOdczyt.hpp, 51

maxKolumna

operacje.cpp, 78

operacje.hpp, 85

maxWiersz

operacje.cpp, 78

operacje.hpp, 86

menu.cpp

generujMenu, 54

obsługaMenu, 55

parametry, 56

parametryKolumny, 57

parametryWiersza, 58

rozszerzArkusz, 59

sortowanie, 60

tworzArkusz, 60

wczytanie, 61

wprowadzWartosc, 62

zapis, 63

zmienTyp, 64

menu.hpp

generujMenu, 66

- obsługaMenu, 66
 - parametry, 67
 - parametryKolumny, 68
 - parametryWiersza, 69
 - rozszerzArkusz, 70
 - sortowanie, 71
 - tworzArkusz, 71
 - wczytanie, 72
 - wprowadzWartosc, 73
 - zapis, 74
 - zmienTyp, 75
- minKolumna
 - operacje.cpp, 79
 - operacje.hpp, 87
- minWiersz
 - operacje.cpp, 80
 - operacje.hpp, 88
- NotNumericValue, 21
- obsługaMenu
 - menu.cpp, 55
 - menu.hpp, 66
- operacje.cpp
 - countCalculateableColumns, 77
 - maxKolumna, 78
 - maxWiersz, 78
 - minKolumna, 79
 - minWiersz, 80
 - sortKolumna, 81
 - sumaKolumna, 82
 - sumaWiersz, 82
- operacje.hpp
 - countCalculateableColumns, 85
 - maxKolumna, 85
 - maxWiersz, 86
 - minKolumna, 87
 - minWiersz, 88
 - sortKolumna, 89
 - sumaKolumna, 90
 - sumaWiersz, 90
- operator[]
 - Column, 15
 - Sheet, 27
- parametry
 - menu.cpp, 56
 - menu.hpp, 67
- parametryKolumny
 - menu.cpp, 57
 - menu.hpp, 68
- parametryWiersza
 - menu.cpp, 58
 - menu.hpp, 69
- PLIK_ACCESS
 - error.hpp, 47
- PLIK_FORMAT
 - error.hpp, 47
- PLIK_ROZMIAR
 - error.hpp, 47
- ProgramowanieObiektowe/arkusz/kolumna/column.cpp, 33
- ProgramowanieObiektowe/arkusz/kolumna/column.hpp, 33
- ProgramowanieObiektowe/arkusz/komorka/adapter.cpp, 34
- ProgramowanieObiektowe/arkusz/komorka/adapter.hpp, 35
- ProgramowanieObiektowe/arkusz/komorka/cell.cpp, 36
- ProgramowanieObiektowe/arkusz/komorka/cell.hpp, 36
- ProgramowanieObiektowe/arkusz/komorka/cellType.hpp, 37
- ProgramowanieObiektowe/arkusz/komorka/intCell.cpp, 38
- ProgramowanieObiektowe/arkusz/komorka/intCell.hpp, 39
- ProgramowanieObiektowe/arkusz/komorka/stringCell.cpp, 39
- ProgramowanieObiektowe/arkusz/komorka/stringCell.hpp, 40
- ProgramowanieObiektowe/arkusz/tablica/tablica.cpp, 41
- ProgramowanieObiektowe/arkusz/tablica/tablica.hpp, 41
- ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.cpp, 42
- ProgramowanieObiektowe/arkusz/tablica/tablica_wysw.hpp, 44
- ProgramowanieObiektowe/error.hpp, 46
- ProgramowanieObiektowe/io/zapisOdczyt.cpp, 48
- ProgramowanieObiektowe/io/zapisOdczyt.hpp, 50
- ProgramowanieObiektowe/main.cpp, 53
- ProgramowanieObiektowe/menu/menu.cpp, 53
- ProgramowanieObiektowe/menu/menu.hpp, 65
- ProgramowanieObiektowe/operacje/operacje.cpp, 76
- ProgramowanieObiektowe/operacje/operacje.hpp, 83
- ProgramowanieObiektowe/utility/utility.cpp, 91
- ProgramowanieObiektowe/utility/utility.hpp, 94
- resize
 - Column, 15
 - Sheet, 28
- rozszerzArkusz
 - menu.cpp, 59
 - menu.hpp, 70
- saveFile
 - zapisOdczyt.cpp, 49
 - zapisOdczyt.hpp, 52
- setValue
 - Cell, 10
 - IntCell, 20
 - StringCell, 31
- Sheet, 22
 - createColumnArray, 24
 - getColumn, 25
 - getHeight, 26
 - getWidth, 26
 - operator[], 27

- resize, [28](#)
- Sheet, [23](#)
- sortKolumna
 - operacje.cpp, [81](#)
 - operacje.hpp, [89](#)
- sortowanie
 - menu.cpp, [60](#)
 - menu.hpp, [71](#)
- StringCell, [29](#)
 - getValue, [31](#)
 - setValue, [31](#)
 - StringCell, [30](#)
- sumaKolumna
 - operacje.cpp, [82](#)
 - operacje.hpp, [90](#)
- sumaWiersz
 - operacje.cpp, [82](#)
 - operacje.hpp, [90](#)
- TABLICA_SIZE
 - error.hpp, [47](#)
- tablica_wysw.cpp
 - DisplaySheet, [43](#)
- tablica_wysw.hpp
 - DisplaySheet, [45](#)
- TABLICA_ZAKR
 - error.hpp, [47](#)
- tworzArkusz
 - menu.cpp, [60](#)
 - menu.hpp, [71](#)
- utility.cpp
 - czyscBufor, [92](#)
 - wprowadzZakres, [93](#)
- utility.hpp
 - czyscBufor, [95](#)
 - wprowadzZakres, [95](#)
- wczytanie
 - menu.cpp, [61](#)
 - menu.hpp, [72](#)
- wprowadzWartosc
 - menu.cpp, [62](#)
 - menu.hpp, [73](#)
- wprowadzZakres
 - utility.cpp, [93](#)
 - utility.hpp, [95](#)
- Wyjatk
 - error.hpp, [47](#)
- zapis
 - menu.cpp, [63](#)
 - menu.hpp, [74](#)
- zapisOdczyt.cpp
 - loadFile, [48](#)
 - saveFile, [49](#)
- zapisOdczyt.hpp
 - loadFile, [51](#)
 - saveFile, [52](#)
- zmienTyp
 - menu.cpp, [64](#)
 - menu.hpp, [75](#)
