

Fundamentos de C#

Exercícios Propostos

**Coleções de Dados**

www.softblue.com.br

# Exercício

Crie uma classe *Produto* com dois fields: nome (string) e valor (double). Implemente a interface IComparable<T> de forma que os produtos possam ser ordenados em ordem crescente de valor quando necessário. Sobrescreva também os métodos Equals() e GetHashCode() e implemente a interface IEquatable<T>, considerando que produtos iguais são produtos que possuem o mesmo nome. E por último sobrescreva também o método ToString(), para mostrar uma representação amigável do produto quando ele for impresso no console.

Na sequência crie uma classe Produtos, responsável por armazenar os produtos criados. Esta classe tem um field *produtos*, do tipo ICollection<Produto>, e os métodos Adicionar(), que adiciona um produto à coleção, e ImprimirProdutos(), que imprime todos os produtos.

Crie uma aplicação que cria os seguintes produtos:

|  |  |
| --- | --- |
| Nome | Valor |
| Laranja | 2,50 |
| Laranja | 2,70 |
| Maçã | 1,45 |
| Mamão | 4,95 |
| Limão | 2,30 |

Experimente adicionar os produtos acima a coleções de diversos tipo, como List<T>, HashSet<T> e SortedSet<T>, e imprima os resultados. Lembre-se que todos estes tipos podem ser atribuídos ao field produtos, pois todos são do tipo ICollection<T>. Procure perceber o que acontece com os elementos quando você muda o tipo de coleção na qual o produto está inserido, com relação à duplicidade de elementos e ordenação.

# Exercício

Crie um programa de votação, onde o usuário deve escolher a sua linguagem de programação preferida dentre as seguintes opções:

1. C#
2. Java
3. C
4. C++
5. Python

O programa solicita votos até que o número 0 seja escolhido. Quando isto acontecer, o programa deverá mostrar a lista de opções em uma tabela, juntamente com o número de votos de cada opção e a porcentagem de votos com relação ao total. A tabela também deve mostrar, no final, o total de votos realizados.

Por fim, o programa deve mostrar qual foi a linguagem de programação mais votada, bem como o número de votos que ela recebeu.

Algumas considerações importantes:

* As opções disponíveis para votação devem ser armazenadas em um dicionário, onde o número da opção é mapeado para a linguagem de programação correspondente.
* Os votos coletados devem ser armazenados em uma lista.
* Se uma opção inválida for digitada, o programa deverá mostrar uma mensagem de erro e solicitar novamente o voto.