

Identidade e Igualdade

1.

$A \ a_1 = \dots;$

$A \ a_2 = \dots;$

a_1 é idêntico a a_2 se a_1 refere o mesmo objeto que a_2 .

Static bool Object.ReferenceEquals (Object o1, Object o2)

↳ método estático de Object que averigua a identidade entre o1 e o2.

bool Object.Equals (Object o)

↳ método virtual de Object que averigua igualdade, normalmente é redefinido pelo utilizador de forma a que este defina a igualdade.

2.

=> Implementações em object
uma a identidade para
verificar a igualdade.

EX:

RT \Leftarrow R n = new R(); // newobj ...

R n2 = new R();

VT \Leftarrow V v = new V(); // initobj v

V v2 = new V();

box (v).Equals(box v); // 2 boxes, mas retornas true, pois têm os mesmos campos

box (v).Equals(box v2); // true

Object.ReferenceEquals(box v, box v); // e o mesmo tipo exato.

n.Equals(n); // true

n.Equals(n2); // false

→ false, porque

nas comparadas

2 versões boxed
que têm endereços

diferentes.

nas RTA => não há
box

mas Equals não foi
redefinido, logo
compara identidade.

compara endereços e nas conteúdo.

Implementação do método Equals em RTs

3.

```
class Point {  
    int x, y;
```

```
    public override bool Equals (object o) {
```

1ª implementação

```
        if (o == null) return false;  
        if (Object.ReferenceEquals (this, o))  
            return true;
```

Errado:
só tenta
compatibilidade
e não o
tipo exato

Não usar ==
pois este operador
pode ser overloaded

```
        if (o is Point) {  
            Point p = (Point) o;  
            return p.x == x && p.y == y;  
        }  
        return false;
```

```
}
```

Implementação de
Object.Equals faz nó
neste ponto.

4.

2^a implementación

```
if (o.GetType() == typeof(Point))  
{  
    ...  
}
```

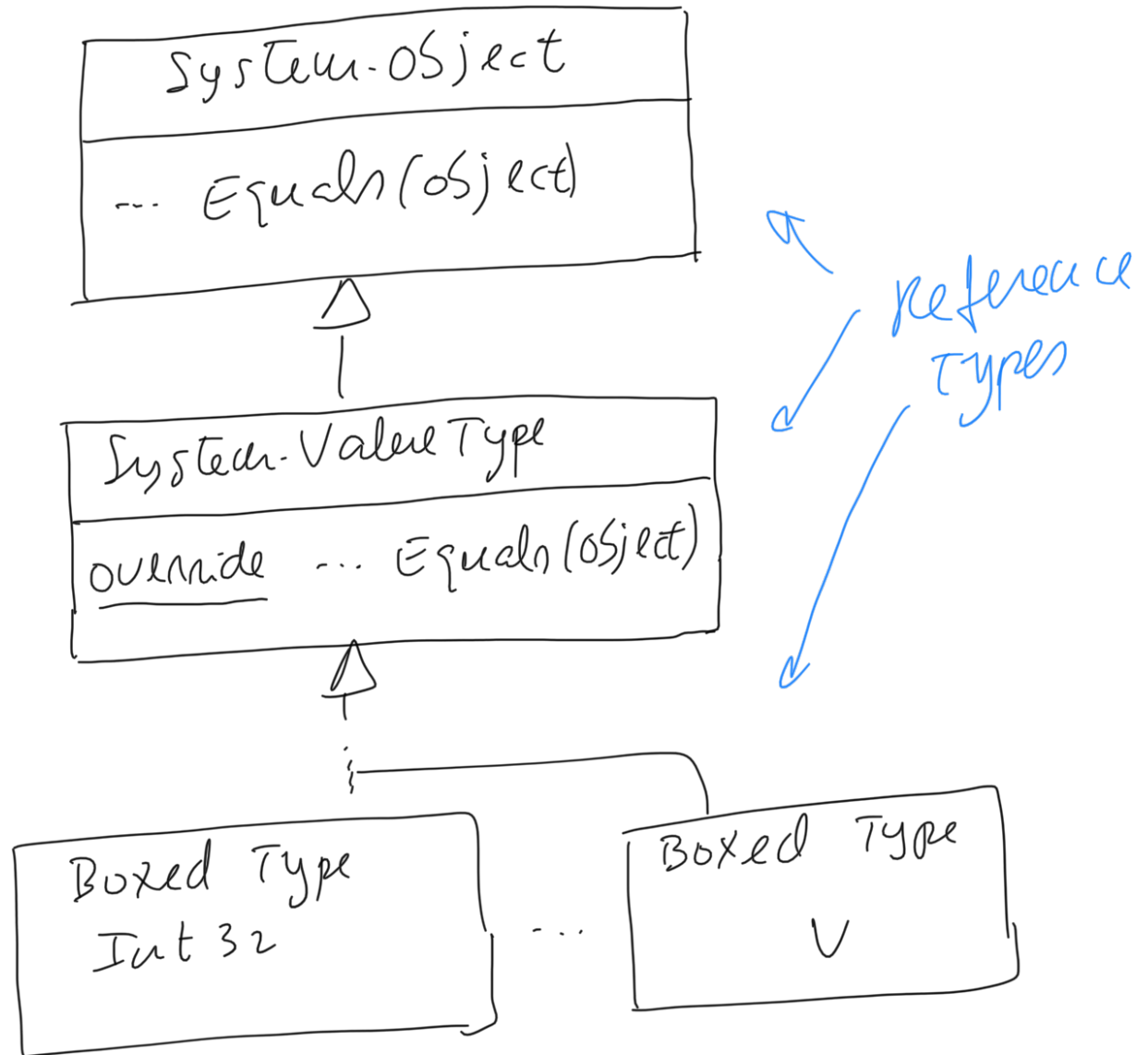
una
acción:
this.GetType()

Testa se es
o mismo tipo
exacto.

Ten problemas
se existiera
clases derivadas

↓
ver ejemplo
código

Equals in Value types 5.



```

struct V {
    int i;
}
  
```

Main:

```

V v1 = new V();
v1.Equals(v1);
  
```

50X 50X true

```

V v2 = new V();
v1.Equals(v2);
  
```

50X 50X false

6.
System.ValueType. Equals(object)
testa a igualdade comparando
os campos dos tipos derivados
de ValueType.

Em tipos usados conhecidos
por ValueType \Rightarrow é feito uso de
reflexão.

Por questões de eficiência, o
programador deve redefinir o Equals
nos seus structs, pois assim é
executado o Equals redefinido e não
o ValueType.Equals (que usa reflexão).