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 О. Хусейнов

Преподаватель В.А Тихомиров

2018

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

**Задания**:

1. Добавить на выводимое сообщение три кнопки и системную пиктограмму с предупреждающим значком. При нажатии на кнопки должна меняться пиктограмма. А при нажатии на одну и ту же кнопку два раза – заканчиваться программа.
2. Сделайте так, чтобы при каждом повторном запуске Вашего приложения случайным образом задавался цвет шрифта в окне.

**Выполнение:**

1) Добавить на выводимое сообщение три кнопки и системную пиктограмму с предупреждающим значком. При нажатии на кнопки должна меняться пиктограмма. А при нажатии на одну и ту же кнопку два раза – заканчиваться программа.

Текст программы приведен в листинге 1.

Листинг 1 – Текст файла Prog1.cpp

|  |
| --- |
| #include "Windows.h"  void msgShow();  int res;  size\_t clickCount = 0;  size\_t clickABORT = 0;  size\_t clickRETRY = 0;  size\_t clickIGNORE = 0;  int WINAPI WinMain  (  HINSTANCE hInstance, // указатель на текущий экземпляр  HINSTANCE hPrevInstance, // указатель на предыдущйи запушенный экземпляр  LPSTR lpCmdLine, // командная строка  int nCmdShow // тип отображения окна программы  )  {  res = MessageBox(NULL, "Системное сообщение с тремя кнопками", "Заголовок", MB\_ABORTRETRYIGNORE | MB\_ICONEXCLAMATION);  while (true)  {  if (res == IDABORT)  {  clickABORT++;  msgShow();  }  else if (res == IDRETRY)  {  clickRETRY++;  msgShow();  }  else if (res == IDIGNORE)  {  clickIGNORE++;  msgShow();  }  else break;  }  return 0;  }  void msgShow()  {  if (clickABORT == 2 || clickRETRY == 2 || clickIGNORE == 2)  exit(0);  clickCount++;  if (clickCount == 2)  {  clickCount = 0;  clickABORT = 0;  clickRETRY = 0;  clickIGNORE = 0;  }  if (clickCount == 0)  res = MessageBox(NULL, "Системное сообщение с тремя кнопками", "Заголовок", MB\_ABORTRETRYIGNORE | MB\_ICONEXCLAMATION);  else  res = MessageBox(NULL, "Системное сообщение с тремя кнопками", "Заголовок", MB\_ABORTRETRYIGNORE | MB\_ICONQUESTION);  } |

Результат работы программы приведены на рисунках 1,2.

На рисунке 1 показано главное выводимое сообщение с тремя кнопками «About», «Retry», «Ignore».

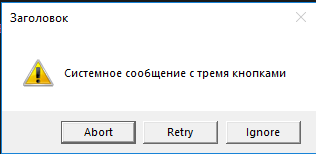


Рисунок 1 – Главное выводимое сообщение

На рисунке 2 показано, как при нажатии на кнопку «About» появляется новое сообщение с измененной пиктограммой.

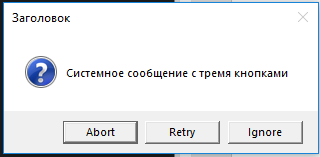


Рисунок 2 –Выводимое сообщение при нажатии на кнопку «About»

Программа закончится, при последовательном нажатии двух одинаковых кнопок.

2) Сделайте так, чтобы при каждом повторном запуске Вашего приложения случайным образом задавался цвет шрифта в окне.

Текст программы приведен в листинге 2.

Листинг 2 – Текст файла Prog2.cpp

|  |
| --- |
| #include <windows.h>  #include "resource.h"  #include <ctime>  BOOL CALLBACK PviewDlgProc(HWND hWnd, UINT wMsg, WPARAM wParam, LPARAM lParam);  int WINAPI WinMain(HINSTANCE hInstance,  HINSTANCE hPrevInstance,  LPSTR lpCmdLine,  int nCmdShow)  {  DialogBox(hInstance, MAKEINTRESOURCE(PVIEW\_DLG), 0, PviewDlgProc);  return 0;  }  //Процедура обработки сообщений диалогового окна  BOOL CALLBACK PviewDlgProc(HWND hWnd,  UINT wMsg,  WPARAM wParam,  LPARAM lParam)  {  switch (wMsg)  {  // Сообщение о инициализации диалогового окна  case WM\_INITDIALOG:  break;  // Сообщение о закрытии диалогового окна  case WM\_CLOSE:  PostQuitMessage(0);  break;  // Сообщение от элементов диалогового окна  case WM\_COMMAND:  {  switch (LOWORD(wParam))  {  default:  return FALSE;  }  break;  }  case WM\_CTLCOLORSTATIC:  {  if ((HWND)lParam == GetDlgItem(hWnd, ID\_COLOR))  {  srand(time(0));  int R = rand() % 255;  int G = rand() % 255;  int B = rand() % 255;  HDC hdc = (HDC)wParam;  SetBkMode(hdc, TRANSPARENT);  SetTextColor(hdc, RGB(R, G, B));  return (int)GetStockObject(NULL\_BRUSH);  }  }  break;  default:  return FALSE;  }  return TRUE;  } |

Результат работы программы приведены на рисунках 4-5.

Первоначальное окно приведено на рисунке 4.

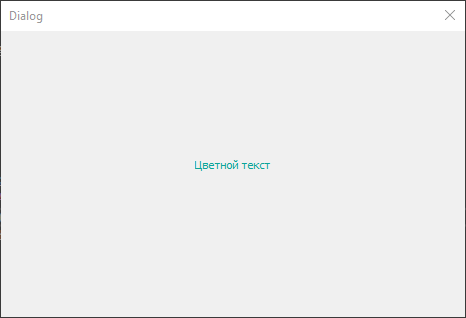


Рисунок 4 – Первоначальное окно

При повторном запуске текст в окне меняет свой цвет (рисунок 5).

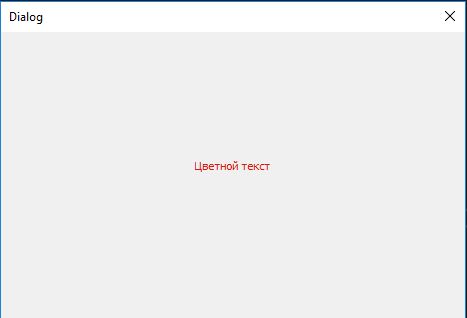


Рисунок 5 – Окно с измененным цветом текста

**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