

Part02

يعني إيه تشيل "دالة" جواه "متغير"؟ وإيه هو الـDelegate؟

تحيل إن عندك (Variable) عادي جداً، بس بدل ما تشيل جواه قيمة زي رقم أو نص، إنت بت Shirley جواه وظيفة (Function). يعني الـDelegate باختصار هو (Pointer) بيمسك الدالة دي، وتقدر تبعتها من مكان للثاني في الكود بقى سهولة.

طب إيه الفايدة؟ ولية يستخدمه؟

1. تمرير الدوال لأنها بيانات (Higher-Order Functions):
تحيل إنك بتعملي دالة بتفلتر بيانات (مثلاً بتجيب الموظفين اللي مرتبهم فوق رقم معين). بدل تكتب دالة جديدة لكل شرط، إنت بتعملي دالة واحدة عامة، وتبعتلها الشرط نفسه كـParameter عن طريق الـDelegate. ده بيقلل تكرار الكود ويحقق مبدأ (DRY) بشكل كبير جداً.

2. مرونة عالية (Flexibility):
الـDelegates بتخلي الكود بقى ديناميكي؛ لأنك بتقدر تحدد الـLogic اللي هيتنفذ وقت التشغيل (Runtime) مش وقت كتابة الكود (Compile-time).

3. الـEvents والـCallbacks:
لو بتعملي Process بتاخذ وقت، وعابيز أول ما تخلص تنادي على دالة تانية تقولها "أنا حلست".
الـDelegates هي اللي بتعملي الربط والسحر ده.

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

Action: بيأخذ منك دالة بتتنفذ أمر معين ومش بترجع أي نتيجة (void).
Func: بيأخذ منك دالة بتتنفذ أمر، ولازم ترجع نتيجة (Return Value).
Predicate: بيأخذ منك دالة بتعملي "اختبار" أو "شرط" معين، ولازم ترجع Boolean (يعني true/false).

في النهاية، استخدامك لـ Delegates مش مجرد "منظرة" في الكود، ده بيغير طريقة تفكيرك
وبيخليلك تكتب (Clean Code)، سهل يتعذر 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.