Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Институт информационных технологий

Факультет повышения квалификации и переподготовки

Кафедра МПСС

**ОТЧЕТ**

По самостоятельной работе №3

по дисциплине ТКП

Тема: «Работа со свойствами элементов управления КАЛЕНДАРЬ»

Минск 2021

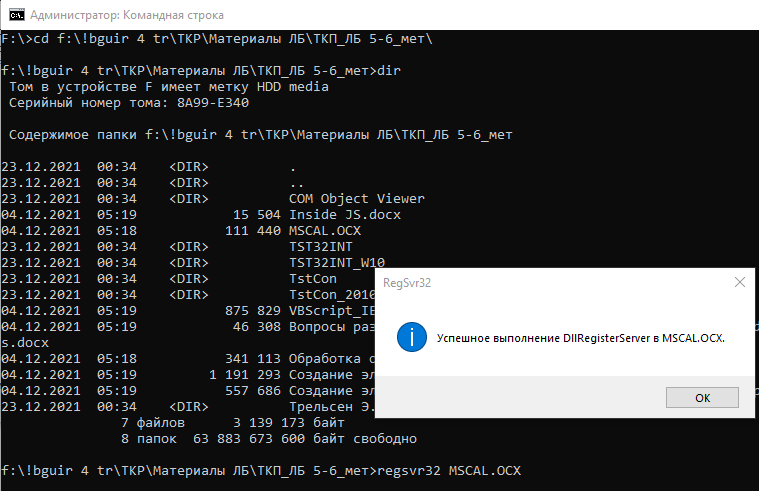
**Цель работы:** Разработать консольное приложение-клиент С++, приложение-клиент на Visual Basic, приложение-клиент на С# для получения (вывода) и установки одного из свойств (по варианту) элемента управления «Календарь». Указание – для доступа к свойству на С++ использовать диспетчерский интерфейс и его методы GetIdsOfName и Invoke.

**Индивидуальное задание вариант 14:**



**Ход работы:**

1. Регистрация календаря в реестре;



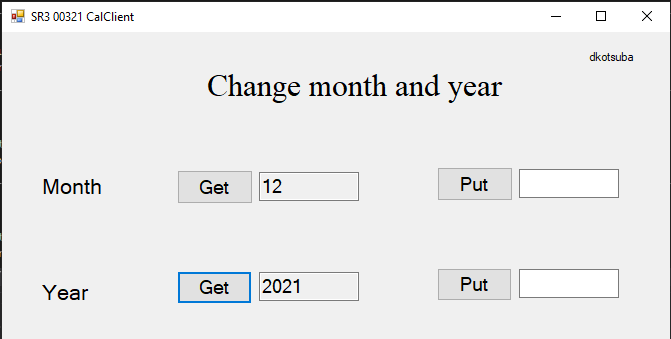
1. Создание клиентского приложения на Visual Basic

Проект **CalClientBasic**

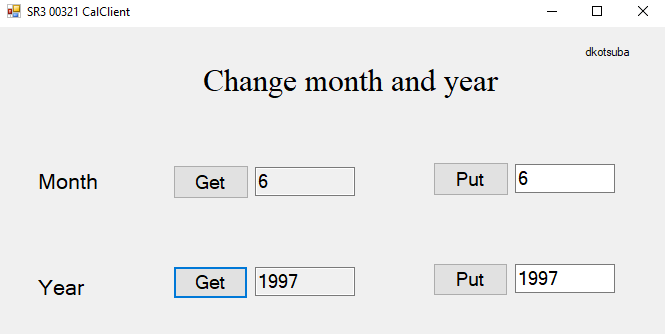
|  |
| --- |
| 'Form1.Designer.vb  <Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()>  Partial Class Form1  Inherits System.Windows.Forms.Form  'Форма переопределяет dispose для очистки списка компонентов.  <System.Diagnostics.DebuggerNonUserCode()>  Protected Overrides Sub Dispose(ByVal disposing As Boolean)  Try  If disposing AndAlso components IsNot Nothing Then  components.Dispose()  End If  Finally  MyBase.Dispose(disposing)  End Try  End Sub  'Является обязательной для конструктора форм Windows Forms  Private components As System.ComponentModel.IContainer  'Примечание: следующая процедура является обязательной для конструктора форм Windows Forms  'Для ее изменения используйте конструктор форм Windows Form.  'Не изменяйте ее в редакторе исходного кода.  <System.Diagnostics.DebuggerStepThrough()>  Private Sub InitializeComponent()  Me.TextBox1 = New System.Windows.Forms.TextBox()  Me.TextBox2 = New System.Windows.Forms.TextBox()  Me.getMonthButton = New System.Windows.Forms.Button()  Me.getYearButton = New System.Windows.Forms.Button()  Me.getMonthTextbox = New System.Windows.Forms.TextBox()  Me.getYearTextbox = New System.Windows.Forms.TextBox()  Me.putYearTextbox = New System.Windows.Forms.TextBox()  Me.putMonthTextBox = New System.Windows.Forms.TextBox()  Me.putYearButton = New System.Windows.Forms.Button()  Me.putMonthButton = New System.Windows.Forms.Button()  Me.Label1 = New System.Windows.Forms.Label()  Me.Label2 = New System.Windows.Forms.Label()  Me.SuspendLayout()  '  'TextBox1  '  Me.TextBox1.BorderStyle = System.Windows.Forms.BorderStyle.None  Me.TextBox1.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.TextBox1.Location = New System.Drawing.Point(40, 142)  Me.TextBox1.Name = "TextBox1"  Me.TextBox1.ReadOnly = True  Me.TextBox1.Size = New System.Drawing.Size(69, 24)  Me.TextBox1.TabIndex = 0  Me.TextBox1.Text = "Month"  '  'TextBox2  '  Me.TextBox2.BorderStyle = System.Windows.Forms.BorderStyle.None  Me.TextBox2.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.TextBox2.Location = New System.Drawing.Point(40, 248)  Me.TextBox2.Name = "TextBox2"  Me.TextBox2.ReadOnly = True  Me.TextBox2.Size = New System.Drawing.Size(69, 24)  Me.TextBox2.TabIndex = 1  Me.TextBox2.Text = "Year"  '  'getMonthButton  '  Me.getMonthButton.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.getMonthButton.Location = New System.Drawing.Point(175, 138)  Me.getMonthButton.Name = "getMonthButton"  Me.getMonthButton.Size = New System.Drawing.Size(76, 34)  Me.getMonthButton.TabIndex = 4  Me.getMonthButton.Text = "Get"  Me.getMonthButton.UseVisualStyleBackColor = True  '  'getYearButton  '  Me.getYearButton.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.getYearButton.Location = New System.Drawing.Point(175, 239)  Me.getYearButton.Name = "getYearButton"  Me.getYearButton.Size = New System.Drawing.Size(75, 33)  Me.getYearButton.TabIndex = 5  Me.getYearButton.Text = "Get"  Me.getYearButton.UseVisualStyleBackColor = True  '  'getMonthTextbox  '  Me.getMonthTextbox.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.getMonthTextbox.Location = New System.Drawing.Point(257, 140)  Me.getMonthTextbox.Name = "getMonthTextbox"  Me.getMonthTextbox.ReadOnly = True  Me.getMonthTextbox.Size = New System.Drawing.Size(100, 29)  Me.getMonthTextbox.TabIndex = 6  '  'getYearTextbox  '  Me.getYearTextbox.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.getYearTextbox.Location = New System.Drawing.Point(257, 240)  Me.getYearTextbox.Name = "getYearTextbox"  Me.getYearTextbox.ReadOnly = True  Me.getYearTextbox.Size = New System.Drawing.Size(100, 29)  Me.getYearTextbox.TabIndex = 7  '  'putYearTextbox  '  Me.putYearTextbox.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.putYearTextbox.Location = New System.Drawing.Point(517, 237)  Me.putYearTextbox.Name = "putYearTextbox"  Me.putYearTextbox.Size = New System.Drawing.Size(100, 29)  Me.putYearTextbox.TabIndex = 11  '  'putMonthTextBox  '  Me.putMonthTextBox.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.putMonthTextBox.Location = New System.Drawing.Point(517, 137)  Me.putMonthTextBox.Name = "putMonthTextBox"  Me.putMonthTextBox.Size = New System.Drawing.Size(100, 29)  Me.putMonthTextBox.TabIndex = 10  '  'putYearButton  '  Me.putYearButton.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.putYearButton.Location = New System.Drawing.Point(435, 236)  Me.putYearButton.Name = "putYearButton"  Me.putYearButton.Size = New System.Drawing.Size(75, 33)  Me.putYearButton.TabIndex = 9  Me.putYearButton.Text = "Put"  Me.putYearButton.UseVisualStyleBackColor = True  '  'putMonthButton  '  Me.putMonthButton.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))  Me.putMonthButton.Location = New System.Drawing.Point(435, 135)  Me.putMonthButton.Name = "putMonthButton"  Me.putMonthButton.Size = New System.Drawing.Size(76, 34)  Me.putMonthButton.TabIndex = 8  Me.putMonthButton.Text = "Put"  Me.putMonthButton.UseVisualStyleBackColor = True  '  'Label1  '  Me.Label1.AutoSize = True  Me.Label1.Font = New System.Drawing.Font("Times New Roman", 24.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))  Me.Label1.Location = New System.Drawing.Point(199, 36)  Me.Label1.Name = "Label1"  Me.Label1.Size = New System.Drawing.Size(310, 36)  Me.Label1.TabIndex = 12  Me.Label1.Text = "Change month and year"  '  'Label2  '  Me.Label2.AutoSize = True  Me.Label2.Location = New System.Drawing.Point(585, 18)  Me.Label2.Name = "Label2"  Me.Label2.Size = New System.Drawing.Size(51, 13)  Me.Label2.TabIndex = 13  Me.Label2.Text = "dkotsuba"  '  'Form1  '  Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)  Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font  Me.ClientSize = New System.Drawing.Size(668, 308)  Me.Controls.Add(Me.Label2)  Me.Controls.Add(Me.Label1)  Me.Controls.Add(Me.putYearTextbox)  Me.Controls.Add(Me.putMonthTextBox)  Me.Controls.Add(Me.putYearButton)  Me.Controls.Add(Me.putMonthButton)  Me.Controls.Add(Me.getYearTextbox)  Me.Controls.Add(Me.getMonthTextbox)  Me.Controls.Add(Me.getYearButton)  Me.Controls.Add(Me.getMonthButton)  Me.Controls.Add(Me.TextBox2)  Me.Controls.Add(Me.TextBox1)  Me.Name = "Form1"  Me.Text = "SR3 00321 CalClient"  Me.ResumeLayout(False)  Me.PerformLayout()  End Sub  Friend WithEvents TextBox1 As TextBox  Friend WithEvents TextBox2 As TextBox  Friend WithEvents getMonthButton As Button  Friend WithEvents getYearButton As Button  Friend WithEvents getMonthTextbox As TextBox  Friend WithEvents getYearTextbox As TextBox  Friend WithEvents putYearTextbox As TextBox  Friend WithEvents putMonthTextBox As TextBox  Friend WithEvents putYearButton As Button  Friend WithEvents putMonthButton As Button  Friend WithEvents Label1 As Label  Friend WithEvents Label2 As Label  End Class |

|  |
| --- |
| 'Form1.vb  Public Class Form1  Dim obj As Object  Public Sub New()  InitializeComponent()  obj = CreateObject("MSCAL.Календарь.7")  End Sub  Private Sub getMonthButton\_Click(sender As Object, e As EventArgs) Handles getMonthButton.Click  getMonthTextbox.Text = obj.Month  End Sub  Private Sub putMonthButton\_Click(sender As Object, e As EventArgs) Handles putMonthButton.Click  obj.Month = putMonthTextBox.Text  End Sub  Private Sub getYearButton\_Click(sender As Object, e As EventArgs) Handles getYearButton.Click  getYearTextbox.Text = obj.Year  End Sub  Private Sub putYearButton\_Click(sender As Object, e As EventArgs) Handles putYearButton.Click  obj.Year = putYearTextbox.Text  End Sub  End Class |

Получим текущий месяц и год



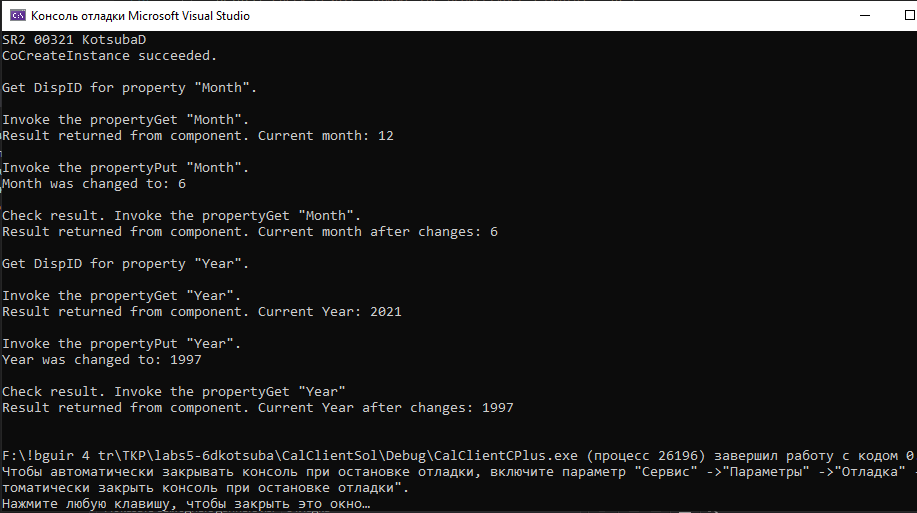
Введем новые данные:



1. Создание клиентского приложения на С++

Проект **CalClientCPlus**

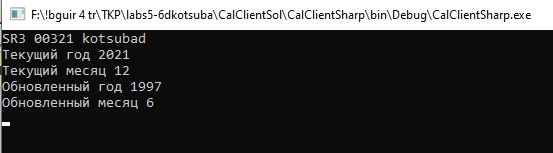
|  |
| --- |
| //CalClientCPlus.cpp  #include <windows.h>  #include <stdio.h>  int main()  {  printf("SR2 00321 KotsubaD \n");  DWORD clsctx;  clsctx = CLSCTX\_INPROC\_SERVER;  HRESULT hr = OleInitialize(NULL);  if (FAILED(hr))  {  printf("Failed to initialize.hr=%X \n", hr);  return 1;  }  //Get the CLSID for the application  wchar\_t progid[] = L"MSCAL.Календарь.7";  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\n");  //Getting the ID for the property Month  printf("Get DispID for property \"Month\".\n\n");  DISPID dispidMonth;  OLECHAR\* name1 = (OLECHAR\*)L"Month";  hr = pIDispatch->GetIDsOfNames(IID\_NULL,  &name1,  1,  GetUserDefaultLCID(),  &dispidMonth);  if (FAILED(hr))  {  printf("Query GetIDsOfNames failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // get month  VARIANT result1;  VariantInit(&result1);  result1.vt = VT\_I2;  DISPPARAMS dispparams1 = { NULL, NULL, 0, 0 };  printf("Invoke the propertyGet \"Month\". \n");  hr = pIDispatch->Invoke(dispidMonth,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_PROPERTYGET,  &dispparams1,  &result1,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  if (result1.vt == VT\_I2)  {  printf("Result returned from component. Current month: %d\n\n", result1.iVal);  }  // put month  VARIANTARG varg2[1];  ::VariantInit(&varg2[0]);  varg2[0].vt = VT\_I2;  varg2[0].intVal = 6;  DISPID dispidNamed2 = DISPID\_PROPERTYPUT;  DISPPARAMS dispparams2;  dispparams2.rgvarg = varg2;  dispparams2.cArgs = 1;  dispparams2.cNamedArgs = 1;  dispparams2.rgdispidNamedArgs = &dispidNamed2;  printf("Invoke the propertyPut \"Month\". \n");  hr = pIDispatch->Invoke(dispidMonth,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_PROPERTYPUT,  &dispparams2,  NULL,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  else // Display log if Invoke() succeeded.  {  printf("Month was changed to: %d\n\n", varg2[0].intVal);  }  printf("Check result. Invoke the propertyGet \"Month\". \n");  hr = pIDispatch->Invoke(dispidMonth,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_PROPERTYGET,  &dispparams1,  &result1,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  if (result1.vt == VT\_I2)  {  printf("Result returned from component. Current month after changes: %d\n\n", result1.iVal);  }  //Getting the ID for the property Year  printf("Get DispID for property \"Year\".\n\n");  DISPID dispidYear;  OLECHAR\* name3 = (OLECHAR\*)L"Year";  hr = pIDispatch->GetIDsOfNames(IID\_NULL,  &name3,  1,  GetUserDefaultLCID(),  &dispidYear);  if (FAILED(hr))  {  printf("Query GetIDsOfNames failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // get Year  VARIANT result3;  VariantInit(&result3);  result3.vt = VT\_I2;  DISPPARAMS dispparams3 = { NULL, NULL, 0, 0 };  printf("Invoke the propertyGet \"Year\". \n");  hr = pIDispatch->Invoke(dispidYear,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_PROPERTYGET,  &dispparams3,  &result3,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  if (result3.vt == VT\_I2)  {  printf("Result returned from component. Current Year: %d\n\n", result3.iVal);  }  // put Year  VARIANTARG varg4[1];  ::VariantInit(&varg4[0]);  varg4[0].vt = VT\_I2;  varg4[0].intVal = 1997;  DISPID dispidNamed4 = DISPID\_PROPERTYPUT;  DISPPARAMS dispparams4;  dispparams4.rgvarg = varg4;  dispparams4.cArgs = 1;  dispparams4.cNamedArgs = 1;  dispparams4.rgdispidNamedArgs = &dispidNamed4;  printf("Invoke the propertyPut \"Year\". \n");  hr = pIDispatch->Invoke(dispidYear,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_PROPERTYPUT,  &dispparams4,  NULL,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  else // Display log if Invoke() succeeded.  {  printf("Year was changed to: %d\n\n", varg4[0].intVal);  }  printf("Check result. Invoke the propertyGet \"Year\" \n");  hr = pIDispatch->Invoke(dispidYear,  IID\_NULL,  GetUserDefaultLCID(),  DISPATCH\_PROPERTYGET,  &dispparams3,  &result3,  NULL,  NULL);  if (FAILED(hr))  {  printf("Invoke call failed.hr=%X \n", hr);  pIDispatch->Release();  return 1;  }  // Display the returned result.  if (result3.vt == VT\_I2)  {  printf("Result returned from component. Current Year after changes: %d\n\n", result3.iVal);  }  } |



1. Создание клиентского приложения на C#

Проект

|  |
| --- |
| //Program.cs  using System;  using MSACAL;  namespace CalClientSharp  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("SR3 00321 kotsubad");  var obj = new Calendar();  Console.WriteLine("Текущий год " + obj.Year);  Console.WriteLine("Текущий месяц " + obj.Month);    obj.Month = 6;  obj.Year = 1997;  Console.WriteLine("Обновленный год " + obj.Year);  Console.WriteLine("Обновленный месяц " + obj.Month);  Console.ReadKey();  }  }  } |



**Выводы:**

В ходе выполнения самостоятельной работы был получен опыт разработки приложений-клиентов на Visual Basic, С++ и C# для получения и установки свойств элемента управления «Календарь».