

Trabalhando com Tipos Referência e Valor

Ricardo Augusto Vicentini
Senior Software Engineer - Nubank

Mais sobre mim

- Desenvolvedor desde 2002
- Entrei na área de desenvolvimento porque sempre gostei muito de *Games*.
- Entusiasta da linguagem C#, escovador de bit 🤪
- Tenho Gatos, Cachorros e curto d+ Games e Avições.





Objetivo do curso

Entender a diferença entre utilizar variáveis por referência e valor, será fundamental para entender como o compilador executa o código que você escrever. Ou seja, sem entender esse conceito fundamental um desenvolvedor terá muitos problemas para descobrir ***bugs*** (Comportamento indesejado no código).

Percurso

Aula 1

Conceitos / Prática

Aula 2

Palavra chave **ref**

Aula 3

ref Struct

Aula 4

Comparação por Valor e por Referência

Aula 5

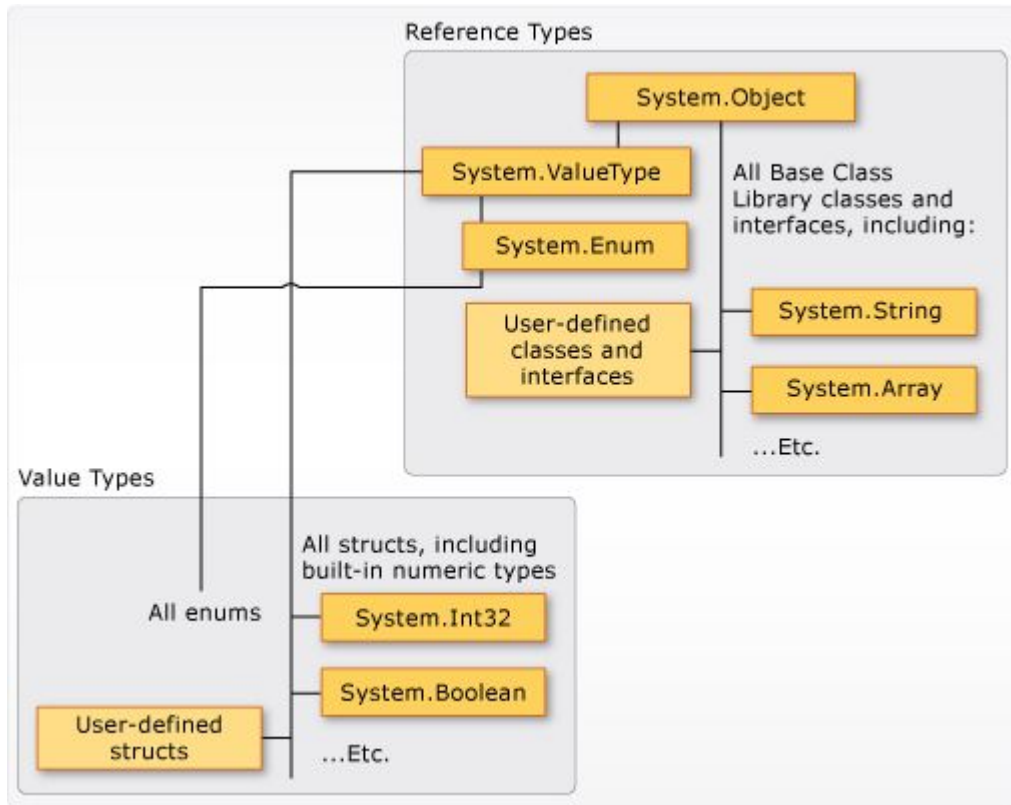
Garbage Collector

Aula 1: Conceitos

Tipos por Referência e Valor



Common Type System (CTS)



Definindo - Value Types

- Contém uma INSTÂNCIA do tipo criado
- A instância sempre é copiada ao atribuir o valor para outra variável
- Alocação na Stack (melhor performance)
- O valor inicial é sempre o valor default de cada tipo

Definindo - Reference Types

- Contém uma REFERÊNCIA para uma instância do tipo criado
- A referência nunca muda ao atribuir o valor para outra variável
- Na STACK fica um ponteiro e a alocação na HEAP
- Seu valor inicial é sempre “Null”
- Requer gerenciamento da Memória através do GC

Value Types

- **Tipos primitivos**
 - **Valores numéricos**
 - **int**
 - **decimal**
 - **double, etc**
 - **Boolean (true/false)**
 - **Char**
 - **Tuples**

Reference Types

- **Classe**
- **Interface**
- **Delegate**
- **Record**
- **object**
- **string**



DIGITAL
INNOVATION
ONE



PARABÉNS

Demos

Criar uma Console application que receba um valor inteiro na
Main;

Criar um método void que receba esse inteiro e altere o seu valor
para qualquer outro;

De volta ao Main exiba no terminal o valor alterado;



Criar uma Console application e uma classe Pessoa com os seguintes atributos “Nome”, “Idade” e “Documento”.

No Main crie uma instância de Pessoa atribuindo a essas propriedades, seu nome e sua idade;

Crie um método void para alterar o Nome do objeto Pessoa;

De volta ao Main exiba no terminal o nome alterado;

Mostrar na prática diferença entre atribuir uma instância para um Value Type e Reference Type.



Criar uma Console application com uma variável do tipo string e atribua seu nome a esta variável;
Crie um método void que receba essa variável e altere esse valor;
No Main escreva no terminal o valor alterado;

Criar uma Console application com uma variável do tipo array de int para armazenar os números pares de 0 a 8;
Crie um método void que receba essa variável e altere o conteúdo desse array para que nele fique armazenado o próximo número inteiro ímpar de cada elemento;
No Main escreva no terminal todos os números desse array;

Criar uma Console application para encontrar um número inteiro em um array;



DIGITAL
INNOVATION
ONE



Level UP!





- O que são Value e Reference Types
- Onde o CLR aloca Value e Reference Types
- Quais os principais objetos tratados como Value Types
- Quais os principais objetos tratados como Reference Types
- Como diferenciá-los
- Documentação:
<https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/types/#the-common-type-system>
- Código das demos:
<https://github.com/ricardovicentini/Demos-Reference-And-Value-Types>



DIGITAL
INNOVATION
ONE

Aula 2: “ref” keyword

Tipos por Referência e Valor

Utilização

**O ref indica que o conteúdo de determinada variável
acessado será acessado por referência.
E pode ser usada em 4 situações**

1. Na declaração dos parâmetros do método e na chamada do método
2. Na declaração do retorno do método
3. No corpo do método para receber um retorno com ref
4. E na declaração de uma Struct

Demos

Exemplificar com ref no parâmetro de entrada o mesmo que fizemos na Demo 1

Demos

Criar um array de string com nomes, permitir que o usuário localize e altere um nome dentro do array



DIGITAL
INNOVATION
ONE



Level UP!



Utilizando palavra chave “ref”

- nos parâmetros de entrada
- no retorno dos métodos
- documentação:

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/ref>

Aula 3: ref struct

Tipos por Referência e Valor

Definição

“ref struct” serve para assegurar que a struct ficará na stack e nunca irá para a Heap.

Limitações

- **ref struct não pode:**
 - ser elemento tipado de um array
 - ser o tipo em campo em uma classe ou não-ref struct
 - implementar interfaces
 - ser convertida para Object e nem para Value Type
 - ser usada em métodos assíncronos



Então, quando usar?

Quando for necessário garantir que a instância da struct não irá acessar a Heap.

Quando for usar tipos do c# que são ref struct, como o caso do ref struct Span

Demos

Vamos construir uma ref struct e tentar utilizá-la em uma classe
Vamos tentar usar um Span em uma não-ref struct



DIGITAL
INNOVATION
ONE



Level UP!



ref struct

- O que é um ref struct
- Onde são alocados
- suas limitações
- documentação:

<https://docs.microsoft.com/en-us/dotnet/api/system.sp-an-1?view=net-5.0>

Aula 4: Comparação

Tipos por Referência e Valor

Demos

Agora que aprendemos a diferença entre Value Types e Reference types, vamos entender o funcionamento do CLR ao comparar esses tipos.

Value Types

instância = a instância

Reference Types

referência = a referência



DIGITAL
INNOVATION
ONE



Level UP!



Aula 5: Garbage Collector

Tipos por Referência e
Valor



DIGITAL
INNOVATION
ONE

GC - Definição

suporte para a criação e destruição de objetos na Heap

Vantagens - GC

- **Segurança**
- **Programador não precisa se preocupar com a liberação de memória**
- **Nem com sobrescrita de memória em uso.**

Desvantagens - GC

- **Performance**
- **Observabilidade**

Arquitetura - GC

- **O GC é dividido em 3 Gerações**
 - **Gen 1**
 - **Objetos de ciclo de vida curto**
 - **Gen 2**
 - **Buffer de alternância entre Gen 1 e Gen 3**
 - **Gen 3**
 - **Objetos com longo ciclo de vida em especial objetos criados como “static”**



DIGITAL
INNOVATION
ONE



Level UP!





Review

- O que são Reference e Value Types e seu comportamento
- Demos sobre alocação na stack e na Heap
- Boas práticas na criação de classes, structs
- A palavra chave ref em parâmetros de entrada
- A palavra chave ref no retorno de métodos e na atribuição de variáveis
- Comportamento do CLR ao comparar value e reference types
- O que é e como é o funcionamento básico do GC

Documentação

<https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/keywords/reference-types>

Github - Demos

<https://github.com/ricardovicentini/Reference-Type-And-Value-Type>

master

1 branch

0 tags

Go to file



ricardovicentini incluido aula 3

036d3de 11 h



Aula 1

commit aula 2



Aula 2

commit aula 2



Aula 3

incluido aula 3



.gitignore

adicionado gitignore



README.md

Update README.md

README.md

Reference Type And Value Type

Este repo contém aulas gravadas para Digital Innovation One, é livre para baixar e aprender! 😊

Mande suas dúvidas e sugestões por meio de Issues e PRs ❤️