

Part02

يعني إيه تشيل "دالة" جوة "متغير"؟ وإيه هو ال Delegate؟

تخيل إن عندك (Variable) عادي جداً، بس بدل ما تشيل جواه قيمة زي رقم أو نص، إنت بتشيل جواه وظيفة (Function). يعني ال Delegate باختصار هو (Pointer) بيمسك الدالة دي، وتقدر تبعته من مكان للتاني في الكود بتاعك بسهولة.

طب إيه الفائدة؟ وليه بنستخدمه؟

1. تمرير الدوال كأنها بيانات (Higher-Order Functions):

تخيل إنك بتعمل دالة بتفلتر بيانات (مثلاً بتجيب الموظفين اللي مرتبهم فوق رقم معين). بدل تكتب دالة جديدة لكل شرط، إنت بتعمل دالة واحدة عامة، وبتبعتها الشرط نفسه ك parameter عن طريق ال Delegate. ده بيققل تكرار الكود ويحقق مبدأ (DRY) بشكل كبير جداً.

2. مرونة عالية (Flexibility):

ال Delegates بتخلي الكود بتاعك ديناميكي؛ لأنك بتقدر تحدد ال Logic اللي هيتنفذ وقت التشغيل (Runtime) مش وقت كتابة الكود (Compile-time).

3. ال Callbacks وال Events:

لو بتعمل Process بتاخذ وقت، وعازب أول ما تخلص تنادي على دالة تانية تقولها "أنا خلصت" Delegates هي اللي بتعمل الربط والسحر ده.

أنواع ال Built-in Delegates:

عشان "مايكروسوفت" تريحنا وتخلي تطبيق ال Functional Paradigm أسهل، وفرتلنا 3 أنواع جاهزة بنستخدمهم كل يوم من غير ما نحس (خصوصاً مع ال LINQ):

Action: بياخد منك دالة بتنفذ أمر معين ومش بترجع أي نتيجة (void).

Func: بياخد منك دالة بتنفذ أمر، ولازم ترجع نتيجة (Return Value).

Predicate: بياخد منك دالة بتعمل "اختبار" أو "شرط" معين، ولازم ترجع Boolean (يعني true/false).

في النهاية، استخدامك للـ Delegates مش مجرد "منظرة" في الكود، ده بيبغير طريقة تفكيرك وبيخليك تكتب (Clean Code)، سهل يتعدل عليه (Maintainable)، والأهم إنه مش بيكرر نفسه.

#SoftwareEngineering #CSharp #DotNet #FunctionalProgramming #Delegates #CleanCode #ProgrammingTips



2-Search about these topics (Parallel Programming and Concurrency - Unit Testing and Test-Driven Development (TDD) - Asynchronous Programming with async and await)?

-Parallel Programming and Concurrency

Concurrency refers to structuring a program to handle multiple tasks at the same time, while

parallel programming executes multiple tasks simultaneously on multiple CPU cores. Concurrency improves responsiveness, whereas parallelism improves performance.

-Unit Testing and Test-Driven Development (TDD)

Unit testing verifies individual units of code to ensure correctness. **Test-Driven Development (TDD)** is a development approach where tests are written before the actual code, leading to better design, fewer bugs, and more maintainable software.

- Asynchronous Programming with async and await

Asynchronous programming allows long-running operations to execute without blocking the main thread. Using **async and await** improves application responsiveness and scalability, especially in I/O-bound operations such as web requests and database calls.