Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Институт информационных технологий

Факультет повышения квалификации и переподготовки

Кафедра МПСС

**ОТЧЕТ**

По лабораторным работам №3-4 и самостоятельной работе №2

по дисциплине ТКП

Тема: «Реализация проекта сервера автоматизации на ATL»

Минск 2021

**Цель работы:** Изучение технологии создания простого компонента c использованием ATL и его использования в клиентском приложении. Создание внутрипроцессного сервера автоматизации в соответствии с примером. Создание клиентского приложения, демонстрирующего использование сервера автоматизации. Исследование влияния способа подключения компонента на скорость взаимодействия клиента и сервера. Создание внутрипроцессного сервера автоматизации в соответствии с индивидуальным заданием. Создание клиентских приложений на Visual Basic, С++ и C#. Тестирование работы сервера.

long у = (long) a + (long) b; // метод Add

long y = (long) a - (long) b; // метод Subtract

long y = (long) a \* (long) b; // метод Multiply

long y = (long) a / (long) b; // метод Divide

**Ход работы:**

1. Создадим проект ATL - COMServer
2. Добавим в проект простой объект ATL MyMath и IWelcome, сборка решения.

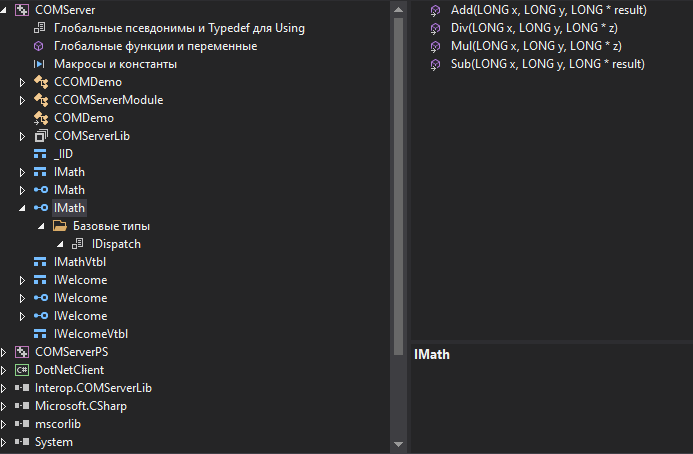
|  |
| --- |
| // COMServer.idl: источник языка IDL для COMServer  //  // Этот файл будет обработан средством MIDL для  // создание библиотеки типов ($handlersafeidlname$.tlb) и кода маршалирования.  import "oaidl.idl";  import "ocidl.idl";  [  object,  uuid(111fe4f4-65aa-4b41-af0e-8fad883469b7),  dual,  nonextensible,  pointer\_default(unique),  custom(0F21F359-AB84-41E8-9A78-36D110E6D2F9, "Interop.Server.IWelcome")  ]    interface IWelcome : IDispatch  {  [id(1), helpstring("Приветствие")] HRESULT Greeting([in] BSTR name, [out,retval] BSTR\* message);  };  ///////////////////////////////////////////////////////////////////////////  [  object,  uuid(509FF51A-AAFC-4F20-AEB9-069F005BC577),  dual,  nonextensible,  pointer\_default(unique),  custom(0F21F359-AB84-41E8-9A78-36D110E6D2F9, "Interop.Server.IMath")  ]  interface IMath : IDispatch  {  [id(1), helpstring("Сложение")]  HRESULT Add([in] LONG x, LONG y, [out, retval]LONG\* result);  [id(2), helpstring("Вычитание")]  HRESULT Sum([in] LONG x, LONG y, [out, retval]LONG\* result);  };  ///////////////////////////////////////////////////////////////////////////  [  uuid(84314509-1c0c-4123-a5c7-a8ad7b1b50ec),  version(1.0),  ]  library COMServerLib  {  importlib("stdole2.tlb");  [  uuid(6bc89994-c72b-4287-947e-cfba0824d0d3),  custom(0F21F359-AB84-41E8-9A78-36D110E6D2F9, "Interop.Server.COMDemo")  ]  coclass COMDemo  {  [default] interface IWelcome;  interface IMath;  };  };  import "shobjidl.idl"; |

1. Добавим методы Add, Sub в интерфейс IMyMath с помощью мастера;

|  |
| --- |
| COMDemo.cpp  […]  STDMETHODIMP CCOMDemo::Greeting(BSTR name, BSTR\* message)  {  CComBSTR tmp("Welcome, ");  tmp.Append(name);  \*message = tmp;  return S\_OK;  }  STDMETHODIMP CCOMDemo::Add(LONG x, LONG y, LONG\* z)  {  \*z = x + y;  return S\_OK;  }  STDMETHODIMP CCOMDemo::Sub(LONG x, LONG y, LONG\* z)  {  \*z = x - y;  return S\_OK;  }  […] |

А также добавим методы Mul и Div.

Обозреватель решений:

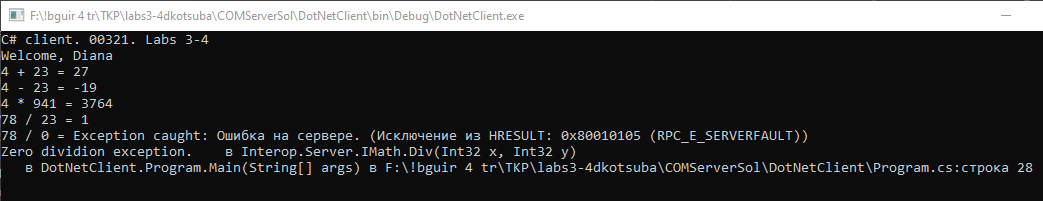


Полный код программы в Приложении 1

1. Создание клиентских приложений на Visual Basic, С++ и C#.

Проект ***DotNetClient***

|  |
| --- |
| // Program.cs  using System;  using System.Runtime.InteropServices;  using Interop.Server;  namespace DotNetClient  {  class Program  {  static void Main(string[] args)  {  var obj = new COMDemo();  IWelcome welcome = obj;  Console.WriteLine("C# client. 00321. Labs 3-4");  Console.WriteLine(welcome.Greeting("Diana"));  IMath math;  math = (IMath)welcome;  Console.Write("4 + 23 = ");  Console.WriteLine(math.Add(4, 23));  Console.Write("4 - 23 = ");  Console.WriteLine(math.Sub(4, 23));  Console.Write("4 \* 941 = ");  Console.WriteLine(math.Mul(4, 941));  Console.Write("78 / 23 = ");  Console.WriteLine(math.Div(78, 23));  Console.Write("78 / 0 = ");  try  {  Console.WriteLine(math.Div(78, 0)); //try to handle null div exception  }    catch (COMException e)  {  Console.WriteLine("Exception caught: {0}", e.Message);  if (e.ErrorCode == -2147417851)  {  Console.WriteLine($"Zero dividion exception. {e.StackTrace}");  }  //Console.WriteLine("Exception caught: {0}", e.InnerException);  //Console.WriteLine("Eerrorcode: {0}", e.ErrorCode);  //Console.WriteLine("HResult: {0}", e.HResult);  //Console.WriteLine("Data: {0}", e.Data);  //Console.WriteLine("trace: {0}", e.StackTrace);  }  //Console.WriteLine("C# client.00321. SR v14");  Console.ReadKey();    //принудительно уменьшим количество ссылок  Marshal.ReleaseComObject(math);  }  }  } |



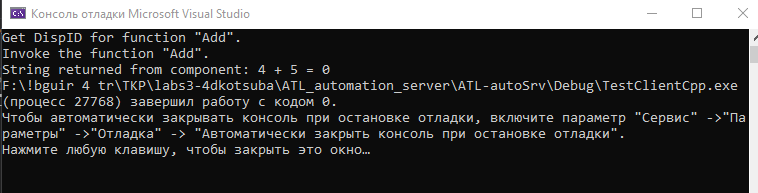
Проект ***VBClientForATLServer***

|  |
| --- |
| 'Form1.vb  Public Class Form1  Const ITERATIONS = 500000  Private Sub BtLate\_Click(sender As Object, e As EventArgs) Handles BtLate.Click  ' Пример позднего связывания  Dim obj As Object  Dim i As Long  Dim Start As DateTime = DateTime.Now  ' Создание объекта с указанным ProgId  'obj = CreateObject("ATLautoSrvLib.CoMyMath.1")  obj = CreateObject("ATLautoSrvLib.CoMyMath.1")  txtResult.Text = obj.Add(txtOp1.Text, txtOp2.Text)  For i = 1 To ITERATIONS  obj.Add(txtOp1.Text, txtOp2.Text)  Next  txtTime1.Text = (DateTime.Now - Start).TotalSeconds  End Sub  Private Sub BtVtbl\_Click(sender As Object, e As EventArgs) Handles BtVtbl.Click  ' Пример связывания через vtbl(очень раннего связывания)  Dim i As Long  Dim obj As New ATLautoSrvLib.CoMyMath  Dim Start As DateTime = DateTime.Now  txtResult.Text = obj.Add(txtOp1.Text, txtOp2.Text)  For i = 1 To ITERATIONS  obj.Add(txtOp1.Text, txtOp2.Text)  Next  txtTime3.Text = (DateTime.Now - Start).TotalSeconds  End Sub  End Class |

|  |
| --- |
| 'Form1.Designer.vb  <Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()>  Partial Class Form1  Inherits System.Windows.Forms.Form  'Form overrides dispose to clean up the component list.  <System.Diagnostics.DebuggerNonUserCode()> \_  Protected Overrides Sub Dispose(ByVal disposing As Boolean)  Try  If disposing AndAlso components IsNot Nothing Then  components.Dispose()  End If  Finally  MyBase.Dispose(disposing)  End Try  End Sub  'Required by the Windows Form Designer  Private components As System.ComponentModel.IContainer  'NOTE: The following procedure is required by the Windows Form Designer  'It can be modified using the Windows Form Designer.  'Do not modify it using the code editor.  <System.Diagnostics.DebuggerStepThrough()>  Private Sub InitializeComponent()  'components = New System.ComponentModel.Container()  'Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font  'Me.ClientSize = New System.Drawing.Size(800, 450)  'Me.Text = "Form1"  Me.txtOp1 = New System.Windows.Forms.TextBox()  Me.txtOp2 = New System.Windows.Forms.TextBox()  Me.txtResult = New System.Windows.Forms.TextBox()  Me.BtLate = New System.Windows.Forms.Button()  Me.BtVtbl = New System.Windows.Forms.Button()  Me.txtTime1 = New System.Windows.Forms.TextBox()  Me.txtTime3 = New System.Windows.Forms.TextBox()  Me.SuspendLayout()  '  'txtOp1  '  Me.txtOp1.Location = New System.Drawing.Point(13, 34)  Me.txtOp1.Name = "txtOp1"  Me.txtOp1.Size = New System.Drawing.Size(148, 20)  Me.txtOp1.TabIndex = 0  Me.txtOp1.Text = "7"  '  'txtOp2  '  Me.txtOp2.Location = New System.Drawing.Point(209, 34)  Me.txtOp2.Name = "txtOp2"  Me.txtOp2.Size = New System.Drawing.Size(148, 20)  Me.txtOp2.TabIndex = 1  Me.txtOp2.Text = "44"  '  'txtResult  '  Me.txtResult.Location = New System.Drawing.Point(399, 34)  Me.txtResult.Name = "txtResult"  Me.txtResult.Size = New System.Drawing.Size(148, 20)  Me.txtResult.TabIndex = 2  '  'BtLate  '  Me.BtLate.Location = New System.Drawing.Point(35, 98)  Me.BtLate.Name = "BtLate"  Me.BtLate.Size = New System.Drawing.Size(98, 30)  Me.BtLate.TabIndex = 3  Me.BtLate.Text = "Late"  Me.BtLate.UseVisualStyleBackColor = True  '  'BtVtbl  '  Me.BtVtbl.Location = New System.Drawing.Point(35, 167)  Me.BtVtbl.Name = "BtVtbl"  Me.BtVtbl.Size = New System.Drawing.Size(98, 30)  Me.BtVtbl.TabIndex = 4  Me.BtVtbl.Text = "Vtbl"  Me.BtVtbl.UseVisualStyleBackColor = True  '  'txtTime1  '  Me.txtTime1.Location = New System.Drawing.Point(209, 108)  Me.txtTime1.Name = "txtTime1"  Me.txtTime1.Size = New System.Drawing.Size(148, 20)  Me.txtTime1.TabIndex = 5  '  'txtTime3  '  Me.txtTime3.Location = New System.Drawing.Point(209, 177)  Me.txtTime3.Name = "txtTime3"  Me.txtTime3.Size = New System.Drawing.Size(148, 20)  Me.txtTime3.TabIndex = 6  '  'Form1  '  Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)  Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font  Me.ClientSize = New System.Drawing.Size(800, 450)  Me.Controls.Add(Me.txtTime3)  Me.Controls.Add(Me.txtTime1)  Me.Controls.Add(Me.BtVtbl)  Me.Controls.Add(Me.BtLate)  Me.Controls.Add(Me.txtResult)  Me.Controls.Add(Me.txtOp2)  Me.Controls.Add(Me.txtOp1)  Me.Name = "VB client, 00321"  Me.Text = "Kotsuba Diana Visual Basic Client Test"  Me.ResumeLayout(False)  Me.PerformLayout()  End Sub  Friend WithEvents txtOp1 As TextBox  Friend WithEvents txtOp2 As TextBox  Friend WithEvents txtResult As TextBox  Friend WithEvents BtLate As Button  Friend WithEvents BtVtbl As Button  Friend WithEvents txtTime1 As TextBox  Friend WithEvents txtTime3 As TextBox  End Class |

Проект ***TestClientCpp***

|  |
| --- |
| //TestClientCpp.cpp  #include <iostream>  #include <windows.h>  #include <stdio.h>  int main()  {  // std::cout << "Hello World!\n";  DWORD clsctx;  clsctx = CLSCTX\_INPROC\_SERVER;  printf("Kotsuba D Test CPlus Client\n");  HRESULT hr = OleInitialize(NULL);  if (FAILED(hr))  {  printf("Failed to initialize.hr=%X \n", hr);  return 1;  }  // Get the CLSID for the application.  wchar\_t progid[] = L"Atl-autoSrv.CoMyMath.1";  CLSID clsid;  hr = ::CLSIDFromProgID(progid, &clsid);  if (FAILED(hr))  {  printf("Failed to get CLSID.hr=%X \n", hr);  return 1;  }  // Create the component.  IDispatch\* pIDispatch = NULL;  hr = ::CoCreateInstance(clsid,  NULL,  clsctx,  IID\_IDispatch,  (void\*\*)&pIDispatch);  if (FAILED(hr))  {  printf("Create instance failed.hr=%X \n", hr);  OleUninitialize();  return 1;  }  printf("CoCreateInstance succeeded.\n");  // First we need to get the IDs for the function names.  printf("Get DispID for function \"Add\".\n");  DISPID dispid;  OLECHAR\* name = (OLECHAR\*)L"Add";  hr = pIDispatch->GetIDsOfNames(IID\_NULL,  &name,  1,  GetUserDefaultLCID(),  &dispid);  if (FAILED(hr))  {  printf("Query GetIDsOfNames failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Prepare the arguments for Fx.  VARIANT vargs[2];  VariantInit(&vargs[1]);  vargs[1].vt = VT\_I4;  vargs[1].lVal = 4;  VariantInit(&vargs[0]);  vargs[0].vt = VT\_I4;  vargs[0].lVal = 5;  DISPPARAMS dispparams = { vargs, NULL, 2, 0 };  VARIANT result;  VariantInit(&result);  result.vt = VT\_I4;  printf("Invoke the function \"Add\".\n");  hr = pIDispatch->Invoke(dispid,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_METHOD,  &dispparams,  &result,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  if (result.vt == VT\_I4)  {  printf("String returned from component: %d + %d = %d", vargs[1].lVal, vargs[0].llVal, result.lVal);  }  // Release the dispatch interface.  pIDispatch->Release();  // Uninitialize the OLE library.  OleUninitialize();  return 0;  } |



**Код программы ATL-autoSrv:**

|  |
| --- |
| // ATL-autoSrv.cpp: реализация экспортов DLL.  #include "pch.h"  #include "framework.h"  #include "resource.h"  #include "ATLautoSrv\_i.h"  #include "dllmain.h"  using namespace ATL;  // Используется, чтобы определить, можно ли выгрузить DLL средствами OLE.  \_Use\_decl\_annotations\_  STDAPI DllCanUnloadNow(void)  {  return \_AtlModule.DllCanUnloadNow();  }  // Возвращает фабрику класса для создания объекта требуемого типа.  \_Use\_decl\_annotations\_  STDAPI DllGetClassObject(\_In\_ REFCLSID rclsid, \_In\_ REFIID riid, \_Outptr\_ LPVOID\* ppv)  {  return \_AtlModule.DllGetClassObject(rclsid, riid, ppv);  }  // DllRegisterServer - добавляет записи в системный реестр.  \_Use\_decl\_annotations\_  STDAPI DllRegisterServer(void)  {  // регистрирует объект, библиотеку типов и все интерфейсы в библиотеке типов  HRESULT hr = \_AtlModule.DllRegisterServer();  return hr;  }  // DllUnregisterServer - удаляет записи из системного реестра.  \_Use\_decl\_annotations\_  STDAPI DllUnregisterServer(void)  {  HRESULT hr = \_AtlModule.DllUnregisterServer();  return hr;  }  // DllInstall - добавляет и удаляет записи системного реестра для конкретного пользователя и компьютера.  STDAPI DllInstall(BOOL bInstall, \_In\_opt\_ LPCWSTR pszCmdLine)  {  HRESULT hr = E\_FAIL;  static const wchar\_t szUserSwitch[] = L"user";  if (pszCmdLine != nullptr)  {  if (\_wcsnicmp(pszCmdLine, szUserSwitch, \_countof(szUserSwitch)) == 0)  {  ATL::AtlSetPerUserRegistration(true);  }  }  if (bInstall)  {  hr = DllRegisterServer();  if (FAILED(hr))  {  DllUnregisterServer();  }  }  else  {  hr = DllUnregisterServer();  }  return hr;  } |

|  |
| --- |
| ; ATL-autoSrv.def: объявляет параметры модуля.  LIBRARY  EXPORTS  DllCanUnloadNow PRIVATE  DllGetClassObject PRIVATE  DllRegisterServer PRIVATE  DllUnregisterServer PRIVATE  DllInstall PRIVATE |

|  |
| --- |
| // ATL-autoSrv.idl: источник языка IDL для ATL-autoSrv  //  // Этот файл будет обработан средством MIDL для  // создание библиотеки типов ($handlersafeidlname$.tlb) и кода маршалирования.  import "oaidl.idl";  import "ocidl.idl";  [  object,  uuid(fb96f5a4-2a4c-4f26-9f0b-14fa7f7c057b),  dual,  nonextensible,  pointer\_default(unique)  ]  interface IMyMath : IDispatch  {  [id(1), helpstring("addition")] HRESULT Add([in] LONG x, [in] LONG y, [out, retval] LONG\* z);  [id(2), helpstring("substraction")] HRESULT Sub([in] LONG x, [in] LONG y, [out, retval] LONG\* z);  [id(3), helpstring("multiply")] HRESULT Mul([in] LONG x, [in] LONG y, [out, retval] LONG\* z);  [id(4), helpstring("divide")] HRESULT Div([in] LONG x, [in] LONG y, [out, retval] LONG\* z);  };  [  uuid(6d2ec9b2-2f7a-4358-8fb4-d333b8edd6fa),  version(1.0),  ]  library ATLautoSrvLib  {  importlib("stdole2.tlb");  [  uuid(4e488da0-e934-4289-9a8a-0d9182fed9c7)  ]  coclass CoMyMath  {  [default] interface IMyMath;  };  };  import "shobjidl.idl"; |

|  |
| --- |
| // dllmain.cpp: реализация DllMain.  #include "pch.h"  #include "framework.h"  #include "resource.h"  #include "ATLautoSrv\_i.h"  #include "dllmain.h"  CATLautoSrvModule \_AtlModule;  // Точка входа DLL  extern "C" BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved)  {  hInstance;  return \_AtlModule.DllMain(dwReason, lpReserved);  } |

|  |
| --- |
| // MyMath.cpp: реализация CMyMath  #include "pch.h"  #include "MyMath.h"  // CMyMath  STDMETHODIMP CMyMath::InterfaceSupportsErrorInfo(REFIID riid)  {  static const IID\* const arr[] =  {  &IID\_IMyMath  };  for (int i=0; i < sizeof(arr) / sizeof(arr[0]); i++)  {  if (InlineIsEqualGUID(\*arr[i],riid))  return S\_OK;  }  return S\_FALSE;  }  STDMETHODIMP CMyMath::Add(LONG x, LONG y, LONG\* z)  {  \*z = x + y;  return S\_OK;  }  STDMETHODIMP CMyMath::Sub(LONG x, LONG y, LONG\* z)  {  \*z = x - y;  return S\_OK;  }  STDMETHODIMP CMyMath::Mul(LONG x, LONG y, LONG\* z)  {  \*z = x \* y;  return S\_OK;  }  STDMETHODIMP CMyMath::Div(LONG x, LONG y, LONG\* z)  {  \*z = x / y;  return S\_OK;  } |

|  |
| --- |
| // MyMath.h: объявление CMyMath  #pragma once  #include "resource.h" // основные символы  #include "ATLautoSrv\_i.h"  #if defined(\_WIN32\_WCE) && !defined(\_CE\_DCOM) && !defined(\_CE\_ALLOW\_SINGLE\_THREADED\_OBJECTS\_IN\_MTA)  #error "Однопотоковые COM-объекты не поддерживаются должным образом платформой Windows CE, например платформами Windows Mobile, в которых не предусмотрена полная поддержка DCOM. Определите \_CE\_ALLOW\_SINGLE\_THREADED\_OBJECTS\_IN\_MTA, чтобы принудить ATL поддерживать создание однопотоковых COM-объектов и разрешить использование его реализаций однопотоковых COM-объектов. Для потоковой модели в вашем rgs-файле задано значение 'Free', поскольку это единственная потоковая модель, поддерживаемая не-DCOM платформами Windows CE."  #endif  using namespace ATL;  // CMyMath  class ATL\_NO\_VTABLE CMyMath :  public CComObjectRootEx<CComSingleThreadModel>,  public CComCoClass<CMyMath, &CLSID\_CoMyMath>,  public ISupportErrorInfo,  public IDispatchImpl<IMyMath, &IID\_IMyMath, &LIBID\_ATLautoSrvLib, /\*wMajor =\*/ 1, /\*wMinor =\*/ 0>  {  public:  CMyMath()  {  }  DECLARE\_REGISTRY\_RESOURCEID(106)  BEGIN\_COM\_MAP(CMyMath)  COM\_INTERFACE\_ENTRY(IMyMath)  COM\_INTERFACE\_ENTRY(IDispatch)  COM\_INTERFACE\_ENTRY(ISupportErrorInfo)  END\_COM\_MAP()  // ISupportsErrorInfo  STDMETHOD(InterfaceSupportsErrorInfo)(REFIID riid);  DECLARE\_PROTECT\_FINAL\_CONSTRUCT()  HRESULT FinalConstruct()  {  return S\_OK;  }  void FinalRelease()  {  }  public:  STDMETHOD(Add)(LONG x, LONG y, LONG\* z);  STDMETHOD(Sub)(LONG x, LONG y, LONG\* z);  STDMETHOD(Mul)(LONG x, LONG y, LONG\* z);  STDMETHOD(Div)(LONG x, LONG y, LONG\* z);  };  OBJECT\_ENTRY\_AUTO(\_\_uuidof(CoMyMath), CMyMath) |

Проект ***COMServerSol***

|  |
| --- |
| // COMDemo.cpp: реализация CCOMDemo  #include "pch.h"  #include "COMDemo.h"  #include <string>  #include <iostream>  // CCOMDemo  STDMETHODIMP CCOMDemo::Greeting(BSTR name, BSTR\* message)  {  CComBSTR tmp("Welcome, ");  tmp.Append(name);  \*message = tmp;  return S\_OK;  }  STDMETHODIMP CCOMDemo::Add(LONG x, LONG y, LONG\* z)  {  \*z = x + y;  return S\_OK;  }  STDMETHODIMP CCOMDemo::Sub(LONG x, LONG y, LONG\* z)  {  \*z = x - y;  return S\_OK;  }  STDMETHODIMP CCOMDemo::Mul(LONG x, LONG y, LONG\* z)  {  \*z = x \* y;  return S\_OK;  }  STDMETHODIMP CCOMDemo::Div(LONG x, LONG y, LONG\* z)  {  if (y == 0) {    throw EXCEPTION\_FLT\_DIVIDE\_BY\_ZERO;  }  \*z = 1;// x / y;      return S\_OK;    }  //STDMETHODIMP CCOMDemo::CreateError()  //{  // CComBSTR someText(L"CreateError Test");  // ICreateErrorInfo\* pCreateError;  // IErrorInfo\* pError;  // HRESULT result = CreateErrorInfo(&pCreateError);  // /\*if (FAILED(result))  // return E\_NOTIMPL;  // result = pCreateError->QueryInterface(&pError);\*/  // if (FAILED(result))  // return E\_NOTIMPL;  // result = pCreateError->SetDescription(someText);  // pError->Release();  // pCreateError->Release();  // return E\_FAIL;  //} |

|  |
| --- |
| // COMServer.cpp: реализация экспортов DLL.  #include "pch.h"  #include "framework.h"  #include "resource.h"  #include "COMServer\_i.h"  #include "dllmain.h"  using namespace ATL;  // Используется, чтобы определить, можно ли выгрузить DLL средствами OLE.  \_Use\_decl\_annotations\_  STDAPI DllCanUnloadNow(void)  {  return \_AtlModule.DllCanUnloadNow();  }  // Возвращает фабрику класса для создания объекта требуемого типа.  \_Use\_decl\_annotations\_  STDAPI DllGetClassObject(\_In\_ REFCLSID rclsid, \_In\_ REFIID riid, \_Outptr\_ LPVOID\* ppv)  {  return \_AtlModule.DllGetClassObject(rclsid, riid, ppv);  }  // DllRegisterServer - добавляет записи в системный реестр.  \_Use\_decl\_annotations\_  STDAPI DllRegisterServer(void)  {  // регистрирует объект, библиотеку типов и все интерфейсы в библиотеке типов  HRESULT hr = \_AtlModule.DllRegisterServer();  return hr;  }  // DllUnregisterServer - удаляет записи из системного реестра.  \_Use\_decl\_annotations\_  STDAPI DllUnregisterServer(void)  {  HRESULT hr = \_AtlModule.DllUnregisterServer();  return hr;  }  // DllInstall - добавляет и удаляет записи системного реестра для конкретного пользователя и компьютера.  STDAPI DllInstall(BOOL bInstall, \_In\_opt\_ LPCWSTR pszCmdLine)  {  HRESULT hr = E\_FAIL;  static const wchar\_t szUserSwitch[] = L"user";  if (pszCmdLine != nullptr)  {  if (\_wcsnicmp(pszCmdLine, szUserSwitch, \_countof(szUserSwitch)) == 0)  {  ATL::AtlSetPerUserRegistration(true);  }  }  if (bInstall)  {  hr = DllRegisterServer();  if (FAILED(hr))  {  DllUnregisterServer();  }  }  else  {  hr = DllUnregisterServer();  }  return hr;  } |

|  |
| --- |
| ; COMServer.def: объявляет параметры модуля.  LIBRARY  EXPORTS  DllCanUnloadNow PRIVATE  DllGetClassObject PRIVATE  DllRegisterServer PRIVATE  DllUnregisterServer PRIVATE  DllInstall PRIVATE |

|  |
| --- |
| // COMServer.idl: источник языка IDL для COMServer  //  // Этот файл будет обработан средством MIDL для  // создание библиотеки типов ($handlersafeidlname$.tlb) и кода маршалирования.  import "oaidl.idl";  import "ocidl.idl";  [  object,  uuid(111fe4f4-65aa-4b41-af0e-8fad883469b7),  dual,  nonextensible,  pointer\_default(unique),  custom(0F21F359-AB84-41E8-9A78-36D110E6D2F9, "Interop.Server.IWelcome")  ]    interface IWelcome : IDispatch  {  [id(1), helpstring("Приветствие")] HRESULT Greeting([in] BSTR name, [out,retval] BSTR\* message);  };  ///////////////////////////////////////////////////////////////////////////  [  object,  uuid(509FF51A-AAFC-4F20-AEB9-069F005BC577),  dual,  nonextensible,  pointer\_default(unique),  custom(0F21F359-AB84-41E8-9A78-36D110E6D2F9, "Interop.Server.IMath")  ]  interface IMath : IDispatch  {  [id(1), helpstring("Сложение")]  HRESULT Add([in] LONG x, LONG y, [out, retval]LONG\* result);  [id(2), helpstring("Вычитание")]  HRESULT Sub([in] LONG x, LONG y, [out, retval]LONG\* result);  [id(3), helpstring("multiply")] HRESULT Mul([in] LONG x, [in] LONG y, [out, retval] LONG\* z);  [id(4), helpstring("divide")] HRESULT Div([in] LONG x, [in] LONG y, [out, retval] LONG\* z);  };  ///////////////////////////////////////////////////////////////////////////  [  uuid(84314509-1c0c-4123-a5c7-a8ad7b1b50ec),  version(1.0),  ]  library COMServerLib  {  importlib("stdole2.tlb");  [  uuid(6bc89994-c72b-4287-947e-cfba0824d0d3),  custom(0F21F359-AB84-41E8-9A78-36D110E6D2F9, "Interop.Server.COMDemo")  ]  coclass COMDemo  {  [default] interface IWelcome;  interface IMath;  };  };  import "shobjidl.idl"; |

**Самостоятельная работа №2**

Создание сервера внутри процесса (Inprocserver) для элемента Автоматизации в соответствии с индивидуальным заданием лабораторной работы TCP\_LB3 и клиентских приложений к нему на VB, С++ и C#. Компонент реализует дуальные интерфейсы. Вызов методов интерфейсов осуществлять через IDispatch и Invoke.

Вариант 14

IFN\_14

double Fun141 (int, int)

int Fun142 (int, int, int)

void Fun143 (double in, double \*out )

**Ход работы:**

1. Создание проекта ***SR2ATLAutoSvrSol*** типа «Проект ATL» -> «Библиотека динамической компоновки (DLL)»
2. Добавление в проект простого объекта ATL SR2MyMath;
3. Сборка решения.
4. Добавление метода fun141, fun142, fun143 в интерфейс ISR2MyMath с помощью мастера.
5. Добавление методов fun142, fun143.
6. Добавление реализации методов в SR2MyMath.cpp;
7. Создание клиентских приложений.

Проект ***SR2ATLAutoSvrSol***

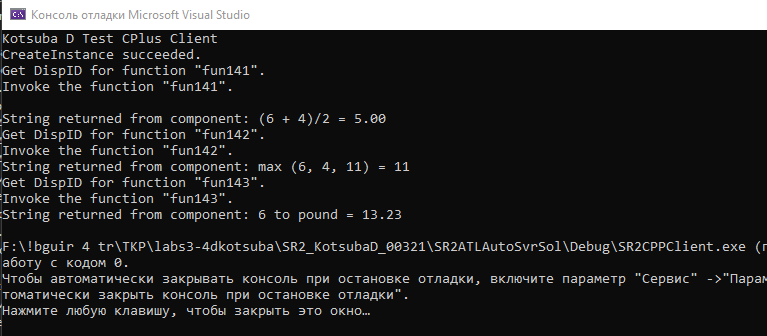
|  |
| --- |
| // SR2ATLAutoSvr.idl: источник языка IDL для SR2ATLAutoSvr  //  // Этот файл будет обработан средством MIDL для  // создание библиотеки типов ($handlersafeidlname$.tlb) и кода маршалирования.  import "oaidl.idl";  import "ocidl.idl";  [  object,  uuid(ef79723f-48f7-4fb1-afc9-c0ddbbc32eac),  dual,  nonextensible,  pointer\_default(unique)  ]  interface ISR2MyMath : IDispatch  {  [id(1), helpstring("Среднее значение")] HRESULT fun141([in] LONG x, [in] LONG y, [out, retval] DOUBLE\* z);  [id(2), helpstring("Макимальное значение")] HRESULT fun142([in] LONG x, [in] LONG y, [in] LONG z, [out, retval] LONG\* output); //LONG x, LONG y, LONG z, LONG\* output  [id(3), helpstring("kg -> pound")] HRESULT fun143([in] DOUBLE x, [out, retval] DOUBLE\* output); //DOUBLE x, DOUBLE\* output  };  [  uuid(95263321-2bc9-4785-aa57-472da3ebc9d5),  version(1.0),  ]  library SR2ATLAutoSvrLib  {  importlib("stdole2.tlb");  [  uuid(a76ea009-d52b-4d40-9444-3301a1357ed9)  ]  coclass SR2MyMath  {  [default] interface ISR2MyMath;  };  };  import "shobjidl.idl"; |

|  |
| --- |
| // SR2MyMath.cpp: реализация CSR2MyMath  #include "pch.h"  #include "SR2MyMath.h"  // CSR2MyMath  STDMETHODIMP CSR2MyMath::fun141(LONG x, LONG y, DOUBLE\* output)  {  \*output = ((double)x + (double)y) / 2.0;    return S\_OK;  }  STDMETHODIMP CSR2MyMath::fun142(LONG x, LONG y, LONG z, LONG\* output)  {  int r;  if (x > y)  r = x;  else  r = y;  if (y > r)  r = z;  \*output = r;  return S\_OK;  }  STDMETHODIMP CSR2MyMath::fun143(DOUBLE x, DOUBLE\* output)  {  //kg to pound  \*output = x \* 2.2046226;  return S\_OK;  } |

|  |
| --- |
| // SR2MyMath.h: объявление CSR2MyMath  #pragma once  #include "resource.h" // основные символы  #include "SR2ATLAutoSvr\_i.h"  #if defined(\_WIN32\_WCE) && !defined(\_CE\_DCOM) && !defined(\_CE\_ALLOW\_SINGLE\_THREADED\_OBJECTS\_IN\_MTA)  #error "Однопотоковые COM-объекты не поддерживаются должным образом платформой Windows CE, например платформами Windows Mobile, в которых не предусмотрена полная поддержка DCOM. Определите \_CE\_ALLOW\_SINGLE\_THREADED\_OBJECTS\_IN\_MTA, чтобы принудить ATL поддерживать создание однопотоковых COM-объектов и разрешить использование его реализаций однопотоковых COM-объектов. Для потоковой модели в вашем rgs-файле задано значение 'Free', поскольку это единственная потоковая модель, поддерживаемая не-DCOM платформами Windows CE."  #endif  using namespace ATL;  // CSR2MyMath  class ATL\_NO\_VTABLE CSR2MyMath :  public CComObjectRootEx<CComSingleThreadModel>,  public CComCoClass<CSR2MyMath, &CLSID\_SR2MyMath>,  public IDispatchImpl<ISR2MyMath, &IID\_ISR2MyMath, &LIBID\_SR2ATLAutoSvrLib, /\*wMajor =\*/ 1, /\*wMinor =\*/ 0>  {  public:  CSR2MyMath()  {  }  DECLARE\_REGISTRY\_RESOURCEID(106)  BEGIN\_COM\_MAP(CSR2MyMath)  COM\_INTERFACE\_ENTRY(ISR2MyMath)  COM\_INTERFACE\_ENTRY(IDispatch)  END\_COM\_MAP()  DECLARE\_PROTECT\_FINAL\_CONSTRUCT()  HRESULT FinalConstruct()  {  return S\_OK;  }  void FinalRelease()  {  }  public:  STDMETHOD(fun141)(LONG x, LONG y, DOUBLE\* z);  STDMETHOD(fun142)(LONG x, LONG y, LONG z, LONG\* output);  STDMETHOD(fun143)(DOUBLE x, DOUBLE\* output);  };  OBJECT\_ENTRY\_AUTO(\_\_uuidof(SR2MyMath), CSR2MyMath) |

C++ клиент. Проект ***SR2CPPClient.cpp***

|  |
| --- |
| //SR2CPPClient.cpp  #include <iostream>  #include <windows.h>  #include <stdio.h>  int main()  {  // std::cout << "Hello World!\n";  DWORD clsctx;  clsctx = CLSCTX\_INPROC\_SERVER;  printf("Kotsuba D Test CPlus Client\n");  HRESULT hr = OleInitialize(NULL);  if (FAILED(hr))  {  printf("Failed to initialize.hr=%X \n", hr);  return 1;  }  // Get the CLSID for the application.  wchar\_t progid[] = L"SR2ATLAutoSvr.SR2MyMath.1";  CLSID clsid;  hr = ::CLSIDFromProgID(progid, &clsid);  if (FAILED(hr))  {  printf("Failed to get CLSID.hr=%X \n", hr);  return 1;  }  // Create the component.  IDispatch\* pIDispatch = NULL;  hr = ::CoCreateInstance(clsid,  NULL,  clsctx,  IID\_IDispatch,  (void\*\*)&pIDispatch);  if (FAILED(hr))  {  printf("Create instance failed.hr=%X \n", hr);  OleUninitialize();  return 1;  }  printf("CreateInstance succeeded.\n");  ////////////////////////////////////////////////////////////////  // First we need to get the IDs for the function names.  printf("Get DispID for function \"fun141\".\n");  DISPID dispid141;  OLECHAR\* name141 = (OLECHAR\*)L"fun141";  hr = pIDispatch->GetIDsOfNames(IID\_NULL,  &name141,  1,  GetUserDefaultLCID(),  &dispid141);  if (FAILED(hr))  {  printf("Query GetIDsOfNames failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Prepare the arguments for Fx fun141.  VARIANT vargs141[2];  VariantInit(&vargs141[1]);  vargs141[1].vt = VT\_I4;  vargs141[1].lVal = 6;  VariantInit(&vargs141[0]);  vargs141[0].vt = VT\_I4;  vargs141[0].lVal = 4;  VARIANT result141;  VariantInit(&result141);  result141.vt = VT\_I4;  DISPPARAMS dispparams141 = { vargs141, NULL, 2, 0 };  printf("Invoke the function \"fun141\".\n");  hr = pIDispatch->Invoke(dispid141,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_METHOD,  &dispparams141,  &result141,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  printf("\nString returned from component: (%d + %d)/2 = %0.2f \n", vargs141[1].lVal, vargs141[0].lVal, result141.date);    //////////////////////////////////////////////////////////////////////////////////  // First we need to get the IDs for the function names.  printf("Get DispID for function \"fun142\".\n");  DISPID dispid142;  OLECHAR\* name142 = (OLECHAR\*)L"fun142";  hr = pIDispatch->GetIDsOfNames(IID\_NULL,  &name142,  1,  GetUserDefaultLCID(),  &dispid142);  if (FAILED(hr))  {  printf("Query GetIDsOfNames failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Prepare the arguments for Fx fun142.  VARIANT vargs142[3];  VariantInit(&vargs142[1]);  vargs142[1].vt = VT\_I4;  vargs142[1].lVal = 6;  VariantInit(&vargs142[0]);  vargs142[0].vt = VT\_I4;  vargs142[0].lVal = 4;  VariantInit(&vargs142[2]);  vargs142[2].vt = VT\_I4;  vargs142[2].lVal = 11;  VARIANT result142;  VariantInit(&result142);  result142.vt = VT\_I4;  DISPPARAMS dispparams142 = { vargs142, NULL, 3, 0 };  printf("Invoke the function \"fun142\".\n");  hr = pIDispatch->Invoke(dispid142,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_METHOD,  &dispparams142,  &result142,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  printf("String returned from component: max (%d, %d, %d) = %d \n", vargs142[1].lVal, vargs142[0].lVal, vargs142[2].lVal, result142.llVal);  //////////////////////////////////////////////////////////////////////////////////  // First we need to get the IDs for the function names.  printf("Get DispID for function \"fun143\".\n");  DISPID dispid143;  OLECHAR\* name143 = (OLECHAR\*)L"fun143";  hr = pIDispatch->GetIDsOfNames(IID\_NULL,  &name143,  1,  GetUserDefaultLCID(),  &dispid143);  if (FAILED(hr))  {  printf("Query GetIDsOfNames failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Prepare the arguments for Fx fun142.  VARIANT vargs143[3];  VariantInit(&vargs143[0]);  vargs143[0].vt = VT\_I4;  vargs143[0].lVal = 6;    VARIANT result143;  VariantInit(&result143);  result143.vt = VT\_I4;  DISPPARAMS dispparams143 = { vargs143, NULL, 1, 0 };  printf("Invoke the function \"fun143\".\n");  hr = pIDispatch->Invoke(dispid143,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_METHOD,  &dispparams143,  &result143,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  printf("String returned from component: %d to pound = %0.2f \n", vargs143[0].lVal, result143.dblVal);  // Release the dispatch interface.  pIDispatch->Release();  // Uninitialize the OLE library.  OleUninitialize();  return 0;  } |



**DOTNETClient**

|  |
| --- |
| //program.cs  using System;  using System.Runtime.InteropServices;  using SR2ATLAutoSvrLib;  namespace SR2DotNetClient  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("SR2 Kotsuba D 00321");  ISR2MyMath math = new SR2MyMath();  Console.WriteLine($"fun141: (151 + 149)/2 = {math.fun141(151, 149)}");  Console.WriteLine($"fun142: 8 \* 3 \* 5 = {math.fun142(81, 74, 18)}");  Console.WriteLine($"fun143: 11 to pound 2 = {math.fun143(11)}");  Marshal.ReleaseComObject(math);  Console.ReadKey();  }  }  } |

