

به نام خدا

جزوه برنامه سازی پیشرفته ۲

مدرس : قاسم زاده

زبان C# یکی از زبان های شی گرا میباشد که مانند VB دارای اشیاء , کلاس , رویداد , مقدار میباشد. از طرفی همانند هر زبان برنامه نویسی دارای انواع رویداد است:

نوع در vb.net	اندازه	نوع
Byte	1	Byte
Char	1	Char
Boolean	1	Bool
Sbyte	1	Sbyte
Int16	2	Short
Int32	2	Int
Single	4	Float
Double	8	Double
Int64	8	Long

تعریف متغیرها: برای تعریف متغیرها از قوانینی که در باقی زبان های برنامه نویسی وجود دارد استفاده میکند , که به شکل زیر تعریف میشوند:

; نام متغیر نوع داده

int x ;

مثال :

نکته : برای مقدار دهی به متغیر ها از " = " استفاده میشود.

نکته : برای تعریف ثابت ها در C# از دستورات #define, Const استفاده میشود.

عملگرها در زبان C# عبارتند از:

۱. عملگرهای محاسباتی .
۲. عملگرهای مقایسه ای .
۳. عملگرهای منطقی .
۴. عملگرهای بیتی .
۵. عملگرهای ترکیبی .

۱. عملگرهای محاسباتی عبارتند از: +, -, *, /, %, ++, -- .

۲. عملگرهای مقایسه ای عبارتند از: <, >, <=, >=, ==, != .

۳. عملگرهای منطقی عبارتند از: &&, ||, ! .

۴. عملگرهای بیتی عبارتند از: &, ^, |, <<, >> .

۵. عملگرهای ترکیبی عبارتند از: این عملگرها از ترکیب دستور جایگزینی و عملگرهای

محاسباتی استفاده میشوند که عبارتند از: *=, /=, +=, -=, %=. .

عملگر؟ از این عملگر در زمانی که عبارت شرطی ما درست یا نادرست باشد , یکی از مقادیر

داده شده در متغیر منظور میشود.

مقدار ۲ : مقدار ۱ ؟ شرط = متغیر

y=3

مثال:

X=y>10?100:200;

در زبان C# ما میتوانیم نوع های مختلف داده را به یکدیگر تبدیل کنیم.

به طور مثال: int به float یا float به bool را انجام دهیم.

برای تبدیل از شی convert استفاده میشود که این شی دارای متدهایی از قبیل :
toboolean , tofloat , tochar و ...

برای نوشتن برنامه در محیط C# فعلا به دو شکل انجام میشود:

۱. استفاده از Console.

۲. استفاده از محیط شی گرا یا application window.

۱. استفاده از console: در این محیط برای نوشتن برنامه باید از یکسری متد استفاده کرد.

مانند:

متد read: از این متد برای خواندن کنسول استفاده میشود.

متد readline: این متد یک خط را خوانده و به خط بعدی میرود.

متد write: از این متد برای نوشتن در کنسول استفاده میشود.

متد writeline: این متد یک خط مینویسد و به خط بعد میرود.

"\n" , "\r": برای رفتن به یک خط پایین تر استفاده میشود.

نوشتن برنامه در محیط console:

```
int x,s;
console.write ("enter x:");
x=convert.ToInt32(console.readline());
s=x*5;
console.write("s=");
console.writeline(s);
console.readkey();
```

دستور شرطی if: برای بررسی شرط در C# به کار میرود.

★ برنامه ای بنویسید که یک عدد صحیح بخونه و اگر زوج بود "even" و اگر فرد بود "odd" را چاپ کند؟

```
Int n ;
N=convert.ToInt32(console.ReadLine());
If (n%2==0)
{
Console.WriteLine("even");
Else
Console.WriteLine("odd");
Console.ReadKey();
```

حلقه for: یکی از حلقه های معین در زبان C# میباشد.

که شکل کلی آن بدین صورت است:

(گام حلقه ; شرط ; مقدار اولیه)for

; دستور

★ با استفاده از حلقه for برنامه ای بنویسید که اعداد فرد را چاپ کند؟

```
Int i;
For(i=1 ; i<20 ; i+=2)
{
Console.WriteLine(i);
Console.ReadKey();
```

★ برنامه ای بنویسید که یک عدد صحیح دریافت و مقسوم علیه های آن را چاپ کند؟

```
Int i,n;
N=convert.ToInt32(console.ReadLine());
For (i=1 ;i<=n ; i++)
```

```
{
If (n%i==0)
Console.WriteLine(i);
Console.ReadKey();
```

حلقه های متداخل for : با چندین حلقه for میتوان حلقه های متداخل ایجاد کرد.

مثال:

```
For (i=1 ;i<=5 ; i++)
{
For (j=1 ; j<=10 ; j++)
Lable 1.text=lable1.text+j.toString();
Lable1.text=lable1.text+"\n";
}
```

★ برنامه ای بنویسید که خروجی زیر را در ورودی چاپ کند؟

```
1
2 1 2
3 2 1 2 3
4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
```

```
Intl,j,k,l;
For (i=1 ;i<=5 ; i++)
{
For (k=1 ; k<=5-i ; k++)
Lable1.text+=" ";
For (j=i ; j>=1 ; j--)
```

```
Lable1.text+=j.toString();
For (l=2 ; l<=l ; l++)
Lable1.text+=l.toString();
Lable1.text+="\n";
```

★ برنامه ای بنویسید که خروجی زیر را در ورودی چاپ کند؟

```
1
2 3
4 5 6
7 8 9 10
Intl,j,k;
For(i=1 ; i<=4 ; i++)
{
For(j=1 ; j<=l ; j++)
{
Lable1.text+=k.toString();
K++;
}
}
Lable1.text+="\n";
```

دستور شرطی switch : از این دستور برای بررسی چندین حالت شرط به کار میرود.

شکل کلی این دستور بدین صورت است:

switch(عبارت)

```
{
case مقدار ۱ ;
Break;
دستور ۱ ;
case مقدار ۲ ;
دستور ۲ ;
Break;
.
.
.
{
Default:
;
دستور آخر
}
```

★ برنامه ای بنویسید که شماره فصل سال را دریافت کند و سپس نام فصل را چاپ کند؟

```
Int n;
N=convert.toint32(textbox1.text);
```


Switch(n)

Case1:

Lable1.text="spring";

Break;

Case2:

Lable1.text="summer";

Break;

Case3;

Lable1.text="outem";

Break;

Case4:

Lable1.text="winter";

Break;

Default:

Lable1.text="out of rang";

حلقه شرطی while : یکی دیگر از حلقه های شرطی میباشد که در صورت برقراری شرط حلقه ادامه میابد.

while (شرط)

شکل کلی این حلقه بدین صورت است:

{

دستورات ;

}

کنترل listbox : یکی از کنترل هایی است که میتوان در آن مقادیر را لیست یا چاپ نمود.

شکل کلی این کنترل بدین صورت است:

listbox1.items.add(رشته) ;

★ برنامه ای بنویسید که اعداد فرد کمتر از ۲۵ را در listBox نشان دهد؟

```
Inti=1;
While(i<20)
{
Listbox1.items.add(i.toString());
i+=2;
}
```

★ برنامه ای بنویسید که یک عدد بگیرد و مقلوب آن را چاپ کند؟

```
Intn,r,s=
0;
N=convert.toint32(textbox1.text);
While(n!=0)
{
R=n%10;
S=s*10+r;
N=n/10;
}
Textbox1.text=s.toString();
```

★ برنامه ای بنویسید که یک عدد صحیح دریافت کرده و آن را به مبنای ۲ ببرد؟

```
Intn,r,s=0,i=1;
N=convert.toint32(textbox1.text);
While (n!=0)
{
R=n%2;
```

```

S=s+r*I;
N=n/2;
I*=10;
}
Textbox1.text=s.toString();

```

کنترل listbox دارای خواصی است که عبارتند از:

۱. items: این خاصیت تعیین میکند که کنترل ظاهر شده چه متدهایی را فراخوانی میکند.
۲. multicolmn: تبیین میکند که آیا کنترل میتواند چند ستون داشته باشد یا خیر.
۳. sorted: تعیین می کند ایا گزینه های کنترل listbox به صورت مرتب شده ظاهر شوند یا خیر.

متدهای این کنترل عبارتند از:

۱. متد add: برای اضافه کردن گزینه ای به انتهای listbox به کار می رود.
۲. متد insert: این متد برای اضافه کردن یک گزینه در مکان خاصی از listbox به کار می رود.

۳. متد count: این متد تعداد گزینه های listbox را تعیین می کند.

۴. متد remove: این متد برای حذف گزینه از listbox به کار می رود.

۵. متد removeat: این متد برای حذف گزینه براساس شمار index به کار می رود.

ارایه ها در زبان C#: برای تعریف ارایه ها در زبان C# به فرم زیر عمل می کنیم .

شکل تعریف ارایه های یک بعدی به صورت زیر است :

نام آرایه [] نوع

Int [] x;

برای تعیین ابعاد به صورت زیر عمل میکنیم:

[بعد] نوع=new نام آرایه

Int [] x=new int[5];

دادن مقدار اولیه به آرایه : برای دادن مقدار اولیه به آرایه می توان به شکل زیر عمل کرد:

Int [] x=new int [5] {5,7,4,6,1};

نکته : اگر مقادیر اولیه از تعداد خانه های آرایه بیشتر باشد,خطا صادر میگردد.

نکته : اگر مقادیر آرایه از تعداد خانه های آرایه کمتر باشد,خانه هایی از آرایه که مقدار نمیگیرند به 0 تبدیل میشوند.

★ با توجه به مقادیری که در آرایه داده شده قرار دارد (برای 5 عضو آرایه) عنصر آرایه را پیدا کرده و در textbox قرار دهید؟

Int [] x=new int [5] {12,16,20,30,14};

Int i,max =12;

For (i=1 ;i<5 ; i++)

{

If (max < x[i])

Max =x[i];

Textbox 1.text =max .tostring();

}

★ برنامه یک ماشین حساب با چهار عمل اصلی؟

مرتب سازی : دو نوع مرتب سازی داریم :

۱. مرتب سازی حبابی (bubble sort).

۲. مرتب سازی انتخاب (selection sort).

بدنه مرتب سازی حبابی : (صعودی)

```
For (i=0 ;i<n-1 ; i++)
{
For ( j=0 ; j<n-1 ; j++)
If (x[ j ] > x[j+1])
Temp=x[ j ];
X[ j ]=x[ j+1 ];
X[ j+1 ]=temp;
}
```

★ با توجه به آرایه داده شده برنامه ای بنویسید که عناصر آرایه را به صورت نزولی مرتب کرده و در 1 listbox نمایش دهد؟

```
Int [ ] x=new int [5] {7 , 4 , 9 , 13 , 16};
Inti,j,temp;
For(i=0 ; i < 5-1 ; i++ )
For ( j=0 ; j < 5-1 ; j++)
If ( x[ j ] < x [ j+1 ] )
{
Temp = x[ j ];
X[ j ]=x[ j+1 ];
X[ j+1 ]=temp;
For ( i=0 ; i < 5 ; i++ )
Listbox 1.items.add( x[ i ] .tostring())
}
```

مرتب سازی انتخاب : در روش مرتب سازی انتخاب در یک لیست بدین شکل عمل میکنیم که ابتدا عنصر اول لیست را در نظر گرفته و سپس از خانه دوم به بعد دنبال کوچکترین عنصر میگردیم. در صورت یافتن کوچکترین عنصر جای آن را با عنصر اول جا به جا می کنیم.این

روند را تا پیمایش آخرین عنصر لیست ادامه میدهیم.این روش برای مرتب سازی صعودی میباشد.برای نزولی عناصر بزرگتر لیست را در اول قرار می دهیم.

مثال :

```

Int [ ] x=new int [5] {7 , 15 , 3 , 25 , 4};
For ( i=0 ; i < n-1 ; i++ )
{
Min =x[ i ] ; cmin = i ;
For ( j=i+1 ; j<n ; j++ )
If ( min > x[ j ])
{
Min = x[ j ];
Cmin = j ;
}
Temp = x[ i ];
X[ i ] = x[ cmin ] ;
X[ cmin ] = temp ;
}
For ( i=0 ; i<n ; i++ )
Listbox 1.items.add( x[ i ].tostring());

```

برای وارد کردن تعداد اعداد مشخص برنامه به صورت زیر است :

بالای formload :

```
Int [ ] x=new int [5];
Inti=0;
```

داخل textbox 1.keypress :

```
Char ch ;
Ch=e.keychar;
If ( ch == '\r' )
{
Lable1.text = ( i+1 ).tostring ( ) ;
X[ i ]= convert . toint32( textbox1.text) ;
i++ ;
}
If ( i<5 )
{
Textbox1.text=" " ;
Textbox1.focus() ;
}
```

در button1_click :

```
For(i=0;i<5;i++)
Listbox1.items.add(x[i].tostring());
```

آرایه های دو بعدی : آرایه هایی هستند که همانند ماتریس ها دارای سطر و ستون هستند.

شکل تعریف این آرایه ها به صورت زیر است :

[بعد ۲ , بعد ۱] نوع داده new = نام آرایه [,] نوع داده

```
Int[ , ] m=new int [3,2];
For ( i=0 ; i<3 ; i++)
```

```
{
For ( j=0 ; j<2 ; j++ )
Listbox 1.items.add( x[ i,j ] .tostring ());
}
```


مقدار اولیه دادن به آرایه های دو بعدی : برای اینکار به شکل زیر عمل میکنیم :

```
Int[ , ] m=new int [3,2] {{4,7},{9,12},{4,8}};
```

برنامه ای بنویسید که دو ماتریس 2×2 داده شده را جمع کرده و به شکل ماتریسی چاپ کند؟ 

```
int [ , ] a=new int [2,2] {{6,7},{12,14}};
int [ , ] b=new int [2,2] {{2,3},{6,9}};
inti,j;
For( i=0 ; i<2 ; i++ )
{
For ( j=0 ; j<2 ; j++ )
Lable 1.text += ( a[ i , j ] + b[ i , j ]).tostring()+" ";
Lable 1.text += " \n ";
}
```

ترانهاده ماتریس : به جا به جایی سطر و ستون در ماتریس ترانهاده گویند.

برنامه ای بنویسید که ماتریس 3×2 دریافتی را در هنگام چاپ به صورت ترانهاده نشان دهد ؟ 


```
Int [ , ] a=new int [ 3,2 ] {{ 4,5 } , { 6,7 } , { 8,9, } };
Inti,j;
For ( i=0 ; i<3 ; i++ )
{
For { j=0 ; j<2 ; j++ )
Lable 1.text += a[i,j] . tostring() + " " ;
Lable 1.text += " \n " ;
}
```



```

}
For ( j=0 ; j<2 ; j++ )
{
For ( i=0 ; i<3 ; i++ )
Lable 1.text += (a[i,j] .toString () +" " );
Lable 1.text += "\n ";
}

```

برنامه ای بنویسید که دو ماتریس 3×3 داده شده را در هم ضرب کرده و نتیجه حاصله را به شکل ماتریسی چاپ کند؟ 

```

Int [ , ] a=new int [ 3,3 ] {{9,8,7},{6,5,4},{3,2,1}};
Int [ , ] b=new int [ 3,3 ] {{10,11,12},{13,14,15},{16,17,18}};
Int[ , ] c=new int [ 3,3 ];
Int i,j,k;
For ( j=0 ; j<3 ; j++ )
{
For ( i=0 ; i<3 ; i++ )
For ( k=0 ; k<3 ; k++)
C[ i,j ] += a[ i,k ] * b[ k,j ];
For ( i=0 ; i<3 ; i++)
For ( j=0 ; j<3 ; j++ )
Lable1.text += "\n ";
}

```

جستجو: انواع جستجو عبارتند از: جستجوی خطی , جستجوی دودویی :

$X = \log n + 1$

X


مثال :

Bool found = false ;

```

While (( L<=h)&&(found=false))
{
If ( key> x[ m ] )
L=m+1;
If (key < x[ m ] )
H=m-1;
If (key == x[ m ] )
Found = true;
}
If (found == true )
Textbox 1.text = " founded ";
Else
Textbox 1.text =" not found";

```

برنامه ای بنویسید که با توجه به لیست نمرات موجود در آرایه ، اگر نمره ۱۸ در داخل لیست بود پیغام پیدا شدن و اگر نبود پیدا نشدن را چاپ کند ؟ 

```

Int [ ] x=new int[ 6 ] { 13,15,16,18,19,20 };
Int l=0 , h=5 , key=18 , m ;
Bool found =false;
While (( L<=h ) && (found == false ))
{
M=(l+h)/2;
If (key > x[ m ] )
L=m+1;
If (k < x[ m ] )
H=m-1;
If (key == x[ m ] );

```

```

Found = true ;
}
If (found==true)
Text box 1.text ="founded";
Else
Textbox1.text="not found";

```

رشته ها : رشته ها در زبان C# براساس string تعریف میشوند .

شکلی آن به صورت زیر است :

نام رشته string ;

متد های کار با رشته :

در زبان C# متدهای مختلفی برای کار با رشته وجود دارند که عبارتند از :

length : این متد طول رشته را بر میگرداند.

```
S1="alireza"
```

```
Int i=s1.length();
```

copy to : این متد تعدادی از کاراکتر های یک رشته را از مکان خاصی از رشته بر میگرداند.

```
S1="alireza";
```

```
S1.copyto(0,s2,0,3);
```

Compare to : این متد برای مقایسه دو رشته بکار میرود.

```
S1="ali";s2="reza";s3="ali";
```

```
Int n=s1.compareto(s2);== -1
```

To upper : حروف کوچک را به بزرگ تبدیل میکند.

```
S1="ali";
```

```
S2=s1.Toupper(s1);== ALI
```

To lower : حروف بزرگ را به کوچک تبدیل میکند.

Trim , trim end , trim start : برای حذف جاهای خالی به کار میروند.

```
S1=" ali";
```

```
S2=s1.trimstart(s1);= "ali"
```

Substring: بخشی از رشته را بر می گرداند (تعداد رشته و شروع) .substring()

```
S1="alireza";
```

```
S2=s1.substring(1,3);= ali
```

Remove : برای حذف بخشی از رشته بکار میروند.

```
remove(تعداد رشته و شروع);
```

```
S1="alireza";
```

```
S2=s1.remove(1,3);= reza
```

Insert: برای جایگزین کردن یک رشته در داخل رشته دیگر در مکان مورد نظر به کار میروند.

★ برنامه ای بنویسید که نام شما را بگیرد و به صورت برعکس در textbox نمایش دهد ؟

★ مثلث خیام زیر را یا آرایه دو بعدی تعریف کنید ؟

۱

۱ ۱

۱ ۲ ۱

۱ ۳ ۳ ۱

۱ ۴ ۶ ۴ ۱

متد ها : استفاده از متد در یک کلاس در واقع همان نقش آفرینی توابع میباشد.

در نتیجه میتوان گفت که در توابع ما میتوانیم مقادیر را به صورت مقداری ویا ارجاعی استفاده کنیم.

ارسال با مقدار : وقتی پارامتری به روش مقداری به متدی ارسال میشود , متد فراخوانده شده یک کپی از مقدار آن پارامتر را دریافت می کند.اگر متد فراخوانی شده در مقدار این کپی از پارامتر تغییراتی ایجاد کند این تغییرات در مقدار اصلی آن تاثیری ندارد.

اما در روش ارجاعی , متد فراخوان به متد فراخوانی شونده این توانایی را می دهد که مستقیما به داده های اصلی در متد فراخوان دستیابی داشته باشد.در واقع میتوان گفت در روش ارجاع ما میتوانیم از طریق نام پارامتر به مقدار اصلی دسترسی داشته باشیم.

برای نوشتن کلاس در C# ابتدا در منوی Project گزینه Add class را انتخاب میکنیم , پنجره ای با نام Add new item ظاهر میشود که در آن نام کلاس را وارد کرده و دکمه Add را میزنیم. در نتیجه محیط کلاس مورد نظر برای نوشتن آماده میباشد.برای نوشتن کلاس در قسمت بالای Form load پیاده سازی میشود.

★ برنامه ای بنویسید که با استفاده از کلاس طول و عرض مستطیل را دریافت و مساحت مستطیل را محاسبه کند و در Textbox قرار دهد؟

در Class :

```
Class rect
{
Private int x;
Private int y;
Private int s;
Public int x;
```

```
{  
Get  
{  
Return x;  
}  
Set  
{  
X=value;  
}  
}  
Public int y;  
{  
Get  
{  
Return y;  
}  
Set  
{  
Y=value;  
}  
}  
Public void area()  
{  
S=x*y;  
}  
Public intarea1()  
{
```

```
Return s;
}
}
```

: Button در

```
Rectrect 1=new rect() ;
Rect1.x=convert.toint32( textbox1.text );
Rect1.y=convert.toint32( textbox2.text );
Rect1.area ();
Textbox3.text=convert.tostring( rect1.area1());
```

برای نوشتن توابع یا متد در C# بدین شکل عمل میکنیم که متد مورد نظر را در بالای Form-load نوشته سپس در بدنه برنامه آن تابع را فراخوانی می کنیم. یعنی با نوشتن اسم تابع و دادن مقادیر پارامتر میتوانیم محاسبه تابع مورد نظر را انجام دهیم.

★ متد یا تابعی بنویسید که در برنامه اصلی یک مقدار صحیح دریافت و سپس با تابع fact فاکتوریل آن را محاسبه و در برنامه اصلی چاپ کند؟

: Button در

```
Int n;
Float f;
N=convert .toint32 (textbox1.text);
F=fact (n);
```

```
Textbox2.text= f.tostring();
```

: form-load بالای

```
Public float fact ( int n) ;
```

```

{
Inti ;
Float x=1 ;
For ( i=1 ; i<=n ; i++ )
X=x * i ;
Return x ;
}

```

★ برنامه ای بنویسید که دو عدد صحیح دریافت و توسط متد Find max بزرگترین را پیدا و چاپ کند؟

متد های بازگشتی : در متد های بازگشتی ابتدا یک شرط خروج و فرمول عمومی آن را باید در نظر بگیریم :

```

Int fact ( int n )
{
If ( n == 1)
Return 1;
Else
Return n * fact (n-1) ;
}

```

★ در متد بازگشتی دو عدد صحیح دریافت و بدون استفاده از ضرب حاصلضرب آن را بدست آورد؟

```

Intmul( int x , int y )
If ( x==1 )
Return y :
Else
Return y+mul( x-1 , y );

```


★ برنامه ای بنویسید که دو عدد بیست رقمی را خوانده و سپس آنها را با یکدیگر جمع کرده و در خروجی نشان دهد ؟

★ توسط متد بازگشتی حاصل تقسیم دو عدد را محاسبه کنید ؟

★ برنامه ای بنویسید که ب.م.م دو عدد را نشان دهد ؟

در Button :

```
Int x,y,z;
X=convert.ToInt32 ( textbox1.text);
Y=convert.ToInt32(textbox2.text);
Z=bmm(x,y);
Textbox3.text=z.ToString();
```

بالای form-load :

```
Int bmm(int x,int y);
If ( x%y==0 )
Return y;
Else
Return bmm( y,x%y );
```

★ برنامه ای بنویسید که یک عدد صحیح دریافت توسط تابع بازگشتی مبنای ۲ آن را بدهد؟

در Button :

```
Long m,n;
N=convert.ToInt32(textbox1.text);
M=mabna(n)/10;
Textbox2.text=m.ToString();
```

بالای Form-load :

```
Long mabna (long n)
```

```

If ( n==0 )
Return 0;
Else
Return 10*mabna(n/2)+(n%2)*10;

```

ارسال آرایه ها به متد یا تابع : آرایه ها را نیز همانند متغیر ها میتوان به متد ها یا توابع ارسال نمود. برای اینکار در هنگام فراخوانی نام متد میتوان نام آرایه را به عنوان آرگومان قرار داده و سپس آن را ارسال نمود.

به عنوان مثال میخواهیم برنامه ای بنویسیم که با توجه به نمرات پنج دانشجو که در آرایه قرار دارد آن را به متد Find max ارسال نموده و بیشترین نمره را پیدا کرده سرانجام در برنامه اصلی چاپ کند؟

```

Float [ ] x=new float [5]{ 17 , 19 , 13 , 14 , 18 };
Float max;
Max=findmax (x);
Textbox1.text=max.toString();
Float findmax (float[ ] x)
{
Float m ;inti ;
M=x[0];
For ( i=1 ; i<5 ; i++ )
{
If ( m<x[ i ] )
M=x[ i ];
}
Return m;
}

```

برای ایجاد منو در C# از کنترل Menu strip استفاده میشود و پس از آوردن این کنترل به فرم در قسمت عبارت Type here نام منو هارا وارد میکنیم.

جدا کننده بین گزینه های منو : برای اینکار در هنگام طراحی منو در خاصیت Text آن علامت (-) قرار میدهیم. برای قرار دادن جدا کننده بین گزینه های منو مکان نما را به محل مورد نظر انتقال داده و کلید Insert را فشار دهید. با انجام این کار یک جای خالی درج میشود که با تنظیم خاصیت Text این گزینه به علامت (-) جدا کننده ای بین این دو گزینه ایجاد میشود.

کلید های شتاب دهنده (میانبر) : استفاده از این کلید ها که معمولا همزمان با گرفتن Alt + حرف اول نام منو میبود ما میتوانستیم اینکار را با قرار دادن علامت & قبل از حرف مورد نظر ایجاد کنیم.

افزودن کلید های Shortcut : برای اینکه بخواهیم هر گزینه یک منو به صورت Shortcut ظاهر شود ما میتوانیم در قسمت خاصیت Shortcutkey کلیک کرده و آن Shortcut مورد نظر را با توجه به موارد گفته شده در آن مشخص کرده و از آن در برنامه استفاده کرد.

افزودن گزینه به منو : برای افزودن گزینه مکان نما را به زیر یکی از گزینه هایی که مد نظر است برده و کلید Insert را کلیک میکنیم. با انجام اینکار یک گزینه جدید ایجاد میشود.

جا به جایی گزینه در منو : برای اینکار دو روش وجود دارد : یک روش حذف گزینه در محل مورد نظر و افزودن همان گزینه در محل جدید. در روش دوم مکان نما را به گزینه ای که باید جا به جا شود منتقل کرده سپس با استفاده از دکمه بالا یا پایین گزینه را به مکان مناسب میبریم. حذف گزینه های منو : برای اینکار کافیست مکان نما را به روی گزینه مورد نظر انتقال داده و کلید Delete را میزنیم. در نتیجه گزینه مورد نظر حذف میشود.

کنترل Color dialog : این کنترل به صورت کادر محاوره ای Color نمایان شده که ما میتوانیم توسط خاصیت Color آن رنگ مورد نظر را به اشیاء مورد نظر ببخشیم.

متد Show dialog : از این متد برای نمایش کادر محاوره ای Color استفاده میشود.

کنترل Font dialog : از این کنترل برای نمایش کادر محاوره ای Font استفاده میشود.

این کنترل دارای خواصی است که عبارتند از :

Font: Font متن را مشخص میکند.

Max size : حداکثر اندازه Font را مشخص میکند.

Min size : حداقل اندازه Font را مشخص میکند.

Show font : کادر محاوره ای Font را مشخص میکند.

★ برنامه ای بنویسید که با استفاده از منوها ساختار زیر را بوجود آورد؟

★ برنامه ای بنویسید که نام و نمره ۵ دانشجو را دریافت و دانشجویی که بیشترین نمره را

دریافت نمود همراه با نام آن را در خروجی چاپ کند؟

بالای Form-load :

```
int[] x = new int[6];
string[] y = new string[6];
int m = 0, a;
```

در Button1 :

```
m++;
if (m < 6)
{
(y[m]) = textBox3.Text;
(x[m]) = Convert.ToInt32(textBox1.Text);
listBox2.Items.Add(y[m]);
listBox1.Items.Add(x[m]);
textBox1.Text = " ";
textBox3.Text = " ";
}
```

در Button 2 :

```
int i,max=0;
for (i = 1; i < 6; i++)
```

```

{
if (max < (x[i]))
{
max = x[i];
a = i;
}
}
label1.Text = max.ToString();
label2.Text = (y[a]).ToString();

```

فایل ها و استریم ها : در زبان C# همانند سایر زبان های برنامه نویسی از دو سازمان فایل ترتیبی و تصادفی استفاده میشود.

سازمان فایل ترتیبی : این نوع فایل ها، رکورد ها به همان ترتیبی که از ورودی خوانده میشود در فایل قرار میگیرد.

برای استفاده از فایل از کلاس File stream استفاده میشود.

با استفاده از این کلاس ما میتوانیم اطلاعات را به صورت باینری در فایل بنویسیم و یا از فایل بخوانیم. این کلاس برای باز کردن فایل ها نیز بکار میرود. از طرفی این کلاس دارای خواصی است که آن ها را بیان خواهیم نمود.

شکل کلی استفاده از کلاس File stream :

FileStream(نام فایل , Filemode , حالت , Fileaccess , نوع دسترسی);

نام فایل : بیان کننده یک نام فایل میباشد.

Filemode : دارای حالت های زیر است :

۱. Appen : این حالت برای اضافه کردن اطلاعات به انتهای فایل بکار میرود.

۲. Creat : این حالت برای ایجاد کردن یک فایل بکار میرود.

۳. Creatnew : این حالت برای ایجاد یک فایل جدید بکار میرود.

۴. Open : این حالت برای باز کردن فایل بکار میرود.

۵. Open or creat : در صورت وجود آن را باز میکند در غیر صورت آن را ایجاد میکند.

File access : دارای حالت های زیر میباشد :

۱. Read : جهت خواندن فایل بکار میرود.

۲. Write : جهت نوشتن فایل بکار میرود.

۳. Read or Write : جهت خواندن , نوشتن فایل بکار میرود.

★ میخواهیم فایلی را با نام A.txt جهت نوشتن با استفاده از FileStream تعریف کنیم؟

```
FileStream out1=new FileStream ("A.txt",filemode.creat,fileaccess.write);
```

خواندن از فایل های ترتیبی : برای خواندن از فایل های ترتیبی میتوانیم از متد Serializable استفاده نماییم.

برای استفاده از این متد باید مراحل زیر را انجام دهیم :

۱. قبل از خواندن اطلاعات با این متد دستور [serializable] را در ابتدای کلاس قرار میدهیم.

۲. یک شی از نوع Binaryformatter برای خواندن اطلاعات بکار میرود.

مثال : `binaryformatter read1=new binaryformatter();`

۳. باز کردن فایلی که میخواهیم اطلاعات را از آن بخوانیم.

مثال :

```
filestream in1=new  
filestream("A.txt",filemode.open,fileaccess.read);
```

۴. خواندن رکورد از فایل با متد Deserialize که دستور زیر این عمل را انجام میدهد :

```
Record rec1=new ( record ) read1.deserialize ( int1 ) :
```

نوشتن اطلاعات در فایل های ترتیبی : برای نوشتن فایل ها در فایل های ترتیبی استفاده از متد Serialize میباشد.

برای انجام اینکار مراحل زیر انجام میشود :

۱. یک شی از نوع Binaryformatter تحت قالب نوشتن میسازیم.


```
label4.Text += st.name + " " + st.fname + " " + (st.av).ToString() +
"\n";
}
read1.Close();
```

در Button2 :

```
n++;
FileStream out1 = new
FileStream("a1.txt", FileMode.Append, FileAccess.Write);
stu st = new stu();
st.name = textBox1.Text;
st.fname = textBox2.Text;
st.av = Convert.ToSingle(textBox3.Text);
w1.Serialize(out1, st);
out1.Close();
```

بانک اطلاعاتی : یکی از روش های ذخیره و بازیابی اطلاعات استفاده از بانک اطلاعاتی Access یا SQL Server میباشد که با استفاده از کنترل هایی که با محیط Access یا SQL Server میتوانند کار کنند , بر اساس جداول طراحی شده ما میتوانیم به اطلاعات داخل جدول دسترسی داشته باشیم.

ابتدا بانک اطلاعاتی Access را در C# مورد بررسی قرار میدهیم :

برای استفاده از بانک اطلاعاتی Access در محیط Access یک جدول ساخته سپس با کنترل هایی که در جدول میتوانند ارتباط ایجاد کنند استفاده نماییم.

کلاس Dataset : از طریق این کلاس ما میتوانیم با جدول بانک اطلاعاتی ارتباط ایجاد کرده و اطلاعات جدول را با استفاده از متد هایی نمایش دهیم. باید در نظر داشت که قبل از استفاده از Dataset از کلاس Dataadapter برای ارتباط با جدول استفاده میکنیم.

خواص متد Dataset :

۱. Table : نام جدول را مشخص میکند.
۲. متد Copy : ساختار و داده های Dataset را کپی میکند.
۳. متد Reset : اطلاعات Dataset را به حالت اولیه بر میگردداند.

۴. متد Merg : اطلاعات دو Dataset را با هم ادغام میکند.

کلاس Dataadapter: این کلاس دارای دو شی oledbdataadapter برای جدول Access و sqlserver برای sqlserver است.

خواص متد Dataadapter :

۱. Deletecommand : برای حذف دستورات Sql بکار میرود.

۲. Selectcommand : دستورات Sql را برای انتخاب رکوردها استفاده میکند.

متد Fill : این متد رکورد ها را از جدول به بانک اطلاعاتی Dataset منتقل مینماید.

متد Update : دستورات Insert , Update یا Delete را به ترتیب برای درج , به هنگام سازی و حذف سطرها در Dataset فراخوانی میکند.

کلاس Datatable : این کلاس در واقع همان شی جدول ما میباشد که ما میتوانیم از طریق آن به اطلاعات داخل جدول دسترسی داشته باشیم.

خواص Data table عبارتند از :

۱. Column : مجموعه ای از ستون های جدول را مشخص میکند.

۲. Rows : مجموعه ای از سطر های جدول را مشخص میکند.

۳. متد Clear : داده های جدول را حذف میکند.

۴. متد Removeat : سطر موجود در جدول را با توجه به اندیس آن حذف میکند.

۵. Count : تعداد سطر ها را شمارش میکند.

مثال:

داخل Form-load:

```
oleDbDataAdapter1.Fill(dataSet11);
oleDbConnection1.Close();
```

داخل Delete :

```
dataSet11 .t1.Rows .RemoveAt (Convert .ToInt32 (textBox3 .Text )-1);
```

```
oleDbDataAdapter1.Update (dataSet11);
```

داخل Add :

```
dataSet11.t1.Addt1Row (textBox1.Text, textBox2.Text);
oleDbDataAdapter1.Update(dataSet11);
```

توضیح برای کار کردن با بانک اطلاعاتی Access :

برای آوردن اشیا بانک اطلاعاتی روی Toolbox راست کلیک کرده سپس گزینه Choose item را انتخاب و در پنجره ظاهر شده با نوشتن SQL به اشیا SQL server دسترسی پیدا کرده و در صورت نوشتن Ole اشیا بانک اطلاعاتی Access در اختیار ما قرار میگیرند. برای اینکار ابتدا شی ole adapter را روی فرم آورده و سپس روی آن کلیک کرده و شروع به تنظیم آن مینماییم. ابتدا بانک آن را مشخص و سپس تست کرده و بعد به قسمت connection string رفته و next را میزنیم. در پیام ظاهر شده گزینه No را انتخاب و سپس در قسمت بعد Next کرده و گزینه Query Build را انتخاب سپس فیلدهای مربوطه را انتخاب مینماییم و سرانجام Next را Finish میزنیم. پس از انجام این مراحل شی oledataconnection فعال میشود. سپس oleadapter را بر اساس Generaldataset تنظیم کرده تا Dataset ساخته شود. شی DataGridView را روی فرم آورده سپس در گوشه آن ، آن را تنظیم و جدول را به آن متصل مینماییم. پس از تنظیم، با ایجاد چند Textbox و Label دکمه Add, delete را روی فرم میسازیم. برای فعال کردن اطلاعات جدول بر روی DataGridView در قسمت داخل Form-load رفته و کد نویسی زیر را انجام میدهیم :

```
Oledbadapter1.fill( dataset11 );
Oledbadapter.close();
```

برای دکمه Add برنامه زیر را مینویسیم :

```
Dataset11.t1.addt1.row( textbox1.text, textbox2.text);
Oledbadapter1.update( dataset11 );
```

و برای دکمه Delete برنامه زیر را مینویسیم :

```
Dataset11.t1.rows.removeat( convert.toint32( textbox3.text ) - 1 );
Oledbdataadapter1.update( dataset11 );
```