

My Project

Создано системой Doxygen 1.8.8

Чт 2 Июн 2016 21:37:17

Оглавление

1	Иерархический список классов	1
1.1	Иерархия классов	1
2	Алфавитный указатель классов	3
2.1	Классы	3
3	Классы	5
3.1	Класс <code>Api</code>	5
3.1.1	Подробное описание	5
3.1.2	Методы	5
3.1.2.1	<code>change_field</code>	5
3.1.2.2	<code>initialize_field</code>	6
3.1.2.3	<code>initialize_settings</code>	6
3.1.2.4	<code>load_field</code>	6
3.1.2.5	<code>print_field</code>	6
3.1.2.6	<code>save_field</code>	6
3.1.2.7	<code>set_cell</code>	6
3.2	Класс <code>Cell</code>	7
3.2.1	Подробное описание	7
3.2.2	Конструктор(ы)	7
3.2.2.1	<code>Cell</code>	7
3.2.3	Методы	8
3.2.3.1	<code>check</code>	8
3.2.3.2	<code>get_status</code>	8
3.2.3.3	<code>get_x</code>	8
3.2.3.4	<code>get_y</code>	8
3.2.3.5	<code>search_living</code>	8
3.2.3.6	<code>set_status</code>	8
3.2.3.7	<code>set_x</code>	9
3.2.3.8	<code>set_y</code>	9
3.3	Класс <code>console_ui</code>	9
3.3.1	Подробное описание	9

3.3.2	Методы	9
3.3.2.1	input_menu	9
3.3.2.2	secondary_menu	10
3.4	Класс exit_window	10
3.5	Класс Field	10
3.5.1	Подробное описание	11
3.5.2	Конструктор(ы)	11
3.5.2.1	Field	11
3.5.3	Методы	11
3.5.3.1	get_cell	11
3.5.3.2	get_cells	11
3.5.3.3	get_x	11
3.5.3.4	get_y	12
3.5.3.5	set_cell	12
3.5.3.6	set_cells	12
3.5.3.7	set_x	12
3.5.3.8	set_y	12
3.6	Класс field_window	12
3.7	Класс figure_window	13
3.8	Класс initialize_window	13
3.9	Класс MainWindow	13
3.10	Структура qt_meta_stringdata_exit_window_t	14
3.11	Структура qt_meta_stringdata_field_window_t	14
3.12	Структура qt_meta_stringdata_figure_window_t	14
3.13	Структура qt_meta_stringdata_initialize_window_t	14
3.14	Структура qt_meta_stringdata_MainWindow_t	15
3.15	Структура qt_meta_stringdata_size_window_t	15
3.16	Класс size_window	15
3.17	Класс TestTest	15
Алфавитный указатель		16

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Api	5
Cell	7
console_ui	9
Field	10
QDialog	
exit_window	10
figure_window	13
initialize_window	13
QObject	
TestTest	15
qt_meta_stringdata_exit_window_t	14
qt_meta_stringdata_field_window_t	14
qt_meta_stringdata_figure_window_t	14
qt_meta_stringdata_initialize_window_t	14
qt_meta_stringdata_MainWindow_t	15
qt_meta_stringdata_size_window_t	15
QWidget	
field_window	12
MainWindow	13
size_window	15

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Api	
Класс Api	5
Cell	
Класс Клетка	7
console_ui	
Класс консольного меню	9
exit_window	10
Field	
Класс Поле	10
field_window	12
figure_window	13
initialize_window	13
MainWindow	13
qt_meta_stringdata_exit_window_t	14
qt_meta_stringdata_field_window_t	14
qt_meta_stringdata_figure_window_t	14
qt_meta_stringdata_initialize_window_t	14
qt_meta_stringdata_MainWindow_t	15
qt_meta_stringdata_size_window_t	15
size_window	15
TestTest	15

Глава 3

Классы

3.1 Класс Api

Класс `Api`.

```
#include <api.h>
```

Открытые члены

- void `initialize_settings` (`Field` &f, int f_x, int f_y)
Задать размеры поля
- void `initialize_field` (`Field` &f)
Инициализировать поле случайными клетками
- void `save_field` (`Field` &f)
Сохранить поле в файл
- void `load_field` (`Field` &f)
Загрузить поле из файла
- void `print_field` (`Field` &f)
Вывести поле в консоль
- void `change_field` (`Field` &f)
Сменить поколение и изменить поле
- void `set_cell` (`Field` &f, int x, int y, std::string status)
Задать клетку поля

3.1.1 Подробное описание

Класс `Api`.

Этот класс предоставляет методы ядра.

3.1.2 Методы

3.1.2.1 void `Api::change_field` (`Field` & f)

Сменить поколение и изменить поле

Аргументы

f	поле
---	------

3.1.2.2 void Api::initialize_field (Field & f)

Инициализировать поле случайными клетками

Аргументы

f	поле
---	------

3.1.2.3 void Api::initialize_settings (Field & f, int f_x, int f_y)

Задать размеры поля

Аргументы

f	поле
f_x	размер поля по оси x
f_y	размер поля по оси y

3.1.2.4 void Api::load_field (Field & f)

Загрузить поле из файла

Аргументы

f	поле
---	------

3.1.2.5 void Api::print_field (Field & f)

Вывести поле в консоль

Аргументы

f	поле
---	------

3.1.2.6 void Api::save_field (Field & f)

Сохранить поле в файл

Аргументы

f	поле
---	------

3.1.2.7 void Api::set_cell (Field & f, int x, int y, std::string status)

Задать клетку поля

Аргументы

f	поле
x	координата клетки по оси x
y	координата клетки по оси y
status	статус клетки

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/core/api.h
- /home/user/LivingCells/sources/livingcells/core/api.cpp

3.2 Класс Cell

Класс Клетка

```
#include <cell.h>
```

Открытые члены

- `Cell` (const int a=0, const int b=0, const bool st=false)
Конструктор
- int `get_x` ()
Получить координату по оси x.
- int `get_y` ()
Получить координату по оси y.
- bool `get_status` ()
Получить состояние клетки
- void `set_x` (const int a)
Установить значение поля x, равное координате по оси x.
- void `set_y` (const int b)
Установить значение поля y, равное координате по оси y.
- void `set_status` (const bool st)
Установить значение поля status, равное состоянию клетки
- int `search_living` (const std::vector< std::vector< `Cell` > > c)
Подсчёт количества соседних живых клеток
- int `check` (`Cell`)
Проверка состояние клетки

3.2.1 Подробное описание

Класс Клетка

Этот класс моделирует объект клетка из которых будет состоять поле.

3.2.2 Конструктор(ы)

3.2.2.1 `Cell::Cell (const int a = 0, const int b = 0, const bool st = false) [inline]`

Конструктор

Аргументы

a	координата по оси x
b	координата по оси y
st	состояние клетки

3.2.3 Методы

3.2.3.1 `int Cell::check (Cell c)`

Проверка состояние клетки

Аргументы

проверяемая	клетка
-------------	--------

3.2.3.2 `bool Cell::get_status () [inline]`

Получить состояние клетки

Возвращает

состояние клетки

3.2.3.3 `int Cell::get_x () [inline]`

Получить координату по оси x.

Возвращает

координата по оси x

3.2.3.4 `int Cell::get_y () [inline]`

Получить координату по оси y.

Возвращает

координата по оси y

3.2.3.5 `int Cell::search_living (const std::vector< std::vector< Cell > > c)`

Подсчёт количества соседних живых клеток

Аргументы

c	двумерный массив клеток
---	-------------------------

3.2.3.6 `void Cell::set_status (const bool st) [inline]`

Установить значение поля status, равное состоянию клетки

Аргументы

st	статус клетки
----	---------------

3.2.3.7 void Cell::set_x (const int a) [inline]

Установить значение поля x, равное координате по оси x.

Аргументы

a	координата по оси x
---	---------------------

3.2.3.8 void Cell::set_y (const int b) [inline]

Установить значение поля y, равное координате по оси y.

Аргументы

b	координата по оси y
---	---------------------

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/core/cell.h
- /home/user/LivingCells/sources/livingcells/core/cell.cpp

3.3 Класс console_ui

Класс консольного меню

```
#include <console_ui.h>
```

Открытые члены

- void [print_menu](#) ()
Вывести главное меню
- void [secondary_menu](#) (Field &f, [Api](#) a)
Вывести промежуточное меню
- void [input_menu](#) (Field &f, [Api](#) a, int x, int y)
Вывести меню задания клеток поля

3.3.1 Подробное описание

Класс консольного меню

Этот класс, посредством консоли, взаимодействует с пользователем.

3.3.2 Методы

3.3.2.1 void console_ui::input_menu (Field &f, [Api](#) a, int x, int y)

Вывести меню задания клеток поля

Аргументы

f	поле
a	объект класса Api
x	размер поля по оси x
y	размер поля по оси y

3.3.2.2 void console_ui::secondary_menu (Field & f, Api a)

Вывести промежуточное меню

Аргументы

f	поле
a	объект класса Api

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/app/console_ui.h
- /home/user/LivingCells/sources/livingcells/app/console_ui.cpp

3.4 Класс exit_window

Граф наследования:exit_window:

Граф связей класса exit_window:

Открытые члены

- exit_window (QWidget *parent)

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/gui/exit_window.h
- /home/user/LivingCells/sources/livingcells/gui/exit_window.cpp

3.5 Класс Field

Класс Поле

```
#include <field.h>
```

Открытые члены

- [Field](#) (const int a=0, const int b=0)
Конструктор
- int [get_x](#) ()
Получить размер поля по оси x.
- int [get_y](#) ()
Получить размер поля по оси y.
- [Cell](#) [get_cell](#) (const int x_cell, const int y_cell)
Получить клетку из поля
- std::vector< std::vector< [Cell](#) > > [get_cells](#) ()

- Получить все клетки поля
- void `set_x` (const int a)
Установить значение поля x, равное размеру поля по оси x.
- void `set_y` (const int b)
Установить значение поля y, равное размеру поля по оси y.
- void `set_cell` (const int x_cell, const int y_cell, const int status)
Задать данные новой клетки
- void `set_cells` (const std::vector< std::vector< `Cell` > > c)
Задать массив клеток

3.5.1 Подробное описание

Класс Поле

Этот класс моделирует объект поле на котором располагаются клетки.

3.5.2 Конструктор(ы)

3.5.2.1 `Field::Field (const int a = 0, const int b = 0) [inline]`

Конструктор

Аргументы

a	размер поля по оси x
b	размер поля по оси y

3.5.3 Методы

3.5.3.1 `Cell Field::get_cell (const int x_cell, const int y_cell) [inline]`

Получить клетку из поля

Аргументы

x_cell	координата клетки по x
y_cell	координата клетки по y

Возвращает

клетки из поля

3.5.3.2 `std::vector<std::vector<Cell> > Field::get_cells () [inline]`

Получить все клетки поля

Возвращает

двумерный массив клеток

3.5.3.3 `int Field::get_x () [inline]`

Получить размер поля по оси x.

Возвращает

размер поля по оси x

3.5.3.4 `int Field::get_y () [inline]`

Получить размер поля по оси y.

Возвращает

размер поля по оси y

3.5.3.5 `void Field::set_cell (const int x_cell, const int y_cell, const int status)`

Задать данные новой клетки

Аргументы

<code>x_cell</code>	координата клетки по x
<code>y_cell</code>	координата клетки по y
<code>status</code>	состояние клетки

3.5.3.6 `void Field::set_cells (const std::vector< std::vector< Cell > > c) [inline]`

Задать массив клеток

Аргументы

<code>c</code>	двумерный массив клеток
----------------	-------------------------

3.5.3.7 `void Field::set_x (const int a)`

Установить значение поля x, равное размеру поля по оси x.

Аргументы

<code>a</code>	размер поля по оси x
----------------	----------------------

3.5.3.8 `void Field::set_y (const int b)`

Установить значение поля y, равное размеру поля по оси y.

Аргументы

<code>b</code>	размер поля по оси y
----------------	----------------------

Объявления и описания членов классов находятся в файлах:

- `/home/user/LivingCells/sources/livingcells/core/field.h`
- `/home/user/LivingCells/sources/livingcells/core/field.cpp`

3.6 Класс `field_window`

Граф наследования:`field_window`:

Граф связей класса `field_window`:

Открытые члены

- field_window (QWidget *parent, [Api](#) a, [Field](#) f)
- void print_field ()

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/gui/field_window.h
- /home/user/LivingCells/sources/livingcells/gui/field_window.cpp

3.7 Класс figure_window

Граф наследования:figure_window:

Граф связей класса figure_window:

Открытые члены

- figure_window (QWidget *parent)

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/gui/figure_window.h
- /home/user/LivingCells/sources/livingcells/gui/figure_window.cpp

3.8 Класс initialize_window

Граф наследования:initialize_window:

Граф связей класса initialize_window:

Открытые члены

- initialize_window (QWidget *parent, [Api](#) *a, [Field](#) *f, int i, int j)

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/gui/initialize_window.h
- /home/user/LivingCells/sources/livingcells/gui/initialize_window.cpp

3.9 Класс MainWindow

Граф наследования:MainWindow:

Граф связей класса MainWindow:

Открытые члены

- MainWindow (QWidget *parent)

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/gui/mainwindow.h
- /home/user/LivingCells/sources/livingcells/gui/mainwindow.cpp

3.10 Структура qt_meta_stringdata_exit_window_t

Открытые атрибуты

- QByteArrayData data [4]
- char stringdata0 [41]

Объявления и описания членов структуры находятся в файле:

- /home/user/LivingCells/sources/build-livingcells-Desktop_Qt_5_5_0_GCC_32bit-Debug/gui/moc↔_exit_window.cpp

3.11 Структура qt_meta_stringdata_field_window_t

Открытые атрибуты

- QByteArrayData data [5]
- char stringdata0 [59]

Объявления и описания членов структуры находятся в файле:

- /home/user/LivingCells/sources/build-livingcells-Desktop_Qt_5_5_0_GCC_32bit-Debug/gui/moc↔_field_window.cpp

3.12 Структура qt_meta_stringdata_figure_window_t

Открытые атрибуты

- QByteArrayData data [4]
- char stringdata0 [46]

Объявления и описания членов структуры находятся в файле:

- /home/user/LivingCells/sources/build-livingcells-Desktop_Qt_5_5_0_GCC_32bit-Debug/gui/moc↔_figure_window.cpp

3.13 Структура qt_meta_stringdata_initialize_window_t

Открытые атрибуты

- QByteArrayData data [4]
- char stringdata0 [59]

Объявления и описания членов структуры находятся в файле:

- /home/user/LivingCells/sources/build-livingcells-Desktop_Qt_5_5_0_GCC_32bit-Debug/gui/moc↔_initialize_window.cpp

3.14 Структура qt_meta_stringdata_MainWindow_t

Открытые атрибуты

- QByteArrayData data [7]
- char stringdata0 [79]

Объявления и описания членов структуры находятся в файле:

- /home/user/LivingCells/sources/build-livingcells-Desktop_Qt_5_5_0_GCC_32bit-Debug/gui/moc_↵_mainwindow.cpp

3.15 Структура qt_meta_stringdata_size_window_t

Открытые атрибуты

- QByteArrayData data [4]
- char stringdata0 [43]

Объявления и описания членов структуры находятся в файле:

- /home/user/LivingCells/sources/build-livingcells-Desktop_Qt_5_5_0_GCC_32bit-Debug/gui/moc_↵_size_window.cpp

3.16 Класс size_window

Граф наследования: size_window:

Граф связей класса size_window:

Открытые члены

- size_window (QWidget *parent, int mode)

Объявления и описания членов классов находятся в файлах:

- /home/user/LivingCells/sources/livingcells/gui/size_window.h
- /home/user/LivingCells/sources/livingcells/gui/size_window.cpp

3.17 Класс TestTest

Граф наследования: TestTest:

Граф связей класса TestTest:

Объявления и описания членов класса находятся в файле:

- /home/user/LivingCells/sources/livingcells/test/tst_testtest.cpp

Предметный указатель

Api, [5](#)

Cell, [7](#)

Cell, [7](#)

check, [8](#)

check

Cell, [8](#)

Field, [10](#)

Field, [11](#)