Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине «Проектирование программ в интеллектуальных системах»

Тема: «Создание интерфейса оконного windows-приложения (окна, диалоговые окна). Автоматический каркас приложения»

**Выполнили:**

Студенты 2 курса

Группы ИИ-21

Литвинюк Т. В.

Кабак Д. Н.

**Проверил:**

Муравьев Г. Л.

Брест 2023

**Цель работы:** 1. Изучить создание Windows-приложений с использований каркаса Hello (или аналогичного каркаса). 2. Изучить технологию работы с диалоговыми окнами и элементами управления (ЭУ). Применением диалогового окна в составе главного, в качестве главного окна. 3. Изучить использование кнопок, окон редактирования.

**Ход работы:**

Каркас Hello похож на типовой каркас приложения, однако он по-другому структурирован, разбит на функции. К стандартному окну с командным меню добавляется пользовательское меню с пунктами Файл, Помощь. В шаблон уже автоматически внесены инструкции для работы с диалоговым окном и пунктами меню.

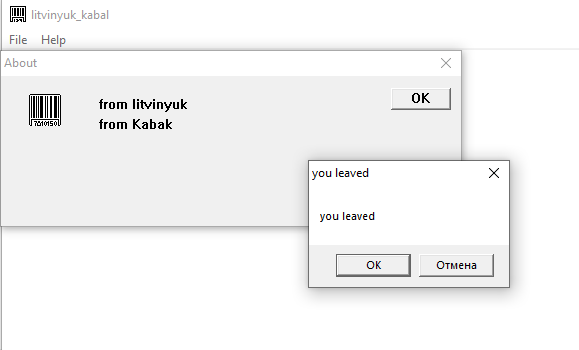
Структура файлов:

* Файл "Hello.cpp" — это файл исходного кода, который содержит функцию main
* Файл "Hello.h" — это заголовочный файл, который содержит объявления функций и структур данных, используемых в приложении. Здесь мы можем определить свои собственные типы данных и функции.
* Файл "stdafx.h" — это файл предкомпилированных заголовков, который ускоряет процесс компиляции проекта. В этом файле содержатся объявления стандартных библиотек и других общих заголовков, которые будут использоваться в проекте.
* Файл "stdafx.cpp" — это файл, который содержит код, используемый при предварительной компиляции заголовков.
* Файл "Hello.vcxproj" — это файл проекта Visual Studio, который содержит информацию о настройках проекта, таких как конфигурации, пути к файлам и настройки компилятора.
* Файлы "Hello.rc" — это файл, который содержит информацию о ресурсах приложения.

1. Создать каркас. Изменить интерфейс приложения (§ 4.2.1-4.2.3 [3]): - изменить названия пунктов меню (вкладка Resource); - поменять название главного окна на свою фамилию; - изменить текст в окне-справке; - по нажатию в окне-справке кнопки ОК выводить окно MessageBox с сообщением выхода из окна-справки; - по завершении работы с приложением выводить окно MessageBox с соответствующим предупреждением.

LRESULT CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)

{

, switch (message)

{

case WM\_INITDIALOG:

return TRUE;

case WM\_COMMAND:

if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)

{

MessageBox(NULL, "you leaved ", "you leaved", 1);

EndDialog(hDlg, LOWORD(wParam));

return TRUE;

}

break;

}

return FALSE;

}

Во вкладке resourse можно легко редактировать свойства окон, кнопок, полей. Поменять их расположения, задать ID.

1. Создать приложение, в котором подсчитываются события - нажатие левой клавиши мыши. А результат - число нажатий - выводится в секции WM\_PAINT только по нажатии правой клавиши мыши.

int num=0;

switch (message)

{

case WM\_COMMAND:

wmId = LOWORD(wParam);

wmEvent = HIWORD(wParam);

// Parse the menu selections:

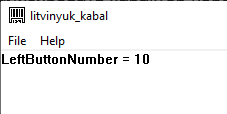
switch (wmId)

{

case IDM\_ABOUT:

DialogBox(hInst, (LPCTSTR)IDD\_ABOUTBOX, hWnd, (DLGPROC)About);

break;

 case IDM\_EXIT:

DestroyWindow(hWnd);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

}

break;

case WM\_LBUTTONDOWN:

LeftButtonNumber++;

break;

case WM\_RBUTTONDOWN:

InvalidateRect(hWnd,NULL,true);

break;

case WM\_PAINT:

if(GetKeyState(VK\_RBUTTON) &0x80)

{

hdc = BeginPaint (hWnd, &ps);

char Str[20];

wsprintf(Str, "LeftButtonNumber = %d", LeftButtonNumber);

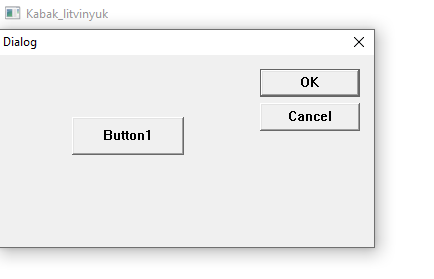
TextOut(hdc, 0, 0, Str, strlen(Str));

ValidateRect (hWnd,NULL);

EndPaint (hWnd, &ps);

}

Для отслеживания нажатий левой кнопкой мыши, используется константа WM\_LBUTTONDOWN. При нажатии ПКМ, происходит перерисовка окна, благодаря блоку case WM\_PAINT.

1. Создать приложение, при запуске которого в качестве главного выводится масштабируемое окно с рамкой и клиентской областью, а за ним сразу же автоматически диалоговое окно с кнопками. При этом, главное окно остается на экране, но теряет активность. Нажатие кнопок диалогового окна приводит к его закрытию.

LRESULT CALLBACK callback2 ( HWND hDlg, UINT Message, WPARAM wParam, LPARAM lParam){

switch (Message) {

case WM\_INITDIALOG:

return FALSE;

case WM\_COMMAND:

switch (wParam) {

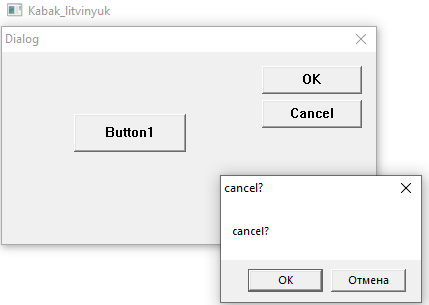
case IDOK:

if (MessageBox(NULL, "ok?", "ok?", 1) == IDOK)

EndDialog(hDlg,TRUE);

break;

case IDCANCEL:

 PostQuitMessage (0);

if (MessageBox(NULL, "cancel?", "cancel?", 1) == IDOK)

EndDialog(hDlg,FALSE);

break;

case IDC\_BUTTON1:

if (MessageBox(NULL, "cancel?", "cancel?", 1) == IDOK)

EndDialog(hDlg,FALSE);

break;

default:

return FALSE;

}

break;

default:

return FALSE;

}

return TRUE;

};

1. Создать приложение, при запуске которого в качестве главного выводится диалоговое окно. Нажатие его кнопок приводит к выводу соответствующих сообщений, а нажатие Cancel - к его закрытию.

#include "stdafx.h"

#include "resource.h"

LRESULT CALLBACK callback (HWND hDlg, UINT Message, WPARAM wParam, LPARAM lParam);

HINSTANCE hInst;

int APIENTRY WinMain(HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPSTR lpCmdLine,

int nCmdShow)

{

DialogBox(hInst, (LPCTSTR) IDD\_DIALOG1, NULL, (DLGPROC) callback ) ;

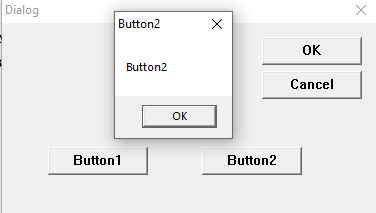
return 0;

}

LRESULT CALLBACK callback (HWND hDlg, UINT Message, WPARAM wParam, LPARAM lParam)

{

switch (Message)

 {

case WM\_INITDIALOG:

return FALSE;

case WM\_COMMAND:

switch (wParam)

{

case IDOK:

MessageBox(NULL, "OK", "OK", 0);

break;

case IDCANCEL:

EndDialog(hDlg, false);

break;

case IDC\_BUTTON1:

MessageBox(NULL, "Button1", "Button1", 0);

break;

case IDC\_BUTTON2:

MessageBox(NULL, "Button2", "Button2", 0);

break;

default:

return FALSE;

}

break;

default:

return FALSE;

}

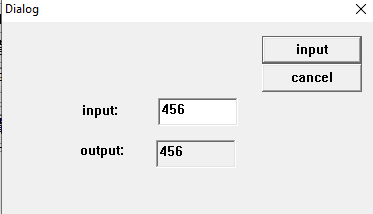
return TRUE;

}

1. Создать приложение, при запуске которого в качестве главного должно выводиться диалоговое окно с окнами редактирования, подсказками, кнопкам), предназначенное для задания исходных данных и их эхо-вывода. По кнопке Ввести производится ввод из поля ввода и эхо-вывод введенного значения.

LRESULT CALLBACK callback (HWND hDlg, UINT Message, WPARAM wParam, LPARAM lParam)

{

 switch (Message)

{

case WM\_INITDIALOG:

return FALSE;

case WM\_COMMAND:

switch (wParam)

{

case IDC\_BUTTON1:

char str[256];

GetDlgItemText(hDlg,IDC\_EDIT1, str, 15);

SetDlgItemText(hDlg,IDC\_EDIT2, str);

break;

case IDCANCEL:

EndDialog(hDlg,TRUE);

PostQuitMessage(0);

break;

default:

return FALSE;

}

break;

default:

return FALSE;

}

return TRUE;

}

**Вывод:** ознакомился с каркасом windows-приложений типа Hello. Научился работать с диалоговыми окнами, кнопками и другими элементами интерфейса. Научился применять диалоговое окно в качестве главного.