// TaypeDlg.cpp : implementation file

//

#include "stdafx.h"

#include "Taype.h"

#include "TaypeDlg.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CTaypeDlg dialog

CTaypeDlg::CTaypeDlg(CWnd\* pParent /\*=NULL\*/)

 : CDialog(CTaypeDlg::IDD, pParent)

{

 //{{AFX\_DATA\_INIT(CTaypeDlg)

 m\_CR = 0.0;

 m\_D = \_T("");

 m\_R = 0.0;

 m\_V = 0.0;

 m\_OPCION = -1;

 //}}AFX\_DATA\_INIT

 // Note that LoadIcon does not require a subsequent DestroyIcon in Win32

 m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CTaypeDlg::DoDataExchange(CDataExchange\* pDX)

{

 CDialog::DoDataExchange(pDX);

 //{{AFX\_DATA\_MAP(CTaypeDlg)

 DDX\_Text(pDX, IDC\_CR, m\_CR);

 DDX\_Text(pDX, IDC\_I, m\_D);

 DDX\_Text(pDX, IDC\_R, m\_R);

 DDX\_Text(pDX, IDC\_V, m\_V);

 DDX\_Radio(pDX, IDC\_SERIE, m\_OPCION);

 //}}AFX\_DATA\_MAP

}

BEGIN\_MESSAGE\_MAP(CTaypeDlg, CDialog)

 //{{AFX\_MSG\_MAP(CTaypeDlg)

 ON\_WM\_PAINT()

 ON\_WM\_QUERYDRAGICON()

 ON\_BN\_CLICKED(IDC\_aceptar, Onaceptar)

 //}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CTaypeDlg message handlers

BOOL CTaypeDlg::OnInitDialog()

{

 CDialog::OnInitDialog();

 // Set the icon for this dialog. The framework does this automatically

 // when the application's main window is not a dialog

 SetIcon(m\_hIcon, TRUE); // Set big icon

 SetIcon(m\_hIcon, FALSE); // Set small icon

 // TODO: Add extra initialization here

 return TRUE; // return TRUE unless you set the focus to a control

}

// If you add a minimize button to your dialog, you will need the code below

// to draw the icon. For MFC applications using the document/view model,

// this is automatically done for you by the framework.

void CTaypeDlg::OnPaint()

{

 if (IsIconic())

 {

 CPaintDC dc(this); // device context for painting

 SendMessage(WM\_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

 // Center icon in client rectangle

 int cxIcon = GetSystemMetrics(SM\_CXICON);

 int cyIcon = GetSystemMetrics(SM\_CYICON);

 CRect rect;

 GetClientRect(&rect);

 int x = (rect.Width() - cxIcon + 1) / 2;

 int y = (rect.Height() - cyIcon + 1) / 2;

 // Draw the icon

 dc.DrawIcon(x, y, m\_hIcon);

 }

 else

 {

 CDialog::OnPaint();

 }

}

// The system calls this to obtain the cursor to display while the user drags

// the minimized window.

HCURSOR CTaypeDlg::OnQueryDragIcon()

{

 return (HCURSOR) m\_hIcon;

}

void CTaypeDlg::Onaceptar()

{

 UpdateData(true);

double I,V;

int i;

char D1[20],D2[20];

m\_D = "Datos obtenidos: \n\n";

switch (m\_OPCION)

{

case 0: {I = m\_V /(m\_CR \* m\_R) ;

 gcvt(I,12,D1);

 m\_D = m\_D + "La corriente en el circuito: " + D1 + "mA \n";

 m\_D = m\_D + "Voltaje en cada resistencia: \n";

 for (i=1;i<=m\_CR;i++)

 { V = m\_V / (m\_CR);

 itoa(i,D1,12); gcvt(V,15,D2);

 m\_D = m\_D + "V" + D1 + " = " + D2 + " V. \n";}

 break;}

case 1: {I = m\_CR \* m\_V / m\_R ;

 gcvt(I,12,D1);

 m\_D = m\_D + "La corriente en el circuito: " + D1 + "mA \n";

 m\_D = m\_D + "Voltaje en cada resistencia: \n";

 for (i=1;i<=m\_CR;i++)

 { V = m\_V ;

 itoa(i,D1,12); gcvt(V,15,D2);

 m\_D = m\_D + "V" + D1 + " = " + D2 + " V. \n";}

 break;}

}

UpdateData(false);

}