

در برنامه‌نویسی (Sealed Class) شده Seal بررسی کلاس‌های

### مقدمه ###

، به حساب Kotlin و C# ابزاری قدرتمند در برخی زبان‌های برنامه‌نویسی، به ویژه در (Sealed Class) شده Seal کلاس‌های می‌آیند. این کلاس‌ها به توسعه‌دهندگان این امکان را می‌دهند که کنترل بیشتری بر وراثت و ارث‌بری داشته باشند.

### ویژگی‌ها ###

1. شده ارث‌بری کنند. این ویژگی باعث می‌شود که کد شما Seal محدودیت وراثت \*\*: تنها کلاس‌های خاصی می‌توانند از کلاس \*\* 1. منظم‌تر و قابل کنترل‌تر باشد.

2. پایداری \*\*: با محدود کردن وراثت، از تغییرات ناخواسته در کلاس‌های فرزند جلوگیری می‌کند و به این ترتیب سازگاری و \*\* 2. پایداری کد را حفظ می‌کند.

### مزایا ###

کاهش پیچیدگی \*\*: با محدود کردن وراثت، کد کمتر پیچیده می‌شود و تعاملات بین کلاس‌ها واضح‌تر است \*\* -

شده ارث‌بری کنند Seal مدیریت بهتر \*\*: توسعه‌دهندگان می‌توانند به راحتی مدیریت کنند که کدام کلاس‌ها می‌توانند از کلاس \*\* -

شده تغییراتی داشته باشد، تاثیر آن بر کلاس‌های فرزند قابل Seal سازگاری بهتر با تغییرات \*\*: بنابراین وقتی که کلاسی \*\* - پیش‌بینی‌تر است.

### نمونه کد (C#) ###

```
public sealed class SealedBaseClass
```

```
{  
    public void Display()  
    {  
        Console.WriteLine("This is a sealed class.");  
    }  
}
```

```
// ارث‌بری کند و خطا خواهد داد SealedBaseClass کلاس زیر نمی‌تواند از //
```

```
// public class DerivedClass : SealedBaseClass
```

```
// {
```

// }

### زبان‌های مختلف ###

بهر راحتی تعریف می‌شوند و کاربردهای فراوانی دارند C# شده در Seal کلاس‌های \*\*:C#\* -

شده ایجاد کرد که این قابلیت مخصوصاً Seal کلاس‌های keyword sealed نیز می‌توان با استفاده از Kotlin در \*\*:Kotlin\* -  
های داده و درخت‌های نمایشی کارآمد است Klasse در هنگام استفاده از

### نتیجه‌گیری ###

شده یک ابزار مفید برای حفظ نظم و کنترل در کدنویسی هستند. با استفاده از این ویژگی‌ها، توسعه‌دهندگان می‌توانند Seal کلاس‌های  
کدهای تمیزتر و قابل پیش‌بینی‌تری بنویسند که از وفور خطاها و مشکلات ناشی از وراثت ناخواسته جلوگیری می‌کند