

OwnExcel

Generated by Doxygen 1.8.17



<b>1 Programowanie Obiektowe</b>	<b>1</b>
1.0.0.1 Projekt arkusza kalkulacyjnego	1
1.1 Interfejs użytkownika	1
1.2 Operacyjna	1
1.2.0.1 obliczenia i zmiany rozmiaru tablicy:	1
1.2.0.2 prezentacja oraz zapisywanie:	2
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Array Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 Array()	6
3.1.3 Member Function Documentation	6
3.1.3.1 average()	6
3.1.3.2 changeValue()	6
3.1.3.3 columns()	8
3.1.3.4 divide()	8
3.1.3.5 getMaxValue()	8
3.1.3.6 getNumberAsString()	9
3.1.3.7 getNumberFromSheet()	9
3.1.3.8 loadDataFromFile()	9
3.1.3.9 multiplication()	9
3.1.3.10 resizeSheet()	10
3.1.3.11 rows()	10
3.1.3.12 saveDataToFile()	10
3.1.3.13 sheetAsChars()	11
3.1.3.14 subtract()	11
3.1.3.15 sum()	11
3.2 ArrayDisplay Class Reference	12
3.2.1 Detailed Description	12
3.2.2 Member Function Documentation	12
3.2.2.1 Display()	12
3.3 Identifier Class Reference	12
3.4 Menu Class Reference	13
3.4.1 Detailed Description	13
3.4.2 Member Function Documentation	13
3.4.2.1 alert()	13
3.4.2.2 clear()	14
3.4.2.3 getch()	14
3.4.2.4 getIdentifier()	14

3.4.2.5 <code>getNumber()</code> . . . . .	15
3.4.2.6 <code>message()</code> . . . . .	15
3.4.2.7 <code>showFunctions()</code> . . . . .	15
3.5 OperationHandler Class Reference . . . . .	16
3.5.1 Detailed Description . . . . .	16
3.5.2 Member Function Documentation . . . . .	16
3.5.2.1 <code>decimalOperation()</code> . . . . .	16
<b>Index</b>	<b>17</b>

## Chapter 1

# Programowanie Obiektowe

### 1.0.0.1 Projekt arkusza kalkulacyjnego

Prosty arkusz kalkulacyjny obsługiwany w konsoli.

## 1.1 Interfejs użytkownika

- [ ] Wyświetlanie informacji na temat funkcji
- [ ] prezentacja arkusza
- [ ] możliwość zmiany rozmiaru tablicy
- [ ] możliwość zmiany danych w arkuszu
- [ ] możliwość wykonywania operacji na arkuszu
- [ ] możliwość wrócenia do menu

## 1.2 Operacyjna

### 1.2.0.1 obliczenia i zmiany rozmiaru tablicy:

- [ ] wprowadzanie danych do arkusza
- [ ] dodawanie komórek
- [ ] odejmowanie komórek
- [ ] mnożenie komórek
- [ ] dzielenie komórek
- [ ] średnia z komórek
- [ ] zapisywanie wyniku do konkretnej komórki
- [ ] zmiana rozmiaru tablicy bez utraty danych

**1.2.0.2 prezentacja oraz zapisywanie:**

- [ ] prezentacja tablicy
- [ ] zapisywanie wyniku do konkretnej komórki
- [ ] odczytywanie danych z pliku
- [ ] zapisywanie danych do pliku

## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Array</a>	This is a class handling all operations on sheet . . . . .	5
<a href="#">ArrayDisplay</a>	This is a class handling displaying sheet . . . . .	12
<a href="#">Identifier</a>	. . . . .	12
<a href="#">Menu</a>	This is a class handling menu operations . . . . .	13
<a href="#">OperationHandler</a>	This is a class handling menu operations . . . . .	16





## Chapter 3

# Class Documentation

### 3.1 Array Class Reference

This is a class handling all operations on sheet.

```
#include <array.h>
```

#### Public Member Functions

- [Array](#) (int [columns](#), int [rows](#))  
*Constructor of class.*
- int [columns](#) ()
- int [rows](#) ()
- char \* [sheetAsChars](#) ()
- float [sum](#) ([Identifier](#) \*identifiers, int length)  
*func that return sum of values depends on each one position in sheet and returns result.*
- float [subtract](#) ([Identifier](#) minued, [Identifier](#) subtrahend)  
*func that subtract two values from sheet and returns result.*
- float [divide](#) ([Identifier](#) divisor, [Identifier](#) dividend)  
*func that divide two values from sheet and returns result.*
- float [multiplication](#) ([Identifier](#) \*identifiers, int length)  
*func that return multiplication of values depends on each one position in sheet and returns result.*
- float [average](#) ([Identifier](#) \*identifiers, int length)  
*func that return average value of values depends on each one position in sheet and returns result.*
- void [changeValue](#) ([Identifier](#) identifier, float value)  
*func that change value single position in sheet and return true or false depends on result.*
- void [resizeSheet](#) (int [columns](#), int [rows](#))  
*func that resize sheet with no data lose.*
- float [getNumberFromSheet](#) ([Identifier](#) identifier)  
*func that return single value from sheet depends on it position.*
- std::string [getNumberAsString](#) ([Identifier](#) identifier)
- void [saveDataToFile](#) ()  
*func that save data from array to file (NOT IMPLEMENTED)*
- void [loadDataFromFile](#) ()  
*func that load data from file (NOT IMPLEMENTED)*
- float [getMaxValue](#) ()

### 3.1.1 Detailed Description

This is a class handling all operations on sheet.

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 Array()

```
Array::Array (
    int columns,
    int rows ) [inline]
```

Constructor of class.

Just in time of creating array object constructor configure sheet variable to proper value of rows and columns. Construction also configure columns and rows properties.

### 3.1.3 Member Function Documentation

#### 3.1.3.1 average()

```
float Array::average (
    Identifier * identifiers,
    int length )
```

func that return average value of values depends on each one position in sheet and returns result.

##### Parameters

in	<i>identifiers</i>	- array that holds position of cells.
in	<i>length</i>	- length of array.

##### Returns

- float value

#### 3.1.3.2 changeValue()

```
void Array::changeValue (
    Identifier identifier,
    float value )
```

func that change value single position in sheet and return true or false depends on result.

**Parameters**

in	<i>identifier</i>	- position of cell that gonna be changed.
in	<i>value</i>	- float value to write up.

**Returns**

- nothing or exception

**3.1.3.3 columns()**

```
int Array::columns ( )
```

**Returns**

numbers of columns

**3.1.3.4 divide()**

```
float Array::divide (
    Identifier divisor,
    Identifier dividend )
```

func that divide two values from sheet and returns result.

**Parameters**

in	<i>divisor</i>	- first number identifier
in	<i>dividend</i>	- second number identifier

**Returns**

- float value or exception.

**3.1.3.5 getMaxValue()**

```
float Array::getMaxValue ( )
```

func search max value stored in sheet.

**Returns**

- max value as float

### 3.1.3.6 `getNumberAsString()`

```
std::string Array::getNumberAsString (
    Identifier identifier )
```

#### Parameters

<i>identifier</i>	- position of cell that value func gonna return.
-------------------	--

#### Returns

number as string value

### 3.1.3.7 `getNumberFromSheet()`

```
float Array::getNumberFromSheet (
    Identifier identifier )
```

func that return single value from sheet depends on it position.

#### Parameters

<i>identifier</i>	- position of cell that value func gonna return.
-------------------	--

#### Returns

return nothing or exception

### 3.1.3.8 `loadDataFromFile()`

```
void Array::loadDataFromFile ( )
```

func that load data from file (NOT IMPLEMENTED)

#### Returns

- nothing

### 3.1.3.9 `multiplication()`

```
float Array::multiplication (
    Identifier * identifiers,
    int length )
```

func that return multiplication of values depends on each one position in sheet and returns result.

**Parameters**

in	<i>identifiers</i>	- array that holds position of cells.
in	<i>length</i>	- length of array.

**Returns**

- float value

**3.1.3.10   resizeSheet()**

```
void Array::resizeSheet (
    int columns,
    int rows )
```

func that resize sheet with no data lose.

**Parameters**

in	<i>columns</i>	- new numbers of columns in sheet.
in	<i>rows</i>	- new numbers of rows in sheet.

**Returns**

return nothing or exception

**3.1.3.11   rows()**

```
int Array::rows ( )
```

**Returns**

number of rows

**3.1.3.12   saveDataToFile()**

```
void Array::saveDataToFile ( )
```

func that save data from array to file (NOT IMPLEMENTED)

**Returns**

- nothing

### 3.1.3.13 sheetAsChars()

```
char* Array::sheetAsChars ( )
```

#### Returns

numbers in sheet in single dimension array of chars

### 3.1.3.14 subtract()

```
float Array::subtract (
    Identifier minued,
    Identifier subtrahend )
```

func that subtract two values from sheet and returns result.

#### Parameters

in	<i>minued</i>	- first number identifier
in	<i>subtrahend</i>	- second number identifier

#### Returns

- float value or exception

### 3.1.3.15 sum()

```
float Array::sum (
    Identifier * identifiers,
    int length )
```

func that return sum of values depends on each one position in sheet and returns result.

#### Parameters

in	<i>identifiers</i>	- array that holds position of cells.
in	<i>length</i>	- holds length of identifiers array

#### Returns

- float value or exception

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

- Header Files/array.h
- Source Files/array.cpp

## 3.2 ArrayDisplay Class Reference

This is a class handling displaying sheet.

```
#include <array_display.h>
```

### Public Member Functions

- void [Display](#) ([Array](#) arr)  
*func that display sheet.*
- template<typename T >  
std::string [to\\_string\\_with\\_precision](#) (const T a\_value, const int n=6)

### 3.2.1 Detailed Description

This is a class handling displaying sheet.

### 3.2.2 Member Function Documentation

#### 3.2.2.1 Display()

```
void ArrayDisplay::Display (  
    Array arr )
```

func that display sheet.

#### Parameters

in	<a href="#">Array</a>	- 2D array that holds sheet.
----	-----------------------	------------------------------

#### Returns

- void or exception

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

- Header Files/array\_display.h
- Source Files/array\_display.cpp

## 3.3 Identifier Class Reference

### Public Member Functions

- [Identifier](#) (int column, int row)



## Public Attributes

- int **Column**
- int **Row**

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

- Models/Identifier.h

## 3.4 Menu Class Reference

This is a class handling menu operations.

```
#include <menu.h>
```

### Static Public Member Functions

- static Operations [showFunctions](#) ()  
*func that get from user selected operation.*
- static [Identifier getIdentifier](#) (const std::string &message)  
*func that get cell identifier from user.*
- static float [getNumber](#) (const std::string &message)  
*func that get number from user.*
- static void [message](#) (const std::string &message)  
*func that send message to user.*
- static void [alert](#) (const std::string &message)  
*func that send alert message to user.*
- static int [getch](#) ()  
*func that getch char without waiting for enter source: <https://cboard.cprogramming.com/faq-board/27714-faq-there-getch-conio-equivalent-linux-unix.html>*
- static void [clear](#) ()  
*func that clear console*

### 3.4.1 Detailed Description

This is a class handling menu operations.

### 3.4.2 Member Function Documentation

#### 3.4.2.1 alert()

```
void Menu::alert (
    const std::string & message ) [static]
```

func that send alert message to user.

**Parameters**

in	<i>message</i>	- message to user.
----	----------------	--------------------

**Returns**

return void

**3.4.2.2 clear()**

```
void Menu::clear ( ) [static]
```

func that clear console

**Returns**

void

**3.4.2.3 getch()**

```
int Menu::getch ( ) [static]
```

func that getch char without waiting for enter source: <https://cboard.cprogramming.com/faq-board/27714-faq-there-getch-conio-equivalent-linux-unix.html>

**Returns**

return int value of char

**3.4.2.4 getIdentifier()**

```
Identifier Menu::getIdentifier (
    const std::string & message ) [static]
```

func that get cell identifier from user.

**Parameters**

in	<i>message</i>	- message to user.
----	----------------	--------------------

**Returns**

return [Identifier](#) of cell

**3.4.2.5 getNumber()**

```
float Menu::getNumber (
    const std::string & message ) [static]
```

func that get number from user.

**Parameters**

in	<i>message</i>	- message to user.
----	----------------	--------------------

**Returns**

return float value

**3.4.2.6 message()**

```
void Menu::message (
    const std::string & message ) [static]
```

func that send message to user.

**Parameters**

in	<i>message</i>	- message to user.
----	----------------	--------------------

**Returns**

return void

**3.4.2.7 showFunctions()**

```
Operations Menu::showFunctions ( ) [static]
```

func that get from user selected operation.

**Returns**

return operation enum

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

- Header Files/menu.h
- Source Files/menu.cpp

## 3.5 OperationHandler Class Reference

This is a class handling menu operations.

```
#include <operation_handler.h>
```

### Static Public Member Functions

- static int [decimalOperation](#) (Operations operation, [Array](#) \*arr)  
*static func that handling decimal operations.*

#### 3.5.1 Detailed Description

This is a class handling menu operations.

#### 3.5.2 Member Function Documentation

##### 3.5.2.1 decimalOperation()

```
int OperationHandler::decimalOperation (
    Operations operation,
    Array * arr ) [static]
```

static func that handling decimal operations.

##### Parameters

in, out	<i>arr</i>	- 2D array that holds sheet.
in	<i>operation</i>	- enum instance that holds user operation.

##### Returns

- -1 if exit, 0 if everything is okey.

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

- Header Files/operation\_handler.h
- Source Files/operation\_handler.cpp

# Index

- alert
  - Menu, [13](#)
- Array, [5](#)
  - Array, [6](#)
  - average, [6](#)
  - changeValue, [6](#)
  - columns, [8](#)
  - divide, [8](#)
  - getMaxValue, [8](#)
  - getNumberAsString, [8](#)
  - getNumberFromSheet, [9](#)
  - loadDataFromFile, [9](#)
  - multiplication, [9](#)
  - resizeSheet, [10](#)
  - rows, [10](#)
  - saveDataToFile, [10](#)
  - sheetAsChars, [10](#)
  - subtract, [11](#)
  - sum, [11](#)
- ArrayDisplay, [12](#)
  - Display, [12](#)
- average
  - Array, [6](#)
- changeValue
  - Array, [6](#)
- clear
  - Menu, [14](#)
- columns
  - Array, [8](#)
- decimalOperation
  - OperationHandler, [16](#)
- Display
  - ArrayDisplay, [12](#)
- divide
  - Array, [8](#)
- getch
  - Menu, [14](#)
- getIdentifier
  - Menu, [14](#)
- getMaxValue
  - Array, [8](#)
- getNumber
  - Menu, [15](#)
- getNumberAsString
  - Array, [8](#)
- getNumberFromSheet
  - Array, [9](#)
- Identifier, [12](#)
- loadDataFromFile
  - Array, [9](#)
- Menu, [13](#)
  - alert, [13](#)
  - clear, [14](#)
  - getch, [14](#)
  - getIdentifier, [14](#)
  - getNumber, [15](#)
  - message, [15](#)
  - showFunctions, [15](#)
- message
  - Menu, [15](#)
- multiplication
  - Array, [9](#)
- OperationHandler, [16](#)
  - decimalOperation, [16](#)
- resizeSheet
  - Array, [10](#)
- rows
  - Array, [10](#)
- saveDataToFile
  - Array, [10](#)
- sheetAsChars
  - Array, [10](#)
- showFunctions
  - Menu, [15](#)
- subtract
  - Array, [11](#)
- sum
  - Array, [11](#)