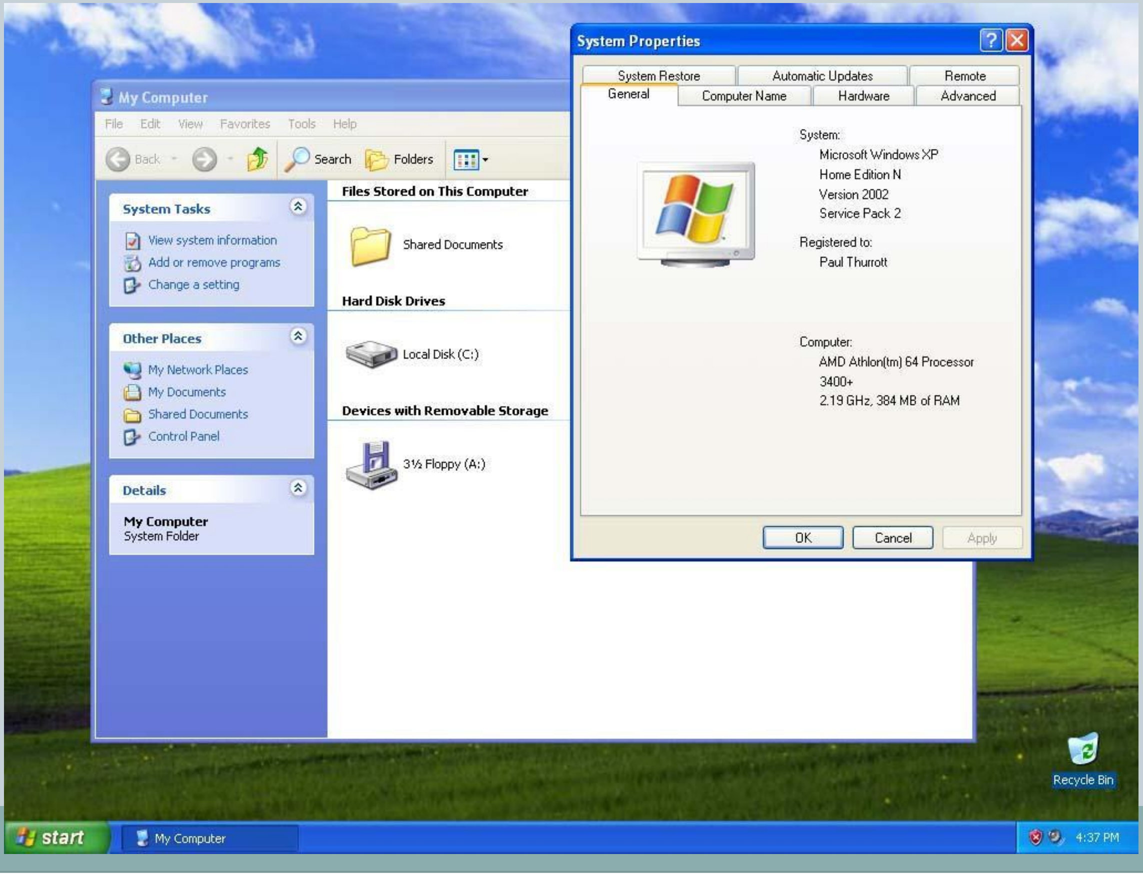
FLTK

HTTP://WWW.FLTK.ORG/

Оконная система, DWM

- XWindows (Linux/UNIX)
- Windows DWM (Win Vista, 7, 8)



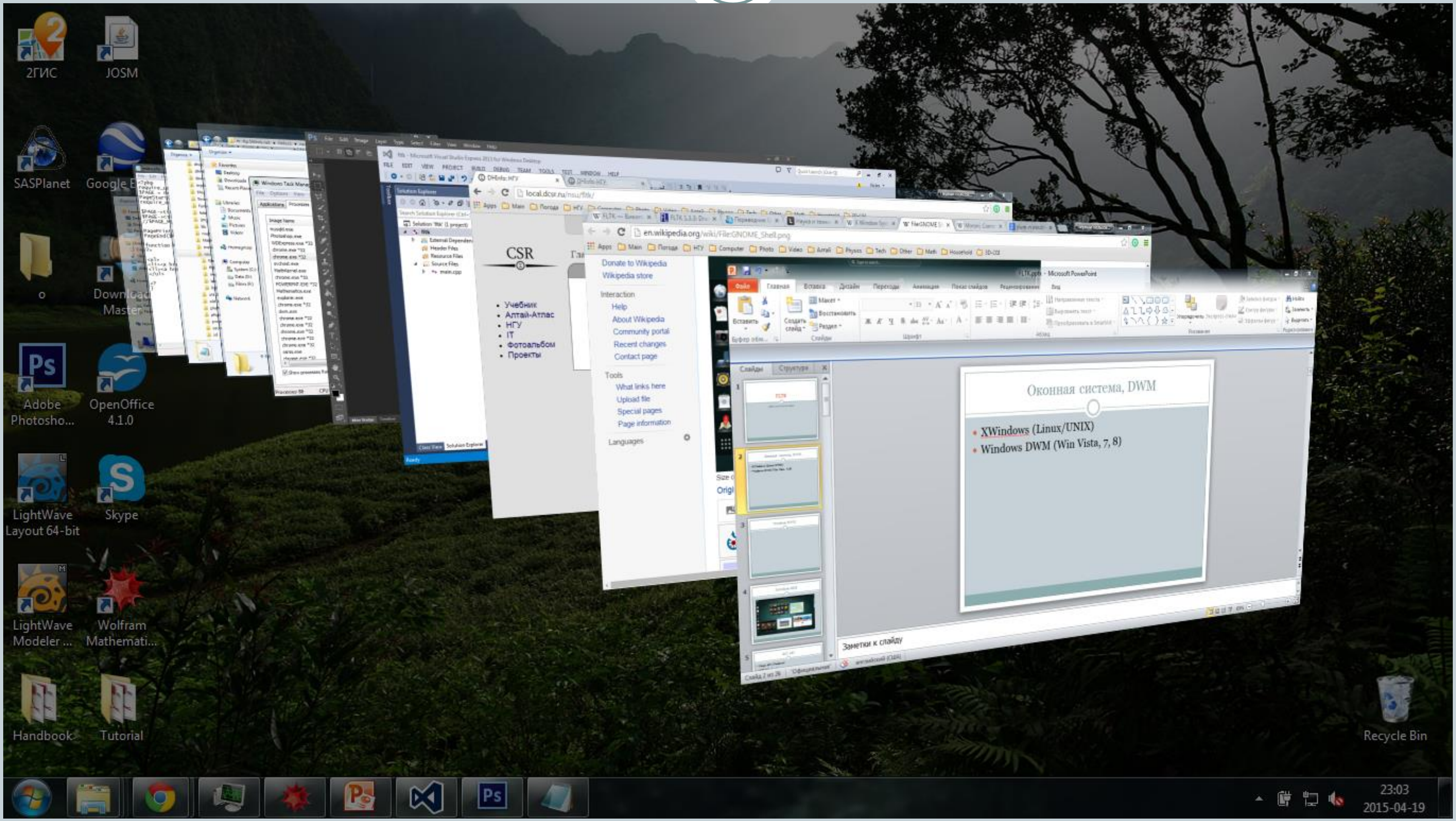
System Properties

System:
Microsoft Windows XP
Home Edition N
Version 2002
Service Pack 2

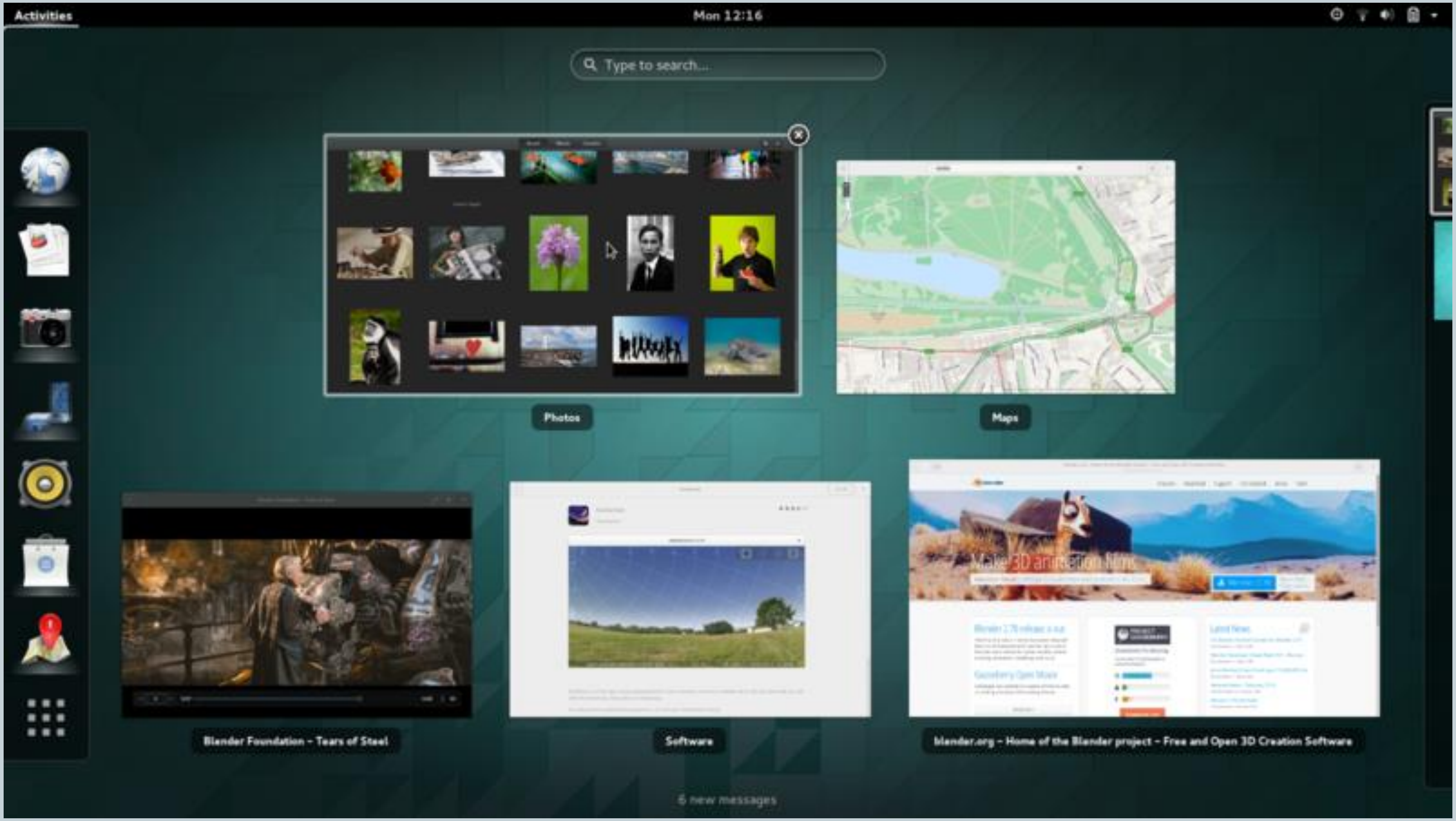
Registered to:
Paul Thurrott

Computer:
AMD Athlon(tm) 64 Processor
3400+
2.19 GHz, 384 MB of RAM

Windows/DWM


2ГИС
JOSM
SASPlanet
Google E
o
Download Master
Adobe Photosho...
OpenOffice 4.1.0
LightWave Layout 64-bit
Skype
LightWave Modeler ...
Wolfram Mathemati...
Handbook
Tutorial
Recycle Bin
CSR
Учебник
Алтай-Атлас
НГУ
IT
Фотоальбом
Проекты
en.wikipedia.org/wiki/File:GNOME_Shell.png
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
Upload file
Special pages
Page information
Languages
Оконная система, DWM
XWindows (Linux/UNIX)
Windows DWM (Win Vista, 7, 8)
Заметки к слайду
23:03
2015-04-19

Xwindows/KDE



Activities
Mon 12:16
Type to search...
Photos
Maps
Blender Foundation – Tears of Steel
Software
blender.org – Home of the Blender project – Free and Open 3D Creation Software
6 new messages

GUI API

- Win32 API (Windows)
- MFC (Windows)
- Qt (Linux, Windows)
- KDE (Linux)
- FLTK (Linux, Windows)
- OpenGL (Linux, Windows)
- DirectX (Windows)

API

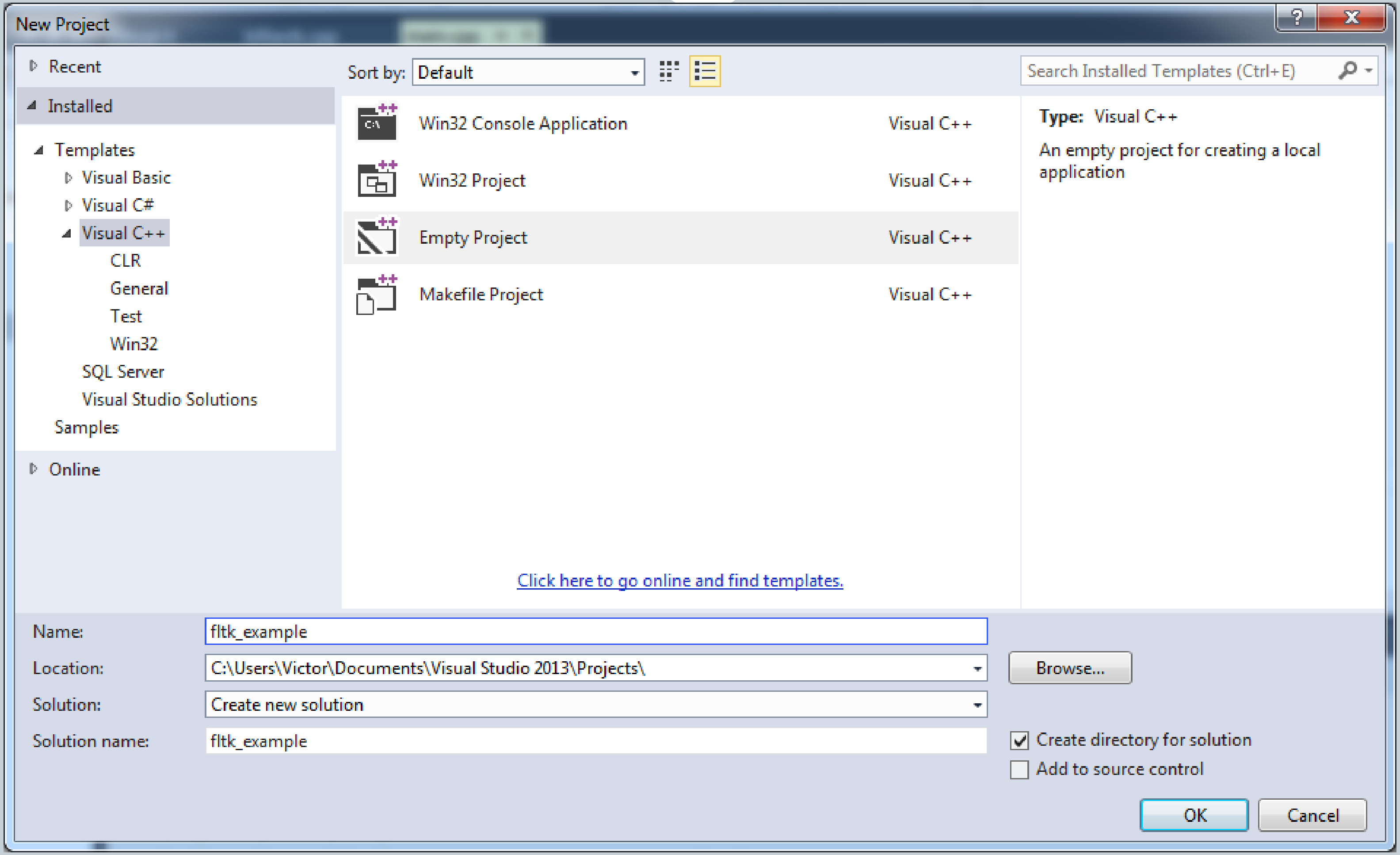
- Работа с файловой системой
- Работа с виртуальной памятью
- Работа с сетью
- Графический вывод (GUI)
- Отслеживание действий пользователя
- Управление системой
- Управление периферийными устройствами
- Управление выполнением программы

FLTK

Fast Light Toolkit — кросс-платформенная библиотека инструментов с открытым исходным кодом (лицензия LGPL) для построения графического интерфейса пользователя (GUI). Платформы:

- UNIX/Linux X11
- Microsoft Windows
- MacOS X

Создание проекта в VS C++ 2013



Sort by: Default

Win32 Console Application	Visual C++
Win32 Project	Visual C++
Empty Project	Visual C++
Makefile Project	Visual C++

Type: Visual C++

An empty project for creating a local application

Click here to go online and find templates.

Name: fltk_example

Location: C:\Users\Victor\Documents\Visual Studio 2013\Projects\

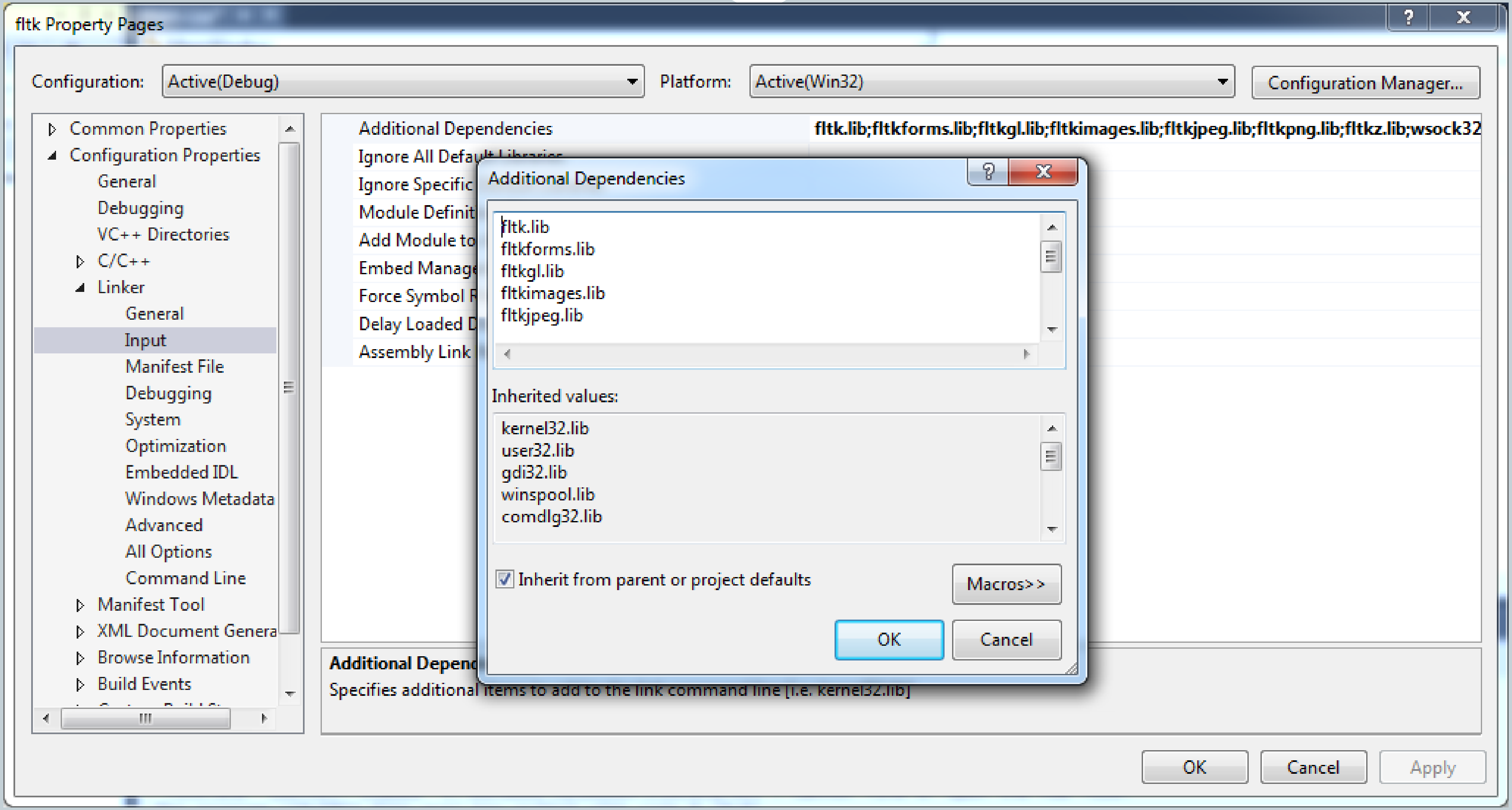
Solution: Create new solution

Solution name: fltk_example

☑ Create directory for solution

□ Add to source control

Установка свойств проекта



fltk Property Pages

Configuration: Active(Debug)

Platform: Active(Win32)

Additional Dependencies: fltk.lib;fltkforms.lib;fltkgl.lib;fltkimages.lib;fltkjpeg.lib;fltkpng.lib;fltkz.lib;wsock32

Additional Dependencies

```
fltk.lib
fltkforms.lib
fltkgl.lib
fltkimages.lib
fltkjpeg.lib
```

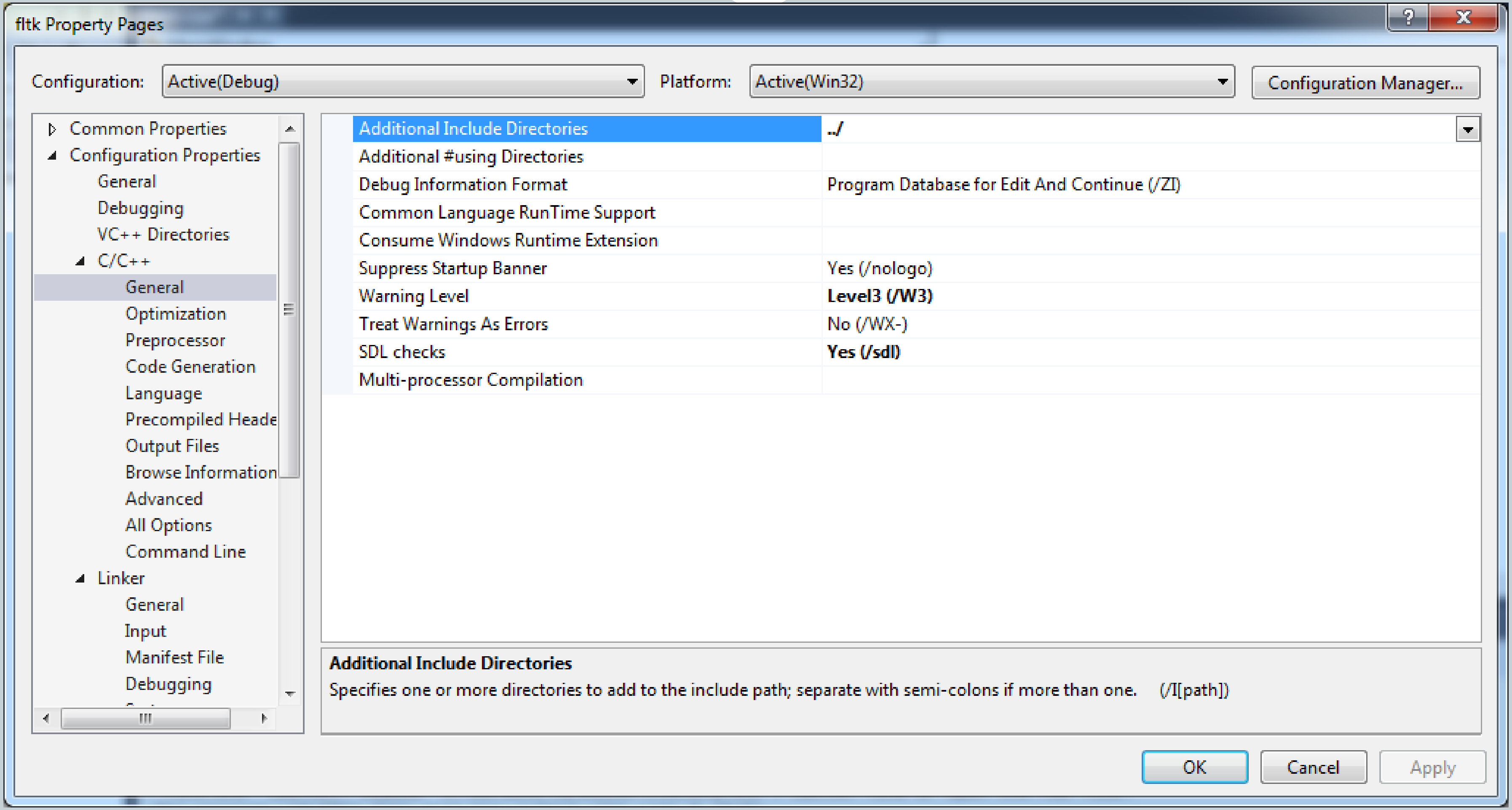
Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
```

☑ Inherit from parent or project defaults

Additional Depend

Specifies additional items to add to the link command line [i.e. kernel32.lib]



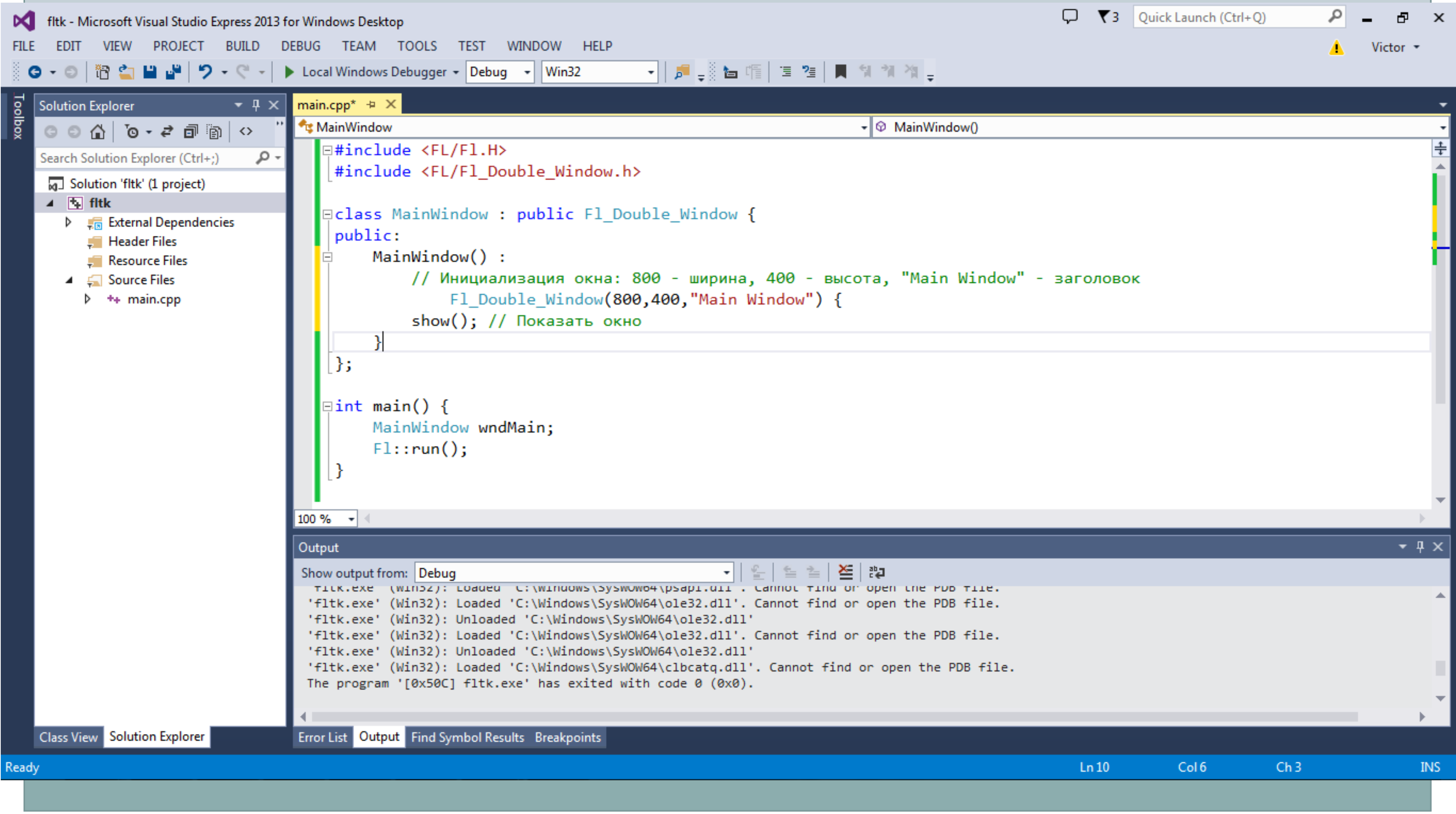
Configuration: Active(Debug)

Platform: Active(Win32)

Additional Include Directories	../
Additional #using Directories	
Debug Information Format	Program Database for Edit And Continue (/ZI)
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
SDL checks	**Yes (/sdl)**
Multi-processor Compilation	

Additional Include Directories

Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

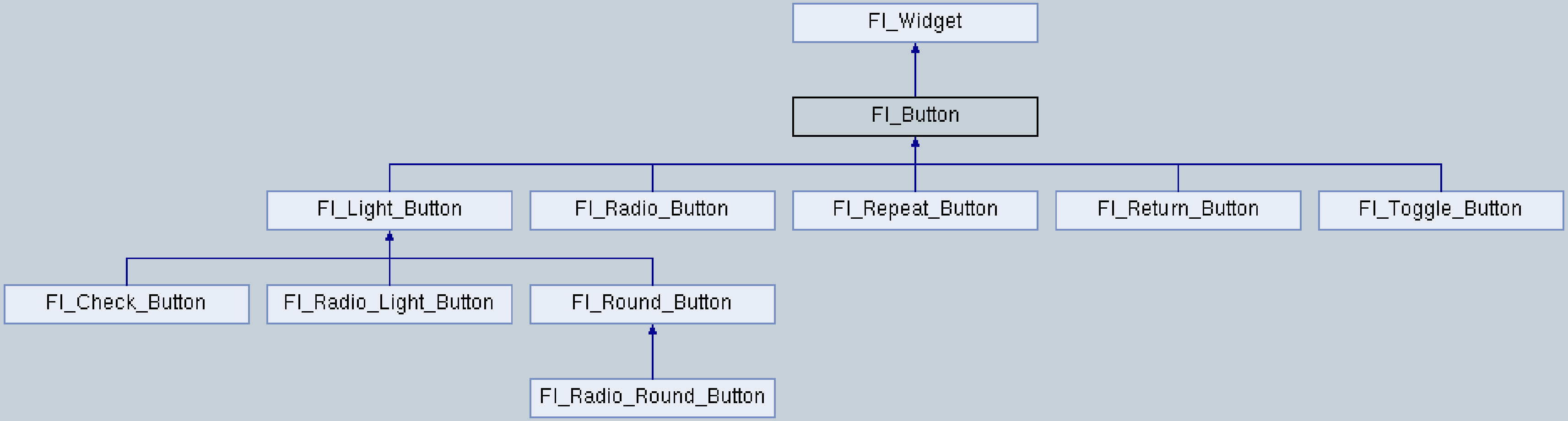


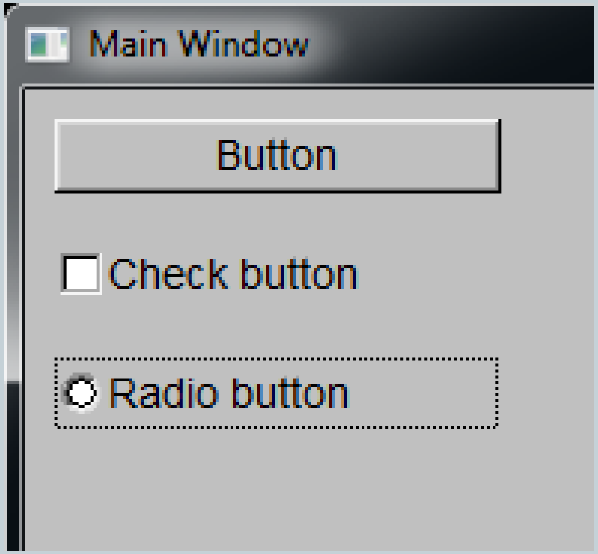
```
#include <FL/Fl.H>
#include <FL/Fl_Double_Window.h>

class MainWindow : public Fl_Double_Window {
public:
    MainWindow() :
        // Инициализация окна: 800 - ширина, 400 - высота, "Main Window" - заголовок
          Fl_Double_Window(800,400,"Main Window") {
        show(); // Показать окно
    }
};

int main() {
    MainWindow wndMain;
    Fl::run();
}
```

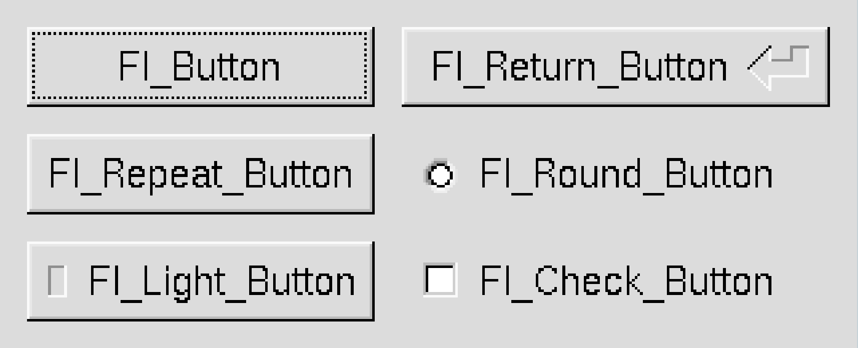
```
'fltk.exe' (Win32): Loaded 'C:\Windows\SysWOW64\psapi.dll'. Cannot find or open the PDB file.
'fltk.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ole32.dll'. Cannot find or open the PDB file.
'fltk.exe' (Win32): Unloaded 'C:\Windows\SysWOW64\ole32.dll'
'fltk.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ole32.dll'. Cannot find or open the PDB file.
'fltk.exe' (Win32): Unloaded 'C:\Windows\SysWOW64\ole32.dll'
'fltk.exe' (Win32): Loaded 'C:\Windows\SysWOW64\clbcatq.dll'. Cannot find or open the PDB file.
The program '[0x50C] fltk.exe' has exited with code 0 (0x0).
```

Кнопки (Buttons)

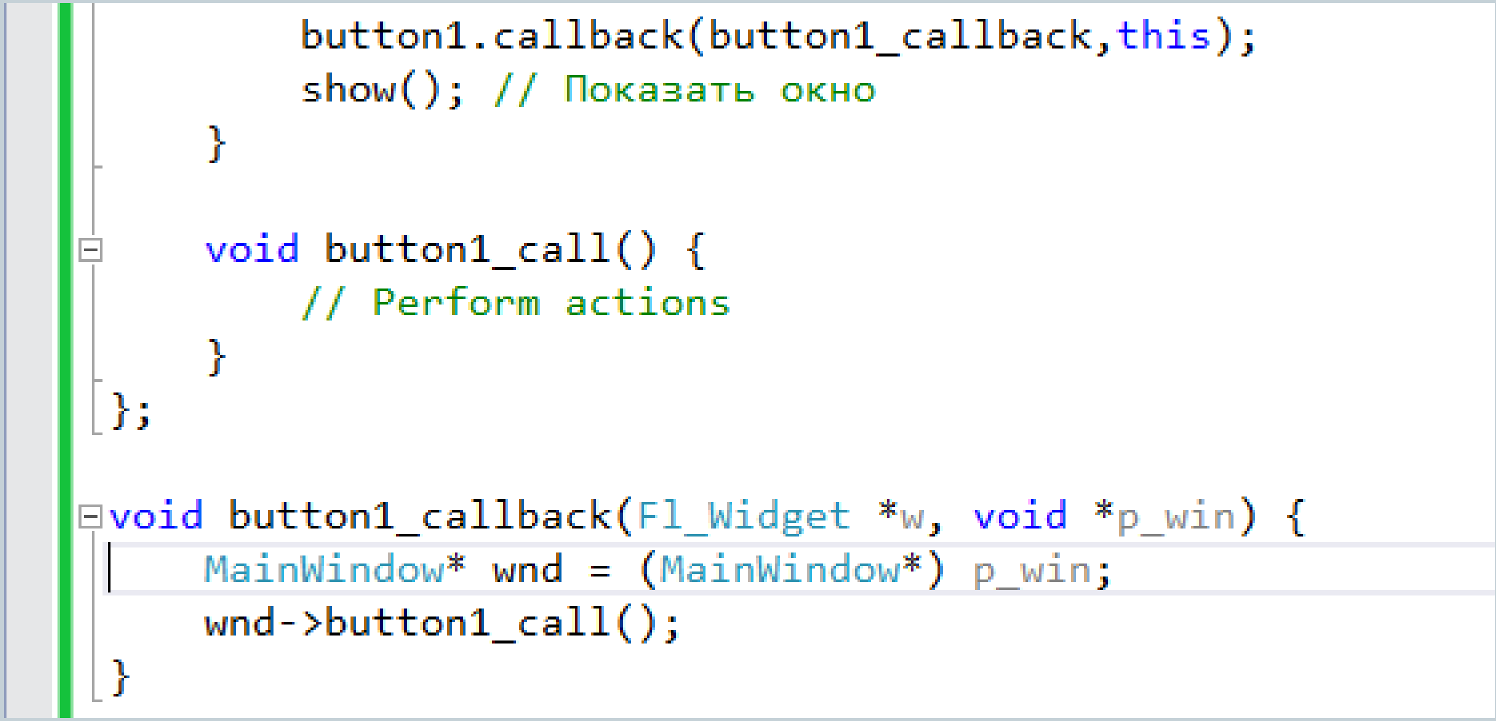
- Fl_Button - A standard push button.
- Fl_Check_Button - A button with a check box.
- Fl_Light_Button - A push button with a light.
- Fl_Repeat_Button - A push button that repeats when held.
- Fl_Return_Button - A push button that is activated by the Enter key.
- Fl_Round_Button - A button with a radio circle.


Fl_Widget
Fl_Button
Fl_Light_Button
Fl_Radio_Button
Fl_Repeat_Button
Fl_Return_Button
Fl_Toggle_Button
Fl_Check_Button
Fl_Radio_Light_Button
Fl_Round_Button
Fl_Radio_Round_Button


Main Window
Button
□ Check button
○ Radio button


Fl_Button
Fl_Return_Button
Fl_Repeat_Button
○ Fl_Round_Button
Fl_Light_Button
□ Fl_Check_Button

Callback



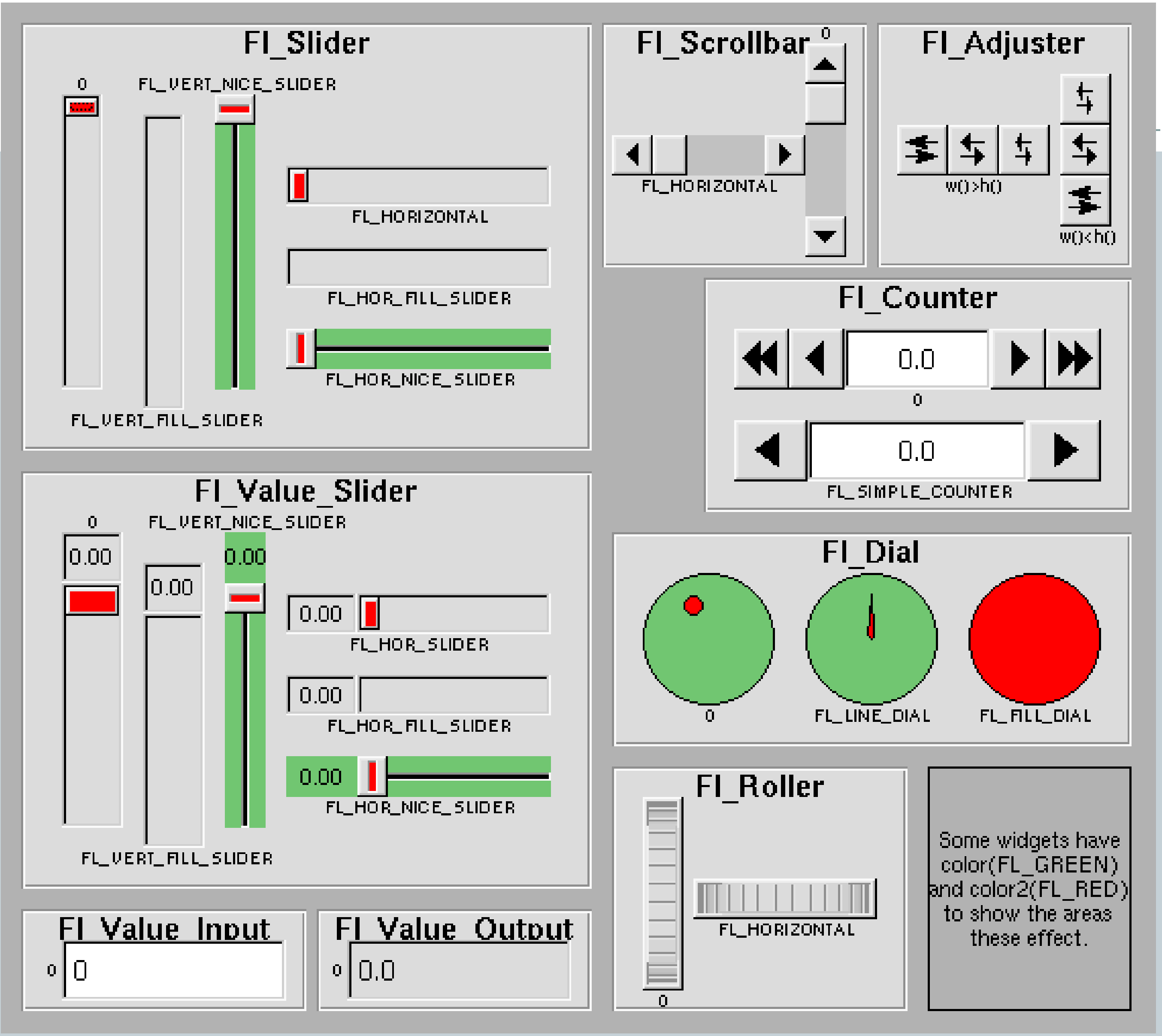
```
        button1.callback(button1_callback,this);
        show(); // Показать окно
    }

    void button1_call() {
        // Perform actions
    }
};

void button1_callback(Fl_Widget *w, void *p_win) {
    MainWindow* wnd = (MainWindow*) p_win;
    wnd->button1_call();
}
```

Текстовые поля (Text)

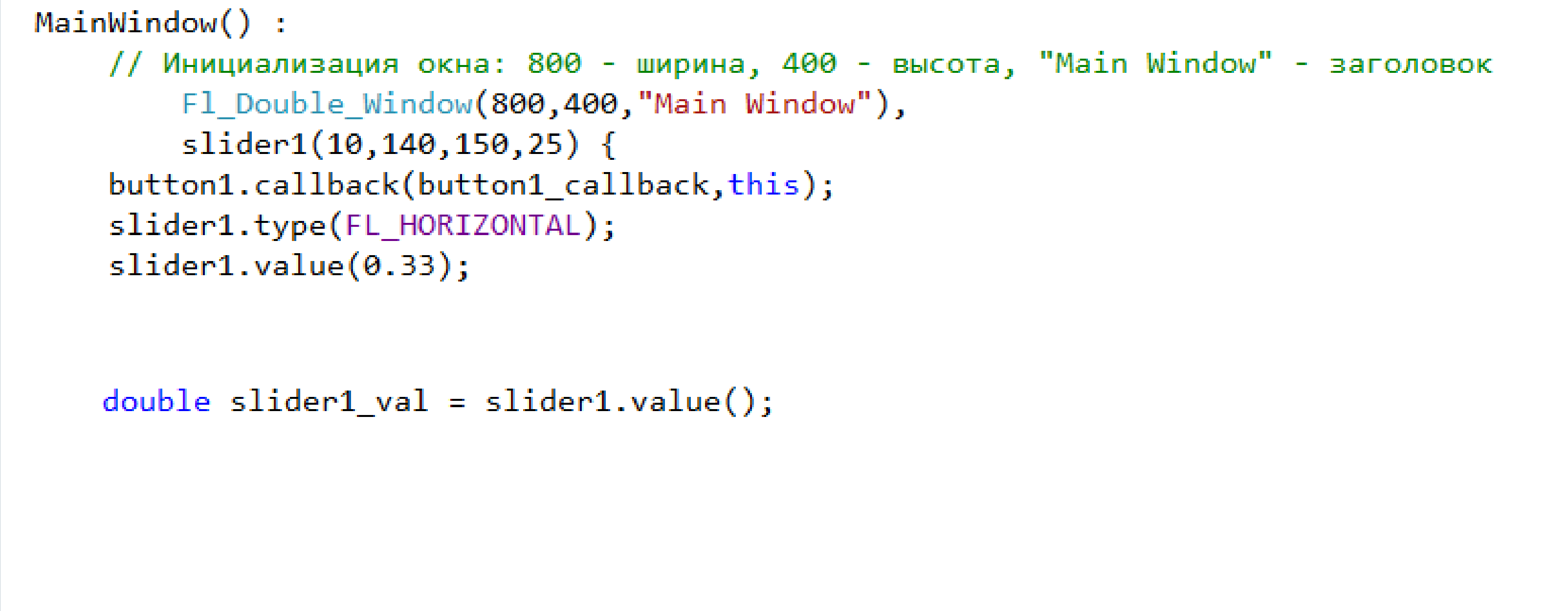
- Fl_Input - A one-line text input field.
- Fl_Output - A one-line text output field.
- Fl_Multiline_Input - A multi-line text input field.
- Fl_Multiline_Output - A multi-line text output field.
- Fl_Text_Display - A multi-line text display widget.
- Fl_Text_Editor - A multi-line text editing widget.
- Fl_Help_View - A HTML text display widget.

Valuators

- Fl_Counter - A widget with arrow buttons that shows the current value.
- Fl_Dial - A round knob.
- Fl_Roller - An SGI-like dolly widget.
- Fl_Scrollbar - A standard scrollbar widget.
- Fl_Slider - A scrollbar with a knob.
- Fl_Value_Slider - A slider that shows the current value.

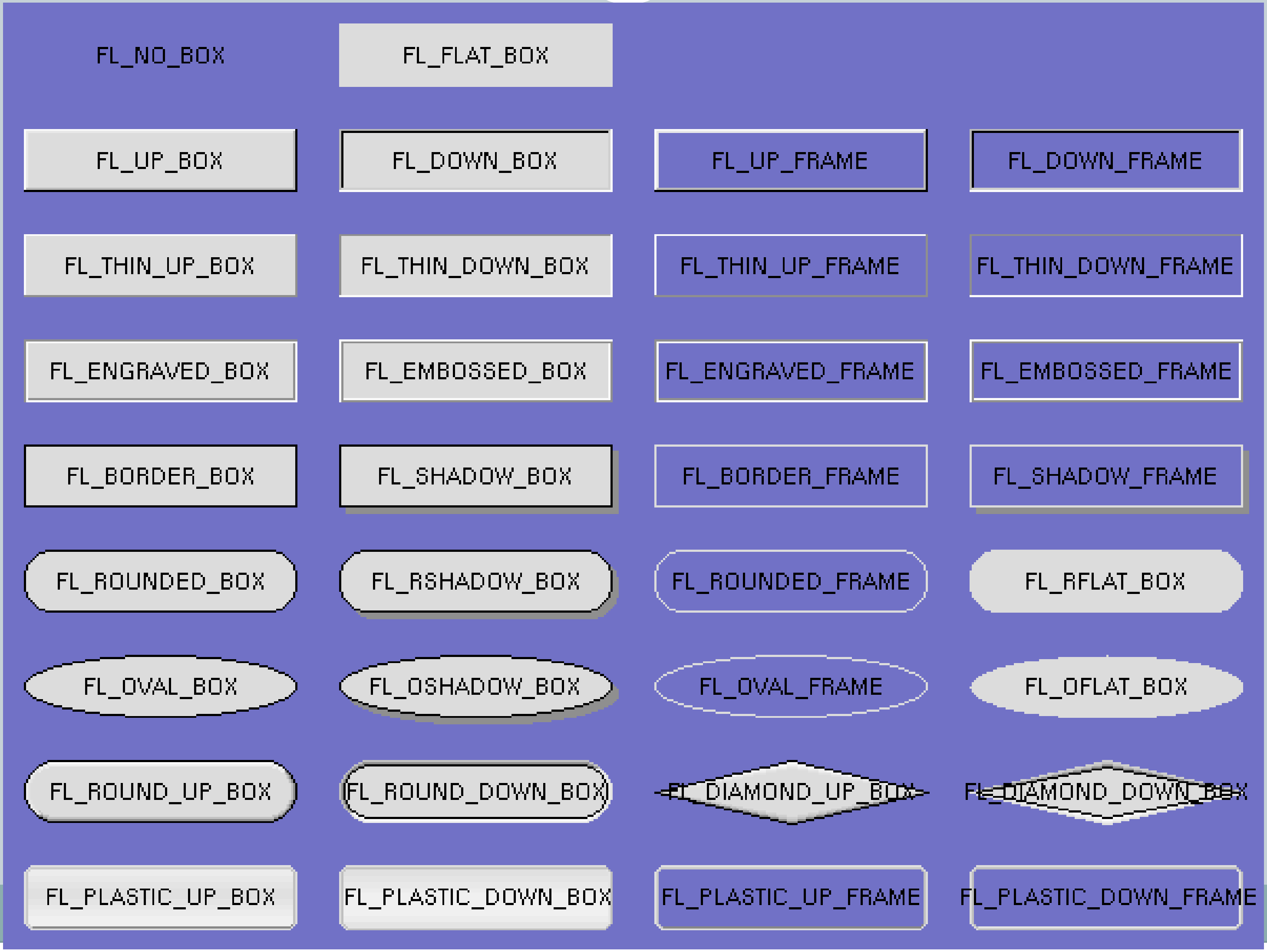


Some widgets have color(FL_GREEN) and color2(FL_RED) to show the areas these effect.

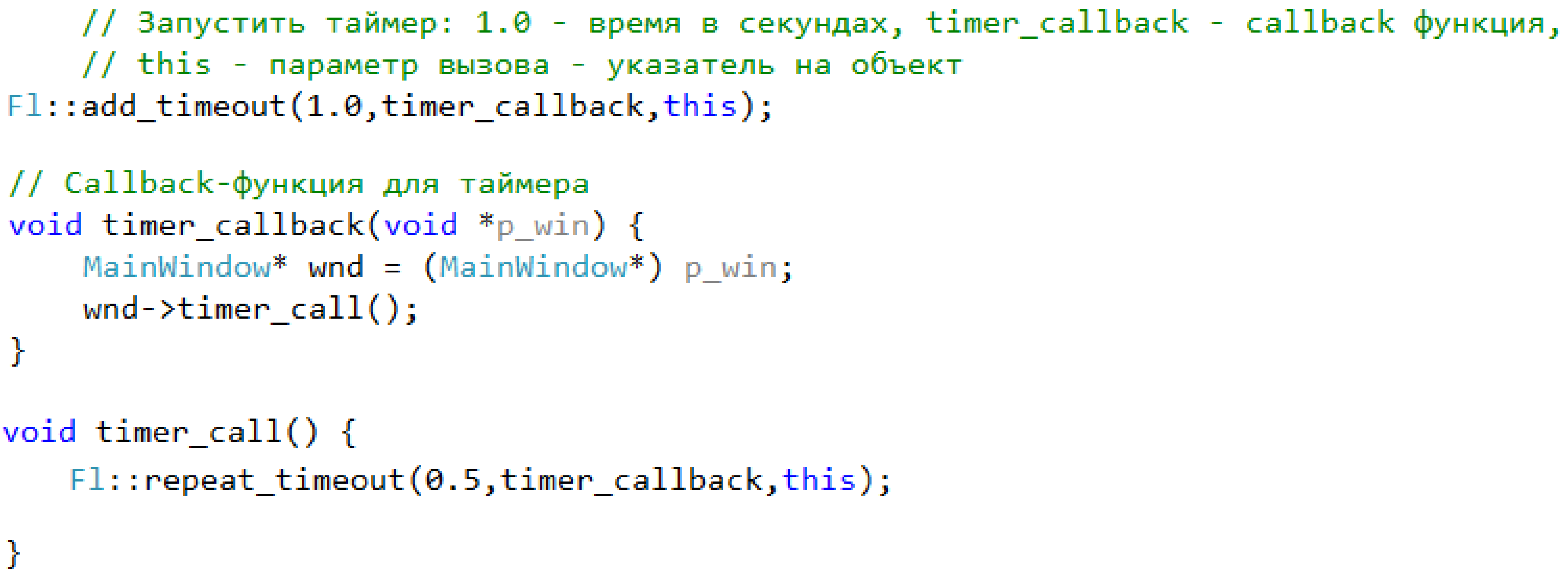


```
MainWindow() :
    // Инициализация окна: 800 - ширина, 400 - высота, "Main Window" - заголовок
        Fl_Double_Window(800,400,"Main Window"),
        slider1(10,140,150,25) {
    button1.callback(button1_callback,this);
    slider1.type(FL_HORIZONTAL);
    slider1.value(0.33);


    double slider1_val = slider1.value();
```

Box стили


FL_NO_BOX
FL_FLAT_BOX
FL_UP_BOX
FL_DOWN_BOX
FL_UP_FRAME
FL_DOWN_FRAME
FL_THIN_UP_BOX
FL_THIN_DOWN_BOX
FL_THIN_UP_FRAME
FL_THIN_DOWN_FRAME
FL_ENGRAVED_BOX
FL_EMBOSSED_BOX
FL_ENGRAVED_FRAME
FL_EMBOSSED_FRAME
FL_BORDER_BOX
FL_SHADOW_BOX
FL_BORDER_FRAME
FL_SHADOW_FRAME
FL_ROUNDED_BOX
FL_RSHADOW_BOX
FL_ROUNDED_FRAME
FL_RFLAT_BOX
FL_OVAL_BOX
FL_OSHADOW_BOX
FL_OVAL_FRAME
FL_OFLAT_BOX
FL_ROUND_UP_BOX
FL_ROUND_DOWN_BOX
FL_DIAMOND_UP_BOX
FL_DIAMOND_DOWN_BOX
FL_PLASTIC_UP_BOX
FL_PLASTIC_DOWN_BOX
FL_PLASTIC_UP_FRAME
FL_PLASTIC_DOWN_FRAME

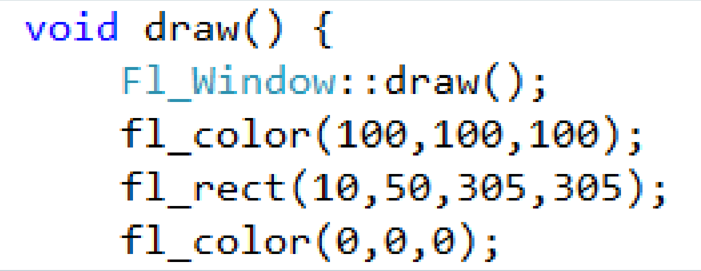
Таймер (Timer)



```
    // Запустить таймер: 1.0 - время в секундах, timer_callback - callback функция,
    // this - параметр вызова - указатель на объект
Fl::add_timeout(1.0,timer_callback,this);

// Callback-функция для таймера
void timer_callback(void *p_win) {
    MainWindow* wnd = (MainWindow*) p_win;
    wnd->timer_call();
}

void timer_call() {
    Fl::repeat_timeout(0.5,timer_callback,this);

}
```

Графика

- Добавьте в класс MainWindow метод draw.



```
void draw() {
    Fl_Window::draw();
    fl_color(100,100,100);
    fl_rect(10,50,305,305);
    fl_color(0,0,0);
```

Графические примитивы

- `void fl_point(int x, int y)`

Draw a single pixel at the given coordinates.

- `void fl_rectf(int x, int y, int w, int h)`

Color a rectangle that exactly fills the given bounding box.

- `void fl_rect(int x, int y, int w, int h)`

Draw a 1-pixel border inside this bounding box.

- `void fl_line(int x, int y, int x1, int y1)`

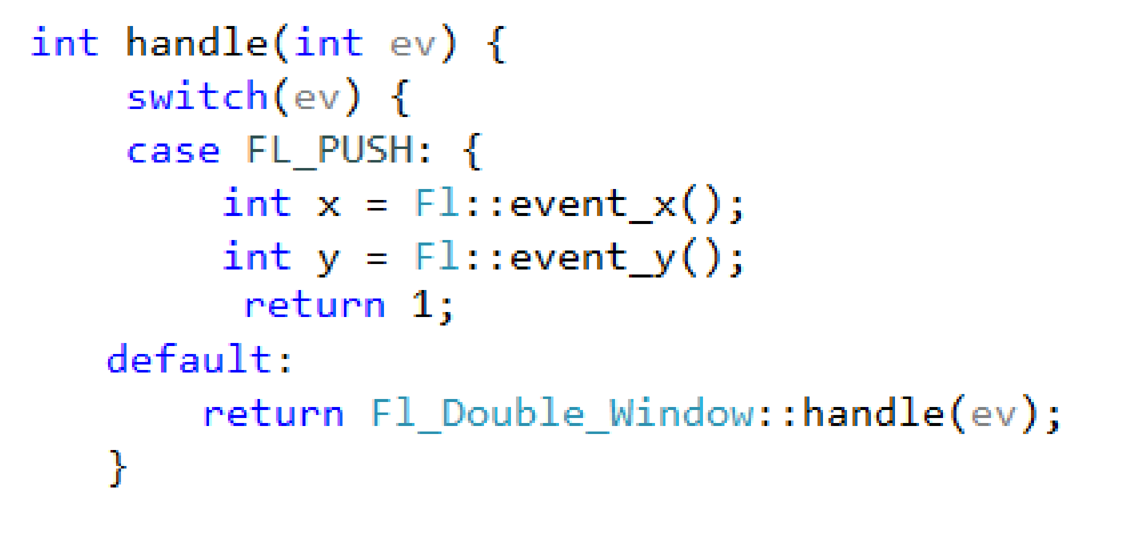
Draw one or two lines between the given points.

- `void fl_circle(double x, double y, double r)`
- `void fl_draw(const char *, int x, int y)`

Обработка сообщений
(Event handlers)

Добавьте в класс MainWindow метод handle.

ev – код события (FL_PUSH, etc.)



```
int handle(int ev) {
    switch(ev) {
    case FL_PUSH: {
        int x = Fl::event_x();
        int y = Fl::event_y();
         return 1;
   default:
       return Fl_Double_Window::handle(ev);
   }
```

Сообщения мыши

- FL_PUSH
- FL_RELEASE
- FL_DRAG
- FL_MOVE
- FL_MOUSEWHEEL

- Fl::event_x()
- Fl::event_y().

События клавиатуры

- FL_KEYBOARD
- FL_KEYDOWN
- FL_KEYUP
- FL_SHORTCUT

- Fl::event_key()
- Fl::event_text()
- Fl::event_length()

- http://www.fltk.org/doc-1.3/
- http://info.dcsr.ru/nsu/fltk/