

Boxing and Unboxing

Wednesday, May 22, 2024 12:03 PM

Boxing and unboxing lets you convert between value and reference types.

- Boxing is the process of converting a value type into an object type
- Unboxing involves manually extracting a value type from an object type

★ While performing boxing the value type is converted into an object, that is managed using managed heap and on the other hand, Unboxing is way to extract the value from those objects.

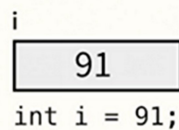
- Boxing is done mostly behind the scenes by the CLR and is implicit
- Unboxing is explicitly done by the programmer

Example

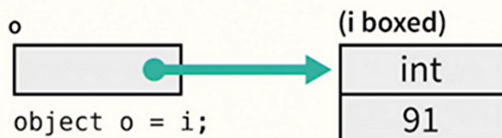
Boxing Example

- `int i = 91;`
- `object o = i;`

On the stack



On the heap

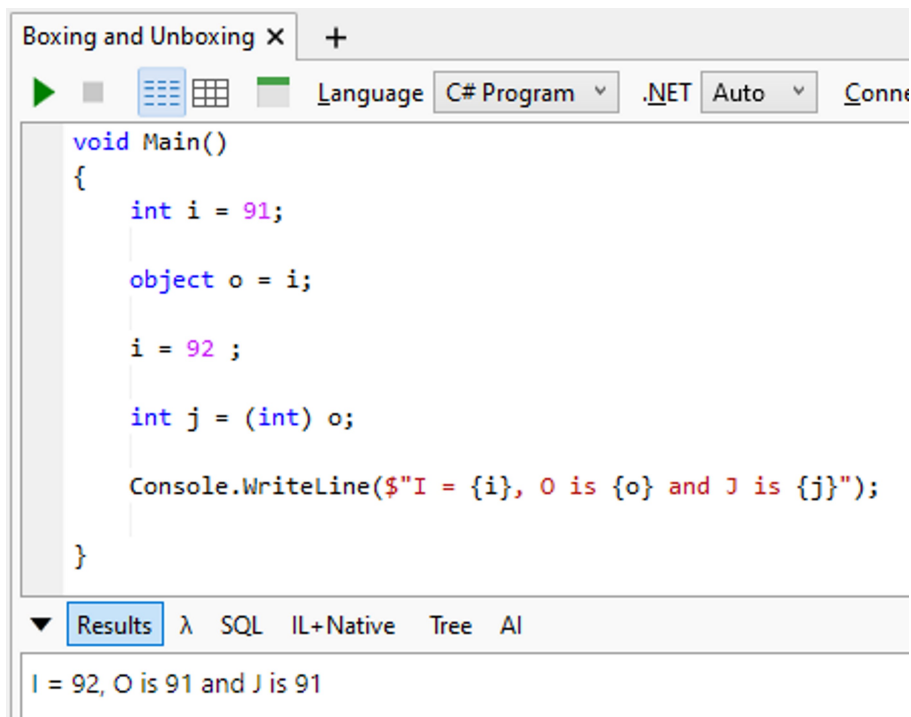


Unboxing Example

- `int i = 91;`
- `object o = i;`
- `i = 92;`
- **`int j = (int) o;`**

I	92
O	91
J	91

Real Code



```
Boxing and Unboxing X +
Language C# Program .NET Auto Connect
void Main()
{
    int i = 91;

    object o = i;

    i = 92 ;

    int j = (int) o;

    Console.WriteLine($"I = {i}, O is {o} and J is {j}");
}
```

Results λ SQL IL+Native Tree AI

I = 92, O is 91 and J is 91

Why not recommended?

- Computationally expensive
- When a value type is boxed, a new object must be allocated and constructed
- Most of the time, the operation is unnecessary

References

<https://www.linkedin.com/learning/nail-your-c-sharp-interview/boxing-vs-unboxing?autoSkip=true&resume=false>