***//Программа GDI-2.* Вывод текста и *простых геометрических фигур******/\*0ператоры препроцессора\*/***#define STRICT //Строгая проверка типов переменных   
#include <windows.h> //Два файла с определениями, макросами   
#include <windowsx.h> //и прототипами функций Windows   
  
**/\*Прототипы используемых в программе функций пользователя\*/**  
BOOL OnCreate(HWND, LPCREATESTRUCT);  
LRESULT CALLBACK WndProc(HWND,UINT, WPARAM, LPARAM); //Оконная функция  
void OnPaint(HWND); //Прототип функции OnPaint   
void OnDestroy(HWND); //Прототип функции OnDestroy  
  
**/\*Глобальные переменные, доступные всем функциям\*/**HPEN hRedPen,hBluePen;//Дескрипторы новых перьев   
HBRUSH hYellowBrush;//Дескриптор новой кист  
  
**/\* Главная функция WinMain\*/**int WINAPI WinMain(HINSTANCE hInst, HINSTANCE, LPSTR, int)  
{  
 char szClassName[]="MainWindow";//Произвольное имя класса главного окна   
 char szTitle[]="Программа GDI-2"; //Произвольный заголовок окна   
 MSG Msg;//Структура Msg типа MSG для временного хранения сообщений Windows   
 WNDCLASS wc;//Структура we типа WNDCLKSS для задания характеристик окна  
  
**/\*3арегистрируем класс главного окна\*/** memset (&wc, 0, sizeof (wc) ); //Обнуление всех членов структуры wc  
 wc.lpfnWndProc=WndProc; //Определяем оконную процедуру для главного окна  
 wc.hInstance=hInst; //Дескриптор приложения   
 wc.hIcon=LoadIcon(NULL,IDI\_APPLICATION); //Стандартная иконка  
 wc.hCursor=LoadCursor(NULL,IDC\_ARROW); //Стандартный курсор мыши   
 wc.hbrBackground=GetStockBrush(WHITE\_BRUSH); //Белая кисть для фона окна   
 wc.lpszClassName=szClassName; //Класс главного окна   
 RegisterClass(&wc); //Вызов функции Windows регистрации класса окна  
  
**/\*Создадим главное окно и сделаем его видимым\*/**  
HWND hwnd=CreateWindow(szClassName,szTitle, //Класс и заголовок окна  
 WS\_OVERLAPPEDWINDOW,10,10,300,150, //Стиль окна, его координаты и размеры   
 HWND\_DESKTOP,NULL,hInst,NULL); //Родитель, меню, другие параметры  
ShowWindow(hwnd,SW\_SHOWNORMAL); //Вызов функции Windows показа окна  
  
**/\*0рганизуем цикл обработки сообщений\*/**  
while(GetMessage(&Msg,NULL,0,0)) //Цикл обработки сообщении: ждать  
 DispatchMessage(&Msg); //сообщения, записать его в msg и передать WndProc   
return 0; //После выхода из цикла обработки сообщений вернуться в Windows  
} //Конец функции WinMain   
  
**/\*0конная функция главного окна\*/**  
LRESULT CALLBACK WndProc(HWND hwnd,UINT msg,WPARAM wParam,LPARAM lParam)  
{   
 switch(msg)  
 { //Переход по значению msg - коду сообщения  
 HANDLE\_MSG(hwnd,WM\_CREATE,OnCreate); //При поступлении сообщения WM\_CREATE  
 HANDLE\_MSG(hwnd,WM\_PAINT,OnPaint); //При поступлении сообщения WM\_PAINT   
 HANDLE\_MSG(hwnd,WM\_DESTROY,OnDestroy);//При завершении пользователем   
 default: //В случае всех остальных сообщений Windows обработка их  
 return(DefWindowProc(hwnd,msg,wParam,lParam));//по умолчанию   
 }//Конец оператора switch   
}//Конец функции WndProc   
  
**//Функция обработки сообщений HM\_CREATE**BOOL OnCreate(HWND,LPCREATESTRUCT)  
{  
 hRedPen=CreatePen (PS\_SOLID, 5, RGB (128,0, 0) ); //Создадим темно-красное перо   
 hBluePen=CreatePen(PS\_SOLID,5,RGB(0,0,255));//Создадим еще синее перо   
 hYellowBrush=CreateSolidBrush(RGB(255,255,0));//Создадим кисть для заливки  
 return TRUE;  
}  
  
**//Функция обработки сообщений WM\_PAINT**void OnPaint(HWND hwnd)   
{  
 PAINTSTRUCT ps; //Структура для функции BeginPaint()  
 TEXTMETRIC tm; //Структура для получения характеристик используемых шрифтов  
 char szText1[] = "Первая строка текста";  
 char szText2[] = "Вторая строка текста";  
 char szText3[] = "Третья строка текста";  
 HDC hdc = BeginPaint(hwnd,&ps); //Получим контекст устройства   
 GetTextMetrics(hdc,&tm); //Получим характеристики (метрики) текущего шрифта   
 int nLineHeight = tm.tmHeight+tm.tmExternalLeading;//Вычислим высоту строки   
 TextOut (hdc, 5,0, szText1, strlen (szText1) ) ; //Вывод арки текста цвета1 по умолчаню  
  
**/\*Варьируем местоположение и цвет выводимого на экран текста\*/**   
 SetTextColor (hdc, RGB (127, 0, 0) ); //Установим в контексте темно-красный цвет текста   
 TextOut(hdc,5,nLineHeight,szText2,strlen(szText2));//Вывод на строку ниже   
 SetTextColor (hdc, RGB (0, 127, 0) ); //Установим в контексте темно-зеленый цвет текста   
 TextOut(hdc,5,nLineHeight\*2,szText3,strlen(szText3));//Вывод езде ниже  
  
**/\*Вывод в окно простых графических фигур\*/**  
 Rectangle(hdc,5,55,105,105); //Фигуры рисуются пером по умолчанию (черным)   
 Ellipse(hdc,85,60,125,100); //и кистью фона по умолчанию (белой)

**/\*Сменим в контексте устройства перо и кисть\*/**  
 HPEN hOldPen=SelectPen (hdc, hRedPen) ; //Старое перо сохраним, новое в контекст   
 Rectangle(hdc,155,55,255,105); //Нарисуем еще прямоугольник в другом месте   
 SelectPen(hdc,hBluePen); //Выберем в контекст новое перо '   
 HBRUSH hOldBrush=SelectBrush (hdc, hYellowBrush) ; //Выберем в контекст новую кисть   
 Ellipse(hdc,235,60,275,100); //Нарисуем круг новыми пером и кистью  
   
**/\*Восстановим контекст устройства \*/**  
 SelectPen(hdc,hOldPen);//Выберем в контекст старое сохраненное перо  
 SelectBrush(hdc,hOldBrush);//Выберем в контекст старую сохраненную кисть  
 EndPaint(hwnd,&ps);//Освобождение контекста устройства  
}

**/\*Функция обработки сообщения WM\_DESTROY\*/**  
void OnDestroy(HWND)  
{  
 DeleteObject(hRedPen);//Уничтожим созданное перо  
 DeleteObject(hBluePen);//Уничтожим созданное перо  
 DeleteObject(hYellowBrush);//Уничтожим созданную кисть.  
 PostQuitMessage(0);//Вызов функции Windows завершения приложения  
}