

C# Assignment Day08 Part02

1- LinkedIn article about abstract class ?



Mahmoud Elsyisy · You
Full stack .Net Web Developer Trainee @ Digital Egypt now · 17:05

لو انت بتذاكر OOP ، أكيد سمعت عن الـ Abstract Class . فيه ناس بتشوفه مجرد "كود ناقص" أو حاجة ملهاش لازمة طالما مش يعترض ناخد منها Object . بس الحقيقة الـ Abstract Class هو اللي بيحط "قواعد اللعبة" لاي سистем احترافي . طيب ايه هو الـ Abstract Class ؟ وليه هو حلقة الوصل المهمة في الـ Software Architecture . يعني ايه؟

بساطة، الـ Abstract Class هو Class غير مكتمل. هو عبارة عن methods بيتحط في الـ Abstract Class ده نفسه . مقدرش أعمل منه instantiation new() (يعني مقدرش أقول (MyAbstractClass)).

الهدف منه مش إنه يستغل لوحده، الهدف إنه يكون Base Class ثانية تورث منه Classes .

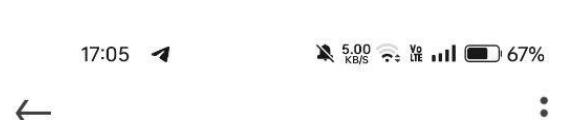
المكونات الأساسية للـ Abstract Class بيتميز بميكس عبقي بين حاجتين:

1) Abstract Methods: دى (Methods) بيتفق عبارة عن "اسم" و "Signature" بس، من غير أي كود داخلى (No Body) . إنت هنا بتقول للـ Subclasses: "لازم تنفذوا الـ Method دي، بس كل واحد بطريقته".

2) Concrete Methods: ودى عادي فيه كود جاهز. أي هيورث من الـ Abstract Class هياخد الكود ده ويستخدمه على طول، وده بتحقق مبدأ الـ Code Reusability .

مثال من الـ Business نفسه عشان الدنيا توضح: تخيل إنك بتعمل سيسنتم لشركة دفع إلكتروني. عندك أنواع كتير من الـ Payment Methods (فوري، كارت ائتمان، محفظة إلكترونية). كلهم مشتركين في حاجات (يـ الـ Amount) (بس طريقة الـ ProcessPayment() بختلف من واحد للثاني). هنا بتعمل

Add a comment...



:Payment اسمه Abstract Class
* بحط فيه الـ Amount كمتغير عادي.
* بعمل Abstract Method اسمها processPayment()
* أي Class جديـ (FawryPayment) لـ اـOverride (ويكتب الـ Logic الخاص بيـ) .

(The Power of Abstraction)
* Enforcing a Contract:
بتجرـ أي developer هيورـ منك إنه يتلزم بـ معين .
* Avoiding Code Redundancy:
الـ code المشـرك بيـكتب مرة واحدة بـ في الـ Abstract Class .
* Flexibility:
بيـسهل عليك التعـامل مع الـ Objects المختلفة باـستخدام الـ Polymorphism .

ملخص سريع:
الـ Abstract Class هو اللي بيـقولك "إيه اللي المفروض الـ عملـ، وبيـسيـب لكلـ Object الحرية يـحدـ هـيـعملـ إـيـ". مع توـفـير شـوـية كـود مشـترك عـشـان مـيـتعـبـكـشـ .

Show translation

Understanding Abstract Classes in C#

What is an Abstract Class?

- Cannot be instantiated.
- Has abstract methods and implemented methods

When to Use It:

- Shared Base Functionality
- Enforce Method Implementation
- Template Design Pattern

Code Example:

```
abstract class Shape
{
    public abstract double CalculateArea();
    public void Display()
    {
        Console.WriteLine("Displaying shape info");
    }
}

class Rectangle : Shape
{
    public override double CalculateArea()
    {
        return Width * Height;
    }
}
```

Output: Area = 50

Add a comment...

2- What we mean by coding against interface rather than class? and if u get it so What we mean by code against abstraction not concreteness?

Coding Against an Interface vs Coding Against a Class

Coding against an interface means:

Writing your code to depend on an interface (or abstract contract) rather than a specific concrete class.

Coding Against Abstraction, Not Concreteness

Abstraction => an interface or abstract class that defines behavior but hides implementation.

Concreteness => a specific class with exact implementation.

Coding against abstraction means:

Your code depends on “**what it should do**”, not “**how it does it**”.

3- What is abstraction as a guideline and how we can implement this through what we have studied?

Abstraction is:

The practice of hiding complex implementation details and exposing only the essential functionality to the user.

It is a design guideline that tells us:

- Focus on what an object does, not how it does it.
- Expose only necessary operations.
- Hide unnecessary details and internal complexity.