**《Windows应用程序原理》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | | |  |  |
|  | |  | | |  |  |
| **实验名称** | 实验四 编写Windows窗口应用程序实现对话框与菜单的应用编程（综合实验） | | | | | |
| **实验时间** | 2024.5.7 | | **实验地点** | DS3304 | | |
| **实验性质** | | | **□验证性 ☑设计性 □综合性** | | | |
| 一、实验目的  1.熟悉Windows窗口应用程序开发环境和工具。  2.掌握如何改变应用程序窗口的外观。  3.学会如何使用对话框来与用户进行交互。  4.学会使用Windows窗口应用程序的基本控件和事件处理机制。  5.通过编写Windows窗口应用程序，掌握对话框和菜单的应用编程技能。 | | | | | | |
| 二、实验项目内容  1. 创建一个包含菜单的Windows窗口应用程序；  2. 在菜单中添加“new”和“About”两个菜单项；  3. 当用户点击“new”菜单项时，弹出一个新建对话框（非模态对话框），允许用户改变字体和背景色  4. 在新建对话框中添加一个按钮，当用户点击按钮时，应用程序窗口的字体和背景色应发生相应变化。  5. 当用户点击“About”菜单项时，弹出一个关于对话框（模态对话框）。 | | | | | | |
| 1. 实验过程或算法（源程序）   1.targetver.h  #pragma once  #include <SDKDDKVer.h>  2.resource.h  #define IDS\_APP\_TITLE 103  #define IDR\_MENU 101  #define IDD\_NEW\_DIALOG 102  #define ID\_FILE\_NEW 1001  #define ID\_HELP\_ABOUT 1002  #define ID\_COLOR\_BUTTON 1003  #define IDR\_MAINFRAME 128  #define IDD\_WINDOWMENU\_DIALOG 102  #define IDD\_ABOUTBOX 103  #define IDM\_ABOUT 104  #define IDM\_EXIT 105  #define IDI\_WINDOWMENU 107  #define IDI\_SMALL 108  #define IDC\_WINDOWMENU 109  #define IDC\_MYICON 2  #ifndef IDC\_STATIC  #define IDC\_STATIC -1  #endif  #ifdef APSTUDIO\_INVOKED  #ifndef APSTUDIO\_READONLY\_SYMBOLS  #define \_APS\_NO\_MFC 130  #define \_APS\_NEXT\_RESOURCE\_VALUE 129  #define \_APS\_NEXT\_COMMAND\_VALUE 32771  #define \_APS\_NEXT\_CONTROL\_VALUE 1000  #define \_APS\_NEXT\_SYMED\_VALUE 110  #endif  #endif  3.frame.h  #define WIN32\_LEAN\_AND\_MEAN // 从 Windows 头文件中排除极少使用的内容  // Windows 头文件  #include <windows.h>  // C 运行时头文件  #include <stdlib.h>  #include <malloc.h>  #include <memory.h>  #include <tchar.h>  4.shiayn4  #include <time.h>  #define MAX\_LOADSTRING 100  LRESULT CALLBACK NewWndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam);  HINSTANCE hInst;  TCHAR szTitle[MAX\_LOADSTRING];  TCHAR szWindowClass[MAX\_LOADSTRING];  HWND hWnd;  HWND newdlghWnd;  HBRUSH hmainBrush = NULL;  ATOM MyRegisterClass(HINSTANCE hInstance);  BOOL InitInstance(HINSTANCE, int);  LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);  INT\_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);  int APIENTRY \_tWinMain(HINSTANCE hInstance,  HINSTANCE hPrevInstance,  LPTSTR lpCmdLine,  int nCmdShow) {  UNREFERENCED\_PARAMETER(hPrevInstance);  UNREFERENCED\_PARAMETER(lpCmdLine);  MSG msg;  HACCEL hAccelTable;  LoadString(hInstance, IDS\_APP\_TITLE, szTitle, MAX\_LOADSTRING);  LoadString(hInstance, IDC\_Q760642, szWindowClass, MAX\_LOADSTRING);  MyRegisterClass(hInstance);  if (!InitInstance(hInstance, nCmdShow)) {  return FALSE;  }  hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_Q760642));  while (GetMessage(&msg, NULL, 0, 0)) {  if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg)) {  if (!IsDialogMessage(newdlghWnd, &msg)) {  TranslateMessage(&msg);  DispatchMessage(&msg);  }  }  }  return(int)msg.wParam;  }  ATOM MyRegisterClass(HINSTANCE hInstance) {  WNDCLASSEX wcex;  wcex.cbSize = sizeof(WNDCLASSEX);  wcex.style = CS\_HREDRAW | CS\_VREDRAW;  wcex.lpfnWndProc = WndProc;  wcex.cbClsExtra = 0;  wcex.cbWndExtra = 0;  wcex.hInstance = hInstance;  wcex.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI\_Q760642));  wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);  wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW + 1);  wcex.lpszMenuName = MAKEINTRESOURCE(IDC\_Q760642);  wcex.lpszClassName = szWindowClass;  wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI\_SMALL));  return RegisterClassEx(&wcex);  }  BOOL InitInstance(HINSTANCE hInstance, int nCmdShow) {  hInst = hInstance;  newdlghWnd = NULL;  hWnd = CreateWindow(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,  CW\_USEDEFAULT, 0, CW\_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);  if (!hWnd) {  return FALSE;  }  ShowWindow(hWnd, nCmdShow);  UpdateWindow(hWnd);  return TRUE;  }  HBRUSH CHColor() {  CHOOSECOLOR cc;  static COLORREF acrCustClr[16];  HBRUSH hbrush;  static DWORD rgbCurrent;  ZeroMemory(&cc, sizeof(cc));  cc.lStructSize = sizeof(cc);  cc.hwndOwner = hWnd;  cc.lpCustColors = (LPDWORD)acrCustClr;  cc.rgbResult = rgbCurrent;  cc.Flags = CC\_FULLOPEN | CC\_RGBINIT;  if (ChooseColor(&cc) == TRUE) {  hbrush = CreateSolidBrush(cc.rgbResult);  rgbCurrent = cc.rgbResult;  return hbrush;  }  return NULL;  }  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) {  int wmId, wmEvent;  PAINTSTRUCT ps;  HDC hdc;  MSG msg;  RECT rc;  HBRUSH hbrush;  TCHAR szText[20] = TEXT("welcome!");  switch (message) {  case WM\_COMMAND:  wmId = LOWORD(wParam);  wmEvent = HIWORD(wParam);  switch (wmId) {  case IDM\_FILE\_NEW:  if (newdlghWnd == NULL)  newdlghWnd = CreateDialog(hInst, (LPCTSTR)IDD\_DIALOG1, hWnd, (DLGPROC)NewWndProc);  ShowWindow(newdlghWnd, SW\_SHOW);  SetForegroundWindow(newdlghWnd);  UpdateWindow(newdlghWnd);  break;  case IDM\_COLOR: {  HBRUSH hb = CHColor();  if (hb != NULL)hmainBrush = hb;  InvalidateRect(hWnd, NULL, TRUE);  }  break;  case IDM\_ABOUT:  DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);  break;  case IDM\_EXIT:  DestroyWindow(hWnd);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  break;  case WM\_PAINT:  hdc = BeginPaint(hWnd, &ps);  GetClientRect(hWnd, &rc);  TextOut(hdc, 100, 100, szText, wcslen(szText));  if (hmainBrush != NULL)  FillRect(hdc, &rc, (HBRUSH)hmainBrush);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  return 0;  }  LRESULT CALLBACK NewWndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) {  int wmId, wmEvent;  wchar\_t\* cppf = \_T("new1");  wchar\_t\* hf = \_T("new2");  switch (message) {  case WM\_INITDIALOG:  SendDlgItemMessage(hWnd, IDC\_LIST1, LB\_ADDSTRING, 0, (LPARAM)cppf);  SendDlgItemMessage(hWnd, IDC\_LIST1, LB\_ADDSTRING, 0, (LPARAM)hf);  break;  case WM\_COMMAND:  wmId = LOWORD(wParam);  wmEvent = HIWORD(wParam);  switch (wmId) {  case IDOK:  DestroyWindow(newdlghWnd);  newdlghWnd = NULL;  break;  }  break;  case WM\_CLOSE:  DestroyWindow(newdlghWnd);  newdlghWnd = NULL;  break;  case WM\_DESTROY:  break;  default:  return FALSE;  }  return TRUE;  }  INT\_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam) {  UNREFERENCED\_PARAMETER(lParam);  switch (message) {  case WM\_INITDIALOG:  return(INT\_PTR)TRUE;  case WM\_COMMAND:  if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL) {  EndDialog(hDlg, LOWORD(wParam));  return(INT\_PTR)TRUE;  }  break;  }  return(INT\_PTR)FALSE;  } | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程     图1 创建一个包含菜单的Windows窗口应用程序    图2 在菜单中添加“new”和“About”两个菜单项    图3 当用户点击“new”时，弹出一个新对话框（非模态对话框）    图4 在新建对话框中添加”color”按钮，当用户点击按钮时，应用程序窗口的字体颜色背景色应发生相应变化    图5 当用户点击“About”菜单项时，弹出一个关于对话框（模态对话框） | | | | | | |