

Curso C# Completo

Programação Orientada a

Objetos + Projetos

Capítulo: Enumerações, composição

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Enumerações (enum)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

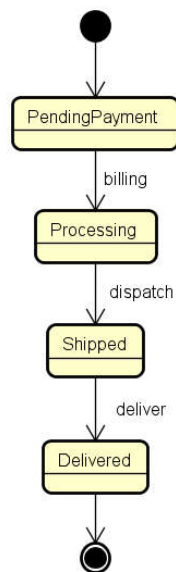
- Definição / discussão
- Exemplo: estados de um pedido
- Conversão de string para enum
- Representação UML

Enumerações

- É um tipo especial que serve para especificar de forma literal um conjunto de constantes relacionadas
- Palavra chave em C#: enum
 - Nota: enum é um tipo valor
- Vantagem: melhor semântica, código mais legível e auxiliado pelo compilador
- Referência: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/enum>

Exemplo

Ciclo de vida de um pedido.



```
enum OrderStatus : int {
    PendingPayment = 0,
    Processing = 1,
    Shipped = 2,
    Delivered = 3
}
```

```
class Order {
    public int Id { get; set; }
    public DateTime Moment { get; set; }
    public OrderStatus Status { get; set; }
}
```

<https://github.com/acenelio/enum1-csharp>

Conversões

Para o tipo original: use casting

```
OrderStatus os1 = (OrderStatus)2;  
int n1 = (int)OrderStatus.Processing;
```

string - enum:

```
string txt = OrderStatus.PendingPayment.ToString(); de enum para string
```

```
OrderStatus os = Enum.Parse<OrderStatus>("Delivered"); de string para enum
```

```
Console.WriteLine(os);  
Console.WriteLine(txt);
```

Notação UML



Vamos falar um pouco de design

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Categorias de classes

- Em um sistema orientado a objetos, de modo geral "tudo" é objeto.
- Por questões de design tais como organização, flexibilidade, reuso, delegação, etc., há várias categorias de classes:

Views

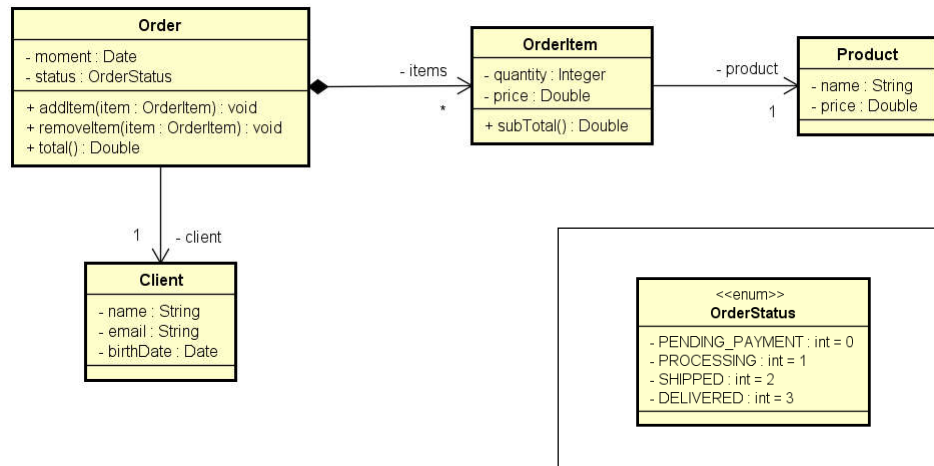
Controllers

Entities

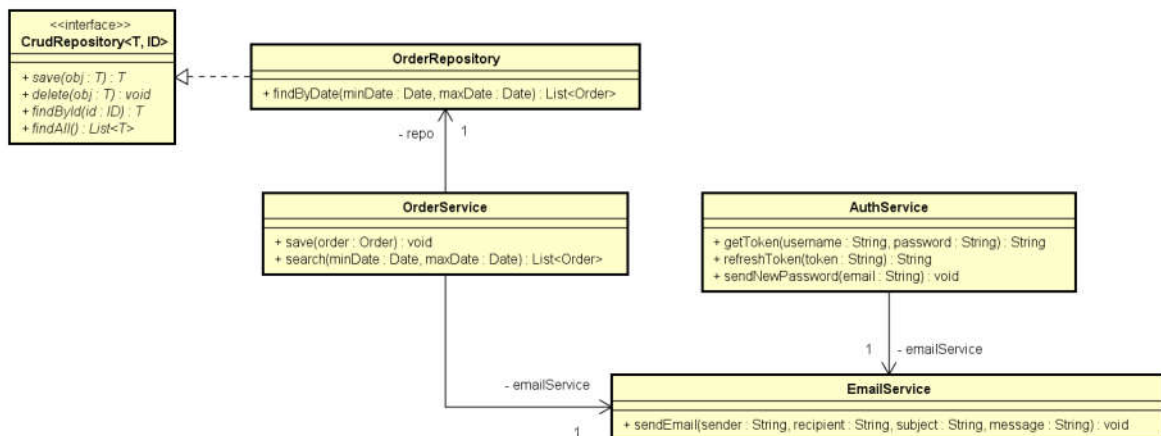
Services

Repositories

Entities



Services



Composição

<http://educandoweb.com.br>

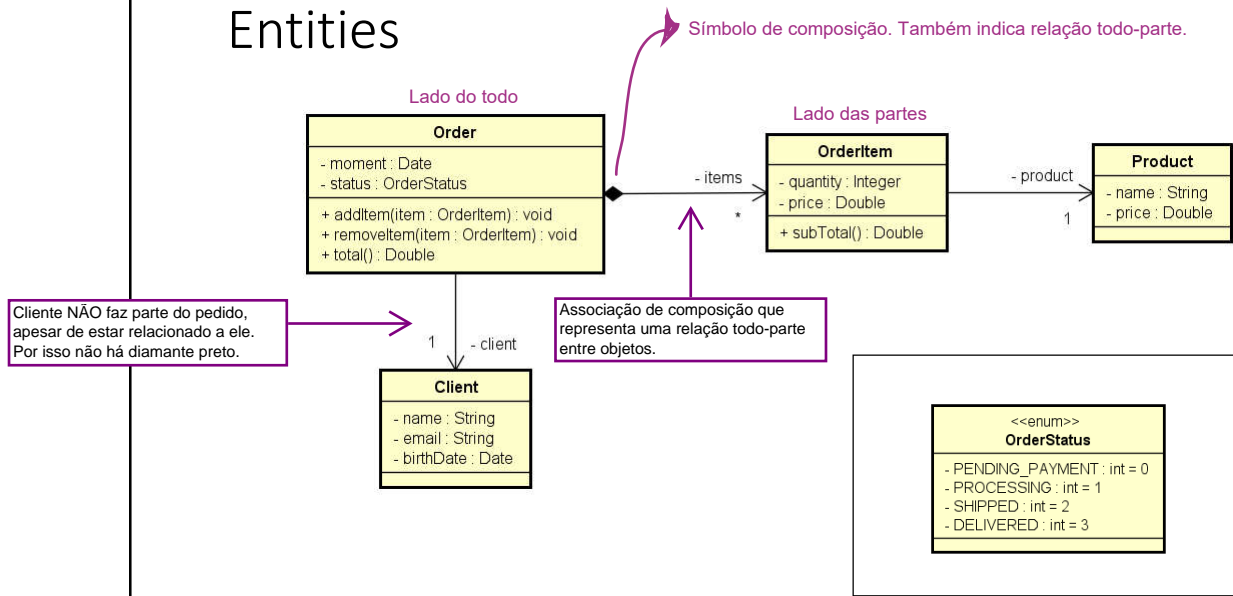
Prof. Dr. Nelio Alves

Composição

- É um tipo de associação que permite que um objeto contenha outro
- Relação "tem-um" ou "tem-vários"
- Vantagens
 - Organização: divisão de responsabilidades
 - Coesão: cada objeto é responsável por uma única coisa
 - Flexibilidade: trabalhar com algo dividido em partes é mais flexível do que trabalhar com algo grande e agrupado.
 - Reuso: o mesmo objeto pode ser reusado em mais de um lugar.
- Nota: embora o símbolo UML para composição (todo-parte) seja o diamante preto, neste contexto estamos chamando de composição qualquer associação tipo "tem-um" e "tem-muitos".

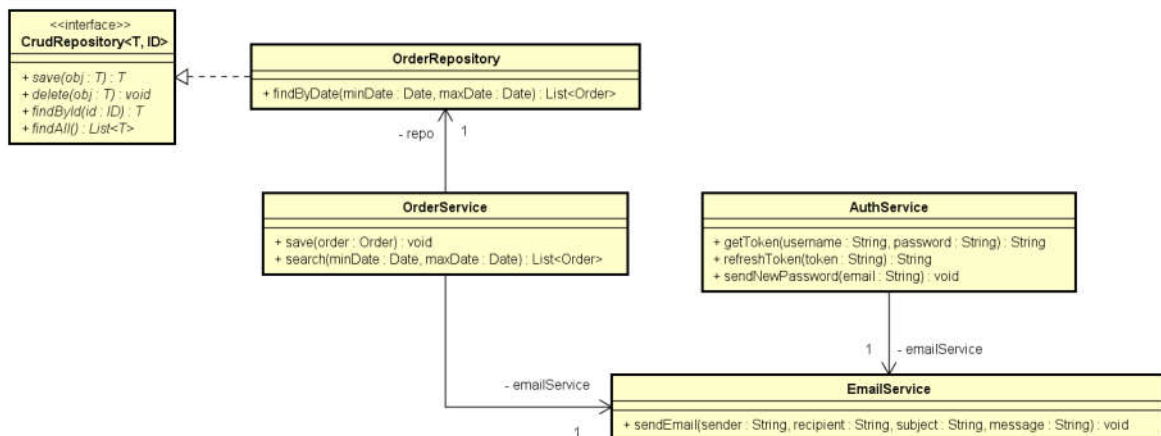
O asterisco (*) indica cardinalidade "vários"

Entities



Composição pode ocorrer também com

Services

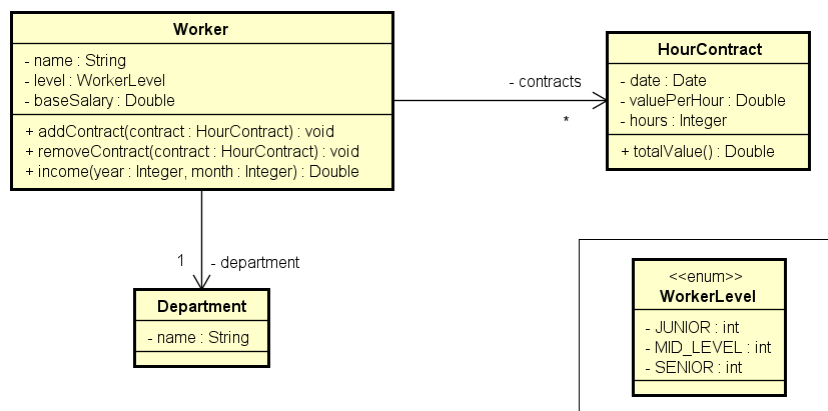


Exercício resolvido 1

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Ler os dados de um trabalhador com N contratos (N fornecido pelo usuário). Depois, solicitar do usuário um mês e mostrar qual foi o salário do funcionário nesse mês, conforme exemplo (próxima página).



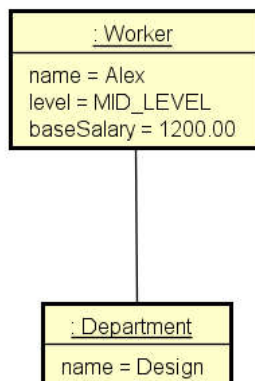
```
Enter department's name: Design
Enter worker data:
Name: Alex
Level (Junior/MidLevel/Senior): MidLevel
Base salary: 1200.00
How many contracts to this worker? 3
Enter #1 contract data:
Date (DD/MM/YYYY): 20/08/2018
Value per hour: 50.00
Duration (hours): 20
Enter #2 contract data:
Date (DD/MM/YYYY): 13/06/2018
Value per hour: 30.00
Duration (hours): 18
Enter #3 contract data:
Date (DD/MM/YYYY): 25/08/2018
Value per hour: 80.00
Duration (hours): 10

Enter month and year to calculate income (MM/YYYY): 08/2018
Name: Alex
Department: Design
Income for 08/2018: 3000.00
```

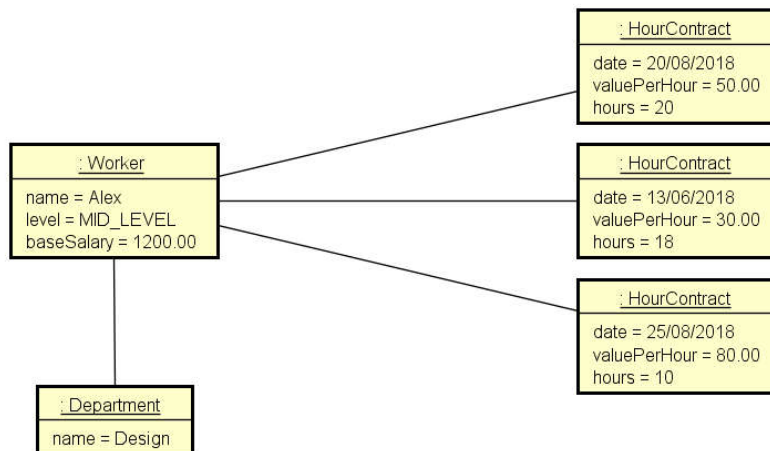
} HourContract

<https://github.com/acenelio/composition1-csharp>

Objects in memory:



Objects in memory:



StringBuilder pertence ao namespace "System.Text".

Responsável por fazer a montagem de uma string de uma forma eficiente.

Mais eficiente do que concatenar strings.

Exercício resolvido 2 (demo StringBuilder)

<http://educandoweb.com.br>

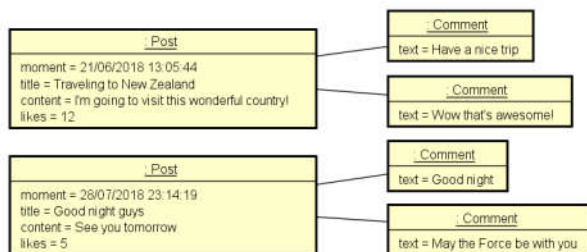
Prof. Dr. Nelio Alves

```
public override string ToString()
{
    StringBuilder sb = new StringBuilder();
    sb.AppendLine(Title);
    sb.Append(Like);
    sb.AppendLine(" Likes - ");
    sb.AppendLine(Moment.ToString("dd/MM/yyyy HH:mm:ss"));
    sb.AppendLine(Content);
    sb.AppendLine("Comments:");
    foreach (Comment c in Comments)
    {
        sb.AppendLine(c.Text);
    }
    return sb.ToString();
}
```

Instancie manualmente os objetos mostrados abaixo e mostre-os na tela do terminal, conforme exemplo.



Console output:



```
Traveling to New Zealand
12 Likes - 21/06/2018 13:05:44
I'm going to visit this wonderful country!
Comments:
Have a nice trip
Wow that's awesome!

Good night guys
5 Likes - 28/07/2018 23:14:19
See you tomorrow
Comments:
Good night
May the Force be with you
```

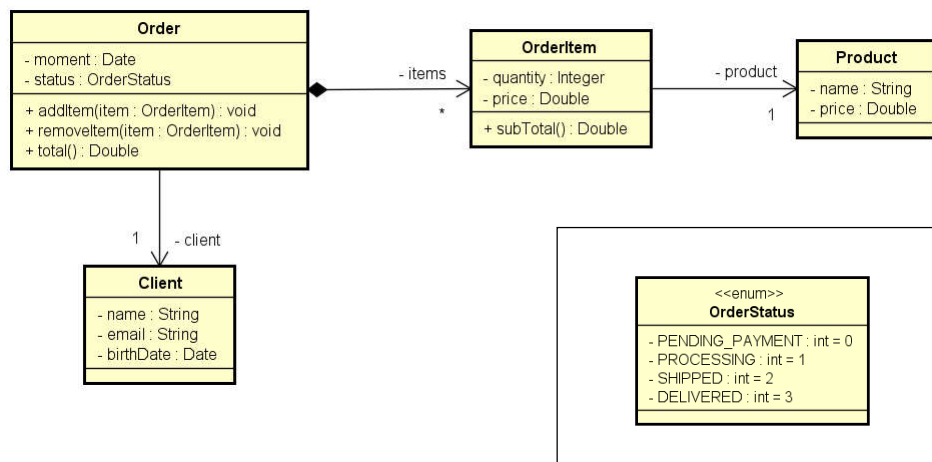
<https://github.com/acenelio/composition2-csharp>

Exercício de fixação

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Ler os dados de um pedido com N itens (N fornecido pelo usuário). Depois, mostrar um sumário do pedido conforme exemplo (próxima página). Nota: o instante do pedido deve ser o instante do sistema: `DateTime.Now`



Enter cliente data:

Name: **Alex Green**

Email: **alex@gmail.com**

Birth date (DD/MM/YYYY): **15/03/1985**

Enter order data:

Status: **Processing**

How many items to this order? **2**

Enter #1 item data:

Product name: **TV**

Product price: **1000.00**

Quantity: **1**

Enter #2 item data:

Product name: **Mouse**

Product price: **40.00**

Quantity: **2**

ORDER SUMMARY:

Order moment: 20/04/2018 11:25:09

Order status: Processing

Client: Alex Green (15/03/1985) - alex@gmail.com

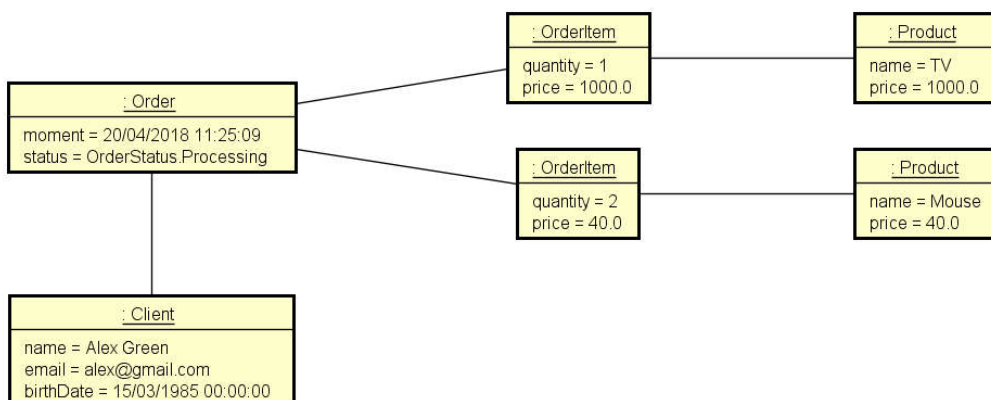
Order items:

TV, \$1000.00, Quantity: 1, Subtotal: \$1000.00

Mouse, \$40.00, Quantity: 2, Subtotal: \$80.00

Total price: \$1080.00

Você deverá instanciar os objetos em memória da seguinte forma:



<https://github.com/acenelio/composition3-csharp>