Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

По дисциплине «Проектирование программ в интеллектуальных системах»

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

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Муравьев Г.Л.

Брест 2023

**Цель работы:** 1. Изучить использование типовых ЭУ (кнопок, окон редактирования, списков). 2. Изу-чить управление меню, технологию создания интерфейсов с использованием меню и диалоговых окон. 3. Изучить типовые диалоговые окна. 4. Изучить создание приложе-ний на базе каркаса Hello.

**Ход работы:**

2. Разработать приложение с диалоговым окном в качестве главного окна (на базе ТКП) с единственной кнопкой ОК и без системных кнопок. Его функция - создать окно с клиентской областью и передать ему управление. При этом диалоговое окно должно остаться на экране. Окно типа главное воспринимает сообщение нажатия левой кнопки мыши (действие – вывод информационного сообщения о событии; закрытие приложения). Приложение также может быть закрыто и системной кнопкой.

Модификации: а) блокировка кнопки ОК ДО при запуске ГО; б) попеременное нажа-тие кнопки ОК должно приводить к визуализации-девизуализации ГО.

Для визуализации клиентского окна необходимо создать его экземпляр функцией CreateWindow и получить новый дескриптор. Соответственно в WinMain( ) следует создать стиль второго окна, создать его экземпляр, создать и активизировать клиентское окно. Нельзя забывать использовать функции ShowWindow, UpdateWindow, иначе окно мы не увидим.Для визуализации-девизуализации можно использовать ShowWindow («Дескриптор окна», «0- спрятать 1- показать»). Для блокировки кнопки можно использовать EnableWindow(«Декскриптор кнопки», «True или False»).

3. Разработать приложение с главным окном. Его функция – запуск при нажатии левой кнопки мыши диалогового окна, содержащего кнопку завершения работы приложения. При запуске ДО разрушать ГО без завершения работы приложения. Собственное разрушение ГО должно быть заблокировано.

Для блокировки закрытия окна,достаточно убрать системные кнопки и функции в обработчике при сообщении WM\_DESTROY,отвечающие за закрытие приложения. Для закрытия окна можно использовать функцию DestroyWindow(), либо же спрятать путем использования функции ShowWindow().

5. Воспроизвести демонстрационный пример..

ЗАДАНИЕ (демонстрационный пример). Разработать приложение на базе ТКП с окном с рамкой в качестве главного, содержащее простейшее пользовательское меню.

Примерный вид интерфейса: два пункта типа POPUP – ВВОД, ВЫВОД, четыре пункта типа MENUITEM – Строка 1, Строка 2 (подпункты пункта ВВОД), Строка 1, Строка 2 (подпункты пункта ВЫВОД).

Данное меню легко создается в ResourceView, путем добавления его через Insert и дальнейшей настройкой (добавлением элементов, присвоения определенных ID, их видов и т.д.). Главное не забыть добавить в WNDCLASSEX его ID, пример: wcex.lpszMenuName=(LPCSTR)IDC\_LAB3\_5.

6. Внести изменения в меню (- подпункт “Строка2” пункта ”ВВОД” переименовать в “Выход” и генерировать сообщение для завершения работы приложения; - по выбору каждого конечного пункта меню выводить комментирующее сообщение; - подпункт “Строка1” пункта ”ВЫВОД” сделать типа POPUP и далее добавить список подпунктов “Дополнительный1”, “ Дополнительный2”).

Изменения эффективней всего будет делать в ResourceView. Обработать сообщение переданное программе по поводу пункта меню делается через просмотр LOWORD(wParam),который передаст ID пункта меню(лучше всего просматривать через Switch->сase).

7. Использование диалогового окна со списком и окошком редактирования в качестве главного

ЗАДАНИЕ (демонстрационный пример). Разработать приложение с диалоговым окном в роли главного. Цель приложения: управление списком строк – фамилий (просмотр, выбор, добавление, удаление, редактирование, расположение в алфавитном порядке). При запуске приложения в качестве главного окна выводится диалоговое, содержащее пустой список, кнопки управления списком и кнопку Завершить (Cancel). Состав ресурсов: - список, окно редактирования, кнопки, рамки – ЭУ типа Group Box и т.д.

Реализуется окно в ResourceView, путем добавления шаблонного диалогового окна и его изменения. Желательно ID ЭУ связать с их действиями для удобство чтения кода программы. Использовались элементы: EditBox,ButtonBox,GroupBox,List,Box. Работа с некоторыми ЭУ производится с помощью функции: SendDlgItemMessage(«Дескриптор окна», «Дескриптор ЭУ», «ID действия»,wParam,lParam). Для работы с ListBox пригодились действия: LB\_GETCOUNT, LB\_GETCURSEL, LB\_ADDSTRING, LB\_DELETESTRING, LB\_GETTEXT.

8. Модифицировать приложение для работы со списком строк, обеспечив константную инициализацию списка заранее заданными строками (фамилиями).

Листинг Программы:

#include "stdafx.h"

#include "resource.h"

LRESULT CALLBACK About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine, int nCmdShow){DialogBox(hInstance, (LPCTSTR)IDD\_LIST, NULL, (DLGPROC)About); return 0;}

char buffer[256];

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

int wmId = LOWORD(wParam); int wmEvent = HIWORD(wParam);

int RecordsAmount = SendDlgItemMessage(hDlg, IDC\_LIST, LB\_GETCOUNT, 0, 0);

int currentNumber = SendDlgItemMessage(hDlg, IDC\_LIST, LB\_GETCURSEL, 0, 0);

switch (message){

case WM\_INITDIALOG:

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_ADDSTRING,0,(LPARAM)"First");

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_ADDSTRING,0,(LPARAM)"Second");

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_ADDSTRING,0,(LPARAM)"Third");

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_ADDSTRING,0,(LPARAM)"Fourth");

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_ADDSTRING,0,(LPARAM)"Fifth");

return TRUE;

case WM\_COMMAND:

switch (wmId) {

case IDOK:

case IDCANCEL:

EndDialog(hDlg, LOWORD(wParam));

PostQuitMessage(0);

return TRUE;

break;

case IDC\_ADD:

GetDlgItemText(hDlg,IDC\_EDIT,buffer,256);

if(strlen(buffer) != 0){

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_ADDSTRING,0,(LPARAM)buffer);} break;

case IDC\_DELETE:

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_DELETESTRING,currentNumber,0);break;

case IDC\_CHANGE:

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_GETTEXT,currentNumber,(LPARAM)buffer);

SendDlgItemMessage(hDlg,IDC\_LIST,LB\_DELETESTRING,currentNumber,0);

SetDlgItemText(hDlg,IDC\_EDIT,buffer);break;}break;}

return FALSE;}

Данное задание делается дублированием команды SendDlgItemMessage(«Деструктор окна», «Деструктор ЭУ»,LB\_ADDSTRING,0,(LPARAM) "Чья-то Фамилия ").

9. Изучить теоретический материал по совместному использованию диалоговых окон и меню.

ЗАДАНИЕ (демонстрационный пример). Разработать приложение на базе ТКП для многократного ввода-вывода строк и фиксации числа введенных строк. Создать интерфейс на основе окна с рамкой и меню. Через меню вызывается диалоговое окно для ввода и окно сообщения для вывода строки. В клиентскую область глав-ного окна выводится число введенных строк. Вид интерфейса показан ниже. Главное окно содержит меню с двумя пунктами. По пункту ВВОД выводится диало-говое окно ввода строки. По пункту ВЫВОД выводится строка и количество строк.

Для чтения с EditBox используется функция GetDlgItemText(). Для закрытия диалоговых окон используется EndDialog().Остальные функции были объяснены в прошлых заданиях.

**Вывод:** 1. Изучил использование типовых ЭУ (кнопок, окон редактирования, списков). 2. Изучил управление меню, технологию создания интерфейсов с использованием меню и диалоговых окон. 3. Изучил типовые диалоговые окна. 4. Изучил создание приложений на базе каркаса Hello.