**قابلیت های Csharp 11 :**

چند قابلیت جدید به این ورژن C# اضافه شده است که به ترتیب با آنها آشنا می شویم :

1. Generic attributes :

در این قابلیت بجای ارسال پارمتری از نوع Type به سازنده‌ی Property، می‌توان کلاس آن attribute را به صورت Generic تعریف کنیم که می‌تواند یک یا چندین نوع را به عنوان پارامتر بپذیرد. یک مزیت مهم این روش نسبت به قبل، امکان تعریف قیود و type safety است، به این ترتیب می‌توان این نوع بررسی‌ها را بجای زمان اجرا و صدور استثناءها، دقیقا در زمان کامپایل انجام داد. کتابخانه مورد استفاده این قابلیت System.Attribute می باشد.

1. INumber Interface :

Interface جدیدی هست که به ورژن 11 زبان سی شارپ اضافه شده و کاربرد آن در Generic متدهایی هستند که توانایی اعمال عملیات ریاضی بر روی انواع Data Type های عددی را داراست.

کتابخانه مورد استفاده این قابلیت System.Numerics می باشد.

1. Abstracting over static Members :

C# 11 و .Net7 دارای اعضای مجازی استاتیک در interface  ها هستند.

این ویژگی به ما این قابلیت را می دهد تا interface  هایی تعریف کنیم که شامل overloaded Operator و یا سایر اعضای static باشند. هنگامی که interface ها رو با اعضای static تعریف می کنیم، می توانیم از آن interface ها به عنوان محدودیت برای ایجاد انواع عمومی که از عملگرها و سایر اعضای static استفاده می کند، نیز استفاده کنیم.حتی اگر interface هایی با overloaded Operator ایجاد نکنیم.

1. Raw string literals :

یکی از جذاب ترین ویژگی های C# 11 قابلیت سهولت کار با رشته های متنی پیچیده است. قبل از C# 11 در صورتی که داخل string خود نیاز به double quotation (") داشتیم یا اگر نیاز بود به خط بعدی بروید و یا در string interpolation نیاز داشتیم در خروجی خود دقیقا مقدار curly bracket ({}) را نمایش بدهیم، روش های پیچیده ای وجود داشت. برای مثال : در صورتی که نیاز بود عبارتی داخل جفت کوتیشن باشد باید آن عبارت را بین \” و \” قرار می دادیم و یا از @ قبل از جمله و در داخل جمله از دوتا کوتیشن جهت نمایش استفاده می کردیم. حال با استفاده از این قابلیت C# 11 تنها کافیست جمله را بین سه تا کوتیشن قرار دهیم و قسمتی از متن که داخل تک کوتیشن نمایش داده شود را به همان صورت بنویسیم.(برای درک بهتر در پروژه اعلام شده نمونه ها در کلاس RawStringLiterals پیاده سازی شده اند).

1. Allow new-lines in all Interpolations :

در C# 11 امکان رفتن به خط جدید در فرمت String InterPolation اضافه شده است. این امکان به شما کمک می کند تا کد های C# و خطوط جدید داخل { } خوانایی و فرمت بهتری داشته باشند.

1. Utf8 string literals :

سی شارپ 11 رشته های UTF-8 را معرفی کرده است که اجازه می دهد تنها کاراکتر های UTF-8 به نمایش بایت آن ها تبدیل کنید.این تبدیل در زمان کامپایل انجام می شود.

1. IntPtr and UIntPtr :

نوع های nint و nuint اکنون به ترتیب با نام مستعار System.IntPtr و System.UIntPtr در سی شارپ 11 هستند.

1. Auto-default Structs :

کامپایلر C# 11 تضمین می‌کند که تمام فیلدهای نوع struct با مقادیر پیش فرض‌شان به عنوان بخشی از اجرای constructor ، مقداردهی اولیه می شوند. این تغییر به این معنی است که هر فیلد یا Property خودکار که توسط constructor (سازنده) مقداردهی نشده است، به صورت اتوماتیک توسط کامپایلر مقداردهی اولیه می‌شود. Structهایی که constructor قطعا همه فیلدها را به آن اختصاص نمی‌دهد اکنون کامپایل می‌شوند، و هر فیلدی که به صراحت مقداردهی اولیه نشده باشد، با مقدار پیش فرض خودش تنظیم می‌شود.

1. File local types :

در سی شارپ 11 ، واژه‌ی کلیدی جدی file، جهت تعریف نوع‌های محدود به یک فایل «File Scoped Types» ارائه شده‌است. این واژه‌ی کلیدی را می‌توان به تعریف هر نوع دلخواهی مانند class, interface, record, struct, enum, delegate اضافه کرد (منهای خواص، فیلدها و رخدادها؛ البته تاسی شارپ 11)

تا آن نوع، دیگر در سایر کلاس‌های فایل‌های برنامه، قابل دسترسی نباشد و سطح دید استفاده‌ی از آن، تنها محدود به فایل جاری محل قرار گیری آن شود. به این ترتیب می‌توان در یک فضای نام مشخص، چندین کلاس هم‌نام را تعریف کرد؛ کاری که در نگارش‌های پیشین C#، میسر نبود. بدیهی دیگر نمی‌توان یک چنین نوع‌هایی را با سطوح دسترسی متداول internal و یا  public، تعریف و ترکیب کرد.

1. List patterns :

List Pattern ها در سی شارپ 11 به توسعه دهندگان و برنامه نویس ها اجازه می دهند تا تطابق میان لیست ها و آرایه ها را به شکل راحت تری انجام دهند. با استفاده از الگوهای قطعه ای توسعه دهندگان می توانند در لیست های صفر و یک عناصر صفر را کنار گذاشته یا آنها را ذخیره سازی کنند.  برای مثال اگر نیاز است بررسی کنیم که آیا مجموعه‌ای با یک مقدار خاص، شروع می‌شود، پایان می‌یابد و یا حاوی آن است، List Patterns مفید واقع خواهند شد.

1. Required Member :

**در C# 11 یک ویژگی جدید به نام Required بعد از سطح دسترسی به فیلد ها و** Property **ها اضافه شده است که با استفاده از آن می توانید سازنده ها و کسانی که از آن شی نمونه سازی می کنند را وادار به مقدار دهی اولیه کنید.**

جهت مشاهده و استفاده از نمونه کدها و مثال ها، از پروژه زیر استفاده نمایید.

<https://github.com/BaharehSmaili/NewCsharp11.git>