Part02

Mohamed Nagy • You
Full Stack Developer | ASP.NET Core
1h •

حد سأل نفسه قبل كدا.. (Variables) بتتخزن فين؟

عارفين الـ Stack والـ Heap اللي بنسمع عنهم دايما في الـ Memory Management؟ تعالوا نعرف هما بيخزنوا الـVariables ازاى جواهم.

تخيل الميموري بتاعة جهازك دي عبارة عن حاجتين:
جيبك (Stack): ده مكان صغير، سريع جدا، ومنظم أي حاجة فيه تحت إيدك علطول.
مخزن كبير (Heap): ده مكان واسع قوي، بس بعيد شوية، وبنحط فيه الحاجات التقيلة.

طيب المتغيرات بتاعتنا بتروح فين؟

الـ Value Types : زي الـ int والـ bool والارقام الصغيرة. دول عاملين زي "الفكة" أو "المفاتيح" حجمهم صغير ومهمين، فبنحطهم في جيبك (Stack) عشان السرعة.. تخلص منهم ترميهم علطول.

الـ Reference Types : زي الـ Classes والـ Strings والـ Objects. دول عاملين زي "عفش البيت" أو "عربية".. ماينفعش تحطهم في جيبك! فبنعمل إيه؟ بنحط العفش نفسه في المخزن (Heap). وبنحط "عنوان المخزن" بس في ورقة صغيرة جوه جيبك (Stack).

مين بينضف لما نخلص؟
(Garbage Collector): الـ Stack (جيبك) بينضف نفسه بنفسه أول ما تخلص (Function ends).
لكن الـ Heap (المخزن) عشان كبير ومكركب، بيحتاج "عامل نظافة" اسمه Garbage Collector (GC) يعدي كل شوية يلم الحاجات اللي مبقاش ليها لازمة عشان الميموري ماتتملاش.

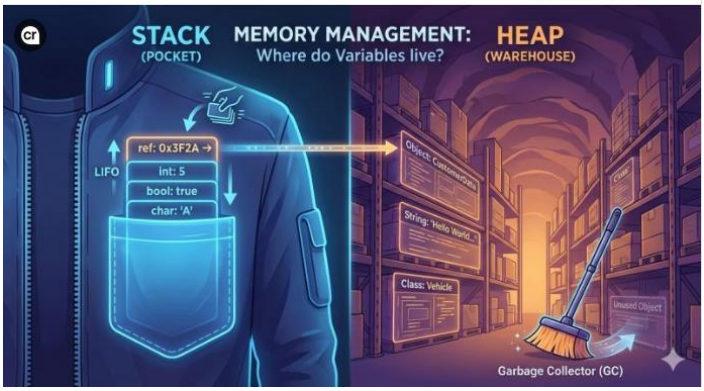
الخلاصة:

الـ Stack بيشيل الحاجة وقيمتها جواه (للخفيف).

الـ Heap بيشيل الحاجة التقيلة، والـ Stack بيشاور عليها بس.

1-

الموضوع بسيط.. بس فهمه بيفرق جداً في أداء الكود بتاعك وإنك تتجنب مشاكل الميموري.
SoftwareEngineering #MemoryManagement #StackVsHeap #Backend#
#ProgrammingTips



2- what's the difference between compiled and interpreted languages and in this way what about Csharp?

Compiled languages are translated completely into machine code before execution, which gives high performance but requires recompilation after changes.

Interpreted languages execute code line by line at runtime, making them more flexible but slower.

C# is considered a hybrid language because it is first compiled into Intermediate Language (IL), then converted to machine code at runtime using JIT compilation inside the CLR.

3- Compare between implicit, explicit, Convert and parse casting?

1-Implicit casting is an automatic conversion from a smaller type to a larger type without data loss.
2-Explicit casting is a manual conversion from a larger type to a smaller type and may cause data loss.
3-Convert is used to convert values between different data types and can handle null values but may throw exceptions if conversion fails.
4-Parse converts a string to a numeric type, does not accept null, and throws exceptions if the value is invalid.

Part03 Bonus

1-Creating objects (Reference Types)?

-Reference types in C# are stored in the heap, and variables hold references to objects rather than the actual values.
-Objects are created using the new keyword and can be initialized directly or through constructors.
-When assigning a reference type, only the reference is copied, so changes affect all references pointing to the same object.
-Memory management is handled automatically by the Garbage Collector.

2-Memory leak?

-A memory leak in C# happens when an application keeps references to objects that are no longer needed, which prevents the Garbage Collector from freeing memory.
-The issue is not the Garbage Collector itself, but leftover references such as static variables, event subscriptions, or undisposed objects.

3-Garbage Collector?

-The Garbage Collector manages memory in C# by removing objects that are no longer referenced.
-It works automatically and uses generations to optimize performance.

4-Customizing Garbage Collector?

-The Garbage Collector cannot be directly controlled, but its performance can be improved by releasing references, using Dispose, and avoiding unnecessary static objects.

5-Unmanaged Resources?

-Unmanaged resources are not handled by the Garbage Collector and must be released manually using IDisposable or a using block.

6-Scientific Notation / FPU (float & double)?

-float and double use scientific notation and rely on the FPU, which makes their results approximate.

7-When and Why to use decimal?

-decimal is used when high precision is required, such as financial calculations, because it avoids rounding errors.

8-Difference between float, double, and decimal?

-float, double, and decimal differ in precision, performance, and intended use cases.

9-Checked & Unchecked block?

-The checked block detects overflow and throws an exception, while unchecked ignores it.

10-Parsing null values?

-Parse does not handle null values, while Convert safely converts null to a default value.

11-Parse vs Convert (Performance)?

-Parse is faster than Convert but less safe, while Convert is slightly slower and handles null values safely.

12- what meant by Csharp is managed code?

-C# is considered managed code because it runs under the control of the CLR, which handles memory management, security, and exception handling.

13- what meant by struct is considered like class before?

-A struct looks similar to a class in syntax but behaves differently. Structs are value types, while classes are reference types, which affects memory allocation and copying behavior.