

## My Project

Generated by Doxygen 1.8.17



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Table</a>	Main table class . . . . .	??
-----------------------	----------------------------	----



## Chapter 2

# Class Documentation

### 2.1 Table Class Reference

Main table class.

```
#include <table.h>
```

#### Public Member Functions

- **Table** (int x, int y)
- int [tab\\_build](#) (int nx, int ny)  
*asks tab\_create to build table*
- int [tab\\_remove](#) ()  
*removes table.*
- int [tab\\_editcell](#) (int axis\_x, int axis\_y, int value)  
*edits chosen by user cell*
- int [tab\\_change](#) (int nx, int ny)  
*adds new data to existing table*
- int [tab\\_show](#) ()  
*Shows content of table.*
- int [save](#) (string name)  
*saves from file to [Table](#) class*
- int [read](#) (string name)  
*takes info from existing table and saves it into file*

#### Private Member Functions

- int [tab\\_create](#) ()  
*creates new table*

#### Private Attributes

- int **X**
- int **Y**
- float \*\* **tab**

### 2.1.1 Detailed Description

Main table class.

this class stores information about axis X and axis Y. It also stores table content. inside of this class are also functions responsible for main functions of program.

**Parameters**

<code>in, out</code>	<code>X</code>	- Axis X lenght
<code>in, out</code>	<code>Y</code>	- Axis Y lenght

**2.1.2 Member Function Documentation****2.1.2.1 read()**

```
int Table::read (
    string name )
```

takes info from existing table and saves it into file

**Parameters**

<code>out</code>	<code>name</code>	- String storing saving file name.
------------------	-------------------	------------------------------------

**2.1.2.2 save()**

```
int Table::save (
    string name )
```

saves from file to [Table](#) class

**Parameters**

<code>out</code>	<code>name</code>	- String storing file name choosed to read from.
------------------	-------------------	--

**2.1.2.3 tab\_build()**

```
int Table::tab_build (
    int nx,
    int ny )
```

asks `tab_create` to build table

Takes X and Y lenght. checks if they are in correct format. then checks if there is already table created, if yes deletes it then asks `tab_create` to create table from checked values X and Y.

**Parameters**

in	<i>nx</i>	- Axis X lenght for new table
in	<i>ny</i>	- Axis Y lenght for new table

**2.1.2.4 tab\_change()**

```
int Table::tab_change (
    int nx,
    int ny )
```

adds new data to existing table

**Parameters**

in	<i>nx</i>	- new column number
in	<i>ny</i>	- new row number

•

**2.1.2.5 tab\_create()**

```
int Table::tab_create ( ) [private]
```

creates new table

creates new table from scratch. running this function is nessecary before running functions working on table content. those operation require table which is created by tab\_create.

**2.1.2.6 tab\_editcell()**

```
int Table::tab_editcell (
    int axis_x,
    int axis_y,
    int value )
```

edits chosen by user cell

**Parameters**

in	<i>axis_x</i>	- column number for changing
in	<i>axis_y</i>	- row number for changing
in	<i>value</i>	- value for replacing



### 2.1.2.7 `tab_remove()`

```
int Table::tab_remove ( )
```

removes table.

Removes table content and X and Y values;

### 2.1.2.8 `tab_show()`

```
int Table::tab_show ( )
```

Shows content of table.

outputs to console content of existing table.

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

- `table.h`
- `file.cpp`
- `table.cpp`

