**Singleton** : این امکان را به ما می دهد که مطمئن شویم که در کل طور عمر برنامه از یک کلاس فقط یک instance، آن هم در اولین فراخوانی ساخته می شود. و بعد از آن هر موقع کلاس موردنظر فراخوانی شود همان شی که در ابتدا ساخته شده در اختیار consumer قرار داده می شود. مثلا در asp.net core وقتی از سرویس AddDbContext استفاده می کنیم، کلاس Context مربوط به بانک اطلاعاتی را به صورت Singleton ایجاد می کنیم. یا به عنوان مثال وقتی اطلاعاتی را به کاربران می خواهیم نشان دهیم که به ازای هر کاربر فرقی در نمایش اطلاعات وجود ندارد و اطلاعات برای همه کاربران یکسان است، پس چرا باید به ازای هر کاربر یک نمونه از کلاس نمایش دهنده اطلاعات ایجاد و حافظه را اشغال کرد.

نکته : یکی از دلایلی که ممکن است نمونه سازی یک class را singleton کنیم این است که ساختن نمونه از آن کلاس بسیار زمان بر است.

**نکته مهم** : چون کلاس های singleton از نوع static هستند، برای garbage collector در اولویت پاک کردن نیستند تا زمانی که مجبور شود، پس اگر احتمال می دهید که کلاستان زیاد new نمی شود (ساختن نمونه جدید از کلاس موجب اشغال حافظه و صرف زمان برای ساخت نمونه می شود) از روش singleton استفاده نکنید.

ویژگی های کلاس singleton :

* متد constructor باید به صورت private باشد تا کاربر نتواند هربار که به یک instance از کلاس احتیاج دارد، با استفاده از کلمه کلیدی new آن را ایجاد کند.
* instance ایجاد شده باید به صورت private باشد و تنها با یک متد static بتوان از آن استفاده کرد.
* متدی که امکان دسترسی به instance را فراهم می‌کند باید به صورت public و static باشد. و new کردن کلاس داخل این متد باید به صورت thread safe باشد.

نکته : همچنین می تواند یک کلاس generic ایجاد که هر کلاس را به عنوان T به آن ارسال کردیم، کلاس را به صورت singleton نمونه سازی کند.