

Part02

ليه بنقول على الـ Strings إنها "Immutable"؟

لو بتتعلم C#, أكيد سمعت جملة "Strings are immutable"، بس يا ترى سألت نفسك: ليه؟ وإيه اللي بيحصل فعلياً في الميموري؟

تعال نكش الشفرة دي في نقاط سريعة عشان تفرق معاك في الـ Performance والـ Design.

يعني إيه Immutability أصلاً؟

ببساطة، الـ String في الـ .NET عامل زي "الوثيقة المختومة". بمجرد ما يتخلق في الـ Heap، مستحيل محتواه يتعدل.

لو حاولت تغيير حرف واحد بس، الـ CLR مش بيعدل الأصل.. هو بياخد نسخة جديدة، يطبق التعديل، ويحجز مكان جديد في الميموري.

إيه اللي بيحصل في الكواليس؟

تحيل أنك كتبت الكود ده:

```
string text = "Hello";
text += ", World";
```

هنا إنت متحيل إنك عدلت text، لكن الحقيقة:

-الـ C# خلقت String جديد تماماً قيمته "Hello, World".

-وجهت المتغير text عشان يشارو عليه.

1- الـ String القديم "Hello" لسه مكانه فيه، الميموري لحد ما الـ Garbage Collector بيشله.

ليه مايكروسوفت صممتها كذا؟

الموضوع مش رخامة، ده عشان هدفين مهمين:

-Thread Safety: بما إن الـ String قيمته ثابتة، فأني عدد من الـ Threads يقدرنا يقرأوا نفس الـ Object في نفس الوقت من غير خوف إن حد يغيره فجأة (Race Conditions).

-Security: تخيل لو الـ String كان Mutable، كان ممكن حد يعدل في الـ Connection String أو الـ Password بعد ما تعملهم Validation! الـ Immutability بتمنع الثغرات دي.

إمتى الموضوع ده يبقى "فخ" للأداء؟ (Performance Trap)

لو عندك Loop بتلف 1000 مرة وتعمل String Concatenation (+)، إنت حرفياً بترمي 1000 Object في الميموري وتحلي الـ GC يشتغل وقت إضافي!

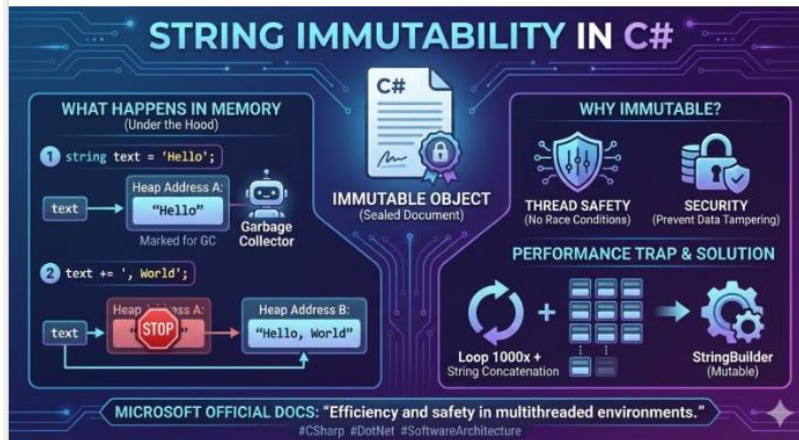
الحل:

في حالات التعديل الكثير، استخدم الـ StringBuilder. هو ده اللي صمموه عشان يكون (Mutable) ويسمح بالتعديل المباشر من غير ما يحجز أماكن جديدة كل مرة.

المصدر:

حسب توثيق مايكروسوفت (Official Docs)، التصميم ده معمول لضمان: "Efficiency and safety in multithreaded environments".

#CSharp #DotNet #SoftwareArchitecture #CodingTips #Performance#



2- What's Enum data type, when is it used? And name three common built-in enums used frequently?

-Enum is a data type that defines a set of named constants.

-Use it when you have a limited set of possible values (like days, months, states).

3- what are scenarios to use string Vs StringBuilder?

-Use String when you have a constant piece of text or when the number of modifications is very small. It is simpler and safer for sharing across different parts of your program.

-Use StringBuilder when you are performing heavy text manipulation, especially inside loops or when concatenating a large number of strings. It significantly boosts performance and reduces memory overhead by modifying the existing object instead of creating new ones.

Part03 Bonus

1-String Handling & Formatting? (Composite formatting- String interpolation)

-Composite Formatting is best used when your string templates are stored externally (like in resource files for translation) or when the format string is determined at runtime. It uses indexed placeholders like {0}.

-String Interpolation (\$) is the modern standard for most coding scenarios. It makes the code cleaner and more maintainable by allowing you to embed variables directly into the string, such as {variableName}. Use it whenever you want to make your code more readable and intuitive.

2-.rdata Section?

-The .rdata section (Read-only Data) is a segment of a PE (Portable Executable) file used to store initialized constants and literal strings that should not be modified during program execution.

3-If/Switch (how the jump happen in switch - Evolution in C# Versions - where upperhand to c#)?

- If statement: Checks conditions sequentially, executes the first true block.
- Switch statement: Jumps to matching case (or default) using **jump table or sequential comparison**.

-Evolution

- Pre-C#7 → basic switch (int, string, char).
- C#7+ → pattern matching with when

```
object obj = 10;

switch (obj)
{
    case int i when i > 0:
        Console.WriteLine("Positive int");
        break;
    case string s:
        Console.WriteLine("It's a string");
        break;
}
```

- C# 8.0+: Switch Expressions (modern style):

```
int day = 3;
string dayName = day switch
{
    1 => "Sunday",
    2 => "Monday",
    3 => "Tuesday",
    _ => "Invalid day"
};
```

- Upperhand:

-If → flexible for boolean logic.

-Switch → cleaner and potentially faster for multiple discrete values; modern C# adds powerful pattern matching and expressions.

4-what meant by user defined constructor and its role in initialization?

- A **user-defined constructor** is explicitly written by the programmer to initialize an object.

- Its main role is to ensure the object starts with **meaningful and safe values**, avoiding uninitialized or garbage data.

5-compare between Array and Linked List?

Array: fixed size, fast random access, slow insertion/deletion except at the end, contiguous memory.

Linked List: dynamic size, slower access by index, fast insertion/deletion at known nodes, uses extra memory for pointers, non-contiguous memory.