ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО

ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ

ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. Н.П. ОГАРЁВА»

Направление: Информатика и вычислительная техника

**ЛАБОРАТОРНАЯ РАБОТА**

по программной инженерии

ЛР – 02069964 –090301– 10 – 19

Лабораторная работа № 10

Выполнил: Проверил:

Федосин С. А.

Студент 441 группы

19.12.2019

Леушкин С.О.

Абдуллаев Д.Д.

Саранск 2019

**Лабораторная работа 10**

Создание объектов Automation и компонентов ActiveX в Delphi

**Цели.**

* ознакомиться с технологией Automation;
* научиться создавать внешний сервер и клиент автоматизации в Delphi XE;
* разработать собственный ActiveX-компонент и приложение с его использованием.

Код программы:

Unit2.pas

|  |
| --- |
| unit Unit2;  interface  uses  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, StdCtrls,  Dialogs,Project2\_TLB;  type  TForm2 = class(TForm)  GroupBox1: TGroupBox;  Edit1: TEdit;  Edit2: TEdit;  Label1: TLabel;  Label2: TLabel;  Button1: TButton;  ListBox1: TListBox;  procedure Button1Click(Sender: TObject);  procedure FormShow(Sender: TObject);  private  { Private declarations }  public  { Public declarations }  end;  var  Form2: TForm2;  S:ITestAutoLab10;  implementation  {$R \*.dfm}  procedure TForm2.Button1Click(Sender: TObject);  var  V: OleVariant; i: Integer; Res: Single;  begin  Res := 0;  V := varArrayCreate([0, ListBox1.Items.Count - 1], varSingle);  for i := 0 to ListBox1.Items.Count - 1 do  V[i] := StrToFloat(ListBox1.Items[i]);  S.AVG(V, Res);  Edit1.Text:= FloatToStrF(Res, ffGeneral, 4, 2);  S.Dispersion(V, Res);  Edit2.Text := FloatToStrF(Res, ffGeneral, 4, 2);  end;  procedure TForm2.FormShow(Sender: TObject);  begin  S := COTestAutoLab10.Create;  end;  end. |

Unit1.pas

|  |
| --- |
| unit Unit1;  {$WARN SYMBOL\_PLATFORM OFF}  interface  uses  ComObj, ActiveX, Project2\_TLB, StdVcl;  type  TTestAutoLab10 = class(TAutoObject, ITestAutoLab10)  protected  procedure AVG(Param1: OleVariant; out Param2: Single); safecall;  procedure Dispersion(Param1: OleVariant; out Param2: Single); safecall;  end;  implementation  uses ComServ, Variants;  procedure TTestAutoLab10.AVG(Param1: OleVariant; out Param2: Single);  var  N1,N2,i : Integer;  A : Single;  begin  A := 0;  N1 := varArrayLowBound(Param1,1);  N2 := varArrayHighBound(Param1,1);  for i := N1 to N2 do  A:=A + Param1[i];  Param2 := A / (N2 - N1 + 1);  end;  procedure TTestAutoLab10.Dispersion(Param1: OleVariant; out Param2: Single);  var  N1,N2,i : Integer;  A,D : Single;  begin  //Ðàñ÷åò ñðåäíåãî  AVG(Param1,A);  //Ðàñ÷åò äèñïåðñèè  D := 0;  N1 := varArrayLowBound(Param1,1);  N2 := varArrayHighBound(Param1,1);  for i := N1 to N2 do  D := sqr(Param1[i] - A);  Param2 := D / (N2 - N1);  end;  initialization  TAutoObjectFactory.Create(ComServer, TTestAutoLab10, Class\_TestAutoLab10,  ciMultiInstance, tmApartment);  end. |

Unit3.pas

|  |
| --- |
| unit Unit3;  {$WARN SYMBOL\_PLATFORM OFF}  interface  uses  ComObj, ActiveX, Project2\_TLB, StdVcl;  type  TTestAutoLab10 = class(TAutoObject, ITestAutoLab10)  protected  procedure AVG(Param1: OleVariant; out Param2: Single); safecall;  procedure Dispersion(Param1: OleVariant; out Param2: Single); safecall;  end;  implementation  uses ComServ;  procedure TTestAutoLab10.AVG(Param1: OleVariant; out Param2: Single);  begin  end;  procedure TTestAutoLab10.Dispersion(Param1: OleVariant; out Param2: Single);  begin  end;  initialization  TAutoObjectFactory.Create(ComServer, TTestAutoLab10, Class\_TestAutoLab10,  ciMultiInstance, tmApartment);  end. |

Project2\_TLB.pas

|  |
| --- |
| unit Project2\_TLB;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //  // WARNING  // -------  // The types declared in this file were generated from data read from a  // Type Library. If this type library is explicitly or indirectly (via  // another type library referring to this type library) re-imported, or the  // 'Refresh' command of the Type Library Editor activated while editing the  // Type Library, the contents of this file will be regenerated and all  // manual modifications will be lost.  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //  // $Rev: 52393 $  // File generated on 01.11.2018 10:46:35 from Type Library described below.  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //  // Type Lib: G:\lab10\Project2 (1)  // LIBID: {D1D7E903-5B81-4A10-A57D-7F02F5D44324}  // LCID: 0  // Helpfile:  // HelpString:  // DepndLst:  // (1) v2.0 stdole, (C:\Windows\SysWOW64\stdole2.tlb)  // SYS\_KIND: SYS\_WIN32  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //  {$TYPEDADDRESS OFF} // Unit must be compiled without type-checked pointers.  {$WARN SYMBOL\_PLATFORM OFF}  {$WRITEABLECONST ON}  {$VARPROPSETTER ON}  {$ALIGN 4}  interface  uses Winapi.Windows, System.Classes, System.Variants, System.Win.StdVCL, Vcl.Graphics, Vcl.OleServer, Winapi.ActiveX;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // GUIDS declared in the TypeLibrary. Following prefixes are used:  // Type Libraries : LIBID\_xxxx  // CoClasses : CLASS\_xxxx  // DISPInterfaces : DIID\_xxxx  // Non-DISP interfaces: IID\_xxxx  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  const  // TypeLibrary Major and minor versions  Project2MajorVersion = 1;  Project2MinorVersion = 0;  LIBID\_Project2: TGUID = '{D1D7E903-5B81-4A10-A57D-7F02F5D44324}';  IID\_ITestAutoLab10: TGUID = '{130FB6C1-258C-4D0A-B039-D74B4DFA3FDE}';  CLASS\_TestAutoLab10: TGUID = '{99EC74C6-74CC-452F-976A-9E93AB725BA0}';  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // Declaration of Enumerations defined in Type Library  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // Constants for enum Enum1  type  Enum1 = TOleEnum;  const  Enum1\_DUMMY = $00000000;  type  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // Forward declaration of types defined in TypeLibrary  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  ITestAutoLab10 = interface;  ITestAutoLab10Disp = dispinterface;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // Declaration of CoClasses defined in Type Library  // (NOTE: Here we map each CoClass to its Default Interface)  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  TestAutoLab10 = ITestAutoLab10;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // Interface: ITestAutoLab10  // Flags: (4416) Dual OleAutomation Dispatchable  // GUID: {130FB6C1-258C-4D0A-B039-D74B4DFA3FDE}  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  ITestAutoLab10 = interface(IDispatch)  ['{130FB6C1-258C-4D0A-B039-D74B4DFA3FDE}']  procedure AVG(Param1: OleVariant; out Param2: Single); safecall;  procedure Dispersion(Param1: OleVariant; out Param2: Single); safecall;  end;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // DispIntf: ITestAutoLab10Disp  // Flags: (4416) Dual OleAutomation Dispatchable  // GUID: {130FB6C1-258C-4D0A-B039-D74B4DFA3FDE}  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  ITestAutoLab10Disp = dispinterface  ['{130FB6C1-258C-4D0A-B039-D74B4DFA3FDE}']  procedure AVG(Param1: OleVariant; out Param2: Single); dispid 201;  procedure Dispersion(Param1: OleVariant; out Param2: Single); dispid 202;  end;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  // The Class CoTestAutoLab10 provides a Create and CreateRemote method to  // create instances of the default interface ITestAutoLab10 exposed by  // the CoClass TestAutoLab10. The functions are intended to be used by  // clients wishing to automate the CoClass objects exposed by the  // server of this typelibrary.  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  CoTestAutoLab10 = class  class function Create: ITestAutoLab10;  class function CreateRemote(const MachineName: string): ITestAutoLab10;  end;  implementation  uses System.Win.ComObj;  class function CoTestAutoLab10.Create: ITestAutoLab10;  begin  Result := CreateComObject(CLASS\_TestAutoLab10) as ITestAutoLab10;  end;  class function CoTestAutoLab10.CreateRemote(const MachineName: string): ITestAutoLab10;  begin  Result := CreateRemoteComObject(MachineName, CLASS\_TestAutoLab10) as ITestAutoLab10;  end;  end. |

Project2.ridl

|  |
| --- |
| // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //  // WARNING  // -------  // This file is generated by the Type Library importer or Type Libary Editor.  // Barring syntax errors, the Editor will parse modifications made to the file.  // However, when applying changes via the Editor this file will be regenerated  // and comments or formatting changes will be lost.  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //  // File generated on 01.11.2018 10:52:26 (- $Rev: 12980 $, 4049718).  [  uuid(D1D7E903-5B81-4A10-A57D-7F02F5D44324),  version(1.0)  ]  library Project2  {  importlib("stdole2.tlb");  interface ITestAutoLab10;  coclass TestAutoLab10;  [  uuid(5B392A86-FD2B-4BE2-90A9-21ECADA24645)  ]  enum Enum1  {  Enum1\_DUMMY = 0  };  [  uuid(130FB6C1-258C-4D0A-B039-D74B4DFA3FDE),  helpstring("Dispatch interface for TestAutoLab10 Object"),  dual,  oleautomation  ]  interface ITestAutoLab10: IDispatch  {  [id(0x000000C9)]  HRESULT \_stdcall AVG([in] VARIANT Param1, [out] float\* Param2);  [id(0x000000CA)]  HRESULT \_stdcall Dispersion([in] VARIANT Param1, [out] float\* Param2);  };  [  uuid(99EC74C6-74CC-452F-976A-9E93AB725BA0),  helpstring("TestAutoLab10 Object")  ]  coclass TestAutoLab10  {  [default] interface ITestAutoLab10;  };  }; |

Результат программы показан на рисунке 1.

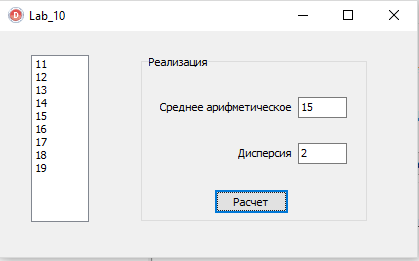


Рисунок 1 – Интерфейс программы.

Контрольные вопросы.

**1. От какого интерфейса унаследован созданный в лабораторной работе интерфейса IТestAutoLab10? Почему полученный интерфейс называется дуальным?**

Если к объекту, имеющему интерфейс IDispatch, добавить свой собственный интерфейс (custom interface), то мы получим двойственный (дуальный) интерфейс. Дуальный интерфейс удобен тем, что предоставляет возможность осуществления как раннего, так и позднего связывания методов COM объекта. Для его реализации объявления методов собственного интерфейса (производного от IUnknown) должны быть идентичны объявлениям методов в диспинтерфейсе.

**2. Перечислите этапы создания внешнего сервера автоматизации.**

Создание внешнего сервера автоматизации в Delphi включает выполнение следующих шагов:

* Создание нового приложения-сервера.
* Создание объекта автоматизации.
* Создание интерфейсов, а также заголовков методов и свойств интерфейсов объекта автоматизации.
* Задание кода методов интерфейса.

Процесс создания сервера автоматизации включает те же действия, что и процесс создания сервера СОМ.

**3. Как создать экземпляр объекта автоматизации в клиентском приложении и получить указатель на его интерфейс?**

При использовании позднего связывания создание экземпляра объекта автоматизации и получение указателя на его интерфейс выполняется с помощью функции

CreateOleObject(const ClassName: string): IDispatch.

**4. Как зарегистрировать форму ActiveForm?**

Данный компонент надо поместить в корневую папку с библиотеками Rad Studio в debug и release версию библиотек в соответствии с разрядностью вашей системы.