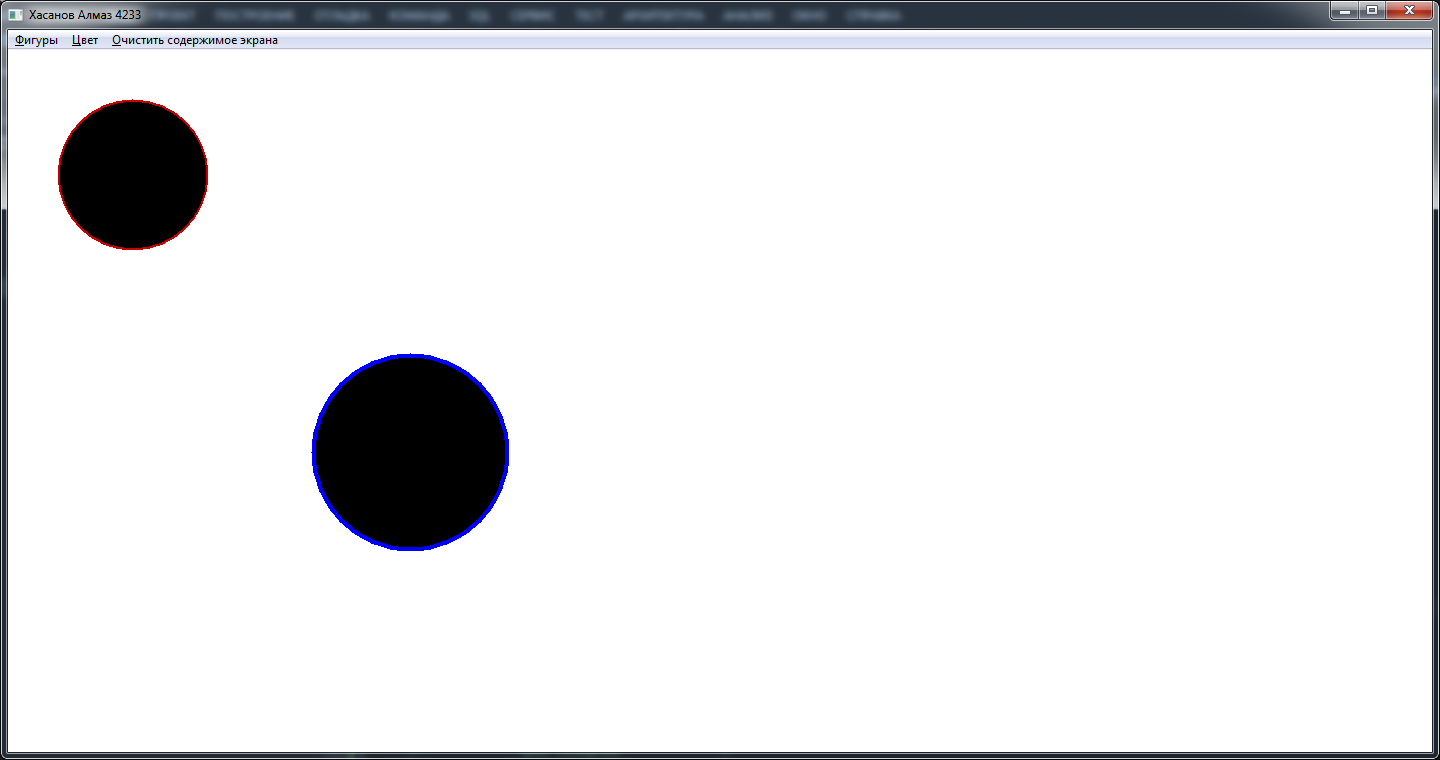
Хасанов Алмаз 4233

Лабораторная работа №5

1. Создал графическое приложение в VisualStudio 2012, изменил каркас программы в соответствии с примером из приложения 7, добавил в меню «Цвет» пункт «Черный», в меню фигуры «Эллипсы», в меню программы «Очистить содержимое окна». Изменил код программы так, чтобы программа корректно реагировала на выбор новых пунктов меню. Изменил заголовок окна на «Хасанов Алмаз 4233».



Исходный код:

#include "windows.h"

#include "resource.h"

/\* объявление функции окна \*/

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

char szWinName[] = "MyWin"; // Имя "класса" окна

/\*----------------------------------------------------\*/

/\* Главная функция \*/

/\*----------------------------------------------------\*/

int WINAPI WinMain (HINSTANCE hThisInst, HINSTANCE hPrevInst,

LPSTR lpszArgs, int nWinMode)

{ HWND hWnd; // дескриптор окна

MSG msg; // сообщение

WNDCLASSEX wcl; // "класс" окна

/\* Определение элементов "класса" окна \*/

wcl.hInstance = hThisInst; //дескриптор данного экземпляра

wcl.lpszClassName = szWinName; // имя "класса" окна \*/

wcl.lpfnWndProc = WindowFunc; // функция окна \*/

wcl.style = 0;

wcl.cbSize = sizeof(WNDCLASSEX);

wcl.hIcon = LoadIcon(NULL,IDI\_APPLICATION);

wcl.hIconSm = LoadIcon(NULL, IDI\_WINLOGO);

wcl.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcl.lpszMenuName = MAKEINTRESOURCE(IDR\_MENU1); // меню

wcl.cbClsExtra = 0; wcl.cbWndExtra = 0;

/\* Фон окна задается белым \*/

wcl.hbrBackground = (HBRUSH) GetStockObject(WHITE\_BRUSH);

/\* Регистрация "класса" окна \*/

if (!RegisterClassEx(&wcl)) return 0;

/\* Создание окна \*/

hWnd = CreateWindow

( szWinName,

"Хасанов Алмаз 4233",

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, CW\_USEDEFAULT,

CW\_USEDEFAULT, CW\_USEDEFAULT,

HWND\_DESKTOP, NULL,

hThisInst, NULL);

/\* Отображение окна \*/

ShowWindow(hWnd, nWinMode);

UpdateWindow(hWnd);

/\* Цикл сообщений \*/

while (GetMessage(&msg, NULL, 0, 0))

{ TranslateMessage(&msg);

DispatchMessage (&msg);

}

return msg.wParam;

}

/\*--------------------------------------------------\*/

HDC memdc; // контекст устройства памяти

HBITMAP hbit; // растр изображения в окне

HBRUSH hbrush, hbrush2; // дескриптор текущей кисти

HBRUSH hOldbrush; // дескриптор прежней кисти

HPEN hRedpen, hBluepen; // дескрипторы перьев

HPEN hOldpen; // дескриптор прежнего пера

int maxX, maxY; // размеры экрана

/\* массивы точек - координаты вершин треугольников \*/

POINT mp1[3]={{150,50},{50,150},{150,150}};// 1-й треуг-к

POINT mp2[3]={{250,150},{350,50},{450,150}};// 2-й треуг-к

COLORREF color= RGB(255,255,0); // цвет кисти

COLORREF color2 = RGB(0,0,0);

/\*--------------------------------------------------------\*/

/\* Функция окна \*/

/\*--------------------------------------------------------\*/

LRESULT CALLBACK WindowFunc (HWND hWnd, UINT message,

WPARAM wParam, LPARAM lParam)

{HDC hdc; // контекст устройства окна

PAINTSTRUCT paintstruct; // характеристики области перерисовки

switch(message)

{ case WM\_CREATE:

/\* получение размеров окна \*/

maxX = GetSystemMetrics (SM\_CXSCREEN);

maxY = GetSystemMetrics (SM\_CYSCREEN);

/\* создание контекста устройства окна \*/

hdc = GetDC (hWnd);

/\* создание совместимого контекста устройства памяти \*/

memdc = CreateCompatibleDC (hdc);

/\* создание совместимого растра \*/

hbit = CreateCompatibleBitmap (hdc, maxX, maxY);

/\* выбор растрового изображения в контекст устройства памяти \*/

SelectObject (memdc, hbit);

/\* заполнение окна белой кистью \*/

PatBlt (memdc, 0, 0, maxX, maxY, PATCOPY);

/\* создание красного и синего перьев \*/

hRedpen = CreatePen (PS\_SOLID, 2, RGB (200, 0, 0));

hBluepen = CreatePen (PS\_SOLID, 4, RGB (0,0,255));

/\* освобождение контекста устройства окна \*/

ReleaseDC (hWnd, hdc);

break;

case WM\_COMMAND: // выбрана команда меню

switch (wParam)

{

case ID\_TREUG:

{

PatBlt (memdc, 0, 0, maxX, maxY, PATCOPY);

hbrush = CreateSolidBrush (color);

hOldbrush = (HBRUSH)SelectObject (memdc,hbrush);

hOldpen = (HPEN)SelectObject (memdc,hRedpen );

Rectangle (memdc, 50, 50, 200, 200);

SelectObject (memdc, hBluepen);

RoundRect (memdc, 250,50, 400, 150, 30, 30);

SelectObject (memdc, hOldpen);

SelectObject (memdc, hOldbrush);

DeleteObject (hbrush);

Polygon(memdc,mp1,3);

InvalidateRect (hWnd, NULL, 0);

break;

}

case ID\_RED:

color = RGB(250,0,0);

break;

case ID\_BLACK:

color = RGB(0,0,0);

break;

case ID\_ELLIPSE:

{

PatBlt (memdc, 0, 0, maxX, maxY, PATCOPY);

hbrush2 = CreateSolidBrush (color);

HBRUSH hOldBrush = (HBRUSH) SelectObject(memdc,hbrush2);

HPEN hOldPen = (HPEN) SelectObject(memdc, hRedpen);

Ellipse (memdc, 50, 50, 200, 200);

SelectObject (memdc,hBluepen );

Ellipse (memdc, 305, 305, 500, 500);

SelectObject(memdc, hOldPen);

SelectObject(memdc, hOldBrush);

DeleteObject (hbrush2);

InvalidateRect (hWnd, NULL, 0);

break;

}

case ID\_CLEAR:

PatBlt(memdc, 0, 0, maxX, maxY, PATCOPY);

InvalidateRect(hWnd, NULL, 0);

break;

}

break;

case WM\_PAINT: //обновление окна

hdc=BeginPaint (hWnd, &paintstruct);

BitBlt (hdc, 0, 0, maxX, maxY, memdc, 0, 0, SRCCOPY);

EndPaint (hWnd, &paintstruct);

break;

case WM\_DESTROY: // "завершить программу"

/\* удаление перьев \*/

DeleteObject (hRedpen);

DeleteObject (hBluepen);

DeleteDC (memdc); // удаление контекста памяти

PostQuitMessage(0);

break;

default:

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

}

return 0;

}

resource.h:

#define IDR\_MENU1 101

#define ID\_TREUG 40003

#define ID\_RED 40004

#define ID\_BLACK 40005

#define ID\_ELLIPSE 40006

#define ID\_CLEAR 40007

файл rc:

#include "resource.h"

IDR\_MENU1 MENU DISCARDABLE

BEGIN

POPUP "&Фигуры"

BEGIN

MENUITEM "&Треугольники", ID\_TREUG

MENUITEM "&Эллипсы", ID\_ELLIPSE

END

POPUP "&Цвет"

BEGIN

MENUITEM "&Красный", ID\_RED

MENUITEM "&Черный", ID\_BLACK

END

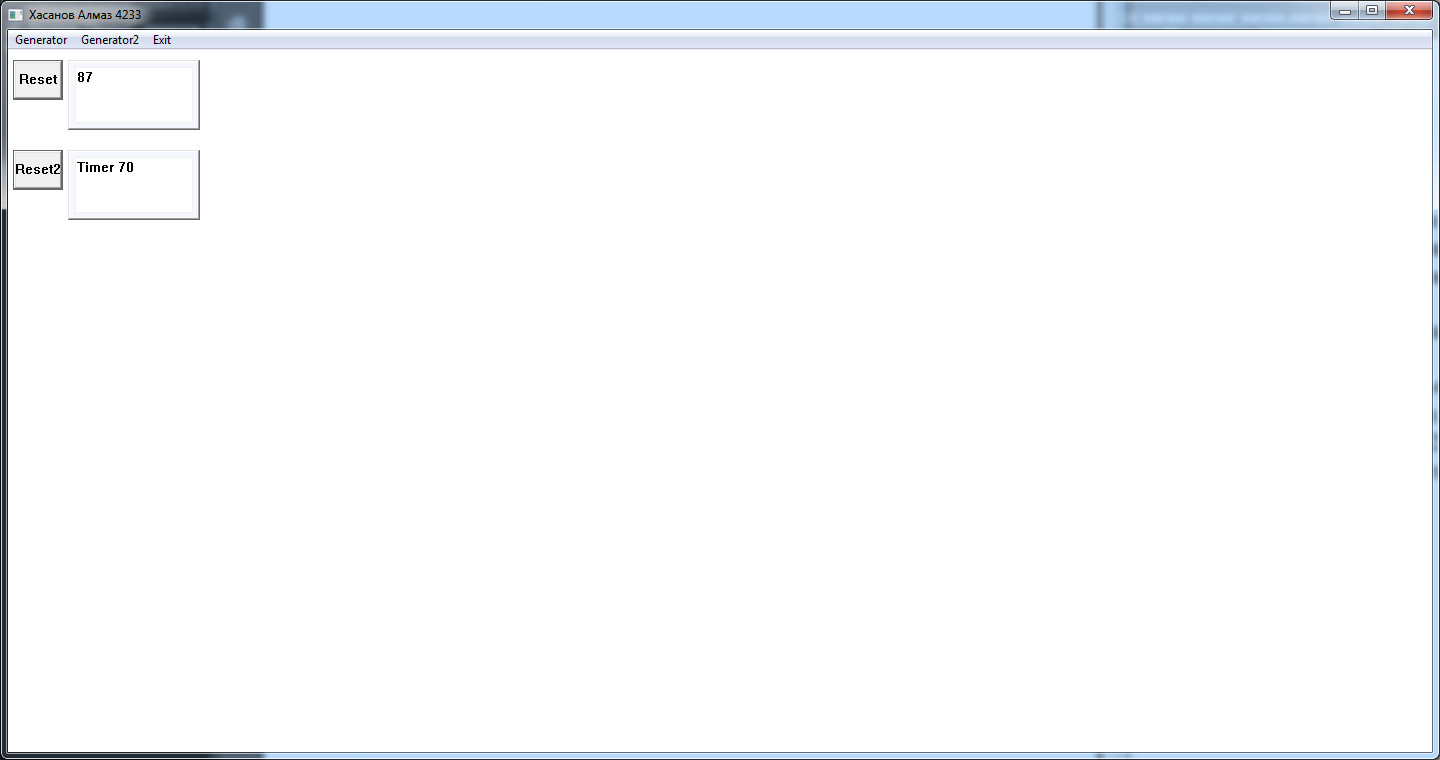
MENUITEM "&Очистить содержимое экрана", ID\_CLEAR

END

1. Создал новое графическое приложение в VisualStudio 2012, изменил каркас программы в соответствии с примером из приложения 8. Изменил заголовок окна на «Хасанов Алмаз 4233». Увеличил задержку таймера до 1 секунды.

Добавил еще один редактор и кнопку с названием «Reset2», которая очищает содержимое второго редактора.

1. Добавил в меню пункт «Exit», подменю «Generator2» с пунктами «Start» и «Stop», реализовал обработчики выбора данных пунктов. Добавил еще один таймер с интервалом 2 секунды, по истечению которого во второй редактор выводится текст «Timer», где вместо N копируется число из первого редактора.



Код:

#include "windows.h"

#include "resource.h"

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

HINSTANCE hInstance; // дескриптор приложения

HWND BtHWnd, BtHWnd2; // кнопка

HWND EdtHWnd, EdtHWnd2; // редактор

char szWinName[] = "MyWin"; // имя класса окна

/\* Главная функция \*/

int WINAPI WinMain(HINSTANCE hThisInst, HINSTANCE hPrevInst, LPSTR lpszArgs, int nWinMode) {

HWND hWnd;

MSG msg;

WNDCLASSEX wcl;

wcl.hInstance = hThisInst;

wcl.lpszClassName = szWinName;

wcl.lpfnWndProc = WindowFunc;

wcl.style = 0;

wcl.cbSize = sizeof(WNDCLASSEX);

wcl.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wcl.hIconSm = LoadIcon(NULL, IDI\_WINLOGO);

wcl.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcl.lpszMenuName = MAKEINTRESOURCE(IDR\_MENU1);

wcl.cbClsExtra = 0;

wcl.cbWndExtra = 0;

wcl.hbrBackground = (HBRUSH) GetStockObject(WHITE\_BRUSH);

if (!RegisterClassEx(&wcl)) return 0;

hWnd = CreateWindow (

szWinName,

"Хасанов Алмаз 4233", // заголовок

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, CW\_USEDEFAULT,

CW\_USEDEFAULT, CW\_USEDEFAULT,

HWND\_DESKTOP, NULL, hThisInst, NULL);

/\* запоминаем дескриптор приложения

для создания дочерних элементов управления \*/

hInstance = hThisInst;

ShowWindow(hWnd, nWinMode);

UpdateWindow(hWnd);

while (GetMessage(&msg, NULL, 0, 0)) {

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return msg.wParam;

}

/\* Функция окна \*/

LRESULT CALLBACK WindowFunc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) {

char msg[100];

int r;

switch (message) {

case WM\_DESTROY: // завершение программы

PostQuitMessage(0);

break;

/\* сообщение от пунктов меню и элементов управления \*/

case WM\_COMMAND:

if (LOWORD(wParam) == IDM\_START) //пункт меню - Start

{

MessageBoxA(hWnd, "Timer started",

"Generation started", MB\_OK);

SetTimer(hWnd, 1, 1000, NULL); // установка таймера

}

if (LOWORD(wParam) == IDM\_STOP) // пункт меню - Stop

{

MessageBoxA(hWnd, "Timer stoped",

"Generation stopped", MB\_OK);

KillTimer(hWnd, 1); // завершение работы таймера

}

if (HWND(lParam) == BtHWnd) // сообщение от кнопки

{

lstrcpy(msg, "0");

/\* помещение текста в поле ввода \*/

SendMessage(EdtHWnd, WM\_SETTEXT, 0, (LPARAM) (LPCTSTR) msg);

}

if (HWND(lParam) == BtHWnd2)

{

lstrcpy(msg, "0");

SendMessage(EdtHWnd2, WM\_SETTEXT, 0, (LPARAM) "");

}

if (LOWORD(wParam) == IDM\_START2)

{

MessageBoxA(hWnd, "Timer started",

"Generation started", MB\_OK);

SetTimer(hWnd, 2, 2000, NULL);

}

if (LOWORD(wParam) == IDM\_STOP2)

{

MessageBoxA(hWnd, "Timer stoped",

"Generation stopped", MB\_OK);

KillTimer(hWnd, 2);

}

if (LOWORD(wParam) == IDM\_EXIT)

DestroyWindow(hWnd);

break;

case WM\_TIMER: // сообщение от таймера

switch(wParam)

{

case 1:

{

int k;

r = rand() % 20;

/\* получение длины текста в поле ввода \*/

k = SendMessage(EdtHWnd, WM\_GETTEXTLENGTH, 0, 0);

/\* получение текста из поля ввода \*/

SendMessage(EdtHWnd, WM\_GETTEXT, k + 1, (LPARAM) msg);

/\* перевод значения из строки в число и суммирование\*/

r += atoi(msg);

/\* перевод из числа в строку \*/

wsprintfA(msg, "%d", r);

/\* помещение текста в поле ввода \*/

SendMessage(EdtHWnd, WM\_SETTEXT, 0, (LPARAM) (LPCTSTR) msg);

break;

}

case 2:

{

int k;

k = SendMessage(EdtHWnd, WM\_GETTEXTLENGTH, 0, 0);

SendMessage(EdtHWnd, WM\_GETTEXT, k + 1, (LPARAM) msg);

r = atoi(msg);

wsprintfA(msg, "Timer %d", r);

SendMessage(EdtHWnd2, WM\_SETTEXT, 0, (LPARAM) (LPCTSTR) msg);

break;

}

}

break;

case WM\_CREATE: // сообщение о создании окна

/\* создание кнопки \*/

BtHWnd = CreateWindow (

"BUTTON", // имя класса окна

"Reset", // заголовок

WS\_CHILD | WS\_VISIBLE | WS\_BORDER, // стиль

5, 10, 50, 40, // координаты и размеры

hWnd, // дескриптор родительского окна

NULL, // меню нет

hInstance, // дескриптор приложения

NULL);

/\* создание редактора(поля ввода) \*/

EdtHWnd = CreateWindow(

"EDIT", // имя класса окна

"0", // заголовок

WS\_CHILD | WS\_VISIBLE | WS\_BORDER | WS\_THICKFRAME,

60, 10, 70, 70, // координаты и размеры

hWnd, // дескриптор родительского окна

NULL, // меню нет

hInstance, // дескриптор приложения

NULL);

BtHWnd2 = CreateWindow (

"BUTTON",

"Reset2",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER,

5, 100, 50, 40,

hWnd,

NULL,

hInstance,

NULL),

EdtHWnd2 = CreateWindow(

"EDIT",

"",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER | WS\_THICKFRAME,

60, 100, 70, 70,

hWnd,

NULL,

hInstance,

NULL);

default:

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

}

return 0;

}

resource.h:

#define IDR\_MENU1 101

#define IDB\_BT1 103

#define IDM\_START 40001

#define IDM\_STOP 40002

#define IDM\_START2 40003

#define IDM\_STOP2 40004

#define IDM\_EXIT 40005

файл rc:

#include "resource.h"

IDR\_MENU1 MENU

BEGIN

POPUP "Generator"

BEGIN

MENUITEM "Start", IDM\_START

MENUITEM "Stop", IDM\_STOP

END

POPUP "Generator2"

{

MENUITEM "Start", IDM\_START2

MENUITEM "Stop", IDM\_STOP2

}

MENUITEM "Exit", IDM\_EXIT

END