Sealed Class

یک کلاس sealed نمیتواند ارث بری شود (این بدان معنی است که این نوع از کلاسها به عنوان کلاس پایه مورد استفاده قرار نمیگیرند). این کلاس مانع ارث بری سایر کلاسها از خودش میشود. بنابراین کلاسهای دیگر نمیتوانند از کلاسی که saeled تعریف شده است ، ارث بری کنند.

در زبان برنامه‌نویسی #C، کلاس‌ها به عنوان بلوک‌های ساختاری برای تعریف اشیاء و رفتارهای آن‌ها استفاده می‌شوند. یکی از ویژگی‌های مهم در طراحی کلاس‌ها، قابلیت وراثت **(Inheritance)** است که به برنامه‌نویسان اجازه می‌دهد تا کلاس‌های جدیدی را بر اساس کلاس‌های موجود ایجاد کنند. با این حال، در برخی موارد ممکن است بخواهیم از وراثت یک کلاس جلوگیری کنیم. در اینجا کلاس‌های \*\*sealed\*\* به کار می‌آیند.

**تعریف کلاس Sealed**

کلاس sealed به کلاسی اطلاق می‌شود که نمی‌توان از آن وراثت گرفت. به عبارت دیگر، اگر یک کلاس به عنوان sealed تعریف شده باشد، هیچ کلاسی نمی‌تواند آن را به ارث ببرد. این ویژگی به برنامه‌نویسان این امکان را می‌دهد که از تغییرات ناخواسته در رفتار کلاس‌های خود جلوگیری کنند و امنیت و یکپارچگی کد را افزایش دهند.

**مزایای استفاده از کلاس Sealed**

1**). افزایش امنیت(:** با جلوگیری از وراثت، می‌توان از تغییرات ناخواسته در رفتار کلاس جلوگیری کرد.

2**). بهینه‌سازی عملکرد(:** کامپایلر می‌تواند بهینه‌سازی‌های خاصی را برای کلاس‌های sealed انجام دهد، زیرا می‌داند که هیچ کلاسی از آن‌ها ارث نمی‌برد.

3**). ساده‌سازی طراحی(:** استفاده از کلاس‌های sealed می‌تواند طراحی کد را ساده‌تر کند و از پیچیدگی‌های اضافی جلوگیری کند.

**معایب استفاده از کلاس Sealed**

1**). عدم انعطاف‌پذیری(:** اگر یک کلاس sealed باشد، نمی‌توان آن را گسترش داد. این ممکن است در برخی موارد محدودیت ایجاد کند.

2**. )کاهش قابلیت استفاده مجدد(:** اگر بخواهید از یک کلاس sealed در پروژه‌های مختلف استفاده کنید، نمی‌توانید آن را به راحتی گسترش دهید.

**نتیجه‌گیری**

کلاس‌های sealed در #C ابزاری مفید برای کنترل وراثت و افزایش امنیت و یکپارچگی کد هستند. با این حال، باید با احتیاط از آن‌ها استفاده کرد، زیرا ممکن است انعطاف‌پذیری و قابلیت استفاده مجدد کد را کاهش دهند. در نهایت، انتخاب بین استفاده از کلاس sealed یا کلاس‌های معمولی بستگی به نیازهای خاص پروژه و طراحی نرم‌افزار دارد.