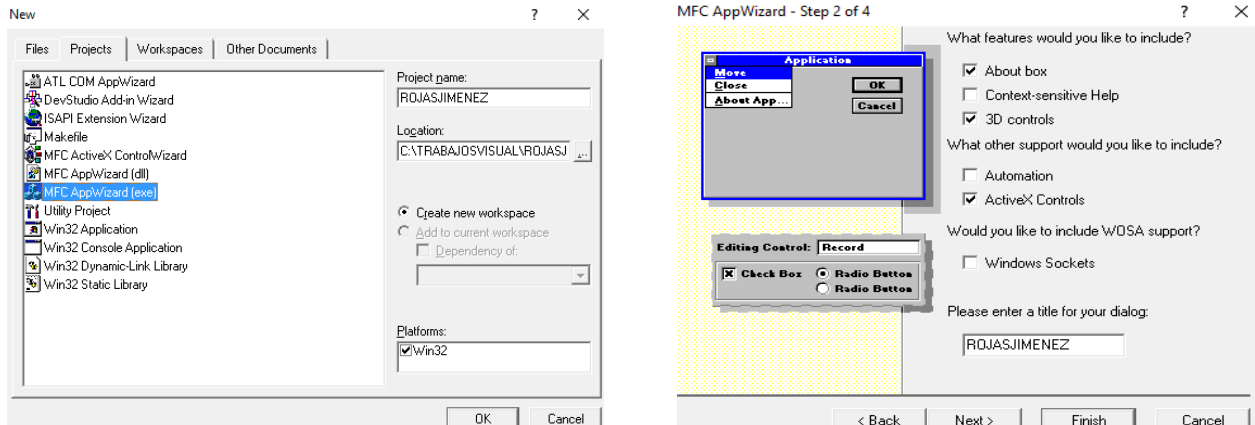


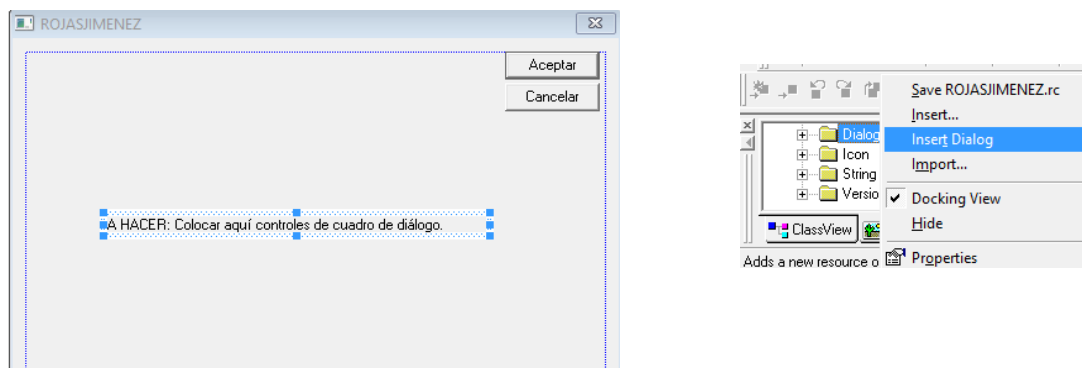
TAREA DE LA SESIÓN 11 Y 12

PROCESAR UNA APLICACIÓN COMUNICACIÓN DE VENTANAS

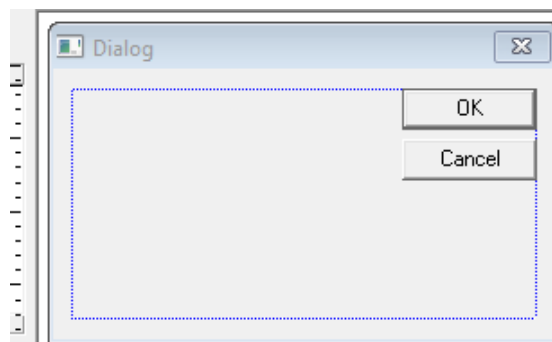
1.- Creamos nuestro programa



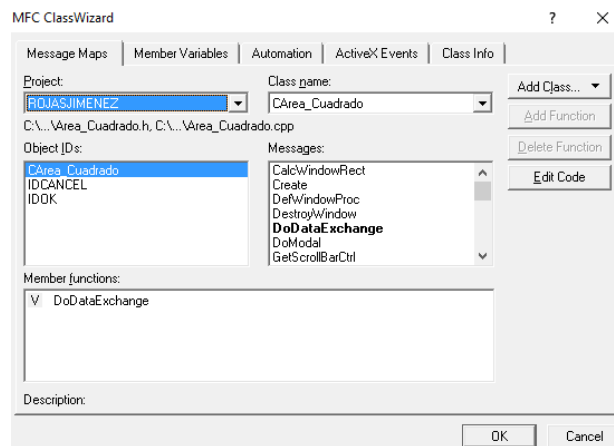
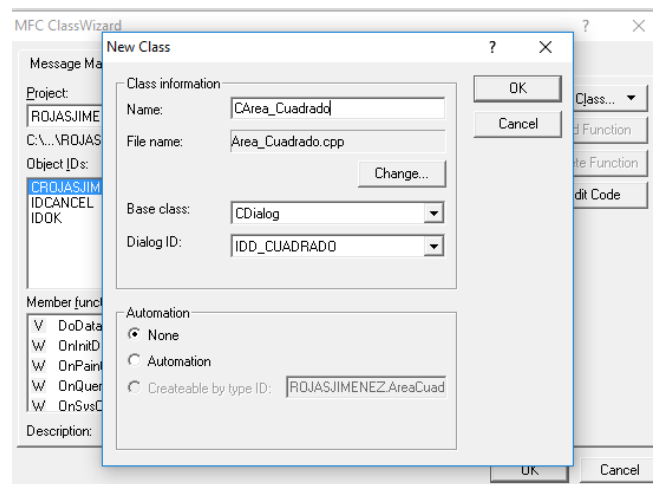
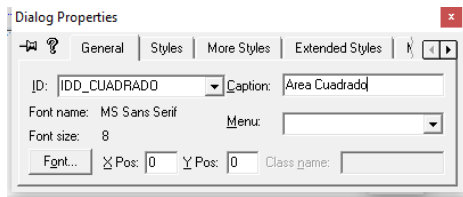
2.- Una vez que estemos en la ventana de dialogo principal ,en la ventana Workspaces, debe estar activa la solapa Resource View; Hacer clic sobre la carpeta Dialog y seleccione Insert Dialog de la siguiente manera:



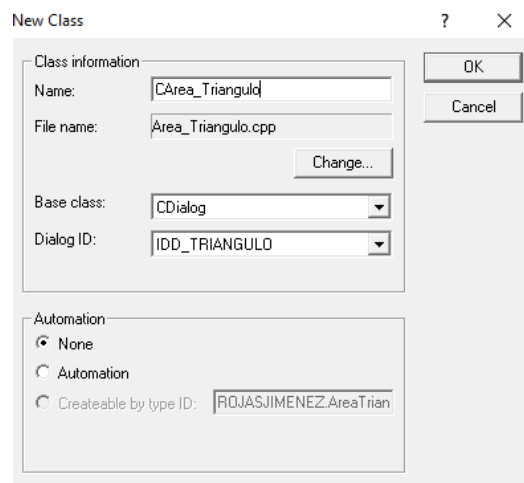
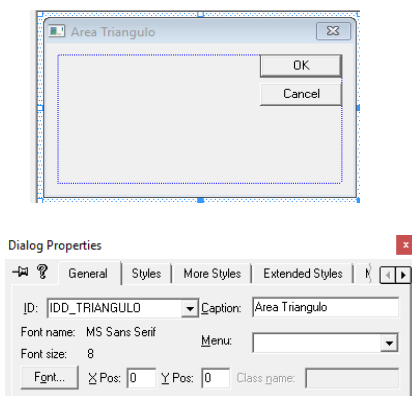
3.- Aparecerá una nueva ventana de diálogo con los botones OK, Cancelar



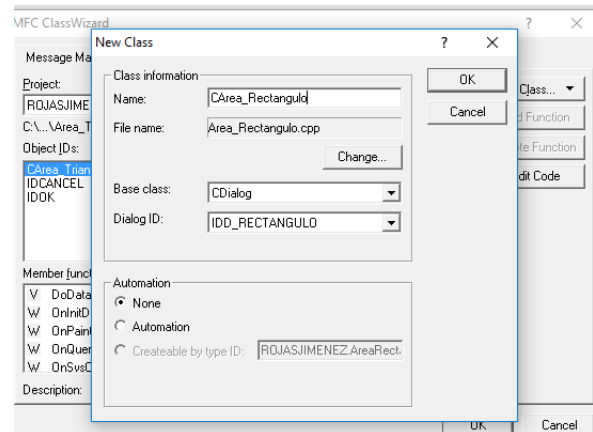
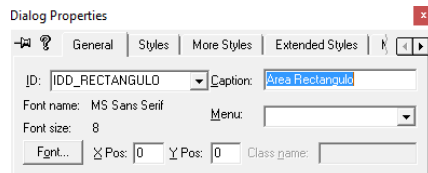
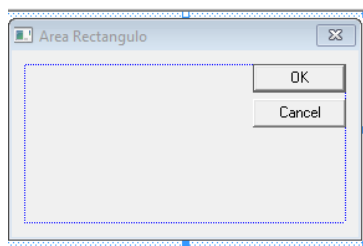
4.- IDD_DIALOG1 será remplazado por IDD_CUADRADO después le damos Ctrl +w y nos saldrá una ventana pidiendo que ingresemos un nombre para la clase escribimos CArea_Cuadrado damos clic en ok



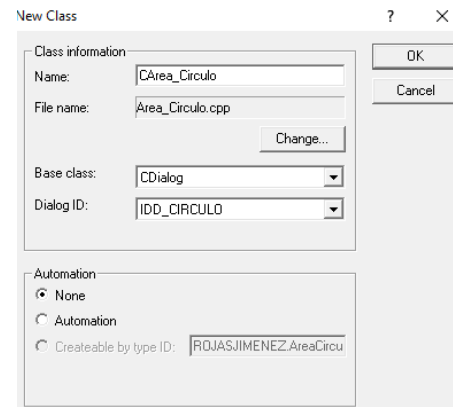
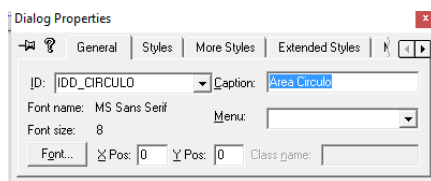
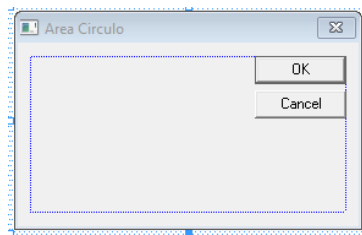
5.-Igualmente creamos la ventana para el triangulo



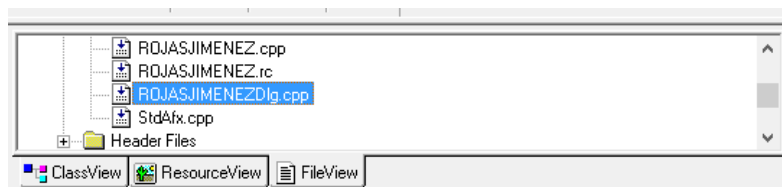
-Luego para el rectangulo



-Y el circulo



6.- Luego nos dirigimos a FileView y hacemos clic en ROJASJIMENEZDlg.cpp e insertamos los #include como se muestra en la imagen



```
// ROJASJIMENEZDlg.cpp : implementation file
//
#include "stdafx.h"
#include "ROJASJIMENEZ.h"
#include "ROJASJIMENEZDlg.h"
#include "Area_Cuadrado.h"
#include "Area_Triangulo.h"
#include "Area_Rectangulo.h"
#include "Area_Circulo.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

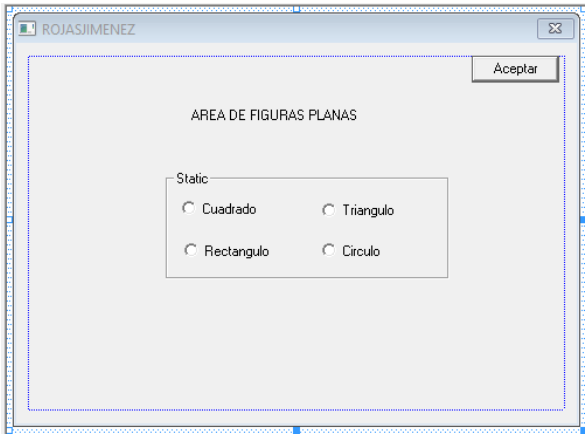
//////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();
};
```

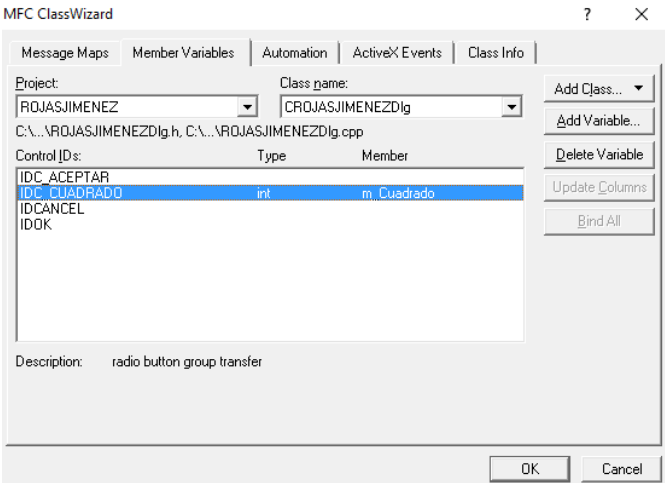
7.-Editamos la ventana principal colocando lo siguiente:

Control	Propiedades	Valor
IDD_VENTANASMULTIPLE S_DIALOG	Caption	Ingresa a una ventana de diálogo desde otra ventana de diálogo
IDC_STATIC	Caption	AREAS DE FIGURAS PLANAS
IDC_STATIC(frame)	Caption	“”
IDC_RADIO1	ID Group Caption	IDC_CUADRADO True Cuadrado
IDC_RADIO2	ID Caption	IDC_TRIANGULO Triangulo
IDC_RADIO3	ID Caption	IDC_RECTANGULO Rectangulo
IDC_RADIO4	ID Caption	IDC_CIRCULO Circulo
IDC_BUTTON	ID Caption Defaultbutton	IDC_ACEPTAR &Aceptar True

-Nuestra ventana debe quedar similar a esta



-Luego agregamos la variable



-Siguiente hacer doble clic en el botón aceptar e insertamos el siguiente codigo

Add Member Function

Member function name:

OnAceptar

OK

Cancel

Message: BN_CLICKED

Object ID: IDC_ACEPTAR

```
void CROJASJIMENEZDlg::OnAceptar()
{
    UpdateData(true);
    switch(m_Cuadrado)
    {
        case 0:
        {
            int r;
            CArea_Cuadrado dlgArea_Cuadrado (this);
            r=dlgArea_Cuadrado.DoModal();
        };break;

        case 1:{int r:CArea_Triangulo dlgArea_Triangulo (this);
            r=dlgArea_Triangulo.DoModal();};break;

        case 2:{int r:CArea_Rectangulo dlgArea_Rectangulo(this);
            r=dlgArea_Rectangulo.DoModal();};break;

        case 3:{int r:CArea_Circulo dlgArea_Circulo (this);
            r=dlgArea_Circulo.DoModal();};break;
        }
        UpdateData(false);
    }
}
```

Colocando eso ya estaría funcionando nuestras ventanas multiples

A cada una de las ventana colocamos sus respectivas propiedades

I.-PARA CREAR LA VENTANA DE DIALOGO AREA DEL CUADRADO

hacemos click en la carpeta Dialog de la pestaña Resource View, y seleccionar IDD_CUADRADO.

Control	Propiedades	Valor
IDD_CUADRADO	Caption	Area del Cuadrado
IDC_STATIC(1)	Caption	Lado :
IDC_STATIC(3)	Caption	Resultado :
IDC_EDIT 1	ID Number	IDC_LADO True
IDC_EDIT 2	ID Number Read-only	IDC_AREA True True
IDC_BUTTON(1)	ID Caption Default_button	IDC_CALCULAR &Calcular True
IDC_BUTTON(2)	ID Caption	IDC_NUEVO &Nuevo

Editamos la ventana y agregamos las variables

Area Cuadrado

Lado

Edit

Calcular

Resultado

Edit

Nuevo

Message Maps

Member Variables

Automation

ActiveX Events

Class Info

Project: ROJASJIMENEZ

Class name: CArea_Cuadrado

C:_Varea_Cuadrado.h, C:_Varea_Cuadrado.cpp

Control ID:s

Type

Member

IDC_AREA

double

m_Area

IDC_CALCULAR

double

m_Lado

IDC_LADO

double

m_Lado

IDC_NUEVO

double

m_Lado

Add Class...

Add Variable...

Delete Variable...

Update Columns

Bind All

Description: double with range validation

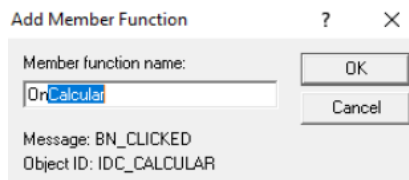
Minimum Value:

Maximum Value:

OK

Cancel

Dar clic en el boton calcular e introducir el siguiente codigo

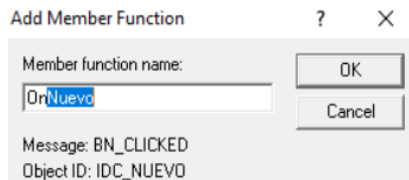


```
BEGIN_MESSAGE_MAP(CÁrea_Cuadrado, CDialog)
    //{{AFX_MSG_MAP(CÁrea_Cuadrado)
    ON_BN_CLICKED(IDC_CALCULAR, OnCalcular)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

// CÁrea_Cuadrado message handlers

void CÁrea_Cuadrado::OnCalcular()
{
    UpdateData(true);
    m_Area=m_Lado*m_Lado;
    UpdateData(false);
}
```

Luego dar clic en el botón nuevo y escribir el codigo



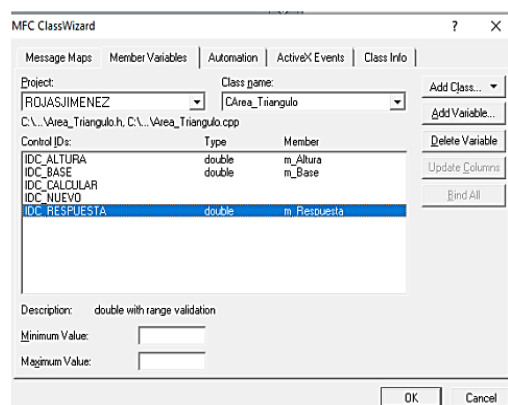
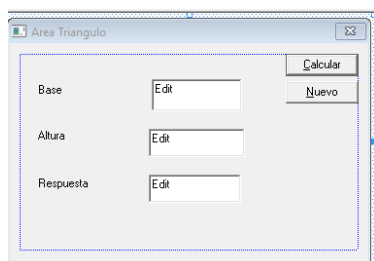
```
void CÁrea_Cuadrado::OnNuevo()
{
    m_Area=m_Lado*0;
    UpdateData(false);
}
```

II.-PARA CREAR LA VENTANA DE DIALOGO AREA DEL TRIANGULO

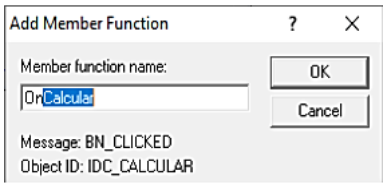
hacemos click en la carpeta Dialog de la pestaña Resource View, y seleccionar IDD_TRIANGULO

Control	Propiedades	Valor
IDD_TRIANGULO	Caption	Area del Triángulo
IDC_STATIC(1)	Caption	Base:
IDC_STATIC(2)	Caption	Altura:
IDC_STATIC(3)	Caption	Respuesta :
IDC_EDIT 1	ID Number	IDC_BASE True
IDC_EDIT 2	ID Number	IDC_ALTURA True
IDC_EDIT 3	ID Number Read-only	IDC_RESPUESTA True True
IDC_BUTTON(1)	ID Caption Default_button	IDC_CALCULAR &Calcular True
IDC_BUTTON(2)	ID Caption	IDC_NUEVO &Nuevo

Editamos la ventana y agregamos las variables

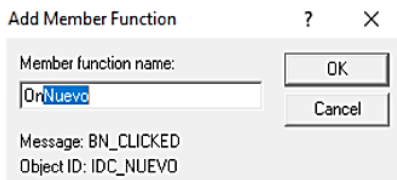


Dar clic en el boton calcular e introducir el siguiente codigo



```
void CArea_Triangulo::OnCalcular()
{
    UpdateData(true);
    m_Respuesta=(m_Altura*m_Base)/2;
    UpdateData(false);
}
```

Luego dar clic en el botón nuevo y escribir el codigo

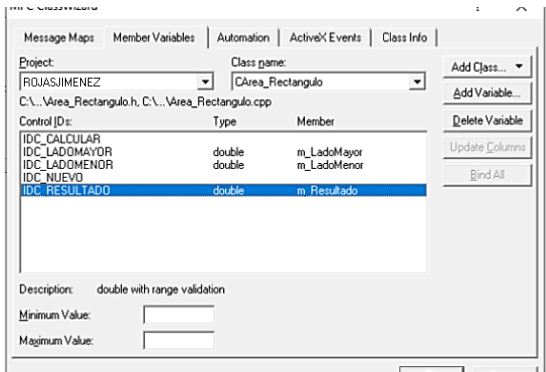
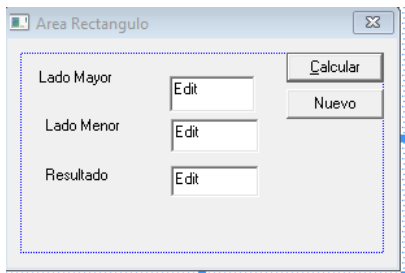


```
void CArea_Triangulo::OnNuevo()
{
    m_Respuesta=m_Altura=m_Base=0;
    UpdateData(false);
}
```

III.-PARA CREAR LA VENTANA DE DIALOGO AREA DEL RECTANGULO

Control	Propiedades	Valor
IDD_RECTANGULO	Caption	Area del Rectangulo
IDC_STATIC(1)	Caption	Lado Mayor:
IDC_STATIC(2)	Caption	Lado Menor :
IDC_STATIC(3)	Caption	Resultado :
IDC_EDIT 1	ID Number	IDC_LADOMAYOR True
IDC_EDIT 2	ID Number	IDC_LADOMENOR True
IDC_EDIT 3	ID Number Read-only	IDC_RESULTADO True True
IDC_BUTTON(1)	ID Caption Default button	IDC_CALCULAR &Calcular True
IDC_BUTTON(2)	ID Caption	IDC_NUEVO &Nuevo

Editamos la ventana y agregamos las variables



Dar clic en el boton calcular e introducir el siguiente codigo

Add Member Function

Member function name:

OnCalcular

OK

Cancel

Message: BN_CLICKED

Object ID: IDC_CALCULAR

```
void CArea_Rectangulo::OnCalcular()
{
    UpdateData(true);
    m_Resultado = m_LadoMayor*m_LadoMenor;
    UpdateData(false);
}
```

Luego dar clic en el botón nuevo y escribir el codigo

Add Member Function

Member function name:

OnNuevo

OK

Cancel

Message: BN_CLICKED

Object ID: IDC_NUEVO

```
void CArea_Rectangulo::OnNuevo()
{
    m_Resultado=m_LadoMayor*m_LadoMenor=0;
    UpdateData (false);
}
```

IV.-PARA CREAR LA VENTANA DE DIALOGO AREA DEL CIRCULO

Control	Propiedades	Valor
IDD_CIRCULO	Caption	Area del Circulo
IDC_STATIC(1)	Caption	Radio:
IDC_STATIC(2)	Caption	Resultado :
IDC_EDIT 1	ID Number	IDC_RADIO True
IDC_EDIT 2	ID Number Read-only	IDC_AREA True True
IDC_BUTTON(1)	ID Caption Default button	IDC_CALCULAR &Calcular True
IDC_BUTTON(2)	ID Caption	IDC_NUEVO &Nuevo

Editamos la ventana y agregamos las variables

Area Circulo

Radio

Edit

Calcular

Resultado

Edit

Nuevo

MFC ClassWizard

Message Maps

Member Variables

Automation

ActiveX Events

Class Info

Project:

ROJASJIMENEZ

Class name:

CArea_Circulo

Add Class...

Control ID:

Type

Member

IDC_CALCULAR

IDC_NUEVO

IDC_RADIO

double

m_Radio

IDC_RESULTADO

double

m_Resultado

Description:

double with range validation

Minimum Value:

Magimum Value:

OK

Cancel

Dar clic en el boton calcular e introducir el siguiente codigo

Add Member Function

Member function name:

OnCalcular

OK

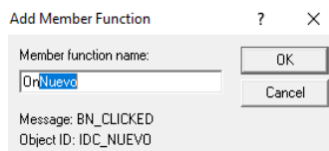
Cancel

Message: BN_CLICKED

Object ID: IDC_CALCULAR

```
void CArea_Circulo::OnCalcular()
{
    UpdateData(true);
    m_Resultado = 3.1415 * m_Radio * m_Radio;
    UpdateData(false);
}
```


A continuación dar clic en el botón nuevo y escribir el código



```
void CArea_Circulo::OnNuevo()  
{  
    m_Resultado=m_Radio=0;  
    UpdateData (false);  
}
```

FINALMENTE EJECUTAMOS Y PROBAMOS LAS SIGUIENTES OPERACIONES DE CADA FIGURA

