

C#AssignmentDay09Part02


1- LinkedIn article about class types ?

23:23

📶 3%

←

⋮



Mahmoud Elsisy • You
Full stack .Net Web Developer Trainee @ Digital Eg...
now • 🌐

ال Class في #C هي البنية الأساسية لأي أليكشن (Object-Oriented). هي عبارة عن ال Blueprint التي يحدد ال Properties وال Methods التي يمكن ال Objects المبنية منه تستخدمها.

إيه هو ال Class بالضبط؟

ال Class عبارة عن نموذج يوصف ال data وال methods مع بعض. بمعنى ثاني، لو عايزين نعمل نموذج ل printer، ممكن نعرف Class باسم Printer فيها properties زي Brand و Model. و methods زي Print () و Scan (). ومن خلالها نقدر create أي عدد من ال objects التي تمثل printers فعلية.

أنواع ال Classes في #C

1. Concrete Class

ده النوع اللي تقدر تعمل منه Objects مباشرة. بيحتوي على methods و properties جاهزة للاستخدام.

مثال:

```
class Car
{
    public string Brand;
    public void Drive() { Console.WriteLine("Car is driving"); }
}
```

هنا نقدر نعمل object :


```
Car myCar = new Car();
```

2. Abstract Class

ده نوع من الكلاسات مينفعش نعمل منه Object مباشرة. بيحتوي على Abstract Methods لازم أي Class يورث منه يطبقها. ودوال عادية ممكن تستخدمها الكلاسات الوراثة منه.

مثال:

```
abstract class Animal
{
    public abstract void MakeSound();
}
```



Add a comment...

23:23

📶 3%

←

⋮

```
public abstract void MakeSound();
public void Eat() { Console.WriteLine("Eating"); }
}
```

أي Class يورث من Animal لازم يعرف MakeSound().

3. Sealed Class

ده كلاس مينفعش أي Class ثاني يورث منه. غالبا بنستخدمه لو عايزين نحمي الكود من التعديل عن طريق الوراثة.

مثال:

```
sealed class Utility
{
    public void DoWork()
    { Console.WriteLine("Working"); }
}
```

أي محاولة لعمل Utility class هندي Error.

4. Static Class

كلاس كل عناصره Static، مينفعش نعمل منه Objects. بنستخدمه لو عايزين مجموعة methods ثابتة أو Constants.

مثال:

```
static class MathHelper
{
    public static int Add(int a, int b)
    { return a + b; }
}
```

نقدر نستخدمه مباشرة:

```
int sum = MathHelper.Add(5, 3);
```

5. Partial Class


ده نوع من الكلاسات بيتقسم على files مختلفة. مفيد لو الشغل على project كبير أو لو الكود بيتولد تلقائي (مثال: Windows Forms).

مثال:

```
partial class Person
{
    public string Name;
}
```

وفي file ثاني:

```
partial class Person
{
    // ...
}
```



Add a comment...

23:23

📶 3%

←

⋮


وفي file ثاني:


```
partial class Person
{
    public void Greet() { Console.WriteLine("Hello " + Name); }
}
```

الخلاصة:

ال Classes هي الأساس لأي أليكشن. Object-Oriented اختيار النوع الصح من الكلاس بيخليك تتحكم في ال Inheritance، Reusability، Code Protection بطريقة أفضل.

Show translation






Concrete Class

Regular class that can be instantiated.

Example:

```
class Car { public void Drive() ... }
```




Abstract Class

Cannot be instantiated, used for inheritance.

Example:

```
abstract class Animal {
    public abstract void MakeSound();
}
```




Sealed Class

Cannot be inherited or extended.

Example:

```
sealed class Utility {
    public void DoWork() (...);
}
```




Static Class


Contains only static members.

Example:

```
static class MathHelper {
    public static int Add(int a, int b) (...);
}
```



Partial Class



Add a comment...

2) What we mean by Generalization concept using Generics ?

Generalization means creating one reusable structure that works for multiple types instead of writing separate code for each type.

How Generics Achieve Generalization?

- Instead of creating multiple versions of a class or method for int, string, double, etc., we write one generic class or method.
- The type parameter T acts as a placeholder for any type.

Benefits of Generalization:

1. Code Reusability → Write once, use for any type.
2. Type Safety → Compiler checks types; no runtime casting.
3. Maintainability → Easier to update one generic implementation.
4. Flexibility → Works with value types, reference types, or user-defined types.

3)What we mean by hierarchy design in real business ?

Hierarchy design means organizing entities into parent-child relationships to reflect “is-a” relationships or levels of specialization.

What It Looks Like in Business?

1. Employees Example
 - Employee → base/general class
 - Manager → inherits from Employee
 - Developer → inherits from Employee

- Intern → inherits from Employee

Benefits of Hierarchy Design:

1. **Reflects Real-world Structure** → Easy for business to relate objects.
2. **Code Reusability** → Common functionality in parent class.
3. **Polymorphism** → Use child objects via parent references.
4. **Maintainability & Extensibility** → Add new specialized types without changing existing code.