

فصل اول: آشنایی با مفاهیم پایه‌ای پردازش داده‌ها

داده‌ها (Data)

تعریف ۱: مجموعه‌ای از مقادیر در مورد یک موضوع یا شیء است که به صورت کمی با یک مقدار عددی و یا به صورت کیفی نشان داده می‌شود.

تعریف ۲: داده‌ها، مقادیر خام و اولیه در مورد یک موضوع هستند.

مثال:

- ✓ نمرات درس ریاضی یک کلاس
- ✓ اسامی دانش‌آموزان یک کلاس
- ✓ نام شهرهای استان محل سکونت شما
- ✓ اندازه دمای محیط

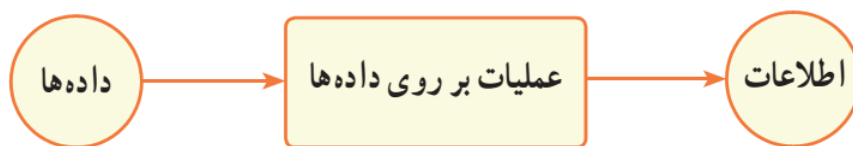
اطلاعات (Information)

اطلاعات، نتایج حاصل از عملیات و محاسبات بر روی داده‌ها می‌باشد که می‌تواند مورد تفسیر و بررسی قرار گیرد و نتیجه بررسی آنها به **دانش** ختم می‌گردد که دانش می‌تواند مبنای تصمیم‌گیری برای انجام کاری شود.

پردازش (Process)

مجموعه محاسبات و عملیاتی که بر روی داده‌ها صورت می‌گیرد را **پردازش** می‌نامند.

ارتباط بین داده، پردازش و اطلاعات



نکات مربوط به فرآیند پردازش داده‌ها

- ✓ **صحت داده‌ها:** یعنی داده‌ها به درستی گردآوری شده باشند و داده اشتباه در بین آنها وجود نداشته باشد.
- ✓ **درستی انجام محاسبات:** یعنی محاسبات و یا به طور کلی عملیاتی که بر روی داده‌ها صورت می‌گیرد با دقت و بدون اشتباه انجام شوند و در حین انجام عملیات لطمه‌ای به داده‌ها وارد نشود.
- ✓ **روش انجام پردازش:** استفاده از روش‌های بهینه و الگوریتم‌های مناسب در عمل پردازش توصیه می‌شود.

کامپیوتر (Computer)

کامپیوتر پردازشگر داده‌ها است.

برنامه (program)

به مجموعه دستوراتی که به کامپیوتر می‌فهماند که چه نوع پردازشی را بر روی داده‌ها انجام دهد و همچنین اطلاعات به دست آمده را چگونه نمایش دهد برنامه می‌گویند

برنامه‌نویسی (programming)

نوشتن دستورات لازم برای کنترل نحوه کار کامپیوتر، به طوری که کامپیوتر بتواند یک کار مشخص را انجام دهد را برنامه‌نویسی می‌گویند.

برنامه نویسی (programmer)

شخصی است که آشنا به دستورات یک زبان برنامه نویسی باشد و با به کارگیری صحیح و مناسب دستورات، برنامه نویسی کند

زبان ماشین (Machine Language)

زبان قابل فهم سخت افزار کامپیوتر، زبان ماشین نام دارد و متشکل از دنباله‌ای از کدهای ۰ و ۱ است
نکته: با توجه به اینکه نوشتن دستورات به زبان ماشین وقت گیر و دشوار است و زبان ماشین هر پردازنده با پردازنده دیگر متفاوت است، زبانهای دیگری طراحی شده‌اند که نوشتن برنامه به آن زبانها برای ما ساده‌تر از زبان ماشین است.

مترجم (compiler)

مترجم یک برنامه کامپیوتری می‌باشد که وظیفه آن، ترجمه و تبدیل دستورات یک زبان سطح بالا، به کدهای زبان ماشین می‌باشد.

انواع زبانهای برنامه نویسی (با توجه به نزدیکی یا دوری به زبان محاوره ای انسان)**الف- زبانهای سطح پایین (Low Level Language)**

زبانهایی که کد آنها به زبان پردازشگر کامپیوتر (CPU) نزدیک باشد و مسلماً از زبان محاوره‌ای ما دور هستند زبانهای سطح پایین نام دارند.

مثال: زبان ماشین و زبان اسمبلی (Assembly Language)

ب- زبانهای سطح بالا (High Level Language)

زبان برنامه نویسی که کد آنها به زبان محاوره‌ای ما نزدیک باشد، زبان سطح بالا نام دارد.

مثال: زبانهای C#, VB, Java

ج- زبانهای سطح میانی (Medium Level Language)

زبانهایی در این دسته قرار می‌گیرند که در آنها دستوراتی برای دسترسی راحت تر به سخت افزار پیش بینی شده باشد و همچنین به زبان عامیانه نزدیک باشند.

مثال: زبان C

نکته: برنامه نویسان از این زبانها برای کار با سخت افزار کامپیوتر و برنامه ریزی وسایلی که در آنها پردازشگر وجود دارد استفاده می‌کنند.

فصل دوم: آشنایی با زبان C#

✓ یک زبان سطح بالا، شی گرا و همه منظوره است که به وسیله شرکت مایکروسافت ابداع و توسعه پیدا کرده است.

✓ در زبان C# در هنگام ترجمه و همچنین اجرای برنامه دقت زیادی بر روی تطبیق و به کارگیری داده ها صورت

می‌گیرد تا از اشتباهات دستوری برنامه نویسی جلوگیری نماید.

✓ زبان C# نسبت به حروف کوچک و بزرگ حساس است.

الگوی یک برنامه ساده به زبان C#

```
نام کلاس class
{
    static void Main()
    {
        ;دستورات مربوط به انجام یک کار (الگوریتم)
    }
}
```

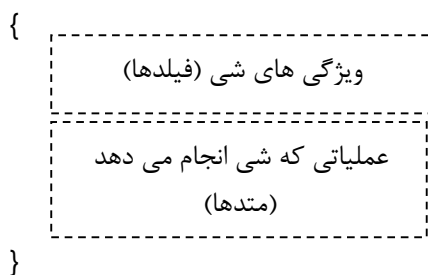
فضای نام (namespace)

منطقه ایی برای تعریف کلاس هاست

کلاس (class)

کلاس یک مفهوم اساسی در برنامه نویسی شی گرا است. ویژگی ها و عملیاتی که یک شی قادر به انجام آن است، در قالب یک کلاس تعریف می شوند.

نام شی Class

**متد (method)**

مجموعه ای از دستورات است که برای انجام یک کار لازم است. عملیاتی که یک شی انجام می دهد بصورت مجموعه ای از دستورات در قالب یک متد نوشته می شود. محل تعریف متد در داخل یک کلاس است.

متد Main()

در برنامه های زبان C# ممکن است متدهای زیادی تعریف و یا مورد استفاده قرار گیرند، اما حتماً باید متدی به نام Main() تعریف شده باشد که **نقطه آغاز اجرای برنامه** است و اجرای یک برنامه از اولین دستور داخل آن شروع می شود و الگوریتم خود را با رعایت قوانین زبان C# در آن می نویسیم.

static void Main()

```

{
    دستور ۱ ;
    دستور ۲ ;
    دستور ۳ ;
    ...
}
  
```

شناسه

نامی که توسط برنامه نویس در برنامه استفاده می شود. مانند انتخاب نام کلاس

روشهای نام گذاری شناسه ها

✓ **روش پاسکال:** در این روش اولین حرف هر کلمه بصورت بزرگ نوشته می شود. مانند SumTortal و

Average. برای نامگذاری **کلاس**، **متد** و **فضای نام** از این روش استفاده می شود.

✓ **روش کوهان شتری:** در این روش اولین حرف کلمات دوم به بعد با حروف بزرگ نوشته می شود مانند

sumTotal و average. برای نامگذاری **متغیرها** از این روش استفاده می شود.

✓ **روش مجارستانی:** در این روش **نوع داده** قبل از نام ذکر می شود. مانند strName و intAge

کلمات کلیدی یا رزرو شده

کلمات شناخته شده برای زبان C# هستند و دارای معنی و مفهوم ثابتی هستند. کلمات کلیدی در برنامه با **حروف کوچک** نوشته می شوند. مانند class، static و void

رشته

حروف، علامتها و عبارتی که ما بین علامتهای نقل قول نوشته می‌شود را رشته می‌نامند. این حروف می‌تواند فارسی، انگلیسی یا به هر زبانی باشد. مانند "Mohammad"

کاراکتر

هر یک از حروف و علامتها را یک کاراکتر می‌نامند. برای مثال رشته Mohammad از ۸ کاراکتر تشکیل شده است. چنانچه فاصله در رشته وجود داشته باشد، فاصله نیز یک کاراکتر محسوب می‌شود

استفاده از کلاس‌ها و متدهای آماده

در زبان C# گنجینه‌ای از کلاس‌های مختلف و کاربردی، از قبل تعریف شده و بصورت آماده وجود دارد که برنامه نویس کافی است آنها را بشناسد و در برنامه استفاده نماید. Console یک کلاس آماده در زبان C# است که عملیات مختلف ورودی و یا خروجی در آن تعریف شده است و در فضای نام System تعریف شده‌اند. چند نمونه از متدهایی که در این کلاس تعریف شده اند عبارتند از:

- ✓ Write () : برای نمایش یک پیام (رشته) روی صفحه
- ✓ WriteLine() : برای نمایش یک پیام روی صفحه و انتقال مکان نما به ابتدای سطر بعد
- ✓ ReadLine() : برای دریافت داده از کاربر
- ✓ ReadKey() : انتظار برای زدن یک کلید

نحوه استفاده از متدهای آماده

متد.کلاس.فضای نام

مثال: نمایش یک پیام روی صفحه

```
System.Console.WriteLine("welcome to c#");
```

نکته: در داخل برنامه می‌توانیم کلمه System را از ابتدای دستورات حذف کنیم به شرطی که در ابتدای برنامه فضای نام System را بصورت زیر معرفی نماییم:

```
using System;
```

ابزارهای برنامه نویسی

- ✓ ویرایشگر متنی مانند notepad یا notepad++
- ✓ مترجم زبان C# با نام csc.exe
- ✓ الگوریتم
- ✓ برنامه .Net Framework

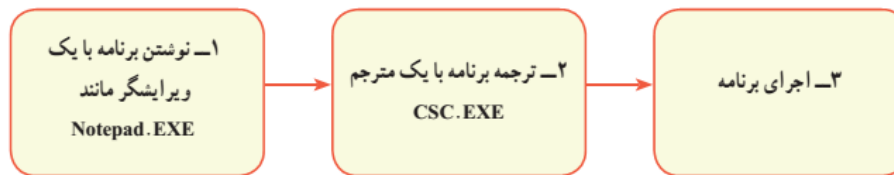
نکته: تمامی این ابزارها در ویندوز ۷ موجود است.

مراحل ایجاد یک برنامه

- ✓ نوشتن برنامه در محیط ویرایشگر و ذخیره آن با پسوند cs
- ✓ ترجمه برنامه با مترجم زبان C# با نام csc.exe بصورت زیر:

نام و پسوند فایل csc

- ✓ اجرای فایل اجرایی برنامه



کاربردهای متد WriteLine()

- ✓ نمایش پیام های متنی (رشته) روی صفحه
- ✓ نمایش نتیجه یک عبارت محاسباتی
- ✓ ایجاد یک خط خالی در خروجی

مثال: برنامه ای بنویسید که نام و نام خانوادگی شما را روی صفحه نمایش دهد

```

using System;
class Program1
{
    static void Main()
    {
        Console.WriteLine("reza");
        Console.WriteLine("ahmadi");
    }
}
  
```

مثال: برنامه ای بنویسید که با استفاده از متد WriteLine() شکلک زیر را نمایش دهد.

```

using System;
class Adamak
{
    static void Main()
    {
        Console.WriteLine("////////");
        Console.WriteLine("| O O |");
        Console.WriteLine("( | ^ | )");
        Console.WriteLine("-----");
        Console.WriteLine("|       |");
    }
}
  
```

```

////////
| o o |
( | ^ | )
| --- |
  
```

مثال: قطعه کد زیر نتیجه عبارت محاسباتی را روی صفحه نمایش می دهد.

```
System.Console.WriteLine(-12 * 2 % 8 - 3 * 2);
```

کار با رنگ ها

ویژگیها و متدهای موجود در کلاس Console برای تغییر رنگ صفحه کنسول عبارتند از:

- ✓ ویژگی BackgroundColor : رنگ زمینه کنسول را تعیین می کند.
- ✓ ویژگی ForegroundColor : رنگ نوشته را تعیین می کند.
- ✓ متد Clear() : صفحه کنسول را با رنگ انتخابی پاک می کند.
- ✓ ConsoleColor : جعبه رنگ ۱۶ تایی در C# است.

مثال: با استفاده از جعبه رنگ ConsoleColor رنگ زمینه را آبی کنید.

```

Console.BackgroundColor = ConsoleColor.Blue ;
Console.Clear() ;
  
```

مثال: کلمه IRAN را با رنگ قرمز روی صفحه بنویسید.

```
Console.ForegroundColor = ConsoleColor.Red ;  
Console.WriteLine("IRAN") ;
```

مثال: حروف کلمه HELLO را در یک خط و با رنگ‌های مختلف نمایش دهید.

```
Console.ForegroundColor = ConsoleColor.Red ;  
Console.Write ("H") ;  
Console.ForegroundColor = ConsoleColor.Blue ;  
Console.Write ("E") ;  
Console.ForegroundColor = ConsoleColor.Green ;  
Console.Write ("L") ;  
Console.ForegroundColor = ConsoleColor.Yellow ;  
Console.Write ("L") ;  
Console.ForegroundColor = ConsoleColor.Cyan ;  
Console.Write ("O") ;
```

درج توضیحات در برنامه

- ✓ توضیحات یک خطی: در ابتدای خط علامت // درج می شود.
- ✓ توضیحات چند خطی: در ابتدای توضیحات علامت /* و در انتهای توضیحات علامت */ درج می شود.