

Mohamed Nagy • You
Full Stack Developer | ASP.NET Core
now •

...

ال Boxing & Unboxing
لما ال بيجي على حساب ال Performance
في Software Design، دائمًا بنصي إن الكود يكون Reusable و Flexible. لكن أحيانًا بنقع في
فخ استخدام النوع object علشان الميثود تستقبل أي نوع من البيانات.
هنا بيظهر مفهوم Boxing & Unboxing.

لما بتضمم Method بتاخد Parameter من نوع Value Types وأنت ناوي تبعتها زي int أو double، يعني بيعمل Boxing، يعني بيحول ال Object ل Value Type ويتحط في ال Heap Memory

الموضوع ده مش بس بيأثر على ال Performance بسبب ال Memory Allocation
لكنه كمان بيكسر مبدأ مهم من مبادئ ال Type Safety وهو Clean Code

التصميم الأنضف والأصح هو استخدام Generics

ال Compiler يتخليك تحقق المرونة (Reusability) من غير ما تضحي بالسرعة، لأن ال Generics
بيتعامل مع النوع الأصلي مباشرة، وده يمنع ال Boxing ويحافظ على ال Type Safety

الخلاصة إن SOLID Principles و Clean Architecture مش بس شكل كود حلو وتنظيم ملفات،
لكن كمان اختيار صحيح لل Data Types وال Design Patterns اللي يحافظ على موارد السيرفر
وتحريك أفضل أداء.

بلاش تستخدم object بداع الكسل، واستخدم Generics علشان كودك يبقى Clean, Safe, and Fast

SoftwareArchitecture #CleanCode #DotNet #CSharp #Performance #Boxing#
#Generics #SoftwareEngineering

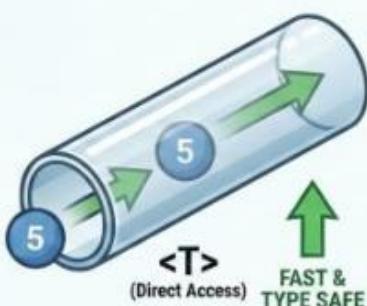
1-

THE PERFORMANCE TRAP (Boxing)



THE CLEAN SOLUTION (Generics)

BOXING & UNBOXING: When “Design” Costs Performance



```
public void ProcessData(object data) { ... }  
  
int x = 5;  
ProcessData(x); // BOXING HAPPENED!
```

```
public void ProcessData<T>(T data) { ... }  
  
int x = 5;  
ProcessData<T>(x); // NO BOXING!
```

2-Search about upcasting and downcasting and its relation to inheritance concept as Design wise?

- a Child object as a Parent (Safe & Automatic). Used for flexibility
- a Parent object as a Child (Manual & Risky). Needs type checking

3-Unmanaged Resources Recovery & I/O Operations?

- Unmanaged resources must be released manually to avoid memory leaks.
- Using statement ensures safe resource cleanup OR IDisposable.

4-Custom Exceptions?

Custom exceptions help separate business errors from system errors and improve maintainability.

5-Global Middleware Try-Catch?

- Global middleware provides a centralized and consistent way to handle errors across the application.