Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «МОП ЭВМ»

ЛАБОРАТОРНАЯ РАБОТА 1

по дисциплине «Операционные системы и оболочки»

## Оконное приложение с использованием WinAPI 32

Студент группы 6ИСб-1 А.М. Закусило

Преподаватель В.А Тихомиров

2018

**Цель работы**: Изучить простейшее оконное приложения Windows, разработанное с использованием WinAPI 32.

**Задания**:

1. Поместить на выводимом сообщении три кнопки «ДА», «НЕТ», «Отмена». При нажатии на кнопку «ДА» должно появляться новое сообщение. При нажатии на кнопку «НЕТ» - меняться пиктограмма на сообщении, при нажатии на «Отмена» - заканчивается работа программы.
2. Комбинация клавиш Shift+M должна вызывать появление системного сообщения окна MessageBox с числом нажатий кнопки мыши.

**Выполнение:**

1. Поместить на выводимом сообщении три кнопки «ДА», «НЕТ», «Отмена». При нажатии на кнопку «ДА» должно появляться новое сообщение. При нажатии на кнопку «НЕТ» - меняться пиктограмма на сообщении, при нажатии на «Отмена» - заканчивается работа программы.

Текст программы приведен в листинге 1.

Листинг 1 – Текст файла Prog1.cpp

|  |
| --- |
| #include "Windows.h"  int WINAPI WinMain(HINSTANCE hInstance,  HINSTANCE hPrevInstance,  LPSTR lpCmdLine,  int nCmdShow)  {  int Ret = MessageBox(0, "Вы студент?", "Соц. опрос", MB\_YESNOCANCEL| MB\_ICONINFORMATION);  while (true)  {  switch (Ret)  {  case IDYES:  MessageBox(0, "Поздравляю", "Соц. опрос", MB\_OK);  exit(0);  case IDNO:  Ret = MessageBox(0, "Вы студент?", "Соц. опрос", MB\_YESNOCANCEL | MB\_ICONEXCLAMATION);  break;  case IDCANCEL:  exit(0);  }  }  } |

Результат работы программы приведены на рисунках 1-3.

На рисунке 1 показано главное выводимое сообщение с тремя кнопками «ДА», «НЕТ», «Отмена».

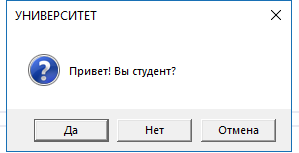


Рисунок 1 – Главное выводимое сообщение

На рисунке 2 показано, как при нажатии на кнопку «ДА» появляется новое сообщение.

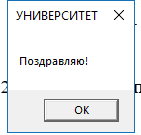


Рисунок 2 –Выводимое сообщение при нажатии на кнопку «ДА»

На рисунке 3 показано, как при нажатии на кнопку «НЕТ», меняется пиктограмма на главном сообщении. Программа закончится, нажав кнопку «Отмена».

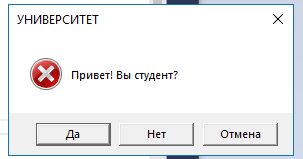


Рисунок 2 –Главное выводимое сообщение с измененной пиктограммой

2) Комбинация клавиш Shift+M должна вызывать появление системного сообщения окна MessageBox с числом нажатий кнопки мыши.

Текст программы приведен в листинге 2.

Листинг 2 – Текст файла Prog2.cpp

|  |
| --- |
| #include "stdafx.h"  #include "Lab1\_2.h"  #define MAX\_LOADSTRING 100  const int window\_width = 700;  const int window\_height = 500;  HINSTANCE hInst; // current instance  WCHAR szTitle[MAX\_LOADSTRING]; // The title bar text  WCHAR szWindowClass[MAX\_LOADSTRING]; // the main window class name  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 wWinMain(\_In\_ HINSTANCE hInstance,  \_In\_opt\_ HINSTANCE hPrevInstance,  \_In\_ LPWSTR lpCmdLine,  \_In\_ int nCmdShow)  {  UNREFERENCED\_PARAMETER(hPrevInstance);  UNREFERENCED\_PARAMETER(lpCmdLine);  LoadStringW(hInstance, IDS\_APP\_TITLE, szTitle, MAX\_LOADSTRING);  LoadStringW(hInstance, IDC\_LAB12, szWindowClass, MAX\_LOADSTRING);  MyRegisterClass(hInstance);  if (!InitInstance (hInstance, nCmdShow))  {  return FALSE;  }  HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_LAB12));  MSG msg;  while (GetMessage(&msg, nullptr, 0, 0))  {  if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))  {  TranslateMessage(&msg);  DispatchMessage(&msg);  }  }  return (int) msg.wParam;  }  ATOM MyRegisterClass(HINSTANCE hInstance)  {  WNDCLASSEXW 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\_LAB12));  wcex.hCursor = LoadCursor(nullptr, IDC\_ARROW);  wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);  wcex.lpszMenuName = MAKEINTRESOURCEW(IDC\_LAB12);  wcex.lpszClassName = szWindowClass;  wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI\_SMALL));  return RegisterClassExW(&wcex);  }  BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)  {  hInst = hInstance; // Store instance handle in our global variable  HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,  (GetSystemMetrics(SM\_CXSCREEN) >> 1) - (window\_width >> 1), (GetSystemMetrics(SM\_CYSCREEN) >> 1) - (window\_height >> 1),  window\_width, window\_height, nullptr, nullptr, hInstance, nullptr);  if (!hWnd)  {  return FALSE;  }  ShowWindow(hWnd, nCmdShow);  UpdateWindow(hWnd);  return TRUE;  }  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)  {  static int count=1;  char szText[] = "Hello World!";  int x = rand() % (window\_width - 100);  int y = rand() % (window\_height - 100);  switch (message)  {  case WM\_COMMAND:  {  int wmId = LOWORD(wParam);  // Parse the menu selections:  switch (wmId)  {  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:  {  PAINTSTRUCT ps;  HDC hdc = BeginPaint(hWnd, &ps);  SetTextAlign(hdc, TA\_CENTER);  TextOut(hdc, window\_width / 2 - strlen(szText), window\_height / 2.5, szText, strlen(szText));  EndPaint(hWnd, &ps);  }  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  case WM\_LBUTTONDOWN:  {  HDC hdc = GetDC(hWnd);  count += 1;  TextOut(hdc, x, y, szText, 13);  }  break;  case WM\_KEYDOWN:  {  char buf[3];  if (GetKeyState(VK\_SHIFT) && LOWORD(wParam) == 77)  {  wsprintf(buf, TEXT("%d"), count);  MessageBox(hWnd, buf, "Hello World Count", MB\_OK | MB\_ICONINFORMATION);  }  }  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  return 0;  }  // Message handler for about box.  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;  } |

Результат работы программы приведены на рисунках 4-5.

Первоначальное окно приведено на рисунке 4.

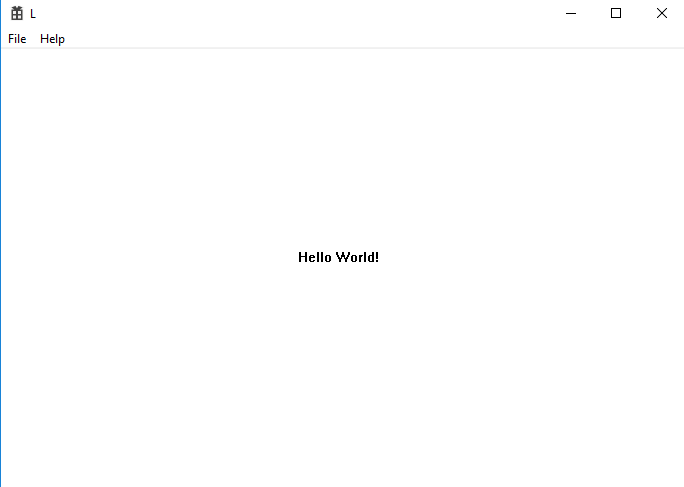


Рисунок 4 – Первоначальное окно

При комбинации клавиш Shift+M вызывается системное сообщение окна MessageBox с числом нажатий кнопки мыши (рисунок 5).

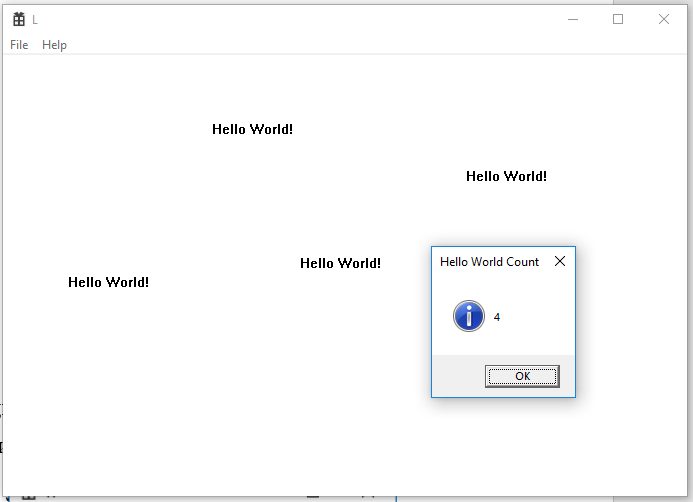


Рисунок 5 – MessageBox с числом нажатий

**Список использованных источников**

1. Саймон, Р. Windows 2000 API Энцеклопедия программиста / Р. Саймон. - М: DiaSoft, 2002

2. Ганеев, Р.М. Проектирование интерфейса пользователя средставми Win32 API / Р.М. Ганеев — М: Телеком, 2007