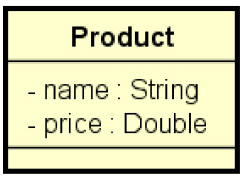
Problema

- Suponha uma classe Product com os atributos name e price. Suponha que precisamos ordenar uma lista de objetos Product.
- Podemos implementar a comparação de produtos por meio da implementação da interface IComparable<Product>
- Entretanto, desta forma nossa classe Product não fica fechada para alteração: se o critério de comparação mudar, precisaremos alterar a classe Product.
- Podemos então usar outra sobrecarga do método "Sort" da classe List:

```
public void Sort(Comparison<T> comparison)
```



Comparison<T> (System)

https://msdn.microsoft.com/en-us/library/tfakywbh(v=vs.110).aspx

```
public delegate int Comparison<in T>(T x, T y);
```

Método Sort com Comparison<T> da classe List:

https://msdn.microsoft.com/en-us/library/w56d4y5z%28v=vs.110%29.aspx

Resumo da aula

```
public void Sort(Comparison<T> comparison)
```

- Referência simples de método como parâmetro
- Referência de método atribuído a uma variável tipo delegate
- Expressão lambda atribuída a uma variável tipo delegate
- Expressão lambda inline

https://github.com/acenelio/lambda1-csharp

Programação funcional e cálculo lambda

http://educandoweb.com.br

Prof. Dr. Nelio Alves