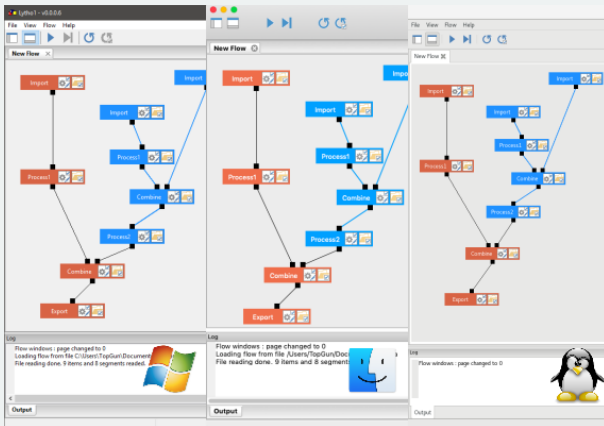




Кроссплатформенная разработка программного обеспечения

Лабораторная работа №4

Основы работы с библиотекой wxWidgets.



Первое приложение wxWidgets

```
#include <wx/wx.h>

class Frame : public wxFrame {
public:
    Frame(const wxString& title);
};

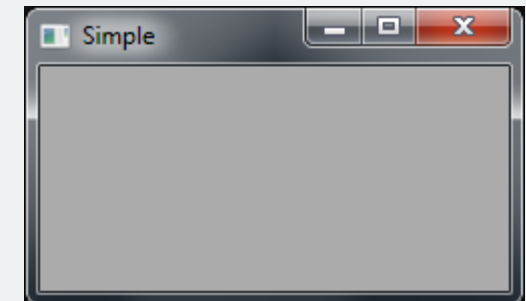
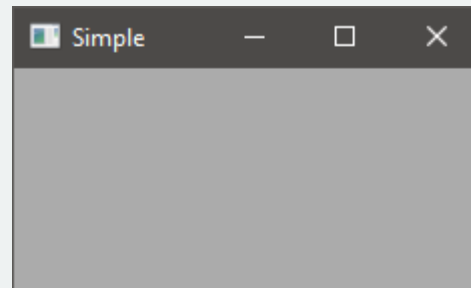
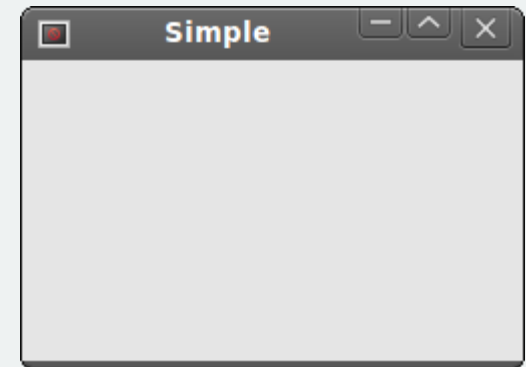
Frame::Frame(const wxString& title)
    : wxFrame(NULL, wxID_ANY, title, wxDefaultPosition, wxSize(250, 150)) {
    Centre();
}

class MyApp : public wxApp {
public:
    virtual bool OnInit();
};

IMPLEMENT_APP(MyApp)

bool MyApp::OnInit() {
    Frame *frame = new Frame(wxT("Simple"));
    frame->Show(true);

    return true;
}
```



Архитектура проекта в терминах wxWidgets

```
#include <wx/wx.h>
```

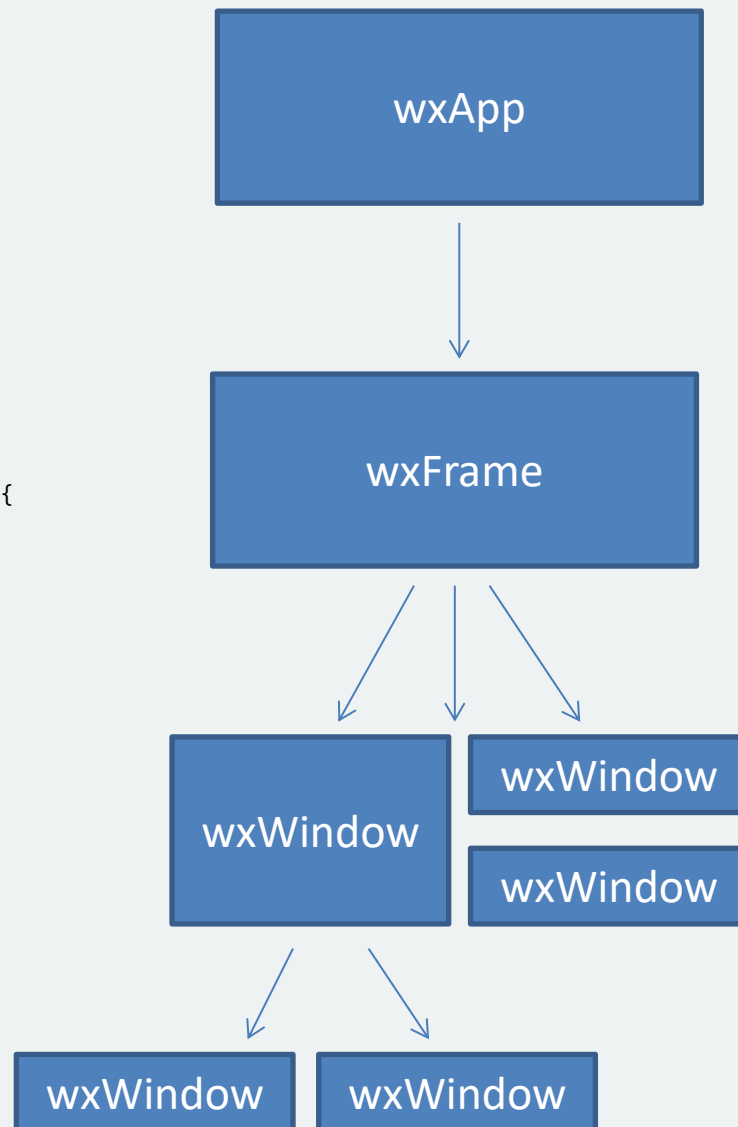
```
class Frame : public wxFrame {  
public:  
    Frame(const wxString& title);  
};
```

```
Frame::Frame(const wxString& title)  
    : wxFrame(NULL, wxID_ANY, title, wxDefaultPosition, wxSize(250, 150)) {  
    Centre();  
}
```

```
class MyApp : public wxApp {  
public:  
    virtual bool OnInit();  
};
```

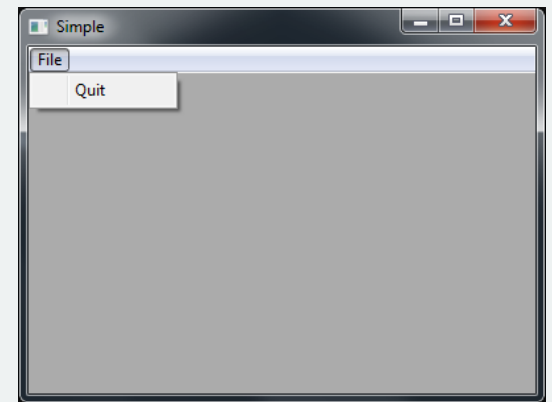
```
IMPLEMENT_APP(MyApp)
```

```
bool MyApp::OnInit() {  
    Frame *frame = new Frame(wxT("Simple"));  
    frame->Show(true);  
  
    return true;  
}
```



Добавление элементов управления: меню

```
class Frame : public wxFrame {  
public:  
    Frame(const wxString& title);  
};  
  
Frame::Frame(const wxString& title)  
    : wxFrame(NULL, wxID_ANY, title, wxDefaultPosition, wxSize(400, 300)) {  
  
    wxMenuBar *p_menubar = new wxMenuBar;  
  
    wxMenu *p_menuFile = new wxMenu;  
    p_menuFile->Append(wxID_EXIT, wxT("&Quit"));  
  
    p_menubar->Append(p_menuFile, wxT("&File"));  
  
    SetMenuBar(p_menubar);  
  
    Centre();  
}
```



Обработка сообщений: задание обработчика (2)

```
class Frame : public wxFrame {  
public:  
    Frame(const wxString& title);  
private:  
    void OnMenu_FileQuit(wxCommandEvent &evt);  
};
```

```
void Frame::OnMenu_FileQuit(wxCommandEvent &evt) {  
    Close(true);  
}
```



Обработка сообщений: ООП стиль

```
Frame::Frame(const wxString& title)
    : wxFrame(NULL, wxID_ANY, ...) {

    p_menubar = new wxMenuBar;
    p_menuFile = new wxMenu;
    p_menuFile->Append(wxID_EXIT, wxT("&Quit"));
    p_menubar->Append(p_menuFile, wxT("&File"));
    SetMenuBar(p_menubar);
```

```
Connect(wxID_EXIT,
        wxEVT_COMMAND_MENU_SELECTED,
        wxCommandEventHandler(Frame::OnMenu_FileQuit));
```

```
Centre();
}
```

Создание панели инструментов: создание иконки

```
Frame::Frame(const wxString& title)
: wxFrame(NULL, wxID_ANY, title, wxDefaultPosition, wxSize(400, 300)) {

    p_menubar = new wxMenuBar;
    p_menuFile = new wxMenu;
    p_menuFile->Append(wxID_EXIT, wxT("&Quit"));
    p_menubar->Append(p_menuFile, wxT("&File"));
    SetMenuBar(p_menubar);
```

```
wxImage::AddHandler(new wxPNGHandler);
wxBitmap bmp_exit(wxT("exit.png"), wxBITMAP_TYPE_PNG);
```

```
Centre();
```

```
}
```

Создание панели инструментов: добавление элементов

```
Frame::Frame(const wxString& title)
: wxFrame(NULL, wxID_ANY, title, wxDefaultPosition, wxSize(400, 300)) {

    p_menubar = new wxMenuBar;
    p_menuFile = new wxMenu;
    p_menuFile->Append(wxID_EXIT, wxT("&Quit"));
    p_menubar->Append(p_menuFile, wxT("&File"));
    SetMenuBar(p_menubar);

    wxImage::AddHandler(new wxPNGHandler);
    wxBitmap bmp_exit(wxT("exit.png"), wxBITMAP_TYPE_PNG);

    wxToolBar *p_toolbar = CreateToolBar();
    p_toolbar->AddTool(wxID_EXIT, wxT("Exit application"), bmp_exit);
    p_toolbar->Realize();

    Centre();
}
```




Создание строки статуса

```
Frame::Frame(const wxString& title)
    : wxFrame(NULL, wxID_ANY, title, wxDefaultPosition, wxSize(400, 300)) {

    ...

    wxToolBar *p_toolbar = CreateToolBar();
    p_toolbar->AddTool(wxID_EXIT, bmp_exit, wxT("Exit application"));
    p_toolbar->Realize();

    CreateStatusBar(3);
    SetStatusText(wxT("Ready!"), 0);

    Centre();
}
```



Обработка мыши: добавление обработчика

```
class Frame : public wxFrame {  
public:  
    Frame(const wxString& title);  
private:  
  
    void OnMouseMove(wxMouseEvent &evt);  
  
    void OnMenu_FileQuit(wxCommandEvent &evt);  
    wxDECLARE_EVENT_TABLE();  
};
```



Обработка мыши: связывание обработчика с событием

```
Frame::Frame(const wxString& title)
: wxFrame(NULL, wxID_ANY, title, ...) {

...

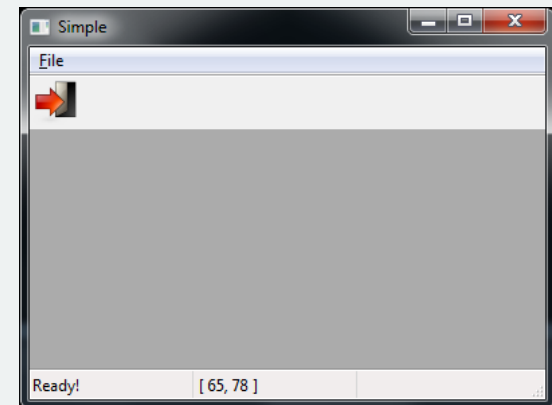
CreateStatusBar(3);
SetStatusText(wxT("Ready!"), 0);

Connect(wxEVT_MOTION, wxMouseEventHandler(Frame::OnMouseMove));

Centre();
}
```

Обработка мыши: описание обработчика

```
void Frame::OnMouseMove(wxMouseEvent &evt) {  
    char buf[32];  
    sprintf(buf, "[ %d, %d ]", evt.GetX(), evt.GetY());  
    SetStatusText(wxString(buf), 1);  
}
```



```
void Frame::OnMouseMove(wxMouseEvent &evt) {  
    SetStatusText(wxString::Format("[ %d, %d ]",  
        evt.GetX(),  
        evt.GetY()),  
        1);  
}
```



Работа с GDI в wxWidgets (1)

```
class Frame : public wxFrame {  
public:  
    Frame(const wxString& title);  
private:  
    ...  
    void OnPaint(wxPaintEvent &evt);  
    ...  
};
```



Работа с GDI в wxWidgets (2)

```
class Frame : public wxFrame {  
public:  
    Frame(const wxString& title);  
private:  
    ...  
    void OnPaint(wxPaintEvent &evt);  
    ...  
};
```

...

```
void Frame::OnPaint(wxPaintEvent &evt) {  
    wxPaintDC dc(this);  
  
    dc.DrawRectangle(100, 100, 200, 100);  
}
```



Работа с GDI в wxWidgets (3)

```
wxBEGIN_EVENT_TABLE(Frame, wxFrame)
    EVT_MENU(wxID_EXIT, Frame::OnMenu_FileQuit)
    EVT_MOTION(Frame::OnMouseMove)
```

```
EVT_PAINT(Frame::OnPaint)
```

```
wxEND_EVENT_TABLE()
```

```
Frame::Frame(const wxString& title)
    : wxFrame(NULL, wxID_ANY, title, ...) {
```

```
...
```

```
Connect(wxEVT_PAINT, wxPaintEventHandler(Frame::OnPaint));
```

```
Centre();
```

```
}
```



Работа с GDI в wxWidgets (4)

```
void Frame::OnPaint(wxPaintEvent &evt) {
    wxPaintDC dc(this);

    wxBrush br(wxColour(255, 255, 255));
    dc.SetBrush(br);

    wxPen pen(wxColour(255, 0, 0), 10, wxPENSTYLE_DOT);
    dc.SetPen(pen);

    dc.DrawRectangle(100, 100, 200, 100);

    dc.SetTextForeground(wxColour(0, 255, 0));
    dc.DrawText(wxT("Hello!"), 50, 50);
}
```


Корректная архитектура графических приложений

Заголовок окна

Строка меню

Панель инструментов

Класс, наследующийся от `wxWindow`

Строка состояния

Объявление главного виджета

```
class Window : public wxWindow {  
public:  
    Window(wxWindow *p_parent);  
private:  
    void OnPaint(wxPaintEvent &evt);  
};
```

```
Window::Window(wxWindow *p_parent)  
    : wxWindow(p_parent, wxID_ANY) {  
    Connect(wxEVT_PAINT, wxPaintEventHandler(Window::OnPaint));  
}
```

```
void Window::OnPaint(wxPaintEvent &evt) {  
    wxPaintDC dc(this);  
  
    wxRect r = GetClientRect();  
    r.Deflate(10);  
    dc.DrawRectangle(r);  
}
```



main_02.cpp



Создание главного виджета

```
Frame::Frame(const wxString& title)
    : wxFrame(NULL, wxID_ANY, title, ... ) {

    ...

    CreateStatusBar(3);
    SetStatusText(wxT("Ready!"), 0);

    Window *p_wnd = new Window(this);

    Centre();
}
```

Создание вложенных элементов меню

```
wxMenuBar *p_menubar = new wxMenuBar;
```

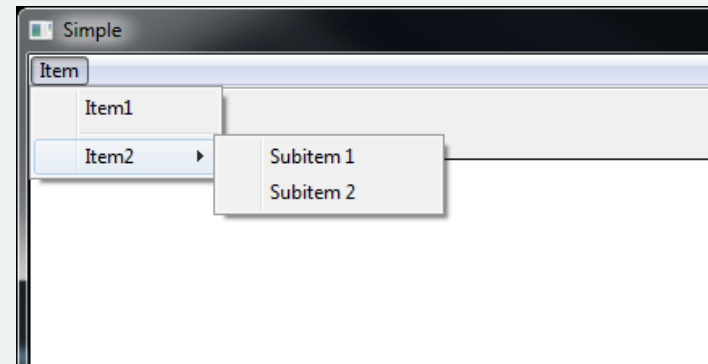
```
wxMenu *p_menuItem = new wxMenu;  
p_menuItem->Append(wxID_ANY, wxT("&Item1"));
```

```
p_menuItem->AppendSeparator();
```

```
wxMenu *p_menuSubitem = new wxMenu;  
p_menuSubitem->Append(wxID_ANY, wxT("Subitem 1"));  
p_menuSubitem->Append(wxID_ANY, wxT("Subitem 2"));
```

```
p_menuItem->AppendSubMenu(p_menuSubitem, wxT("&Item2"));
```

```
p_menubar->Append(p_menuItem, wxT("&Item"));  
SetMenuBar(p_menubar);
```



Задание

Написать программу с использованием wxWidgets, которая рисует схмотехнические элементы: резистор, биполярный pnp-транзистор, источник тока. Информация о выбранном элементе и его положении сохраняется в момент закрытия программы и подгружается в момент её запуска.

В программе должны быть пункты меню File и Element.

В меню File следующие пункты:

- «Open...» - загрузить файл с информацией об элементе (с вызовом диалога)
- «Save As...» - сохранить информацию об элементе в файл (с вызовом диалога)
- Разделитель
- «Exit» - выход с подтверждением

В меню Element следующие пункты:

- resistor – при выборе пункта меню программа рисует резистор
- npn BJT – при выборе пункта меню программа рисует биполяр
- current source – при выборе пункта меню программа рисует источник тока

Когда двигаем мышкой – в строке состояния выводятся координаты курсора.

Когда кликаем мышкой – меняется положение рисуемого элемента.