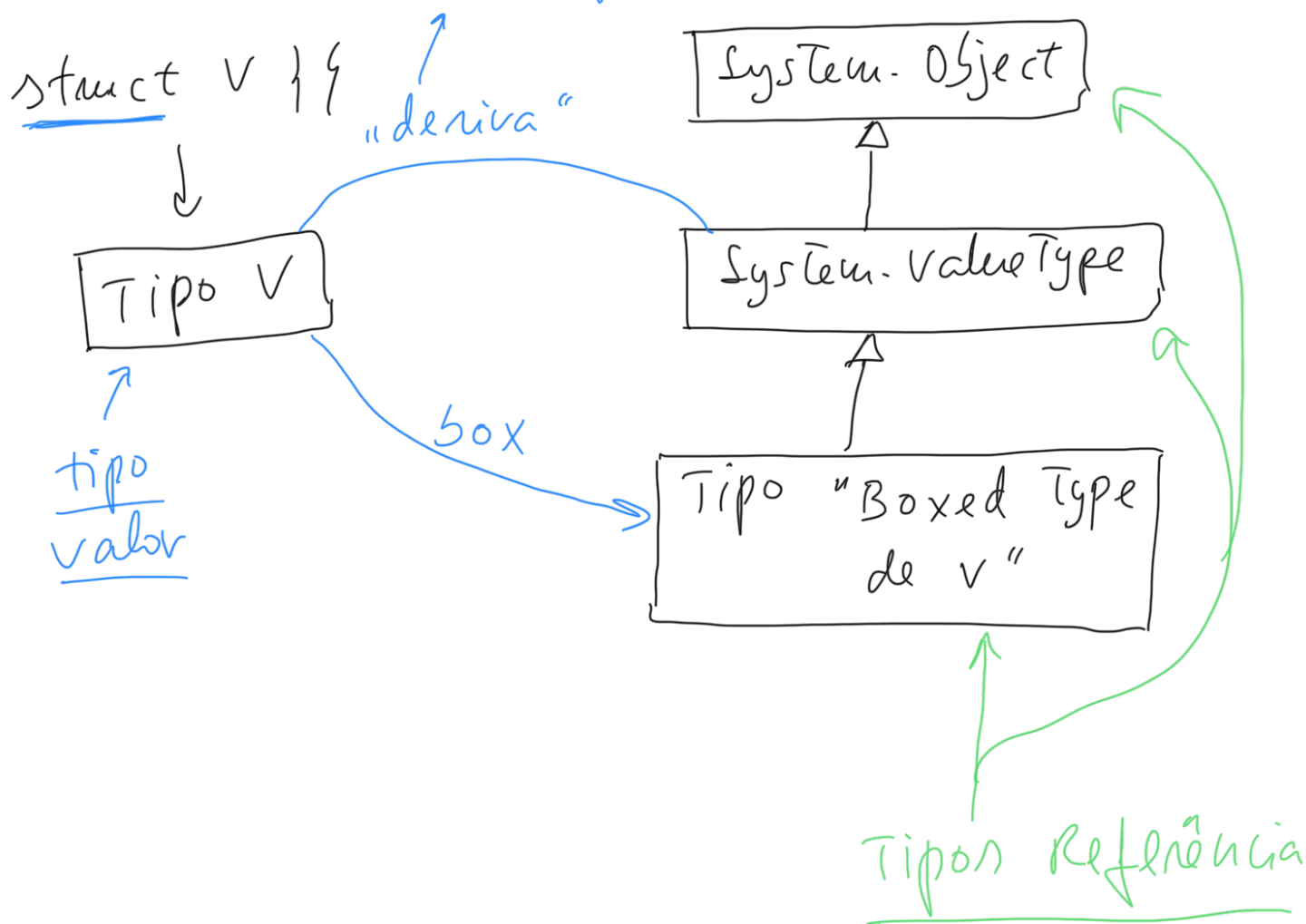


Conversão entre tipos valor e correspondente tipo referência: Box e unbox

Quando é criado um tipo valor, 1.
São na realidade criados 2 tipos:
resolvido pelo compilador



2.

Na realidade, os tipos valor não derivam fisicamente de `System.ValueType`; o seu "boxed type" é que deriva de `System.ValueType`.

O compilador/AVE associa o tipo valor ao seu "boxed type".

Ex:

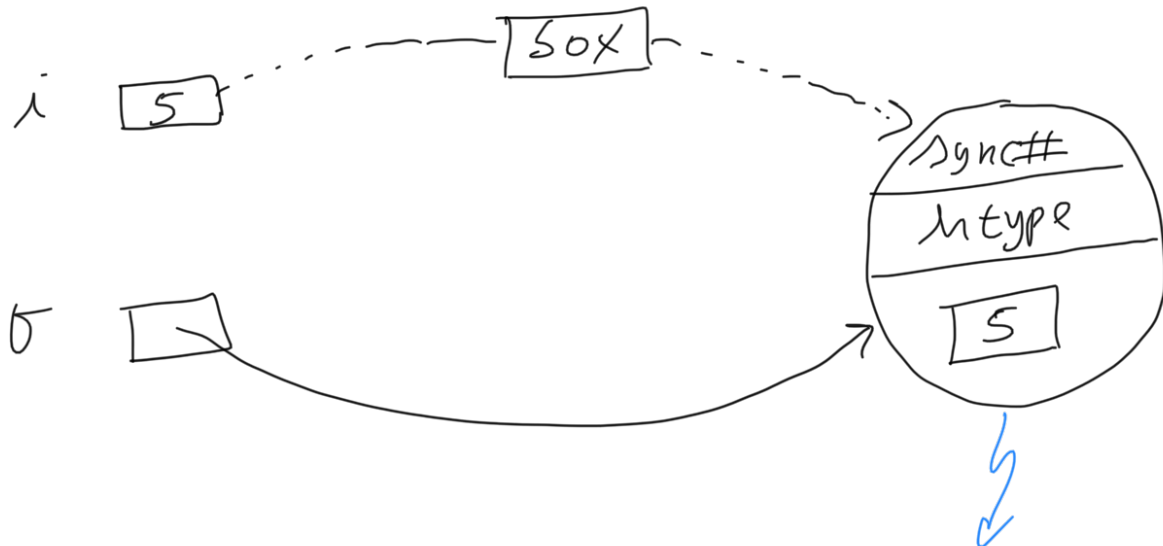
```

    int i = 5;
    ↑
    System.Int32
    object o = i; // VT Int32 → RT object ⇒ box
    ↑
    System.Object
  
```

3.

STACK

HEAP



Instruções IL:

`ldloc.0`

`box Int32`

Tipo exato:
Boxed Type
de `Int32`

4.
Box: Conversão de uma instância
de VT para instância de RT

unBox: contrário; em C#, o unBox é
conseguído fazendo no código um cast
explícito para o tipo valor original
daquela instância.

- Box é um operador de casting implícito;
o compilador só aceita tipos compatíveis
com o tipo valor:

- object

- System.ValueType

⇒ são todos RT

- Qualquer interface
implementada

EX:

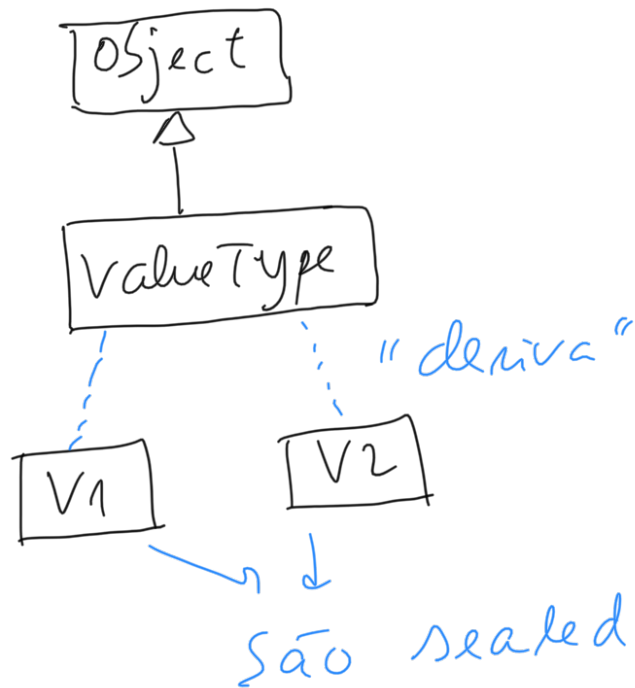
```

interface IPrintable { ... }
struct V1 : IPrintable { int x; }
struct V2 : IPrintable { int y; }

class Prog {
    static void Print(V1 v) { ... }
    static void Print2(IPrintable p) { ... }
}

main:
    V1 v1 = new V1(); // initok
    V2 v2 = v1; X Error: V1 & V2
                    has no
                    comparison
                    operator
    V2 v2 = new V2(); // initok
    Prog.Print(v2); X
    Prog.Print(v1); ✓
    Prog.Print2(v2); ✓ → box
    Prog.Print2(v1); ✓ → box

```



Operação Sox

1. Recebe a instância (ldloc.0) de tipo valor a converter + token que identifica o tipo valor
2. Calcula e reserva espaço no heap para alojar o objeto = $\text{sizeof}(\text{campos}) + \text{header}$

3. copia os valores dos campos
do stack para o heap
4. retorna a referência para o objeto.
- 7.

Unbox

- instância IL unbox.any
- Ao contrário do box, não faz alojamento no heap
 - Obtém o ponteiro para os campos e copia os seus valores para o stack.

Normalmente, de seguida existe uma cópia por valor para uma variável local (stloc)

→ Um objeto (representação boxed)
apenas pode ser unboxed para uma variável que
tem o mesmo tipo que o valor boxed.
⇒ em caso de falha, lança
InvalidCastException



```
EX: struct V {
    int i;
    void Inc() { ++i; }
}
```

main:

```
V v = new V();
```

```
object o = v; // box → 1°)
```

```
v.i = 1; → 2°)
```

```
((V)o).Inc(); // unbox → 3°)
```

```
V v2 = (V)o; // unbox → 4°)
```

```
Console.WriteLine(v2.i); // 0
```


9.

STACKHEAP