مبانی برنامه نویسی

C#

www.arvinac.com

فهرست منابع این کتاب

فصل اول	آشنایی با زبان سی شارپ و .NET
فصل دوم	آشنایی با محیط Visual Studio
فصل سوم	قواعد برنامه نویسی در سی شارپ
فصل چهارم	شروع برنامه نویسی
فصل پنجم	آشنایی با انواع داده ها ، متغییرها و فضای نام و شی گرایی در سی شارپ
فصل ششم	کار با داده ها در سی شارپ
فصل هفتم	انواع عملگر ها در سی شارپ
فصل هشتم	کار با کنترلر ها در سی شارپ
فصل نهم	ساختار های شرطی و حلقه ها در سی شارپ
فصل دهم	آرایـــــــــه ها در سی شارپ

گردآورنده : آموزشگاه مهندسی آروین

فصل اول

آشنایی با زبان سی شارپ و .NET

فصل اول : مقدمه و آشنایی با زبان سی شارپ

برنامه نویسی با C# در محیط نرم افزاری Visual Studio کد نویسی می شود.

زبان برنامه نویسی c# یکی از قدرتمند ترین زبان های برنامه نویسی در کل جهان بوده و به کاربران امکان می دهد تا برنامه های مورد نیاز خود را ایجاد نمایند.

زبان سی شارپ که بر پایه سادگی ، مدرن بودن و همه منظوره بودن و هم چنین شیء گرایی شکل گرفته است توانسته همه ساله توجه برنامه نویسان را به خود جلب نماید.

C# توسط شرکت مایکروسافت ارائه شده که همچون دیگر زبان های برنامه‌نویسی دارای دو نوع کامپایلر است یک نوع آن command lineنام دارد که محیطی شبیه محیط Dos را داراست و دیگری Visual ست که محیط ویندوز مانندی دارد یعنی درحقیقت محیط گرافیکی است شاید برایتان جالب باشد که بدانید چنانچه شایع شده است مایکروسافت در مصاف با جاوا، به ارائه برنامه نویسی به زبان C# پرداخته و جالب تر آن است که شباهت های بین این دو زبان بسیار چشمگیر است مایکروسافت در رابطه با میزان استفاده و گسترش زبان فوق ابراز امیدواری نموده بود که گسترش چشمگیر در میان جامعه برنامه نویسان حکایت از صحت پیش بینی ها دارد. این زبان بگونه ای طراحی شده که نه تنها وابستگی به یک Platformخاص را ندارد، بلکه در اغلب موارد وابستگی Runtime نیز ندارد.

فصل اول : آشنایی با .NET

کمپانی ماکروسافت در جولای ۲۰۰۰ پلتفرم .NET را به جهانیان معرفی نمود.

این پروژه عظیم که زبان C#.NET بخشی از آن بود به مرور زمان جای خود را در بین علاقه مندان این رشته باز کرد ، و به یکی از قدرتمند ترین زبان های برنامه نویسی در دنیای امروز تبدیل شد !

به طوریکه از آن پس صحبت از نرم افزار های دیگری مانند Delphi , Oracle کمتر به گوش می رسد.

با استفاده از زبان C# می توان نرم افزار هایی با رویکرد های متفاوت تهیه و تولید کرد.

نرم افزار هایی مانند حسابداری ، انبار داری ، فروش ، حقوق و دستمزد و کسب و کار و مانند اینها را می توان از طریق سی شارپ طراحی نمود.

کمپانی بزرگ ماکروسافت نرم افزا ر قدرتمند .NET را برای نرم افزار های .NET ارائه کرده و شما می توانید با نصب ویژوال استودیو می توان زبان هایی مانند ویژوال بیسیک و سی شارپ در این محیط آشنا شوید.

محیط **.NET** یک محیط مجتمع برای اجرا و توسعه ی برنامه های کاربردی ویندوز ، برنامه های اینترنتی ، گزارش گیری و حتی دستگاه های موبایل است ، که در نسخه های جدید این برنامه ، پک هایی برای دستگاه های مختلف موبایل طراحی شده است.

C	C++	C#1
1980	1990	2000

فصل اول : آشنایی با .Net

.NET یک محیط مبتنی بر اصول طراحی شی ء گرایی برنامه ریزی و پایه گذاری شده است.

از چار چوب .NET می توان از هر سیستم عاملی بعنوان سیستم عامل میزبان استفاده نمود.

کار کردن با محیط .NET بسیار آسان می باشد و گزینه های امنیتی نیز در آن پیش بینی شده است.

.NET اقیانوسی از مفاهیم و اطلاعات می باشد که در زمینه های مختلف جهت کاربرانی با اهداف مختلف طراحی شده است.

.NET مشتمل بر حدود ۵۰۰۰ کلاس مستقل از زبان برنامه نویسی و مستقل از هر بستر و هر پلت فرمی می باشد.

منظور از مستقل از هر زبان برنامه نویسی اینست که شما می توانید برنامه خود را به هر زبانی مانند زبان های جدول زیر بنویسید.

C
C++
C#
J#
F#
VB.NET

فصل اول : آشنایی با .NET

وقتی برنامه ای را در محیط .NET می نویسیم ، برنامه شما به یک برنامه بینا بینی یعنی نه یک زبان سطح بالا و نه یک زبان سطح پایین تبدیل خواهد شد.

این زبان به نام (Microsoft Intermediate Language) **IL** نام دارد.

در صورتیکه زبانی به غیر از .NET را کامپایل می کنیم به زبان ماشین صفر و یک تبدیل شده که به کمک سیستم عامل اجرا می شود.

فایل IL فایلی است با پسوند .exe or .dll

سپس فایل il به کمک JIT (Just in time Compiler) ، تحویل ماشین کد یا Native Code شده و ماشین کد توسط CLR یعنی زبان کد هنگام اجرا یا همان Commone Language Runtime به زبان ماشین تبدیل می شود و در نهایت برای اجرا توسط سیستم عامل به cpu تحویل داده می شود.

.Net Frame Work چیست ؟

این برنامه زیر بنای اصلی Microsoft .net می باشد .

این برنامه ترکیبی از Class + Services می باشد.

این برنامه لایه ای را در بین برنامه های کاربردی و سیستم عامل شکل می دهد.

CLR

زبان مشترک زمان اجرا

اجرای کدها ، کامپایل ، مدیریت حافظه ، جمع آوری زباله و مدیریت نخ ها را به عهده دارد.

به زبان ساده تر ، مدیریت کل عمر یک چرخه کاربردی بر عهده CLR است.

FCL

کتاب خانه ای از صد ها کلاس پایه است.

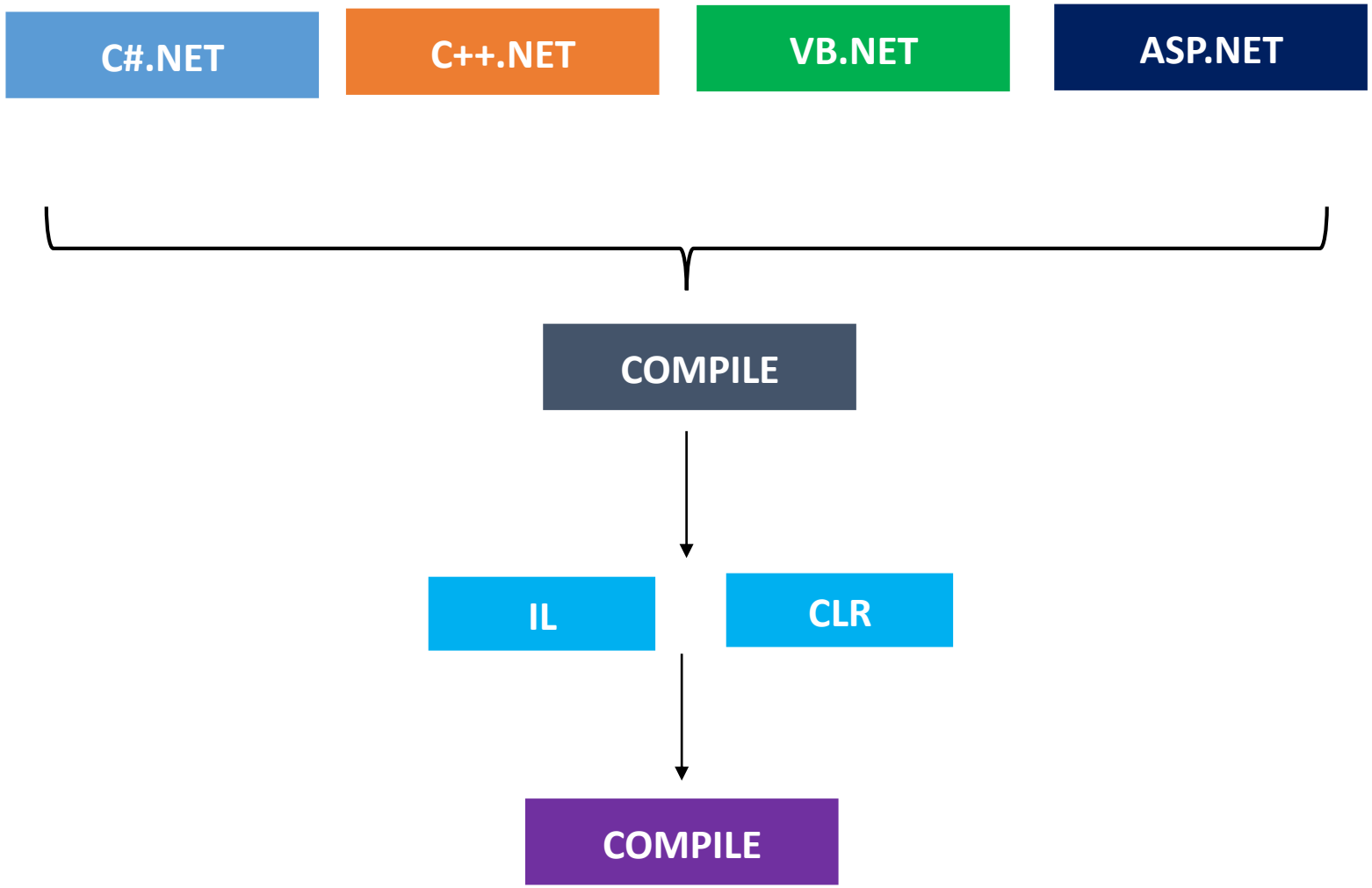
این کتاب خانه بالغ بر ۵۰۰۰ کلاس می باشد ، و همه زبان های NET. مانند :

C#.NET

VB.NET

C++.NET

ASP.NET


C#.NET
C++.NET
VB.NET
ASP.NET
COMPILE
IL
CLR
COMPILE

فصل دوم

آشنایی با VISUAL Studio

نحوه اجرای نرم افزار Visual studio 2013 و نحوه کار با آن

Start → All program → Microsoft Visual Studio 2013 → Visual Studio 2013

چنان چه برای بار اول از این نرم افزار استفاده می نماییم ، بهتر است در منوی **chose your default in vironment** کزینه **General Development Setting** را انتخاب نماییم.

سپس بر روی دکمه Start Visual Studio کلیک نمایید.

پس از باز نمودن برنامه ، ستون سمت چپ دارای آپشن های متعددی می باشد که شامل :

New Project , Open Project , …

ایجاد یک پروژه :

جهت ایجاد یک پروژه جدید در Visual Studio مراحل زیر را انجام دهید

File → New → Project

و یا از طریق دکمه های میانبر CTR + SHIFT + N

بعد از انتخاب یک پروژه جدید ، کادر محاوره ای NEW PROJECT باز می شود .که باید در آپشن Visual c# → Template را انتخاب نمود.

سپس در سمت راست گزینه Windows Form Application را انتخاب می نماییم.

در قسمت Name یک نام برای پروژه خود انتخاب نمایید.

در قسمت Location مسیر پیش فرض ذخیره سازی را تعیین می نماییم ، که توصیه می شود مسیر پیش فرض را تغییر ندهیم.

نهایتا بر روی دکمه ok کلیک می کنیم.

بعد از انجام مراحل قبل ، پروژه جدید نیز ایجاد می گردد و نرم افزار فرم اولیه را نیز نمایش می دهد.

نکته :

بعد از ایجاد پروژه جدید ، فایل های برنامه در پانل Solution که در سمت راست می باشد ، قرار گرفته است.

چنانچه این پانل را مشاهده نمی کنید به منوی View → Solution رفته و آن را فعال نمایید و یا از دکمه های : Alt+Ctl+L

در این پانل منابع تمام فرم ها ، تنظیمات قابل اجرا و ... دیده می شود.

بررسی پانل Solution

در بالا ترین قسمت این پانل ، نام Solution قرار دارد.

در یک Solution می توان چندین پروژه داشت.

در زیر نام Solution ، نام پروژه قرار دارد.

فایل Form1.cs ، فایل سورس کد های پروژه هستند و می توان کد های خود را در آن وارد نماییم.

{ برای ورود به محیط کد نویسی از کلید F7 استفاده می نماییم و با فشردن مجدد F7 مجددا به محیط فرم باز خواهیم گشت }

چنان چه قصد داشته باشیم به مسیر Solution حود برویم ، در پانل Solution بر روی نام Solution خود راست کلیک کرده و گزینه Open Folder In File Explore

پانل Properties

این پانل در پایین پانل Solution قرار دارد.

هر کنترلی که روی فرم قرار بگیرد از طریق این پانل قابل تنظیم است.

بعنوان مثال قصد داریم رنگ پس زمینه فرم را عوض نماییم ، بدین منظور بر روی فرم کلیک کرده و پانل Properties نیزفعال می گردد و سپس بر آیکون Alpha Betical کلیک کرده و خصوصیت Back color را انتخاب کرده و رنگ مورد نظر را برای فرم انتخاب می کنیم.

در سمت چپ محیط کاری چندین سربرگ وجود دارد.

سربرگ Toolbox

این سربرگ را انتخاب نموده که یک جعبه ابزاری که حاوی انواع کنترلر ها می باشد نمایش داده میشود.

چنانچه toolbox را نمی بینید باید به منوی view → Toolbox را انتخاب نماییم و یا می توانیم از کلید های کنترلی **Alt + ctr + X** هم استفاده نماییم.

در این پنجره ، کنترلر ها به چندین قسمت دسته بندی شده اند.

مثلا در All Window Form دسترسی به تمام کنترلر ها دسترسی خواهیم داشت.

و یا در قسمت Data ابزار های مربوط به پایگاه داده قرار دارند.

فصل سوم

قواعد برنامه نویسی در سی شارپ

قواعد برنامه نویسی در سی شارپ

هر زبان برنامه نویسی دارای قوانینی می باشد که برای شروع به کار با آن زبان ، می بایست آن قوانین را خوب فرا گرفت.

به قوانین برنامه نویسی نیز Syntax گفته می شود.

یکی از مهمترین **syntax** های زبان c# اینست که حتما باید در انتهای هر دستئر از **Semicolon(;)** استفاده کرد که در غیر اینصورت در هنگام کامپایل برنامه با خطا مواجه خواهیم شد.

برای نام گذاری متغییرها در سی شارپ می توانیم از قواعد زیر استفاده نمود

A-Z	a-z	0-9	_

هر نام باید با یک علامت و یا یک حرف شروع شود.

هم چنین نباید از کاراکتر های خاص مانند موارد زیر استفاده شود

?	!	%	@

هم چنین نباید از کارکتر – space در بین نام ها استفاده کرد.

{ زبان سی شارپ یک زبان Case Sensetive می باشد و به حروف کوچک و بزرگ حساس است }

مثلا دو حرف زیر در سی شارپ هیچ وقت باهم برابر نیستند

RESULT ≠ result

در سی شارپ ۷۹ کلمه تحت عنوان کلمات کلیدی و یا Keyword وجود دارد ، مانند . class , return , namespace

این کلمات در سی شارپ ذخیره شده اند و برای نام گذاری شناسه ها قابل استفاده نمی باشند.

نکته : در سی شارپ کلمات کلیدی (Keyword) ها با رنگ آبی نمایش داده می شوند.

برای دیدن این کلمات کلیدی بر روی دکمه F1 در محیط VISUAL کلیک کرده و بر روی گزینه

C# Keyword

نیز کلیک نموده تا کلمات کلیدی را ببینید.

درج توضیحات در سی شارپ

در این درس به بررسی توضیحات و یا همان **Comment** ها می پردازیم.

در برخی موارد برنامه نویس نیاز دارد که توضیحاتی را در برنامه وترد نماید و هم چنین نمیخواهد که این توضیحات اجرا و چاپ گردد.

در اینجا به دو روش می توانیم توضیحات را به برنامه اضافا نماییم :

- در روش اول ، اگر شما می خواهید توضیحات فقط به یک خط که برنامه در آن نوشته شده است اضافه گردد با علامت (//) این کار را انجام می دهیم.

```
// this is a my code ⟶ comment
```

روش دوم : چنانچه بخواهیم توضیحات به بیش از یک خط اعمال گردد از (/* */) استفاده می کنیم

```
/* this is a codes */
```

فصل چهارم

شروع برنامه نویسی

شروع برنامه نویسی در سی شارپ : نوشتن اولین برنامه

برای نوشتن یک برنامه در محیط Visual Studio به آدرس زیر می رویم :

File →New → Project →Visual c# →Windows Forms Application

بعد از نوشتن نام پروژه، محل ذخیره سازی آن را ایجاد می نماییم.

سپس در پروژه ایجاد شده به سربرگ **Toolbox** رفته و زیر مجموعه **Common Control** را انتخاب می نماییم ، سپس بر روی گزینه **Button** کلیک و آن را ایجاد نمایید.

برای تغییر نام دکمه ایجاد شده به پانل **propertice** رفته و به آپشن Text رفته و نام دکمه فعلی را عوض می نماییم و هم چنین خاصیت **name** را نیز می توان عوض نمود.

بعد از ایجاد دکمه جدید ، دو بار بر روی آن کلیک نموده تا به محیط کد نویسی آن وارد شویم.

مانند تصویر زیر عبارت messagebox را تایپ نموده و می نویسیم.

```
10
11  namespace APPPARAND2
12  {
13      public partial class Form1 : Form
14      {
15          public Form1()
16          {
17              InitializeComponent();
18          }
19
20          private void btnhello_Click(object sender, EventArgs
21          {
22              MessageBox.Show("Hello World!");
23          }
24      }
25  }
26
```

بجای تایپ messagebox می توان عبارت mbox را تایپ کرده و با دوبار فشردن کلید Tab ساختار آن را بطور کامل مشاهده نمود.

برای اینکه برنامه اجرا گردد باید آن را کامپایل نمود که به مسیر زیر می رویم

Build → Build solution را انجام می دهیم.و یا از کلید های ترکیبی ctl+shift+b می باشد.

بعد از انجام این کار پانل output در پایین کد نمایش داده می شود.و اگر خطایی در اجرای برنامه وجود نداشته باشد پیغام Succeeded نمایش داده خواهد شد.

اگر پانل **output** نمایش داده نمیشود بایستی از منوی View آن را فعال نمود.

برای اجرای برنامه به مسیر زیر بروید :

Debug → start without debugging

نحوه تغییر رنگ فرم به رنگ دلخواه

برای این کار از دستور زیر استفاده می کنیم

```
This.backcolor=color.blue
```

بعد از نوشتن کد باید از دکمه Start استفاده نمود.

برای اینکه دکمه دقیقا به مرکز فرم منتقل گردد باید از سمت راست با پانل **properties** رفته و به

position start → Center Screen

برای اینکه فرم با کلیک کردن بر روی دکمه بسته شود می توان از پانل Propertice زیر مجموعه

Text → Exit

Name -> Btnexit

سپس با دو بار روی دکمه به رویداد کلیک دکمه می رویم :

```
This . Close ( ) ;
```

در اینجا **close** یک متد محسوب میشود.در اینجا متد ها با آیکون مکعب مربع نمایش داده می شوند.

دقت نمایید بعد از نام متد می بایست از ( ) استفاده نمایید و در پایان نیز از ; استفاده نماید.

نکته : اگر برای بار اول برنامه نوشتیم باید آنر Debug I نماییم و در مراحل بعد می توان از Start و یا دکمه میانبر **F5** برای اجرای برنامه استفاده نمود.

چنانچه بخواهیم با دکمه ESC از فرم خارج شویم باید در پانل Proper ice از مسیر زیر این کار را انجام دهیم

Cancell button → btnexit

فصل پنجم

آشنایی با انواع داده ها ، متغییرها و
فضای نام و شی گرایی در سی شارپ

فصل پنجم

آشنایی با داده ها و متغییر ها در سی شارپ

آشنایی با انواع داده ها

هر زبان برنامه نویسی برای پردازش داده ها ، به انواع مختلفی از داده ها نیاز داد .

در سی شارپ نیز این قاعده وجود دارد.

متغییر ها در سی شارپ

گاهی اوقات در برنامه لازم است که عدد و یا حروفی را ذخیره نموده تا بعدا بتوان آن ها را فراخوانی نمود و عملیاتی را بر روی آن انجام دهیم که برای این کار از متغییر ها استفاده می شود.

متغییر و یا Variable مکانی در حافظه موقت است که می تواند مقداری را در خود نگهدارد که البته این مقدار قابل تغییر است.

زمانیکه مقداری را در متغییر قرار می دهید ، مقدار قبلی آن نیز از بین خواهد رفت.

متغییر ها با نام خود فراخوانی میشوند.

استفاده از دو نام برای یک متغییر مجاز نمی باشد.

قبل از استفاده از متغییر باید آن را اعلان و یا Declare نماییم.

منظور از نام گذاری متغییر ، تعیین نوعی و مقداری است که می تواند بپذیرد.

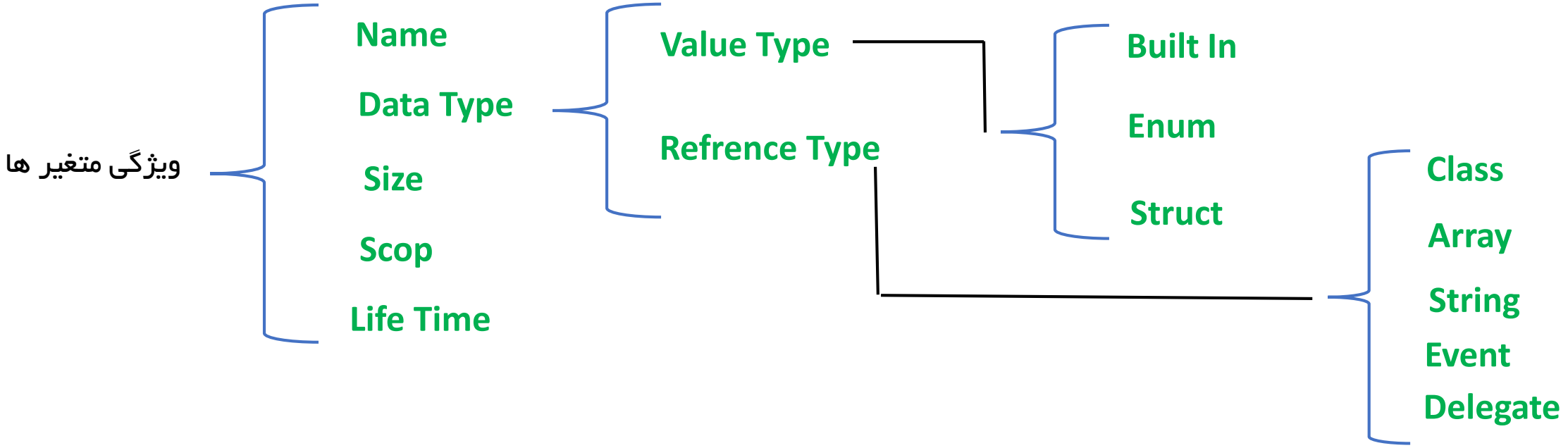
در سی شارپ ، می بایست ابتدا مقدار اولیه یک متغییر را مشخص نمود و سپس از آن استفاده کرد.

قوانین نام گذاری متغییر ها در سی شارپ

نام متغییر باید با یکی از حروف _ , a-z , A-Z شروع شود.

استفاده از اعداد در نام متغییر مجاز نمی باشد .

از کارکتر های خاص مانند (... , . , % , space , @ , !) نمی توان به هیچ عنوان استفاده نمود.



تفاوت Value Type ها با Reference Type ها

Value Type

این متغیر ها در Stack ذخیره می شود ، به زبانی ساده تر هم خودشان و هم مقدارشان در پشته و یا Stack ذخیره میشوند.

Reference Type

این متغییر ها ، آدرس آنها در Stack و مقدار آن در Hip ذخبره میشود.

برای بررسی متغییر ها در سی شارپ باید از طریق دکمه F1 به HELP رفته و به انتهای صفحه می رویم و بر روی گزینه Built –In Types Table کلیک کرده که به این ترتیب فهرستی از متغییر های سی شارپ نمایش داده می شود.

نکته : برای دسترسی به help این نرم افزار باید حتما به اینترنت وصل شویم.

نحوه نام گذاری متغییر ها در سی شارپ

ابتدا مقدار متغییر را درج می نماییم و سپس یک space و بعد یک نامی را برای آن درج می کنیم.

در اینجا مقدار int ، یک مقدار صحیح را نشان می دهد.

```
Int number = 10 ;
```

مقدار long بازه بزرگتری از اعداد صحیح را شامل میشود.برای این مقدار در انتها حرف L را قبل از سیمیکالن باید تایپ نمود.

```
Long my long = 40 L;
```

مقدار Float برای داده های عددی اعشاری استفاده می شود.و باید در انتهای مقدار این متغییر حرف F را تایپ نمود.

```
Float my float = 29.5 F ;
```

مقدار متغییر Double برای تعریف متغییر های عددی اعشاری استفاده میشود، با این تفاوت اعداد اعشاری با دقت بالاتری را می توان توسط این متغییر تعریف کرد.

```
Double mydouble = 2.545 ;
```

مقدار متغییر Char برای تعریف متغییری از نوع کاراکتر تعریف می شود که این گونه متغییر ها حروف را در خود ذخیره می کنند.

```
Char mychar = ‘a‘ ;
```

مقدار متغییر String که برای تعریف متغییر هایی که شامل رشته ای از کاراکتر ها هستند استفاده می شود که باید نام در بین دو double cote قرار گیرد.

```
Sring mystr =“ali”;
```

مقدار متغییر Bool که می تواند تنها دو مقدار درست و غلط را در بر گیرد.

```
Bool mybool = true ;
```

استفاده از کلمه کلیدی var

زمانیکه بجاب نوع داده ای متغییر از کلمه کلیدی var استفاده می کنیم.

در اینجا وقتی که از کلمه کلیدی var استفاده می کنیم ، سی شارپ مقدار متغییر را تعیین میکند.

```
Var myvariable = 25 ;

Myvariable=5 ;
```

که در اینجا ، سی شارپ ۲۵ را یک مقدر صحیح میشناسد.

نکته : در ارجاعات بعدی نمی توان مقداری غیر از عدد صحیح به آن نسبت داد.

فضای نام Name Space

فضایی برای دسته بندی کلاس ها بر حسب وظایف آنها می باشد.

فضای نام را می توان مشابه Folder ها دانست.

علامت یک Name Space بصورت { } می باشد، که باید در ابتدای فرم و با دستور **Using** به برنامه معرفی گردد.

در سی شارپ دو نوع کلاس وجود دارد :

نوع اول که از قبل در .NET نوشته شده اند و می توان در هر کجای برنامه که نیاز شد از آن استفاده نمود.

گروه دوم کلاس هاییی هستند که خود برنامه نویس آنها را ایجاد کرده و سپس از آن استفاده می نماید.

کلاس های نوشته شده در .NET در کتابخانه (Framework Common Library) **FCL** قرار دارند و هر کدام از این کلاس ها در فضای مربوط به خود دسته بندی شده اند.پس بنابر این برای استفاده از این کلاس ها باید در ابتدای برنامه فضای نام مربوطه را فراخوانی کرد.

فضاهای نام Name Space

مهمترین فضای نام موجود در سی شارپ ، فضای نام System است و این فضای نام خود شامل فضاهای نام دیگریست که خود در برگیرنده انواع کلاس های استثنایی ، کلاس های پایه ، کتابخانه توابع ریاضی و ... می باشد.

فضای نام Sysrem.Collections نام دارد که برای طراحی انواع کالکشن ها استفاده میشود.

فضای نام System. Data می باشد که این فضای نام برای اتصال سی شارپ به پایگاه داده مانند Oracle , Sql Server نیز استفاده شده .

فضای نام System.diognestics می باشد که برای اشکال زدایی و رفع باگ استفاده می شود.

فضای نام System. Drawing برای عملیات گرافیکی می باشد.مانند ترسیم مستطیل ، کمان ، بیضی و ... می باشد.

فضای نام System. Globalization است که برای تعرف واحد های پولی رایج ، تعریف زبان های مختلف ، تاریخ های مختلف و هر آنچه مرتبط با فرهنگ ای مختلف است استفاده میشود.

فضای بعدی System.Io که یرای ورودی و خروجی داده استفاده میشود.

فضای System. Security برای تنظیم امنیت داده ها استفاده میشود.

فضای نام System.threading که برای کار با نخ ها و دسترسی و مدیریت آن استفاده میشود.

فضای نام System.Web برای کار با asp.net و صفحات وب می باشد.

فضای نام System.Windows.Forms برای کار با کلاس های مورد استفاده در فرم مانند ... , textbox , Buttom می باشد.

فضای بعدی System.xml هست که برای پردازش اسناد xml از آن استفاده می کنییم.

فصل پنجم

مفهوم شی ء گرایی و کلاس ها در سی شارپ

مفهوم Class از کلمه **Classification** و یا دسته بندی کردن گرفته شده است.

هر برنامه را می توان با استفاده از کلاس ها دسته بندی نمود و بدین ترتیب برنامه ساختار یافته تر شده و کار با این چنین برنامه ها بسیار راحت تر می باشد.

مفهوم شی ء گرایی یا Object Oriented Programming (Oob)

در برنامه نویسی شی ء گرایی ، هر واقعیت بیرونی را به صورت مجموعه ای از موجودیت ها یا اشیا ء در نظر بگیریم.

بعنوان مثال در دانشگاه ، سه نمونه از اشیاء را که می توان در نظر گرفت عبارت اند از :

- **دانشجو**
- **استاد**
- **درس**

بنابر این برنامه ای که برای دانشگاه می نویسیم می تواند ، دانشجو و استاد و درس داشته باشد.

بنابراین سعی کنید که قبل از نوشتن هر برنامه ای ، آن برنامه شامل کلاس هایی باشد و آن کلاس باید شامل متد ها یی باشد.

مثلا یک دانشجوظ متد هایی از قبیل نام و شماره دانشجویی باشد و اعمالی که یک دانشجو انجام می دهد به عنوان یک متد شناخته می شود.

برای مثال متد ثبت نام برای کلاس دانشجو می باشد.

کلاس ها و فضای نام در سی شارپ

همانطور که قبلا هم اشاره شده Net. شامل ۵۰۰۰ تا کلاس می باشد.

در این درس قصد داریم که بررسی نماییم که در Net. کلاس ها در کجا قرار دارند.

حرف اول نام کلاس را می بایست بصورت بزگ تایپ نمود.

بعنوان مثال قصد داریم که کلاس proccess را فراخوانی نماییم ، بدین ترتیب حرف اول کلاس که P هست را بزرگ تایپ نموده و سپس بقیه متن را تایپ می تماییم ، بعد از تایپ کامل نام کلاس (مثلا همان Proccess) از ترکیب کلید های alt+shift+f10 منویی مبنی بر نام فضای کلاس در زیر نام کلاس درج می گردد.

در این منو دو گزینه وجود دارد :

usingSystem.Diagnostics

usingDiognostics.Proccess

کلاس نخست به ما پیشنهاد می کند که دستور using را در فرم قرار دهد که با انجام این کار دستود using System.Diognostic در ابتدای فرم نمایان می شود.

در هر دو حالت پس از افزودن فضای نام ، کلاس سبز رنگ می شود یعنی همان .Proccess

کلاس ها و فضای نام در سی شارپ

مثالی دیگر :

بر فرض می خواهیم از کلاس Sqlconnection استفاده نماییم.

ابتدا نام SqlConnection را تایپ نموده و بلافاصله مستطیلی با نام **SqlDbType** نمایان می گردد.

می توان با کلید های ترکیبی **Alt + shift + f10** از منوی ترکیبی که در زیر آن باز می گردد استفاده نمود.

که فضای نام مربوط به کانکشن نیز می آید.

با انتخاب **System.Data.SqlClient.SqlConnection** در ابتدای کلاس می آید که نام کلاس سبز رنگ می شود.

فصل ششم

کار با داده ها در سی شارپ

نحوه تبدیل انواع داده ها به یکدیگر

برای شروع متغییری از نوع string را به متغییری از نوع int تبدیل خواهیم کرد.

در سی شارپ برای تبدیل انواع داده ها می توان از دو روش استفاده کرد :

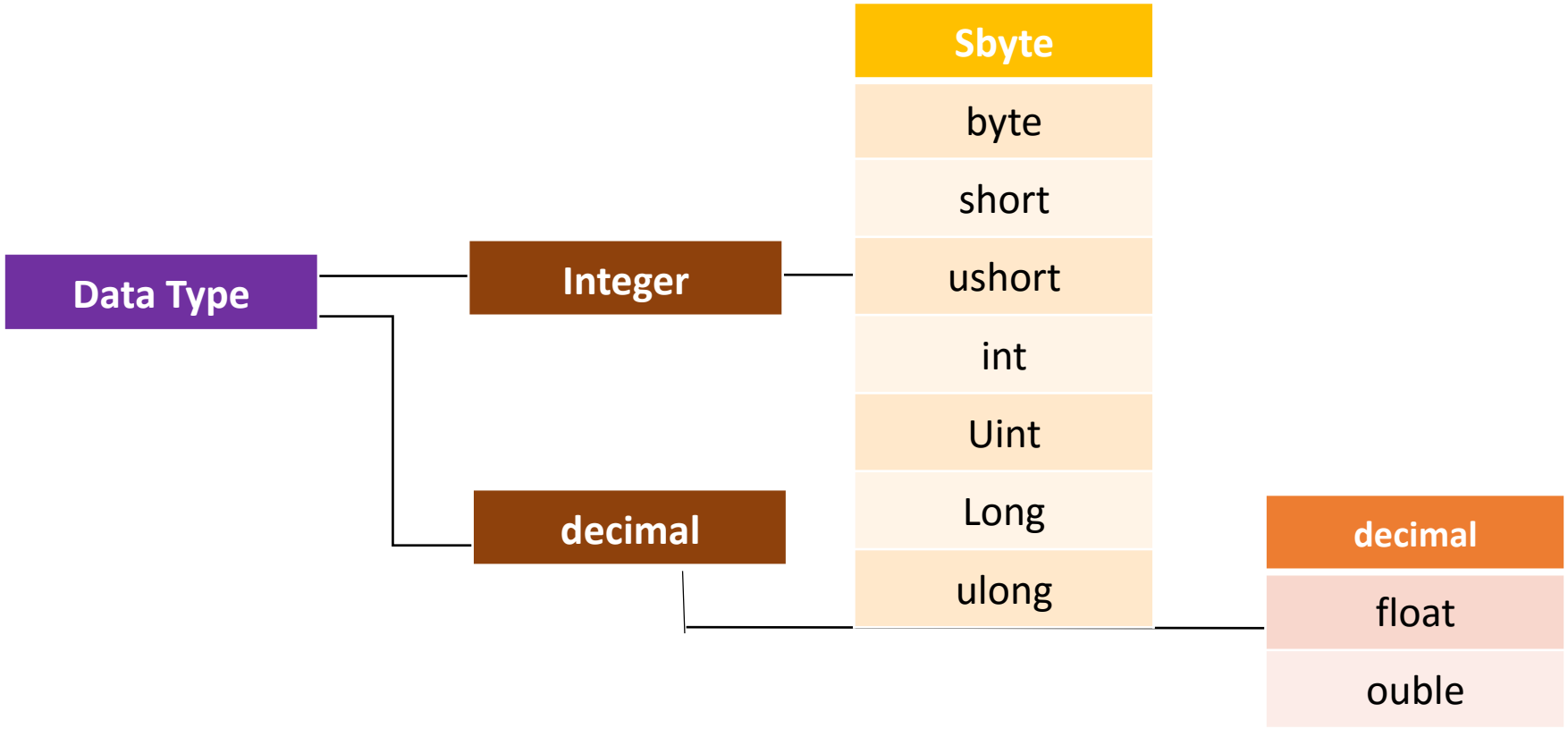
روش اول : استفاده از کلاس Convert

```
String mystr = “32” ;

Int myint = Convert.ToInt32(mystr);
```

در اینجا بعد از نقطه می توان تبدیل های مختلف را مشاهده نمود.

روش دوم : استفاده از متد parse

```
Int myint2 ;

Myint2 = int.parse(mystr);
```


Data Type
Integer
decimal
Sbyte
byte
short
ushort
int
Uint
Long
ulong
decimal
float
ouble

متغیر هایی که ابتدای نام آن ها از حرف u استفاده شده به معنای Unsigned یعنی بدون علامت منفی می باشد.این نوع متغییر ها فقط اعداد مثبت را می پذیرند.

نوع داده ای Short به اندازه ۲ بایت از حافظه را اشغال می کند .

نوع داده ای int به اندازه ۴ بایت از حافظه را اشغال می کند .

نوع داده ای long به اندازه ۸ بایت از حافظه را اشغال می کند .

نوع داده ای byte به اندازه ۱ بایت از حافظه را اشغال می کند .

نوع داده ای float به اندازه ۴ بایت از حافظه را اشغال می کند .

نوع داده ای decimal به اندازه ۱۶ بایت از حافظه را اشغال می کند .

نحوه استفاده از داده های رشته ای و کارکتری

داده های کاراکتری ویا همان **char** فقط شامل یک کاراکتر بوده و می توانند شلمل اعداد ، حروف و علائم باشند.و مقدار آن داخل
Singlecote قرار میگیرد.

```
Char mychar ;
Mychar = ‘a’ ;
```

داده های رشته ای ویا همان **string** ، که میتواند شامل یک رشته از اعداد، حروف و یا علامت باشند و مقدار آن همیشه بین
double Cote قرار می گیرد.

نکات کاربردی داده ها (۱)

اولین نکته در رابطه با داده ها در سی شارپ در مورد int می باشد.

همانگونه که قبلا هم گفته شد ، متغییر int یک نوع عددی صحیح را نمایش میدهد.

یک مثال کاربردی :

```
Int I = 123 ;
Int j = i*25000;
Msgbox.show(j.tostring());
```

در اینجا آرگومان ورودی در متد **show** حتما باید از نوع رشته ای باشد.

و متغییر **j** با استفاده از دستور **Tostring** به رشته تبدیل خواهد شد.

نکات کاربردی داده ها (۱)

متد **Checked** : از این متد برای پیغام خطای overflow استفاده مینماییم.

```
Int j = checked (i*25000) ;
```

ممکن است جواب درستی از متغییر ها دریافن نکنیم ، مثلا با ضرب ۴*۸ جواب ۲۵۰۰ را داشته باشیم و این جواب درست نمی باشد ولی برنامه بدون خطا کار میکند . اگر از checked استفاده نماییم ، می توانیم یک پیغام خطا بصورت overflow داشته باشیم که نشان دهنده جواب نادرست از برنامه می باشد.

نکات کاربردی داده ها (۱)

نحوه فراخوانی متغییر های نوع char :

```
Char alpha = ‘a’ ;
Int x = alpha ;
Messagebox.show(x.tostring() ) ;
```

در اینجا متغییر **x** کد اسکی حرف a را نمایش میدهد.

نحوه درج شماره برای خطوط در محیط برنامه نویسی سی شارپ :

Tools → Option→texteditor →alllanguages→display→linenumbers

نکات کاربردی داده ها (۲)

مثالی از متغییر های نوع اعشاری

```
Float myflo = 25/2 f ;
```

نکته : در زبان سی شارپ نوع پیش فرض اعداد اعشاری Double می باشد.

در اینجا چنانچه بخواهیم یک عدد را بصورت اعشاری درج نماییم باید با درج **f** در انتهای عدد این کار را انجام دهیم ، مه اصطلاحا به این نوع درج کردن **Cast** میگویند.

این نوع **Cast** کردن برای اعداد دسیمال نیز صادق است که با درج حرف **m** این کار انجام میگیرد.

```
Decimal mydecimal = 25/2 m ;
```

چند روش برای مقدار دهی متغییر های نوع int

```
Int a ;
a=14;
Int a,b =5;
```

برررسی میدان دید متغییر ها (Scop)

منظور از scop اینست که متغییر در چه مرحله ای از برنامه میتواند اجرا گردد.

```
Int myresault ;
My result = 25 ;
```

در اینجا به این نوع متغییر ها ، متغییرهای لوکال گفته می شود.یعنی در همان رویداد ، نام myresault نیز بصورت لوکال می آید.که در حالت evenhandler می باشد.

متغییر های Privet :

این مغییر ها در خارج از Event Handler ها تعریف میشود.

طبق استاندارد ماکروسافت بهتر است متغییر privet را قبل ازevent handler تعریف کنیم.

بهتر است متغیرهای privet در ابتدای فرم تعریف شوند.

```
Private string mystring ;
```

که اگر در هر جای برنامه مثلا mystring را تایپ نماییم ، این نام بصورت private تعریف میشود.ودر تمام فرم های پروژه در دسترس می باشد.

بررررسی میدان دید متغییر ها (Scop)

متغییر های نوع publice داخل یک کلاس تعریف شده ، سپس برای دسترسی به این نوع متغییرها حتما میبایست شیئی از کلاس مورد نظر ایجاد کنیم.و این گونه متغییرها در همه جا قابل دسترسی اند.

متغییرهای نوع Protect :

این نوع متغییرها فقط داخل کلاسی که تعریف شده اند نوشته میشوند ، ودر همع جا قابل دسترسی نیستند.

فصل هفتم

انواع عملگر ها در سی شارپ

عملگر های ریاضی در سی شارپ

```
Int  x = 10 ;
Int  y = 25 ;
Int additional = x+y;
Int subtraction = x-y ;
Int multiple = x*y;
Int divition = x/y ;
Int remiender = x%y ;
```

در سی شارپ مانند بقیه زبان ها ، علگرهای ریاضی دارای اولویت هستند

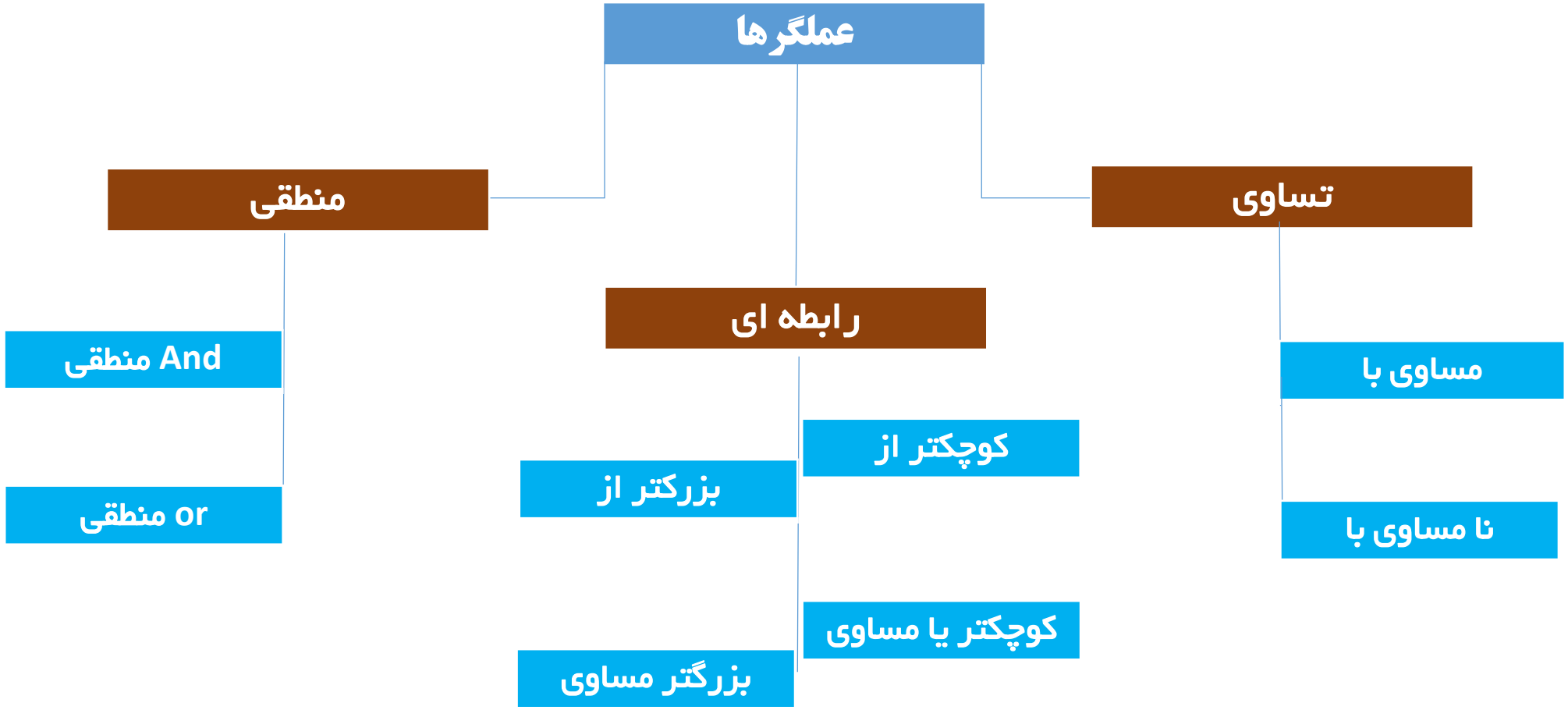
* / % + -

این عملگر ها را با استفاده از پرانتز میتوان تغییر داد..

در متغییرهای از نوع رشته ای تنها از عملگر + برای چسباندن دو رشته استفاده میشود.

فصل هفتم

انواع عملگرها در سی شارپ



داده های از نوع بولی

این داده های تنها دو مقدار True و یا False را نیز قبول میکند.

```
Bool mybool = true;
Messagebox.show(mybool.tostring() );
```

نتیجه True

عملگر not :

این عملگر باعث معکوس شدن جواب میشود. که نمای آن ! است.

```
Bool mybool = true;
Messagebox.show( ((!mybool).tostring() );
```

نتیجه Fals

عملگر تساوی

عملگر مساوی به صورت == می باشد

```
 private void button1_Click(object sender, EventArgs e)
{
        bool myresult;
        int mycount = 17;
        myresult = (mycount==17);
        MessageBox.Show(myresult.ToString());
}
```

عملگر نا مساوی

عملگر نا مساوی به صورت != می باشد

```
 private void button1_Click(object sender, EventArgs e)
{
        bool myresult;
        int mycount = 17;
        myresult = (mycount!=17);
        MessageBox.Show(myresult.ToString());
}
```

عملگر رابطه ای و منطقی

این عملگر ها شامل :

<= و >= و > و < می باشند.

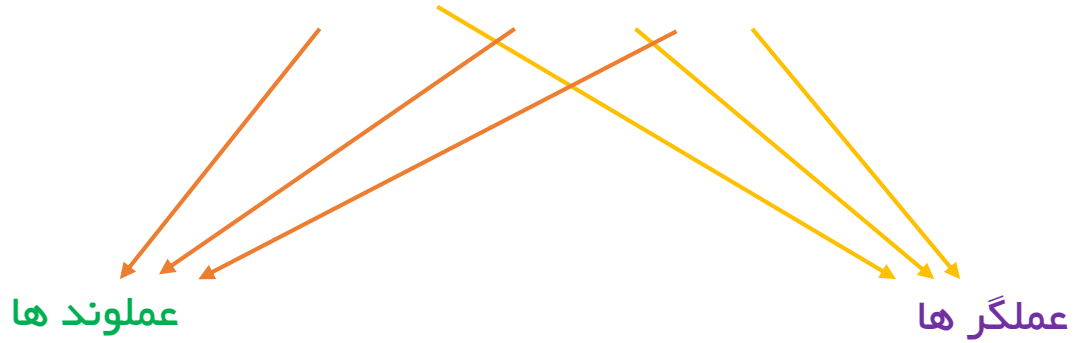
این عملگر ها پس از انجام مقایسه مقدار True و یا False را بر می گردانند.

```
private void button1_Click(object sender, EventArgs e)
        {
          bool result;
          int count = 40;
          result = (count > 0  && count < 10);
          result = (count < 0 || count > 10);
          MessageBox.Show(result.ToString());
        }
```

عملوند چیست ؟

هر عبارت که بین دو عملگر قرار بگیرد و یا بعد از یک عملگر بیاید یک عملوند محسوب می گردد.

25 * 14 + 6 / 8



عملگر های ++ و --

برای اضافه کردن یک مقدار عددی به یک متغییر از ++

برای کم کردن یک مقدار عددی از یک منغییر از --

فصل هشم

کار با کنترلر ها در سی شارپ

خصوصیات یک فرم :

برای تنظیمات خواص فرم ، از جمله رنگ بک گراند و ... باید به پانل PROPERTY رفته و تغییرات را اعمال کنیم.

کنترلر Text Box

در نرم افزار ویژوال ، کنترلرها در پانل سمت چپ بنام Toolbox قرار گرفته اند .

کنترلر ها ، ابزار هایی هستند که بر روی فرم قرار گرفته و هر کدام کار خاصی را انجام می دهند .

مثلا : از کنترلر Lable برای نمایش متن و از کنترلر text Box برای ورودی های کاربر مورد استفاده قرار می گیرد .

مشخصات کنترلر ها در پانل Properties نمایش داده میشوند .

برای تنظیم ارتفاع کنترلر text box ، به پانل properties رفته و سپس گزینه Multi line را انتخاب می نماییم !

گزینه Accept return باید همیشه False باشد ،و باعث میشود که مشکلات تارتباط با پایگاه داده را که برنامه نویسان را دچار مشکل میکند برطرف نماید.

گزینه character Casing برای ورودی هایی که به بزرگی و کوچکی حروف حساس هستند.

کنترلر Label

برای نمایش اطلاعات و داده های خروجی استفاده می نمایم.

خصوصیات کنترلر Lable

Name : برای درج نام یک لیبل استفاده میشود.

Text : برای تعیین متن لیبل بکار میرود.

کنترلر Combo box

توسط این کنترلر می توان یک لیست بازشو، از اطلاعات را ایجاد نمود .

خصوصیات کنترلر ComboBox

Item : اصلی ترین خصوصیت این کنترل item است ، که برای تعیین عناصر موجود در لیست بکار میرود.

در این قسمت ، می بایست لیستی از اطلاعات را وارد نمود.

Data Sourc : برای اتصال بانک اطلاعاتی به برنامه است.

DroupDownStyle : برای تعیین سبک نمایش combobox استفاده میشود.

- Droup Down List : کاربر نمیتواند متنی را اضافه نماید.
- Simple : کل لیست نمایش داده میشود.
- Droup Down : کاربر می تواند متنی را وارد نماید.

کنترلر Combo box

خصوصیات کنترلر ComboBox

Display Member : مربوط به نمایش اطلاعات ستونی است که قصد نمایش آن را داریم.

Value Member : مربوط به اطلاعات ستونی است که قصد ذخیره سازی آن را داریم.

بعنوان مثال : اگر بخواهیم بصورت پیش فرض یکی از مقادیر لیست نمایش داده شود ، باید index آن را بدانیم.

در سی شارپ ، hindex ها از ۰ شروع می شوند.

```
private void Form1_Load(object sender, EventArgs e)
{
comboBox1.SelectedIndex = 0;
}
```

یک مثال :

یک **textbox** , **button** , **combobox** ایجاد کرده و به **combobox** اطلاعاتی را بدهید.سپس بر روی دکه اگر کلیک کردید ، مقادیر **combo** را داخل **textbox** نمایش دهد .

پاسخ :

```
private void button1_Click(object sender, EventArgs e)
{
textBox1.Text = comboBox1.SelectedItem.ToString();
//textBox1.Text = comboBox1.SelectedIndex.ToString();
{
```

کنترلر Check Box

از این کنترل برای تاثییر و یا عدم تاثیر یک گزینه استفاده میشود.

حالت های منطقی که شامل درست و نادرست بودن است با استفاده از این کنترلر بررسی میشود.

خصوصیات مهم Check Box

Checked : مشخص می کند که آیا کنترلر checked انتخاب شده باشد یا خیر !

Exm:

```
private void button1_Click(object sender, EventArgs e)
            {
             label1.Text = checkBox1.Checked.ToString();
            }
```

خصوصیات مهم Check Box

Appearance: برای تعین ظاهر check box بکار میرود.

Auto check: در صورتیکه کاربر بر روی متن کلیک نمایید تغییر حالت نشان داده میشود که اگر در حالت False باشد کاربر حق انتخاب نخواهد داشت.

Check align: برای تعین محل قرار گیری متن مورد کنترل می باشد و بیشتر جاهایی کاربرد دارد که متن فارسی نوشته شده باشد.

Three State: اگر مقدار این قسمت حالت true باشد ، مقدار چک باکس می تواند سه حات درست ، غلط و نامعین باشد.مثلا اگر قصد سوال از کاربر را داشته باشیم و هیچ یک از حالات درست و غلط را نخواهیم ، می بایست حالت نامعین را داشته باشیم.

در حالت نامعین به حالت مربع نشان داده میشود و در حالن درست بصورت تیک دار، و در حالت نادرست هیچ علامتی داخل مربع نمایش داده نمی شود.

خصوصیات مهم List Box

خصوصیت مهم این کنترلر فهرستی از گزینه هاست ، که کاربر می تواند هریک از آنها را انتخاب نماید.

در این کنترلر ، اگر تعداد لیست ها زیاد باشد ، اسکرول برای پیمایش آنها ایجاد میگردد.

خصوصیات دیگر این کنترلر Itemes است که برای تعیین گزینه های مورد نظر کاربرد دارد.

از دیگر خصوصیات این کنترلر Selection Mode می باشد که در اینجا اگر گزینه Multiple Extended را انتخا ب نماییم ، به کاریر حق انتخاب چند گزینه را میدهیم.که کاربر با نگه داشتن کلید های shift و یا control می تواند گزینه ها را انتخاب نماید.

در حالت Multi Simple کاربر می تواند بدون نگه داشتن Shift , control گزینه ها را انتخاب نماید.

د حالت one کاربر فقط یک گزینه را میتواند انتخاب نماید.

و در نهایت با انتخابnone کاربر هیچ گزینه ای را نمیتواند انتخاب نماید.و اطلاعات بصورت readonly می باشد.

خصوصیات مهم List Box

خصوصیت Sorted که برای مرتب کردن لیست استفاده می شود در پانل موجود می باشد.

خصوصیت AlwayeVisible می باشد که اگر این خصوصیت را با True تنظیم نماییم در زمان نمایش اگر لیست بیش از حد مجاز بود ، Scrool خواهیم داشت و اگر در حالت True تنظیم نماییم ، اسکرول نخواهیم داشت.

کنترلر Picture Box

از این کنترلر برای نمایش تصاویر استفاده میشود.

مهمترین خصوصیت این منترلر ، image می باشد که برای درج تصویر میباشد.

خصلت بعدی ،Size Mode است.که با استفاده از این خصوصیت ، میتوان تصویر درون picture تراز نمود .

خصوصیت Wait Onload که برای لود شدن تصویر استفاده میشود و تا زمانیکه تصویر لود میشود ، یک حالت ساعت شنی نمایش داده میشود.

یک مثال کاربردی : با استفاده از یک دکمه ، تصویری را به درون یک Picture Box لود نمایید !

```
private void button1_Click(object sender, EventArgs e)
}
pictureBox1.Image = Image.FromFile(@"G:\Picture Gallery\Gallery1\Wallpaper
Collection\Wallpaper-2012\Wallpaper-3\a.jpg");
{
```

کنترلر Timer

چنانچه قصد داریم عملیاتی در بازه زمانی خاص از اول تا انتهای برنامه اجرا شود استفاده می نماییم.

وقتی این کنترلر را درگ مینماییم بر روی صفحه ، این کنترلر نمایش داده نمیشود و در پایین محیط ، نمایش داده میشود.

به این گونه کنترلرها Component گفته میشود.

مهمترین خصوصیات تایمر Enabled است. که فعالیت و یا عدم فعالیت این کامپوننت را تعیین میکند.

خصوصیت Interval است که زمان لازم برای اجرای عملیات را مشخص می نماید.

یک مثال کاربردی : فرمی طراحی نمایید که بعد از ۴ ثانیه ، رنگ فرم به آبی تغییر نماید !

ابتدا خصوصیت intrerval = 4000 و خصوصت enabled = true .

سپس بر روی دکمه timer کلیک کرده و برنامه زیر را مینویسیم.

```
private void timer1_Tick(object sender, EventArgs e)
    {
    this.BackColor = Color.Blue;
    }
```

کنترلر Web Browser

از این کنترل برای نمایش صفحات وب استفاده می شود.

این کنترل بصورت Navigate است ک از آن برای آدرس سایت استفاده میشود.

ذکر یک مثال :

ابتدا یک textbox و یک دکمه و سپس یک webbrowser را فعال مینمایید.

سپس در رویداد کنترلر دکمه کد زیر را وارد نمایید :

```
private void button1_Click_1(object sender, EventArgs e)
    {
    webBrowser1.Navigate(textBox1.Text);
    }
```

کنترلر Menu Strip

منوها از مهمترین اجزای یک برنامه گرافیکی می باشد.

برای پیاده سازی منوها به مسیر زیر بروید :

Toolbox -> Menu and Toolbars - > Menue Strip

چنانچه بخواهیم در فرم راست کلیک نماییم ، میتوانیم از آپشن :menu Context نیز استفاده نماییم.

برای شروع در فرم مورد نظر خصوصیت right to left را حتما باید تنظیم نماییم.و سپس برای فرم عنوانی در نظر میگیریم .و بعد خصوصیت StartPosition را در فرم را به حالت Center Screen تنظیم می نماییم.

سپس در toolbox آپشن MenuStrip را اضافه می نماییم ، که با این کار یک منو در بالای فرم اضافه و همچنین در پایین فرم نیز menu strip اضافه شده است.

کنترلر Menu Strip

در منوی ساخته شده در قسمت typrhere میتوان عنوان منو را وارد نمود.

بعد از وارد نمودن عنوان منو ، می توان زیر منویی را نیز اضافه نمود.

برای ایجاد یک Seprater ، میتوان در منوی Type Hrer رفتهخ و کاراکتر – را وارد و بع Enter نمود.

برنامه نویسی منوها (۱)

بهتر است برای منو های اصلی نام آن را MNU در نظر بگیریم و برای منوی فرعی نامitm

اگر در منو از خاصیت خروج خواستیم استفاده نماییم ، ابتدا نام آن را نام گذاری نموده و سپس به محیط کد نویسی آن میرویم.سپس کد زیر را وارد می نماییم :

```
 private void itmexit_Click(object sender, EventArgs e)
{
Application.Exit();
}
```

برنامه نویسی منوها (۱)

اگر خواستیم در منو ها بازدن دکمه ، یک فایل exe باز شود از کد زیر استفاده می نماییم :

```
private void itmcalc_Click(object sender, EventArgs e)
{
System.Diagnostics.Process.Start("calc") .WaitForExit();
}
```

در این مثال ، ماشین حساب اجرا خواهد شد.

متد waitForExit باعث میشود که برنامه فقط یک بار اجرا شود.

برنامه نویسی منوها (۲)

برای اینکه تصویری را از روی سیستم بر روی فرم نمایش دهیم ، می توانیم در یک منو نامی با عنوان تغییر عکس ایجاد نموده و سپس کد زیر را وارد نماییم :

ابتدا کنترلر Pictur Box را انتخاب نموده و خصوصیات زیر را اعمال می نماییم :

۱- borderstyle - > fixed single

۲- sizemode -> StechImage

۳- Doc -> fill

در صفحه بعد کد زیر را وارد می نماییم :

```
private void itmchangpic_Click(object sender, EventArgs e)
{
            string fileName = " ";
            OpenFileDialog open = new OpenFileDialog();
            if (open.ShowDialog()==DialogResult.OK)
}
                fileName = open.FileName;
{
            pictureBox1.ImageLocation = fileName;
{
```

کنترلر contextmenustript

این کنترلر همان منویی است که با راست کلیک د برنامه نمایش داده میشود.

با اضافه نمودن این کنترلر به فرم ، امکان انتخاب راست کلیک کردن را خواهیم داشت.

بطور مثال ، مقادیری شامل : save , save as , select all را در کنترلر وارد نماییم.

و اگر خروجی بگیریم متوجه خواهیم شد که با راست کلیک کردن موس هیچ اتفاق خاصی نخواهد افتاد.

در اینجا به پنل propertice رفته و سپس خاصیت Context Menu Stript را انتخاب و گزینه

context meny stript 1 را انتخاب میکنید تا راست کلیک فعال شود.

و همچنین میتوانیم یک textbox طراحی نماییم و برای آن یک contentmenuestript در نظر بگیریم.

کنترلر status strip

با اضافه نمودن این کنترل ، نوار وضعیت در پایین فرم نمایش داده میشود.

چنانچه بر روی مستطیل سمت راست این منو کلیک نمایم ، آپشن هایی نیز نمایان میشود.

با استفاده از status Lable پیغام های مناسبی را میتوان به کاربر نمایش داد که بطور مثال ساعت سیستم و یا نام کاربری که به سیستم وارد شده است.

در اینجا چنانچه بخواهیم رویداد مربوط به ساعت سیستم را نشان دهیم ، می بایست به رویداد فرم رفته و کد زیر را وارد نماییم .

کد مربوطه ،در صفحه بعد آورده شده است.

```
public Form1()
 }
InitializeComponent();
 {

private void Form1_Load(object sender, EventArgs e)
{
toolStripStatusLabel1.Text = DateTime.Now.ToString();
}
```

کار با خصوصیت های یک کنترلر :

در اینجا به ۳ روش میتوانیم برای کار با متدها و خصوصیات کنترلر عمل نماییم :

اولین روش : با استفاده از پانل propertice می توان به خصوصیات یک کنترلر دسترسی داشت.

دومین روش : با استفاده از Smart Tag میتوان به خصوصیات یک کنترلر دسترسی داشت.

بعنوان مثال یک picturbox را بر روی فرم مورد نظر drag نمایید و در بالای آن یک کادر کوچکی نمایش داده میشود که همان Smart Tag می باشد.

Smart tag یک کادر هوشمند بوده که بر روی اغلب کنترلر ها وجود دارد.

سومین روش : در این روش ، می توان کد نویسی انجام داد.

فصل نهم

ساختار های شرطی و حلقه ها

ساختار های شرطی در سی شارپ !

این ساختار ، یکی از مهمترین ساختارهایی است که در تمام زبان های برنامه نویسی مورد استفاده قرار می گیرد.

بعنوان مثال ، برای مقایسه دو عدد و بری بدست آوردن بزرگترین عدد می بایست از ساختار های شرطی استفاده نمود.

بطور کلی ، به مجموعه ای از دستور العمل ها ، که امکان انتخاب و تصمیم گیری از بین یک یا چند موضوع را برای ما فراهم می نمایند را ساختار های تصمیم گفته می شود.

یکی از مهمترین دستورات شرطی if می باشد .

از دستور if برای تصمیم گیری در برنامه استفاده میشود.

```
private void button1_Click_1(object sender, EventArgs e)
    {
            int num;
            num = int.Parse(textBox1.Text);
            if(num<=10)
            {
                MessageBox.Show("very bad");
            }
            else
            }
            MessageBox.Show("good");
            }
   }
```

دستور else if

چنانچه بخواهیم چندین شرط را در سی شارپ چک کنیم می بایست از دستور else if استفاده نماییم !

```
If()
{}
Elseif()
{}
Else
{}
```

دستور do while

در این ساختار شرط حلقه در پایان بررسی میشود.

do { command}

while(condition) ;

```
private void button1_Click(object sender, EventArgs e)
{
    int sum = 0;
    int i = 1;
    do
    {
        sum += i;
        i++;
    } while (i<=10);
}
```

حلقـــه for

حلقه for یکی از پرکاربرد ترین ساختارهای برنامه نویسی است.

حلقه for یک حالت کوتاه تر و منسجم تری از حلقه while است .

do { command}

while(condition) ;

```
for (int i=1; i<10; i++)
{
Commands
}
```

فصل دهم

آرایه ها

تعریف آرایه

آرایه ها این امکان را فراهم می نماید که متغیر های هم نوع داشته باشیم .

نحوه تعریف آرایه مشابه تعریف متغییر ها می باشد .با ابن تفاوت که بین نوع داده و آرایه از [] استفاده می گردد.

روش دسترسی به اعضای آرایه :

برای دسترسی به اعضای آرایه ، از نام آرایه به همراه شماره آرایه که به آن اندیس و یا index گفته می شود، استفاده می نماییم.

اندیس مشخص می کند که قصد دسترسی به چندمین عضو از آرایه را دارید !

اندیس ها از **صفر** شروع می شوند.

دستورات **Try Catch** برای مدیریت خطاها در سی شارپ بکار میرود.

برای این کار ، دستورات را انتخاب نموده و روی دستورات راست کلیک می نماییم و گزینه Surraound Width را انتخاب و سپس Try را نیز انتخاب می نماییم .

در این مثال یک دکمه را ایجاد نموده و به رویداد آن وارد میشویم و سپس دستورات زیر را وارد می نماییم.

```
private void button1_Click(object sender, EventArgs e)
 {
 try
 }
 int[] myarray = new int[] { 1, 2, 3, 4, 5 };

 MessageBox.Show(myarray[6].ToString());
    {
    catch (IndexOutOfRangeException error)
    }
 MessageBox.Show(error.Message);
  }
}
```

پیمایش آرایه ها توسط دستور For

فرض کنید که می خواهیم عملیات یکسانی را بر روی تمامی اعضای آرایه انجام دهیم.

بعنوان مثال ، می خواهیم تمامی اعضای آرایه را درون یک ListBox نمایش دهیم.برای این کار یک دکمه و یک listbox ایجاد نموده و سپس وارد رویداد Button شده و دستورات زیر را وارد می نماییم:

```
private void button1_Click_1(object sender, EventArgs e)
  {
  int[] myarray = { 1, 2, 3, 4, 5, 6 };
  for (int i = 0; i < myarray.Length; i++)
      {
      listBox1.Items.Add(myarray[i]);
      }
 }
```

نکات تکمیلی در مورد آرایه ها

در این مثال از دستور Sort برای مرتب کردن یک آرایه استفاده می نماییم.

```
private void button1_Click_2(object sender, EventArgs e)
{
            int[] myarray = new int[] { 1, 22, 3, 4, 5, 6 };
            MessageBox.Show(myarray[1].ToString());
            Array.Sort(myarray);
            MessageBox.Show(myarray[1].ToString());
}
```

نکات تکمیلی در مورد آرایه ها

در این مثال از دستور Riverse برای معکوس کردن آرایه استفاده می نماییم.

```
private void button2_Click(object sender, EventArgs e)
{
            int[] myarray = new int[] { 1, 22, 3, 4, 5, 6 };
            MessageBox.Show(myarray[2].ToString());
            Array.Reverse(myarray);
            MessageBox.Show(myarray[2].ToString());
}
```

کپی کردن یک آرایه در آرایه دیگر

استفاده از متد Coppy.

ابتدا می بایست آرایه مقصد را تعریف کرده ، سپس مشخص نماییم برای چند خانه از آرایه عمل کپی انجام گیرد.

ابتدا یک دکمه و یک listbox را درج کرده و سپس به رویداد button رفته و کد های زیر را وارد می نماییم.

در این مثال ، ابتدا با استفاده از متد clear لیست باکس را خالی می نماییم.

```
private void button1_Click_3(object sender, EventArgs e)
  {
   listBox1.Items.Clear();
   int[] a = { 2, 3, 5, 6, 77 };
   int[] b = new int [a.Length] ;
   Array.Copy(a, b, 3);
   for (int i = 0; i <=2; i++)
       {
      listBox1.Items.Add(b[i]);
      }
  }
```

آرایه های چند بعدی

معمولا آرایه های دوبعدی به بالا را شامل میشود .

آرایه های دوبعدی بصورت یک ماتریس می باشد که شامل سطر و ستون است.

رایه های سه بعدی را می توان بصورت یک مکعب در نظر گرفت.

مثالی از آرایه های دو بعدی :

```
private void button1_Click_4(object sender, EventArgs e)
 {
   int[,] matrix = { { 1, 4 }, { 2, 8 }, { 4, 6 } };
   for (int i = 0; i < 2; i++)
       {
      for (int j = 0; j < 1; j++)
          {
           listBox1.Items.Add(matrix.GetValue(i, j).ToString());
          }
       }
       }
```

آرایه های سه بعدی

این آرایه ها را بصورت یک مکعب میتوان در نظر گرفت.برای این نوع آرایه ها از سه حلقه for استفاده می نماییم.