3 نمره کلاسی ( کسر 5/0 نمره به ازای هر غیبت)

5 نمره پروژه های کلاسی (به شرط تحویل به موقع)

12 نمره پایانی (تئوری و عملی)

.Net یک چارچوب (Frame Work) است. البته برای Windows.

نسخه هایی از آن مثل مونو (Mono) ، برای Linux و Mac وجود دارد.

امکاناتی برای کار با وسایل هوشمند مثل موبایل و PDA (دستیار دیجیتالی شخصی) و WEB با کمک ASP دارد.

شامل کتابخانه های گوناگون حاوی توابع و کلاس ها از جمله توابع کنترل ورودی و خروجی ، چاپ ، شبکه و ... .

این توابع و کلاس ها به گروه ها و فضای نامهای مختلفی دسته بندی شده اند.

شامل CTS ( سیستم نوع داده عمومی ) و CLR ( زبان عمومی مشترک زمان اجرا ) است.

برنامه هنگام کامپایل در Visual Studio، به سطح میانی مایکروسافت (MSIL) که مختص وینوز و C#، نیست ترجمه و سپس توسط کامپایلر JIT (Just-in-Time) ، برای اجرا توسط سیستم عامل، ترجمه میشود.

کد MSIL در فایلهایی به نام اسمبلی (.exe, .dll) ذخیره میشوند.

برای توزیع نرم افزار تولیدشده ، کپی آن کافی است. البته به شرط نصب .Net Frame Work روی سیستم مقصد.

مدیریت حافظه توسط Garbage Collection(GC) بصورت اتومات انجام میشود. تقریبا شبیه Recycle Bin ویندوز.

**انواع داده :**

byte(Byte), short(Int16), int(Int32), long(Int64), float(Single), double(Double), ushort, ulong, uint, string(String) (+, Length, Substring, Replace, IndexOf, String.Compare, … ) , DateTime (d=DateTime.now , System.Globalization.PersianCalendar, p.GetDayOfMonth(d)), bool,Boolean, var

s= "c:\\temp"; // s=@"c:\temp"; "\r" , "\n"

% Operator for integral and floating point.

نام شناسه ها میتوانند فارسی هم باشند.

**تبدیل داده ها:**

f=(float)15/2; i=(int)f; // Casting or Boxing

d=double.Parse(s), i=int.Parse(s), … , Convert

**کلاس Math :** توابع آماده ریاضی مثل

 Abs, Sin, Asin, ..., BigMul, Ceiling, Floor, DivRem, Exp, Log, ...

و ثابت هایی مثل PI ,E

**دستورات :**

if, switch(int, bool, string, char, ~~float~~ ), for, foreach (for Arrays and Collections) ,do while, while, break, continue

foreach (string s in System.IO.Directory.GetDirectories("c:\\"))

 listBox1.Items.Add(s);

هر دستور میتواند در چند خط بدون هیچ علامت خاصی، با پایان ";" قرار بگیرد.

This.Close();

Application.Exit();

**فرم های چندگانه:**

Form2 frm2 = new Form2();

Form3 frm3 = new Form3();

private void Form1\_Load(object sender, EventArgs e)

{

 this.IsMdiContainer = true;

 frm2.MdiParent = this;

 frm3.MdiParent = this;

}

**منوسازی menuStrip و contextMenuStrip و toolStrip و statusStrip:**

private void tileToolStripMenuItem\_Click(object sender, EventArgs e)

{

 LayoutMdi(MdiLayout.TileHorizontal);

}

**لطفا با فرم های آماده About Box و MDI Parent Form و ... کار کنید.**

**آرایه ها :**

int[] ar=new int[10]; //0 to 9

int[] ar={1, 2, 3 }; // new int[]{1, 2, 3 };// new int[3]{1, 2, 3 };

int[] ar=new int[~~5~~]{1, 2, 3 };

~~ar={1, 2, 3 };~~ ar=new int[]{1, 2, 3 };

Array.Sort(ar); i=ar.GetUpperBound(0); Array.Reverse(ar);

int [,] ar= new int[ 2, 3 ];

string[] num = textBox1.Text.Split(','); //textBox1.Text.Split(',',' ');

Array.Resize<int>(ref a, 10);

**آرایه دندانه دار (Jagged Array):**

int[][] jaggedar = new int[2][];

jaggedar[0] = new int[3];

jaggedar[1] = new int[5];

**آرایه کنترلی :**

 **-ایجاد شیء :**

private void Form1\_Load(object sender, EventArgs e)

{

 Button btn = new Button() { Text = "Ok", Parent = this, Top =100, Left = 100 };

 btn.Click += btn\_Click;

}

void btn\_Click(object sender, EventArgs e)

{

 MessageBox.Show("Hello");

}

 **-آرایه کنترلی:**

TextBox txt1, txt2, txt3;

private void Form1\_Load(object sender, EventArgs e)

{

 txt1 = new TextBox() { Left = 20, Width =100, Parent = this, Top = 20 };

 txt2 = new TextBox() { Left = 20, Width = 100, Parent = this, Top = txt1.Top + txt1.Height+20};

 txt3 = new TextBox() { Left = 20, Width = 100, Parent = this, Top = txt2.Top + txt2.Height+20};

 Button[] btns = new Button[4];

 string[] label = new string[] { "+", "-", "/", "\*" };

 for (int i = 0; i < btns.Length; i++)

 {

 btns[i]=new Button(){Top=20+i\*30, Left=txt1.Left+txt1.Width+20, Parent=this,Text=label[i] };

 btns[i].Click += btns\_Click;

 }

}

void btns\_Click(object sender, EventArgs e)

{

 float n1,n2;

 n1=float.Parse(txt1.Text);

 n2=float.Parse(txt2.Text);

 switch (((Button)sender).Text )

 {

 case "+":

 txt3.Text = (n1 + n2).ToString();

 break;

 case "-":

 txt3.Text = (n1 - n2).ToString();

 break;

 case "\*":

 txt3.Text = (n1 \* n2).ToString();

 break;

 case "/":

 txt3.Text = (n1 / n2).ToString();

 break;

 }

}

 **-آرایه ای از اشیاء:**

object[] obj = new Object[3];

Button b = new Button() { Text = "B1", Top = 10, Left = 10 };

TextBox t = new TextBox() { Text = "T1", Top = 100, Left = 10 };

int x = 10;

obj.SetValue(b, 0); obj.SetValue(t, 1); obj.SetValue(x, 2);

for (int i = 0; i < obj.Length; i++)

 if (obj[i].GetType() == typeof(Button))

 ((Button)obj[i]).Parent = this;

 else

 if (obj[i].GetType() == typeof(TextBox))

 ((TextBox)obj[i]).Parent = this;

 else

 MessageBox.Show(((int)obj[i]).ToString());

**داده های شمارشی (Enum):**

enum WeekDays

{

 sat, sun, mon, tue, wed, thu, fri

}

WeekDays w;

double s = 100000;

for (w = WeekDays.sat; w <= WeekDays.thu; w++)

{

 s += 10000;

 listBox1.Items.Add(w.ToString());

}

MessageBox.Show(s.ToString());

**ثابت ها (Constants):**

const float pi = 3.14F;

**متدها یا توابع :**

پارامتر آرایه ای باید بدون ذکر اندازه باشد(در توابع). پارامتر آرایه ای باید بدون ذکر اندازه باشد(در توابع).

long sum(int[] a)

{ . . . }

Recursive

///

ref (نیاز به مقدار دهی اولیه دارد), out(نیاز به مقدار دهی اولیه ندارد)

Over Loading

params int [] x

 پشتیبانی از آرگومان پیش فرض در نسخه 2010 به بعد (Optional parameters).

**ساختارها (Structures):**

میتوانند اعضائی از نوع متد و Property داشته باشند.

 Property میتوانند علاوه بر کنترل داده های دریافتی، فقط خواندنی یا فقط نوشتنی باشند.

struct Student

{

 const float e = 2.718F;

 private String \_n;

 private float \_a;

 //public string n { set { \_n = value; } }

 public string getn()

 {

 return \_n;

 }

 public string n { get { return \_n; } }

 public float a

 {

 get { return \_a; }

 set

 {

 if (value < 0 || value > 20)

 MessageBox.Show("Error");

 //throw new Exception("نمره باید بین 0 تا 20 باشد");

 else

 \_a = value;

 }

 }

 public Student(string x, float y)

 {

 \_n = x; \_a = y;

 }

}

**کلاس ها (Class) :**

**Encapsulation, Reusibility (یک کلاس برای چند پروژه )**

**Constructor (Public Class\_Name) , :base(...)**

**Destructor ( ~ Tilde sign + Class\_Name)**

class Circle

{

 private Double \_Radius;

 public Double Radius

 {

 get { return \_Radius; }

 set

 {

 if (value < 0)

 //throw new Exception("Radius Is Negative");

 \_Radius = 0;

 else

 \_Radius = value;

 }

 }

 public Double Area()

 {

 return (Math.PI \* Radius \* Radius);

 }

 public Double Pyramid()

 {

 return (2 \* Math.PI \* Radius);

 }

 public Circle(Double Radius) // سازنده

 {

 this.Radius = Radius;

 }

 public Circle()

 {

 Radius = 0;

 }

 ~Circle()

 {

 System.Windows.Forms.MessageBox.Show("خاتمه");

 }

}

**Inheritance ( Derived and Base Class), Polymorphism**

class Cylinder : Circle //Derived

{

 public double Height;

 public double Area()

 {

 return (Pyramid() \* Height + 2 \* base.Area());

 }

 public double Volume()

 {

 return Height \* base.Area();

 }

 public Cylinder(double Height)

 {

 Radius = 0;

 this.Height = Height;

 }

 public Cylinder(double Radius,double Height)

 {

 this.Radius = Radius;

 this.Height = Height;

 }

 //public Cylinder(double Radius,double Height):base(Radius)

 //{

 // this.Height = Height;

 //}

 public Cylinder()

 {

 Radius = Height = 0;

 }

}

کلاس ها میتوانند متداخل تعریف شوند. ساختارها هم میتوانند.

دسترسی به اعضای کلاس پایه با base و کلاس جاری با this امکان پذیر است.

ساختار ها نوع داده مقداری (Value) ولی کلاس ها نوع داده ارجاعی (Reference)هستند. ساختارها در حافظه Stack ولی کلاس ها در حافظه Heap ذخیره میشوند. ساختارها وراثت ندارند و معمولا برای داده های کوچک استفاده میشوند و Destructor ندارند.

عضو static، فقط توسط نام کلاس (Math.Sin) قابل دسترس است نه شیئ . میتوانند توسط همه اشیاء، استفاده شود. مثلا حداقل طول رمز کاربران.

عضو protected فقط در بدنه کلاس مشتق شده قابل استفاده است نه در شیئ. ولی عضو internal در هر دو، اما فقط در همین اسمبلی.

از کلاس abstract (انتزاعی)، نمیتوان object گرفت (Instance کرد). متدهای تعریف شده بصورت abstract در آن، باید در کلاس های مشتق شده، override شوند.

از کلاس sealed (مهر و موم) بر عکس abstract (انتزاعی یا مجرد)، اجازه Derive کردن کلاس را نداریم ولی اجازه Instance کردن Object را داریم.

<T> in front of class name -- (int)((object)x)

**نحوه Over Load کردن عملگرها:**

مثلا کلاس اعداد موهومی و عملگر + یا کلاس لیست مشتق شده از ListBox با امکان جمع دو لسیت یا کلاس ساعت با عملگر جمع و تفریق

class Mobham

{

 public double n, i;

 public static Mobham operator +(Mobham m1, Mobham m2)

 {

 Mobham m=new Mobham();

 m.n = m1.n + m2.n;

 m.i = m1.i + m2.i;

 return m;

 }

 public override string ToString()

 {

 return n.ToString() + "\*" + i.ToString() + "i";

 }

}

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

class My\_ListBox : ListBox

{

 public static My\_ListBox operator +(My\_ListBox l1, My\_ListBox l2)

 {

 My\_ListBox l = new My\_ListBox();

 l.Items.AddRange(l1.Items);

 l.Items.AddRange(l2.Items);

 return l;

 }

}

**Interfaces :**

یک قرارداد برای تعیین نوع رفتار کلاس. فقط متد ها و خاصیت ها (Properties)، بدون پیاده سازی (Implementation) .

interface I\_2DShapes

{

 double Width { get; set; }

 double Height { get; set; }

 double Area();

}

public class MyCircle : I\_2DShapes

{

 private double \_Width, \_Height;

 public double Width{

get { return \_Width; }

set { \_Width = value; }

}

 public double Height{

get{ return \_Height; }

set { \_Height = value; }

}

 public double Radius {

get { return Height / 2.0; }

set { Height = Width = 2 \* value; }

}

 public double Area()

 {

 return (Math.PI \* \_Width / 2.0 \* \_Width / 2.0);

 }

}

**فضای نام (Namespace) :**

هدف : ساده تر کردن استفاده از کلاس و استفاده از کلاس های هم نام. مثلا شیر در فضای نام حیوانات یا لبنیات یا تأسیسات.

بیش از 3500 کلاس در چندین گروه (Namespace) در .Net وجود دارد. هر فضای نام در درون خود فضاهای نام دیگری دارد. مثلا System شامل System.Data و System.Windows و System.Xml و ... است.

استفاده از Object Browser .

کلاس ها میتوانند partial تعریف شوند. درچند قسمت از یک فضای نام (در یک فایل) یا در فایل های متعدد البته با فضای نام های یکسان.

**لیست پیوندی یا آرایه ای (Array List):**

داده هایی با انواع مختلف هم میتواند داشته باشد.

using System.Collections; لازم است.

ArrayList ar = new ArrayList();

for (int i = 0; i < 10; i++)

 ar.Add(i);

Button b = new Button();

b.Parent = this;

TextBox t = new TextBox();

t.Parent = this;

ar.Add(b); ar.Add(t);

ListBox lst = new ListBox();

lst.Items.Add(21);

lst.Items.Add(22);

ar.AddRange(lst.Items);

//for (int j = 0; j < ar.Count; j++)

// if (ar[j].GetType().Equals(typeof(int)))

// listBox1.Items.Add(ar[j]);

foreach (object x in ar)

 if (x.GetType().Equals(typeof(int)))

 listBox1.Items.Add(x);

**جدول قابل جستجوی Hash Table :**

Hashtable ht = new Hashtable();

ht.Add("1", "One");

MessageBox.Show(ht["1"].ToString()); // One

Hashtable h = new Hashtable();

//h.Add("Pencil", "Medad");

ListBox lst;

lst = new ListBox();

lst.Items.Add("Ketab");

lst.Items.Add("Ketabcheh");

h.Add("Book", lst);

lst = new ListBox();

lst.Items.Add("Khodkar");

lst.Items.Add("Ghalam");

h.Add("Pen", lst);

string itm = "Book";

if (h.Contains(itm)) //ContainsValue , ContainsKey

 for (int i = 0; i < ((ListBox)h[itm]).Items.Count; i++)

 MessageBox.Show(((ListBox)h[itm]).Items[i].ToString());

**فایل ها:**

StreamWriter, StreamReader, FileStream

private void button1\_Click(object sender, EventArgs e)

{

 StreamWriter sw = new StreamWriter(@"e:\test.txt");

 //sw.Write(richTextBox1.Text);

 foreach (string line in richTextBox1.Lines )

 {

 sw.WriteLine(line);

 }

 sw.Close();

 //richTextBox1.SaveFile(@"e:\test.txt", RichTextBoxStreamType.RichText);

}

private void button2\_Click(object sender, EventArgs e)

{

 StreamReader sr = new StreamReader(@"e:\test.txt");

 while(!sr.EndOfStream)

 richTextBox1.Text += sr.ReadLine() + "\n";

 sr.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

 FileStream fs = new FileStream(@"e:\test.dat", FileMode.OpenOrCreate , FileAccess.Write);

 for (byte i = 0; i < 100; i++)

 {

 fs.WriteByte(i);

 }

 fs.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

 FileStream fs = new FileStream(@"e:\test.dat",FileMode.Open,FileAccess.Read,FileShare.None);

 byte[] ar=new byte[fs.Length];

 fs.Read(ar, 0,(int) fs.Length);

 foreach (byte b in ar)

 {

 listBox1.Items.Add(b);

 }

 fs.Close();

}

**کادرهای محاوره ای :**

Message Box, Open File Dialog, Save File Dialog, Font Dialog,

Color Dialog, Folder Browser Dialog, Print Dialog

if (x < 0)

 if (MessageBox.Show("Negative Number", "Error", MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button2) == DialogResult.Yes)

 textBox1.Text = Math.Sqrt(x).ToString();

private void OpenBTN\_Click(object sender, EventArgs e)

{

 System.IO.StreamReader sr;

 openFileDialog1.Filter = "Text Files|\*.txt|Doc Files|\*.doc";

 if (openFileDialog1.ShowDialog() == DialogResult.OK)

 {

 sr = new System.IO.StreamReader(openFileDialog1.FileName);

 richTextBox1.Text = sr.ReadToEnd();

 sr.Close();

 }

}

private void FontBTN\_Click(object sender, EventArgs e)

{

 FontDialog fd = new FontDialog();

 fd.Font = richTextBox1.SelectionFont;

 fd.ShowColor = true;

 if (fd.ShowDialog() == DialogResult.OK)

 {

 richTextBox1.SelectionFont = fd.Font;

 richTextBox1.SelectionColor = fd.Color;

 }

}

int l;

private void PrintBTN\_Click(object sender, EventArgs e)

{

 printDialog1.Document = printDocument1;

 if (printDialog1.ShowDialog() == DialogResult.OK)

 {

 printDocument1.PrinterSettings = printDialog1.PrinterSettings;

 l = 0;

 printDocument1.Print();

 }

}

private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

 e.Graphics.DrawRectangle(Pens.Blue, e.MarginBounds);

 Font fnt = richTextBox1.Font;

 float lm = e.MarginBounds.Left, tm = e.MarginBounds.Top;

 while (l < richTextBox1.Lines.Length)

 {

 e.Graphics.DrawString(richTextBox1.Lines[l], fnt, Brushes.Black, new PointF(lm, tm));

 l++;

 tm += fnt.GetHeight();

 if (tm > e.MarginBounds.Bottom - fnt.GetHeight()) break;

 }

 if (tm > e.MarginBounds.Bottom - fnt.GetHeight())

 {

 tm = e.MarginBounds.Top; e.HasMorePages = true;

 }

 else e.HasMorePages = false;

}

میتوان از e.Graphics.MeasureString(txt, fnt).Width برای بدست آوردن عرض متن txt با فونت fnt استفاده کرد.

**گرافیک (Graphics) :**

Graphics G = this.CreateGraphics();

G.DrawLine(Pens.Blue, 0, 0, 100, 100);

G.DrawEllipse(Pens.Red, 0, 0, 200, 200);

G.DrawPie(Pens.Red, new Rectangle(0, 0, 100, 200), 0, 90);

G.DrawImage(Image.FromFile(@"C:\Users\Public\Pictures\Sample Pictures\Penguins.jpg"), 100, 100, 200, 100);

SolidBrush b = new SolidBrush(Color.Green);

//TextureBrush b = new TextureBrush(Image.FromFile(@"C:\Users\Public\Pictures\Sample Pictures\Penguins.jpg"));

//HatchBrush b = new HatchBrush(HatchStyle.LightHorizontal, Color.Red, Color.Blue);

//LinearGradientBrush b = new LinearGradientBrush(new Point(0, 0), new Point(30, 30), Color.Red, Color.Blue);

G.DrawString("Hello", new Font("Times New Romans", 100), b, 100, 100);

G.FillEllipse(b, new Rectangle(200, 200, 100, 100));

//printForm1.Print();

**کنترل استثناء ها (Exceptions):**

int x = 10, y;

try

{

 y = Int32.Parse(textBox1.Text);

 MessageBox.Show((x / y).ToString());

 return;

}

catch (DivideByZeroException)

{

 MessageBox.Show("تقسیم بر صفر");

}

catch (FormatException ex)

{

 MessageBox.Show(ex.Message); //throw new Exception("خطا");

}

finally

{

 MessageBox.Show("Test");

مثلا در یک حلقه و اجرای دستور continue قبل از رسیدن به //finally

}

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

private void button1\_Click(object sender, EventArgs e)

{

 System.IO.Stream fn = System.IO.File.Open(@"C:\a\file.dat", System.IO.FileMode.Create);

 System.IO.BinaryWriter bw = new System.IO.BinaryWriter(fn);

 for (int i = 0; i < 10; i++)

 {

 bw.Write(i);

 }

 bw.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

 System.IO.Stream fn = System.IO.File.Open(@"C:\a\file.dat", System.IO.FileMode.Open);

 System.IO.BinaryReader br = new System.IO.BinaryReader(fn);

 int i = 0;

 while (true)

 {

 try

 {

 i = br.ReadInt32();

 listBox1.Items.Add(i);

 }

 catch (Exception)

 {

 break;

 }

 }

 br.Close();

 //try

 //{

 // while (true)

 // listBox1.Items.Add(br.ReadInt32());

 //}

 //catch

 //{

 // throw new Exception("فایل به پایان رسیده");

 //}

 //finally

 //{

 // br.Close();

 //}

}

**کار با پایگاه داده (Data Base) :**

روش های ویزاردی توصیه نمی گردد.

بسته به نوع پایگاه داده

 using System.Data.SqlClient

یا using System.Data.OleDb

یا using System.Data.Odbc اضافه شود.

//OdbcConnection con;

//OleDbConnection con;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

DataGridView dgv;

CurrencyManager CM;

DataView DV;

Properties.Settings S = new WindowsFormsApplication4.Properties.Settings();

private void Form1\_Load(object sender, EventArgs e)

{

 //con = new SqlConnection("server=site9-ostad; database=Test;integrated security=true;");

 //con = new SqlConnection("Data Source=site9-ostad; Initial Catalog=Test;integrated security=true;");

 //con = new SqlConnection("Data Source=site9-ostad; Initial Catalog=Test;user id=ali; password=123");

 con = new SqlConnection(S["ConStr"].ToString());

 try

 {

 con.Open();

 da = new SqlDataAdapter("Select \* From St\_Info", con);

 ds = new DataSet();

 da.Fill(ds,"T1");

 dgv=new DataGridView() {Parent=this,Dock=DockStyle.Fill };

 dgv.DataSource = ds;

 dgv.DataMember = "T1";

 con.Close();

 dgv.SelectionChanged += new EventHandler(dgv\_SelectionChanged);

 }

 catch (Exception)

 {

 MessageBox.Show("Error In Openinig Database...");

 }

}

void dgv\_SelectionChanged(object sender, EventArgs e)

{

 try

 {

 if (dgv.RowCount > 1)

 {

 textBox1.Text = dgv.SelectedRows[0].Cells[0].Value.ToString();

 textBox2.Text = dgv.SelectedRows[0].Cells[1].Value.ToString();

 textBox3.Text = dgv.SelectedRows[0].Cells[2].Value.ToString();

 }

 }

 catch (Exception)

 {

 }

}

private void dgvRefresh()

{

 ds.Tables["T1"].Clear();

 con.Open();

 da = new SqlDataAdapter("Select \* From St\_Info", con);

 da.Fill(ds, "T1");

 con.Close();

}

private void btnAdd\_Click(object sender, EventArgs e)

{

 con.Open();

 //SqlCommand AddCMD = new SqlCommand("Insert Into St\_Info (St\_ID,St\_Name,St\_Avg) Values (" +

 // long.Parse(textBox1.Text) + ",'" + textBox2.Text + "'," + float.Parse(textBox3.Text) + ");", con);

 SqlCommand AddCMD = new SqlCommand("Insert Into St\_Info (St\_ID,St\_Name,St\_Avg) Values (@i,@n,@a);", con);

 AddCMD.Parameters.AddWithValue("@i", Convert.ToInt64(textBox1.Text));

 AddCMD.Parameters.AddWithValue("@n", textBox2.Text);

 AddCMD.Parameters.AddWithValue("@a", Convert.ToSingle(textBox3.Text));

 AddCMD.ExecuteNonQuery();

 con.Close();

 dgvRefresh();

}

private void btnDel\_Click(object sender, EventArgs e)

{

 con.Open();

 da = new SqlDataAdapter();

 da.DeleteCommand = new SqlCommand("Delete St\_Info Where St\_ID=" + long.Parse(textBox1.Text), con);

 da.DeleteCommand.ExecuteNonQuery();

 con.Close();

 dgvRefresh();

}

private void btnUpdate\_Click(object sender, EventArgs e)

{

 con.Open();

 SqlCommand UpdateCMD = new SqlCommand("Update St\_Info set St\_Name='" + textBox2.Text

 + "',St\_Avg=" + float.Parse(textBox3.Text) + " Where St\_ID=" + long.Parse(textBox1.Text) , con);

 UpdateCMD.ExecuteNonQuery();

 con.Close();

 dgvRefresh();

}

**Class Library & Windows Forms Control Library (DLLs) :**

محافظت از دست کاربر – DLL مستقل از زبان – عدم نیاز به کامپایل مجدد.

در Class Library : قبل از کلاس مورد نظر [ToolboxBitmap(@”bitmap file name”)] برای آیکن Toolbox

**Thread یا نخ :**

نحوه استفاده ازApplication.DoEvents()

using System.Threading; لازم است.

int i = 0; Thread t;

private void button1\_Click(object sender, EventArgs e)

{

 Control.CheckForIllegalCrossThreadCalls = false;

// اگر با اشیاء (کنترل ها) کار می کنیم.

 t = new Thread(new ThreadStart(test));

 i = 0;

 t.Start();

}

public void test()

{

 while (i < 1000000)

 {

 i++;

 textBox1.Text = i.ToString();

 }

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

 if (t!=null && t.ThreadState==ThreadState.Running) t.Abort();

}

private void button2\_Click(object sender, EventArgs e)

{

 if (t.ThreadState == ThreadState.Suspended) t.Resume();

else t.Suspend();

}

private void button3\_Click(object sender, EventArgs e)

{

 if (t.ThreadState==ThreadState.Running) t.Abort();

}

در Task Manager قابل رؤیتند.

Thread.Sleep(100); // Suspends the current thread for a specified time.

**Garbage Collection (جمع آوری زباله ها) :**

مسئول از بین بردن اشیاء غیر قابل استفاده.

طراحان C# به دلایل زیر وظیفه از بین بردن اشیاء را به ما نداده اند:

1)فراموشی ما

2)سعی در از بین بردن اشیاء در حال استفاده

3)سعی مجدد در از بین بردن اشیاء

System.GC.Collect(); // از بین بردن اشیاء غیر قابل استفاده

**عامل ها (Delegates) :** اشاره گر به یک تابع .

public delegate double func(double x);

private void button1\_Click(object sender, EventArgs e)

{

 func f = new func(Math.Sin); // func f = Math.Sin;

 MessageBox.Show(f(30 \* Math.PI / 180).ToString());

}

+++++++++++++++++++++++++++++++

static double CosD(double d)

{

 double r = d \* Math.PI / 180;

 return (Math.Cos(r));

}

public delegate double func(double x);

private void button1\_Click(object sender, EventArgs e)

{

 func f = CosD;

 MessageBox.Show(f(60).ToString());

}

+++++++++++++++++++++++++++++++

public delegate double func(double x);

private void button1\_Click(object sender, EventArgs e)

{

 func[] f = new func[4];

 f[0] = Math.Sin;

 f[1] = Math.Cos;

 f[2] = Math.Tan;

 f[3] = CoTan;

 for (int i = 0; i < 4; i++)

 MessageBox.Show(f[i](30 \* Math.PI / 180).ToString());

}

**عامل های چند طرحی :**

static void test1(ref int x)

{

 x++;

}

static void test2(ref int x)

{

 MessageBox.Show(x.ToString());

}

public delegate void Multi\_Cast(ref int x);

private void button1\_Click(object sender, EventArgs e)

{

 int a = 5;

 Multi\_Cast f = test1;

 f += test2;

 //f -= test1;

 f(ref a);

}

++++++++++++++++++++++++++++++++++

func f = delegate(double x) //Function Without Name

{

 double r = x \* Math.PI / 180;

 return (Math.Cos(r));

};

MessageBox.Show(f(60).ToString());

**عملگر Lambda :**

func f = (double x) =>

{

 double r = x \* Math.PI / 180;

 return (Math.Cos(r));

};

MessageBox.Show(f(60).ToString());

+++++++++++++++++++++++++++++++++++

func f = (double x) => Math.Cos(x \* Math.PI / 180);

+++++++++++++++++++++++++++++++++++

func f = x => Math.Cos(x \* Math.PI / 180);

+++++++++++++++++++++++++++++++++++

func f = x => x \* x;

MessageBox.Show(f(5).ToString());

**Extention ها :** می توانیم به کلاسهای موجود مثل TextBox متدهائی اضافه کنیم. در داخل یک کلاس (public static) با هر نامی، تعریف مربوطه را به صورت زیر قرار می دهیم.

public partial class Form1 : Form

{

 public Form1()

 {

 InitializeComponent();

 }

 private void button1\_Click(object sender, EventArgs e)

 {

 textBox1.Tag = "Ali";

 textBox2.Text = textBox1.GetTag();

 MessageBox.Show(textBox2.ToString(2));

 MessageBox.Show(button1.ToString(3));

 }

}

public static class My\_Extensions

{

 public static string GetTag(this Object Obj)

 {

 return ((TextBox)Obj).Tag.ToString();

 }

 public static string ToString(this Object Obj, int x)

 {

 if (Obj is TextBox)

 return ((TextBox)Obj).Text.Substring(0, x);

 else

 if (Obj is Button)

 return ((Button)Obj).Text.Substring(0, x);

 else return null ;

 }

}

**Indexer ها :** تقریبا حالت Default Property

public class My\_Indexers

{

 public int x;

 public string y;

 public string GetMembers()

 {

 return y + "\_" + x.ToString();

 }

 public object this[int i]

 {

 get

 {

 object Obj = null;

 switch (i)

 {

 case 0:

 Obj = x;

 break;

 case 1:

 Obj = y;

 break;

 }

 return Obj;

 }

 set

 {

 switch (i)

 {

 case 0:

 x = (int)value;

 break;

 case 1:

 y = (string)value;

 break;

 }

 }

 }

}

private void button1\_Click(object sender, EventArgs e)

{

 My\_Indexers Ind = new My\_Indexers();

 Ind[0] = 10;

 Ind[1] = "Ali";

 MessageBox.Show(Ind.GetMembers());

 MessageBox.Show(Ind[1].ToString() + "\_" + Ind[0].ToString());

}