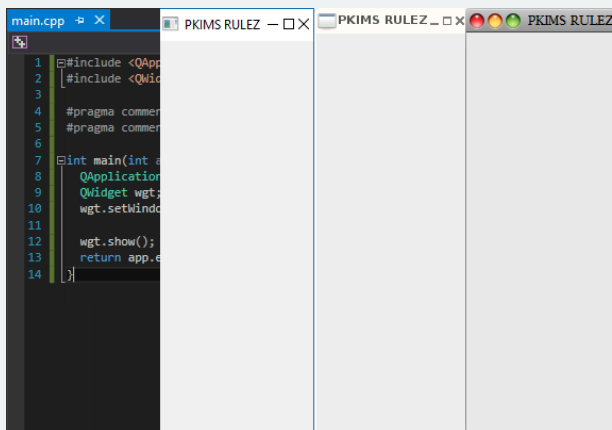




Кроссплатформенная разработка программного обеспечения



Лабораторная работа №5

Основы работы с библиотекой wxWidgets.

Часть 2.



Меню и панели инструментов: работа с изображениями

```
wxImage::AddHandler(new wxPNGHandler);  
wxBitmap bmp_exit(wxT("exit.png"), wxBITMAP_TYPE_PNG);  
  
wxToolBar *p_toolbar = CreateToolBar();  
p_toolbar->AddTool(wxID_EXIT, bmp_exit, wxT("Exit application"));  
p_toolbar->Realize();
```



Использование изображений в формате XPM

...

```
#include "exit.xpm"
```

...

```
wxImage::AddHandler(new wxPNGHandler);
```

```
wxBitmap bmp_exit(wxT("exit.png"), wxBITMAP_TYPE_PNG);
```

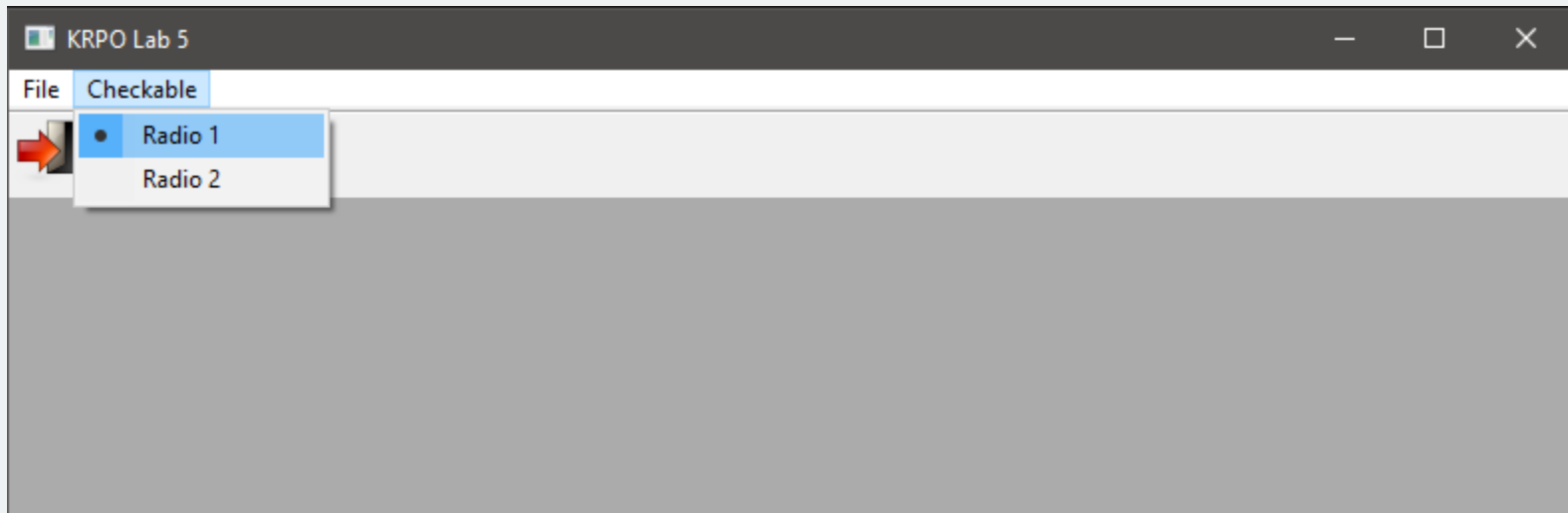
```
wxToolBar *p_toolbar = CreateToolBar();
```

```
p_toolbar->AddTool(wxID_EXIT, exit_xpm, wxT("Exit application"));
```

```
p_toolbar->Realize();
```


Управление поведением элементов меню (1)

```
wxMenu *p_menuRadio = new wxMenu;  
p_menuRadio->AppendRadioItem(wxID_RADIO1, wxT("&Radio 1"));  
p_menuRadio->AppendRadioItem(wxID_RADIO2, wxT("&Radio 2"));  
p_menubar->Append(p_menuRadio, wxT("&Checkable"));
```





Управление поведением элементов меню (2)

```
Connect(wxID_RADIO1, wxEVT_COMMAND_MENU_SELECTED,  
        wxCommandEventHandler(Frame::OnMenu_Chackable_SetRadio1));  
  
Connect(wxID_RADIO2, wxEVT_COMMAND_MENU_SELECTED,  
        wxCommandEventHandler(Frame::OnMenu_Chackable_SetRadio2));  
...
```

```
void OnMenu_Chackable_SetRadio1(wxCommandEvent &e);
```

```
void OnMenu_Chackable_SetRadio2(wxCommandEvent &e);
```



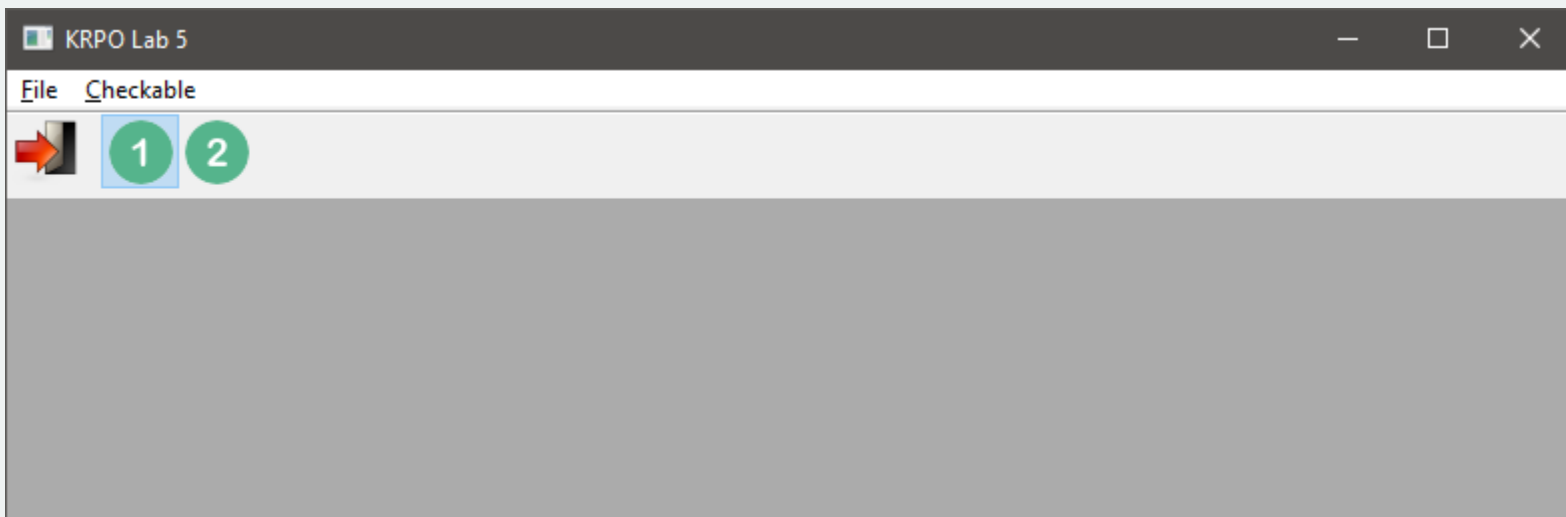
Управление поведением элементов меню (3)

```
Connect(wxID_RADIO1, wxEVT_COMMAND_MENU_SELECTED,  
        wxCommandEventHandler(Frame::OnMenu_Chackable_SetRadio1));  
  
Connect(wxID_RADIO2, wxEVT_COMMAND_MENU_SELECTED,  
        wxCommandEventHandler(Frame::OnMenu_Chackable_SetRadio2));  
...
```

```
void Frame::OnMenu_Chackable_SetRadio(wxCommandEvent &evt) {  
    switch (evt.GetId()) {  
        case wxID_RADIO1:  
            wxMessageBox(wxT("Radio 1!"));  
            break;  
        case wxID_RADIO2:  
            wxMessageBox(wxT("Radio2!"));  
            break;  
    }  
}
```

Синхронизация состояний элементов управления (1)

```
wxToolBar *p_toolbar = CreateToolBar();  
  
p_toolbar->AddTool(wxID_EXIT, wxT("Exit application"),  
                 bmp_exit, wxT("Exit application"));  
  
p_toolbar->AddSeparator();  
p_toolbar->AddRadioTool(wxID_RADIO1, wxT("Radio1"), bmp_r1);  
p_toolbar->AddRadioTool(wxID_RADIO2, wxT("Radio2"), bmp_r2);  
p_toolbar->Realize();
```





Синхронизация состояний элементов управления (2)

```
class Frame : public wxFrame {  
    int radioid;  
    ...  
    void OnMenu_Radio1Update(wxUpdateUIEvent &evt);  
    void OnMenu_Radio2Update(wxUpdateUIEvent &evt);  
};
```

```
Connect(wxID_RADIO1, wxEVT_UPDATE_UI,  
        wxUpdateUIEventHandler(Frame::OnMenu_Radio1Update));
```

```
Connect(wxID_RADIO2, wxEVT_UPDATE_UI,  
        wxUpdateUIEventHandler(Frame::OnMenu_Radio2Update));
```



Синхронизация состояний элементов управления (3)

```
void Frame::OnMenu_Radio1Update(wxUpdateUIEvent &evt) {  
    evt.Check(1 == radioid);  
}
```

```
void Frame::OnMenu_Radio2Update(wxUpdateUIEvent &evt) {  
    evt.Check(2 == radioid);  
}
```



Синхронизация состояний элементов управления (4)

```
void Frame::OnMenu_Radio1Update(wxUpdateUIEvent &evt) {  
    evt.Check(1 == radioid);  
}
```

```
void Frame::OnMenu_Radio2Update(wxUpdateUIEvent &evt) {  
    evt.Check(2 == radioid);  
}
```

```
void Frame::OnMenu_Chackable_SetRadio(wxCommandEvent &evt) {  
    switch (evt.GetId()) {  
        case wxID_RADIO1:  
            wxMessageBox(wxT("Radio 1!"));  
            radioid = 1;  
            break;  
        case wxID_RADIO2:  
            wxMessageBox(wxT("Radio2!"));  
            radioid = 2;  
            break;  
    }  
}
```



Создание пользовательских диалогов в wxWidgets

```
class Dialog : public wxDialog {
public:
    Dialog(wxWindow *parent);
};
```

```
Dialog::Dialog(wxWindow *parent)
    : wxDialog(parent, wxID_ANY, wxT("About"),
        wxDefaultPosition,
        wxSize(300, 200)) {

    CenterOnScreen();
}
```



Вызов пользовательских диалогов в wxWidgets

```
#include "dialog.h"
```

```
...
```

```
void Frame::OnMenu_Help_About(wxCommandEvent &e) {  
    Dialog dlg(this);  
    dlg.ShowModal();  
}
```

Использование элементов GUI: wxStaticText

```
wxStaticText *p_text = new wxStaticText(this,  
                                         wxID_ANY,  
                                         wxT("Я тут буду всегда"),  
                                         wxPoint(20, 50));
```



```
new wxStaticText(this,  
                 wxID_ANY,  
                 wxT("Я тут буду всегда"),  
                 wxPoint(20, 50));
```



Где располагать указатель на компонент и нужен ли он?

```
class Dialog : public wxDialog {  
    wxStaticText *p_text;  
  
    ...  
};
```

```
Dialog::Dialog(...) {  
    p_text = new wxStaticText(...
```

```
Dialog::Dialog(...) {  
    wxStaticText *p_text = new wxStaticText(...
```

```
Dialog::Dialog(...) {  
    new wxStaticText(...
```


Использование менеджеров компоновки в wxWidgets (1)

```
wxBoxSizer *p_box = new wxBoxSizer(wxVERTICAL);
```

```
wxButton *p_btnOk = new wxButton(this, wxID_ANY, wxT("Ok"));
```

```
wxButton *p_btnCancel = new wxButton(this, wxID_ANY, wxT("Cancel"));
```

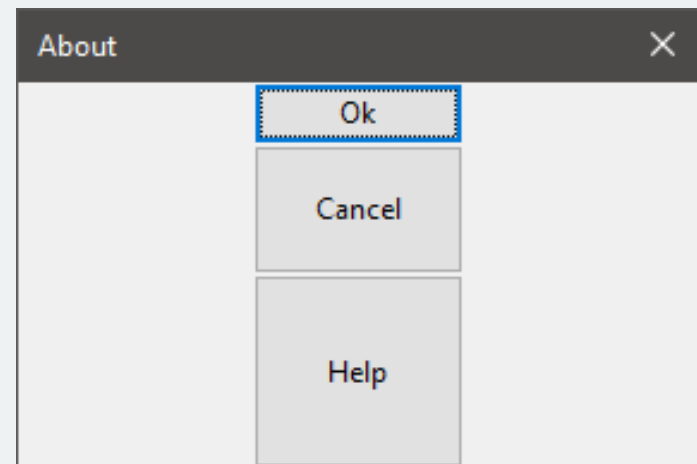
```
wxButton *p_btnHelp = new wxButton(this, wxID_ANY, wxT("Help"));
```

```
p_box->Add(p_btnOk, 1, wxCENTER);
```

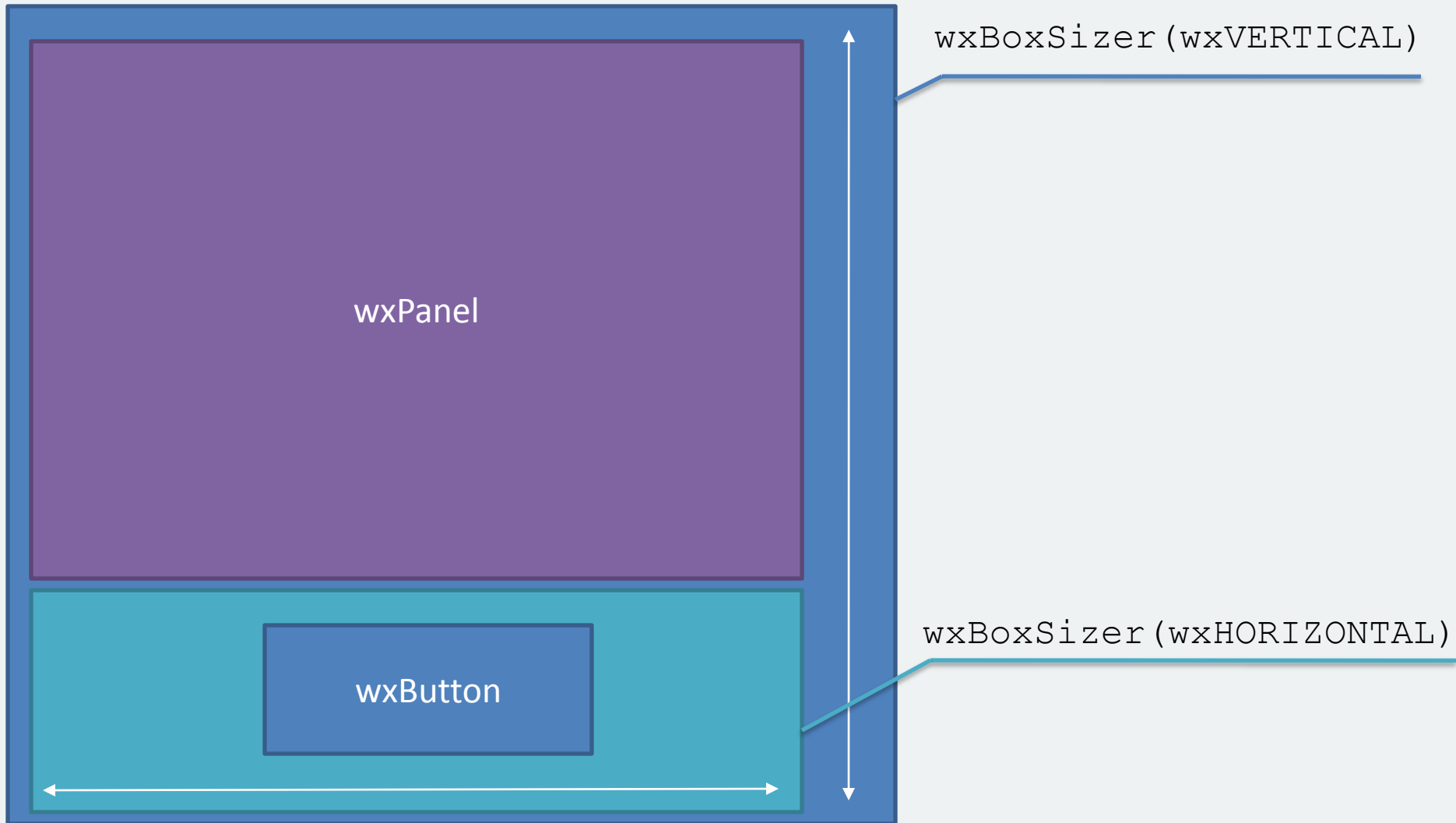
```
p_box->Add(p_btnCancel, 2, wxCENTER);
```

```
p_box->Add(p_btnHelp, 3, wxCENTER);
```

```
SetSizer(p_box);
```

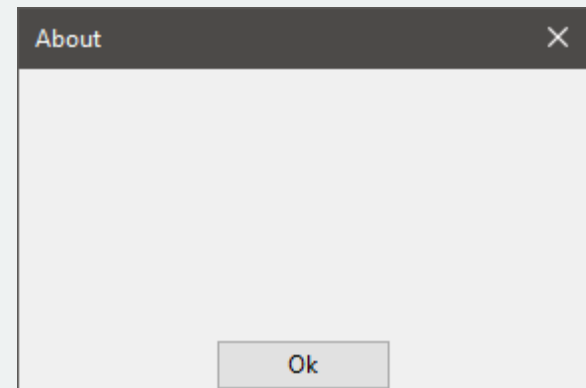


Использование менеджеров компоновки в wxWidgets (2)



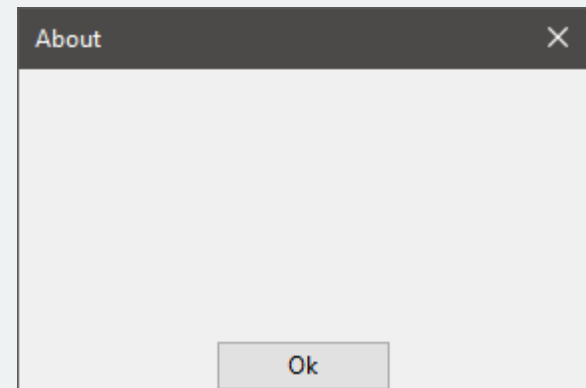
Использование менеджеров компоновки в wxWidgets (3)

```
wxBoxSizer *p_vbox = new wxBoxSizer(wxVERTICAL);  
wxBoxSizer *p_hbox = new wxBoxSizer(wxHORIZONTAL);  
  
wxPanel *p_panel = new wxPanel(this, wxID_ANY);  
  
wxButton *p_btnOk = new wxButton(this, wxID_ANY, wxT("Ok"));  
  
p_hbox->Add(p_btnOk, 1, wxCENTER);  
  
p_vbox->Add(p_panel, 1, wxEXPAND | wxALL);  
p_vbox->Add(p_hbox, 0, wxCENTER);  
  
SetSizer(p_vbox);
```



Использование менеджеров компоновки в wxWidgets (3)

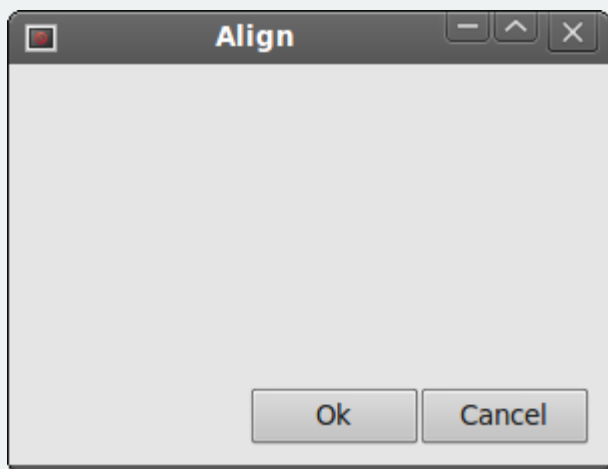
```
wxBoxSizer *p_vbox = new wxBoxSizer(wxVERTICAL);  
wxBoxSizer *p_hbox = new wxBoxSizer(wxHORIZONTAL);  
  
wxPanel *p_panel = new wxPanel(this, wxID_ANY);  
wxButton *p_btnOk = new wxButton(this, wxID_OK, wxT("Ok"));  
  
p_hbox->Add(p_btnOk, 1, wxCENTER);  
  
p_vbox->Add(p_panel, 1, wxEXPAND | wxALL);  
p_vbox->Add(p_hbox, 0, wxCENTER);  
  
SetSizer(p_vbox);
```



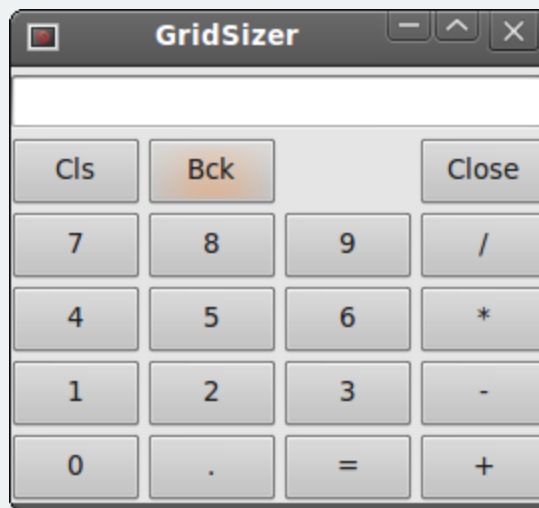


Менеджеры компоновки wxWidgets

wxBoxSizer



wxGridSizer



wxFlexGridSizer



Стандартные диалоги в wxWidgets : работа с файлами

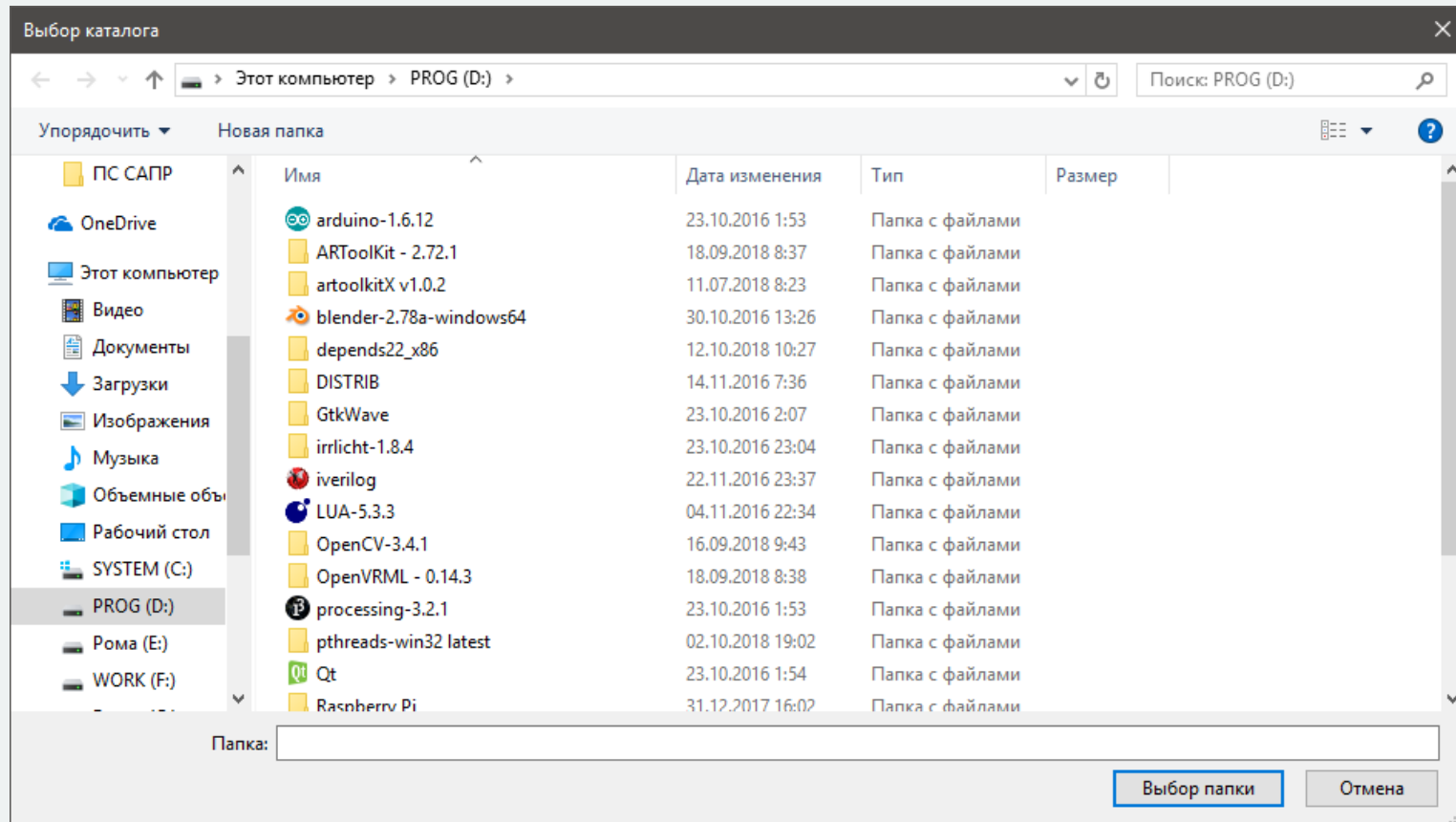
```
wxFileDialog dlg(this,  
                wxT("Choose file to open"),  
                wxEmptyString,  
                wxEmptyString,  
                wxT("Text files (*.txt)|*.txt|All|*.*"),  
                wxFD_OPEN,  
                wxDefaultPosition);  
  
if (dlg.ShowModal() == wxID_OK)  
    wxMessageBox(dlg.GetPath(), wxT("Open file"),  
                wxOK | wxICON_INFORMATION);
```

```
42  enum  
43  {  
44      wxFD_OPEN           = 0x0001,  
45      wxFD_SAVE          = 0x0002,  
46      wxFD_OVERWRITE_PROMPT = 0x0004,  
47      wxFD_NO_FOLLOW     = 0x0008,  
48      wxFD_FILE_MUST_EXIST = 0x0010,  
49      wxFD_CHANGE_DIR   = 0x0080,  
50      wxFD_PREVIEW      = 0x0100,  
51      wxFD_MULTIPLE     = 0x0200  
52  };  
53
```

Стандартные диалоги в wxWidgets : выбор каталога

```
wxDirDialog dlg(this, wxT("Выбор каталога"), wxT("D:\\\\"));
```

```
if (dlg.ShowModal() == wxID_OK)  
    wxMessageBox(dlg.GetPath(), wxT("Path"),  
                wxOK | wxICON_INFORMATION);
```





Задание на лабораторную работу

Реализовать программу чтения и визуализации результатов моделирования в одном из форматов.

| Номер варианта | Поддерживаемый формат |
|-----------------|-----------------------|
| 1, 4, 7, 10, 13 | PSF |
| 2, 5, 8, 11, 14 | CSV |
| 3, 6, 9, 12, 15 | CSDF |



Форматы выходных файлов: PSF

HEADER

```
"PSFversion" "1.00"  
"simulator" "KSI"  
"runtype" "Transient Analysis"
```

TYPE

```
"node" FLOAT DOUBLE PROP(  
"key" "node"  
)  
"branch" FLOAT DOUBLE PROP(  
"key" "branch"  
)  
"sweep" FLOAT DOUBLE
```

SWEEP

```
"time" "sweep"
```

TRACE

```
"group" GROUP 3  
"1" "node"  
"0" "node"  
"2" "node"
```

VALUE

```
"time" 0.000000e+00  
"group"  
0.000000e+00  
0.000000e+00  
0.000000e+00  
"time" 1.000000e-09  
"group"  
0.000000e+00  
0.000000e+00  
0.000000e+00  
"time" 2.000000e-09
```

Форматы выходных файлов: CSV

```
; Additional parameters:  
; Temperature (TEMP) = 2.50000000e+001  
; Temperature (TNOM) = 2.50000000e+001  
; Local inaccuracy = 9.99999700e-003  
; reltol = 1.00000000e-003  
; Acceleration level = without acceleration
```

| TIME | 'v(1)' | 'v(2)' |
|-----------------|-----------------|-----------------|
| 0.00000000e+000 | 0.00000000e+000 | 0.00000000e+000 |
| 2.00000000e-009 | 0.00000000e+000 | 0.00000000e+000 |
| 5.00000000e-009 | 0.00000000e+000 | 0.00000000e+000 |
| 5.02386518e-009 | 5.96629438e-002 | 1.39067797e-003 |
| 5.07159553e-009 | 1.78988831e-001 | 6.88855948e-003 |
| 5.12889016e-009 | 3.22225398e-001 | 2.04634393e-002 |
| 5.18236485e-009 | 4.55912118e-001 | 3.96611966e-002 |
| 5.28931422e-009 | 7.23285558e-001 | 9.54911926e-002 |
| 5.42992725e-009 | 1.07481814e+000 | 2.01060109e-001 |
| 5.60837057e-009 | 1.52092642e+000 | 3.80746951e-001 |
| 5.81707047e-009 | 2.04267617e+000 | 6.45518006e-001 |
| 6.06470662e-009 | 2.66176655e+000 | 1.02159440e+000 |
| 6.35438100e-009 | 3.38595250e+000 | 1.52822094e+000 |
| 6.69637078e-009 | 4.24092695e+000 | 2.19562050e+000 |
| 7.00000000e-009 | 5.00000000e+000 | 2.83483203e+000 |
| 7.14454208e-009 | 5.00000000e+000 | 3.10826704e+000 |
| 7.38134698e-009 | 5.00000000e+000 | 3.50881305e+000 |
| 7.85495678e-009 | 5.00000000e+000 | 4.07983339e+000 |
| 8.25644207e-009 | 5.00000000e+000 | 4.38750412e+000 |
| 8.67525042e-009 | 5.00000000e+000 | 4.59960723e+000 |
| 9.11298514e-009 | 5.00000000e+000 | 4.74340123e+000 |



Форматы выходных файлов: CSDF

```
#H
SOURCE='SYMSPICE'
TITLE='* # file name: F:\rc
SUBTITLE=''
TIME='08:37:36' DATE='9/11/2018'
ANALYSIS='TR'
TEMPERATURE=' 2.500000E+001'
SWEEPVAR='TIME'
COMPLEXVALUES='NO' FORMAT='1 VOLTSorAMPS;EFLOAT'
XBEGIN=' 0.000000e+000' XEND=' 1.000000e-007'
NODES='      2'
#N 'v(1)' 'v(2)'

#C      0.00000000e+000      2      0.00000000e+000      0.00000000e+000
#C      2.00000000e-009      2      0.00000000e+000      0.00000000e+000
#C      5.00000000e-009      2      0.00000000e+000      0.00000000e+000
#C      5.02386518e-009      2      5.96629438e-002      1.39067797e-003
#C      5.07159553e-009      2      1.78988831e-001      6.88855948e-003
#C      5.12889016e-009      2      3.22225398e-001      2.04634393e-002
#C      5.18236485e-009      2      4.55912118e-001      3.96611966e-002
#C      5.28931422e-009      2      7.23285558e-001      9.54911926e-002
#C      5.42992725e-009      2      1.07481814e+000      2.01060109e-001
```