



Fundamentos de C#

Exercícios Propostos

Arrays

1 Exercício

Escreva um programa que solicite a entrada de 5 números inteiros via console, os quais deverão ser armazenados em um array. Depois itere sobre os elementos do array e imprima as seguintes informações na tela:

- 1) Média entre os elementos do array.
- 2) Maior valor presente no array.
- 3) Menor valor presente no array.

Para finalizar, ordene os elementos do array em ordem decrescente e imprima os elementos.

2 Exercício

Crie uma classe `Carta` que contenha o valor e o naipe (o naipe pode ser um *enumeration*). Crie também uma classe `Baralho`, que contém um array de cartas. A classe `Baralho` deve ter os seguintes métodos:

`Carta[] Distribuir(int qtde)`: Distribui cartas do baralho. A quantidade de cartas distribuídas é passada como parâmetro, e o retorno do método é um array de cartas distribuídas. As cartas que são distribuídas deixam de fazer parte do baralho, pelo menos conceitualmente.

`void Embaralhar()`: Embaralha as cartas do baralho. Cartas já distribuídas não fazem parte do processo de embaralhamento.

`void MostrarCartas()`: Imprime as cartas do baralho. Cartas já distribuídas não devem ser mostradas, pois não fazem mais parte do baralho.

Algumas considerações importantes:

- A quantidade a ser distribuída no processo de distribuição deve ser maior que zero e menor ou igual ao número de cartas existentes no baralho. Caso a quantidade seja inválida, o método deve lançar uma exceção do tipo `ArgumentOutOfRangeException`.
- O processo de embaralhamento pode ser feito utilizando outro array, que recebe os elementos do array original de forma randomizada.
- Durante a randomização no processo de embaralhamento, é possível que a mesma carta seja sorteada mais de uma vez. Portanto é preciso controlar quais cartas já foram sorteadas e fazer esta checagem, para evitar situações deste tipo. Uma dica é utilizar um array para armazenar os índices de elementos já sorteados.

Crie um método `Main()` na aplicação que instancia as cartas e o baralho e faça chamadas aos métodos `Distribuir()`, `Embaralhar()` e `MostrarCartas()` para validar a implementação realizada.

3 Exercício

Faça um programa que cria uma matriz de inteiros de duas dimensões e atribui valores a cada uma das posições desta matriz. O número de linhas, de colunas e os valores a serem atribuídos para cada posição devem ser lidos via console.

Na sequência, imprima a matriz e calcule a soma dos elementos de cada coluna. Esta soma deve ser armazenada em um array. Por exemplo, se a matriz tem tamanho 3x4, você deverá criar um array de 4 posições, onde cada posição armazenará a soma dos 3 valores da coluna. Os valores das somas também deverão ser mostrados.