

Roteiro

1. Crie o ArrayList AL1 sem definir a capacidade inicial.
2. Imprima a capacidade e a quantidade de elementos.
3. Adicione os números 19, 7 e 11.
4. Imprima a capacidade e a quantidade de elementos.
5. Adicione os números 5, 7 e 17.
6. Imprima a capacidade e a quantidade de elementos.
7. Imprima os elementos do ArrayList usando o comando foreach.
8. Adicione o número 5 nas posições 0, 2 e 10. Ocorreu algum problema? Se sim, anote qual foi o motivo e resolva esse problema.
9. Imprima os elementos do ArrayList usando o comando for.
10. Imprima a primeira e a última posição que contém o número 7.
11. Imprima todas as posições que contém o número 5.
12. Adicione os números 5, 23, 47, 5 e 5.
13. Imprima os elementos do ArrayList usando o comando while.
14. Imprima as posições do número 5 retornada pelos métodos: BinarySearch(), IndexOf() e LastIndexOf()
15. Ordene os elementos do ArrayList.
16. Imprima os elementos do ArrayList (use o comando que você desejar).
17. Remova o número 23.
18. Imprima os elementos do ArrayList (use o comando que você desejar).
19. Remova o elemento da posição 7.
20. Imprima os elementos do ArrayList (use o comando que você desejar).
21. Remova os elementos das posições 2, 3 e 4.
22. Imprima os elementos do ArrayList (use o comando que você desejar).
23. Inverta os elementos do ArrayList.
24. Imprima os elementos do ArrayList (use o comando que você desejar).
25. Remova todos os elementos do ArrayList.
26. Imprima os elementos do ArrayList (use o comando que você desejar).
27. Imprima a capacidade e a quantidade de elementos.

Lista de exercícios

Coleções .NET

Atenção, cada uma das questões deve ter 3 soluções, uma para **ArrayList**, outra para **Queue** e outra para **Stack**. Para todos os exercícios, considere soluções não-destrutivas, ou seja, os dados devem ser mantidos na ordem original.

- 1 - Crie um ArrayList e adicione 10 valores inteiros digitados pelo usuário. Ao final, imprima todos os elementos.
- 2 - Crie um ArrayList e adicione 10 strings digitados pelo usuário. Ao final, imprima todos os elementos.
- 3 - Crie um ArrayList e adicione os números ímpares no intervalo entre 1 a 100. Calcule a soma dos números usando o comando **for**.
- 4 - Crie um ArrayList e adicione os números pares no intervalo entre 1 a 100. Calcule a soma dos números usando o comando **while**.
- 5 - Crie um ArrayList e adicione os números no intervalo entre 1 a 100. Calcule a soma dos números usando o comando **do while**.

6 - Crie um ArrayList e adicione os números pares no intervalo entre 1 a 100. Calcule a soma dos números usando o comando **foreach**.

7 - Crie um ArrayList contendo os seguintes elementos (5, 13, 19, 31, 3, 7, 11, 5, 57, 13, 5). Faça uma função que apague TODAS as ocorrências de um determinado elemento. Use essa função para apagar todas as ocorrências do número 5 e 13.

Dica: você pode criar um ArrayList e já inicializá-lo com alguns elementos conforme o exemplo abaixo:

```
ArrayList AL = new ArrayList() {1, 2, 3, 4, 5};
```

8 – Faça um programa que **leia n números** inteiros e os armazene em um ArrayList. Calcule a soma e a média aritmética (use o comando FOR e depois o FOREACH).

9 – Faça um programa que **leia n números** inteiros e os armazene em um Queue. Calcule a soma e a média aritmética (use o comando FOREACH para iterar sobre os elementos).

10 – Faça um programa que **leia n números** inteiros e os armazene em um Stack. Calcule a soma e a média aritmética (use o comando FOREACH para iterar sobre os elementos).

11 – Faça um programa que preencha um ArrayList com elementos de diferentes tipos (int, double, float, boolean, String). Tente calcular a soma dos elementos. Evidentemente, isso irá provocar uma mensagem de erro. Que mensagem o Visual Studio retorna?

Dica: você pode criar um ArrayList e já inicializá-lo com alguns elementos conforme o exemplo abaixo:

```
ArrayList AL = new ArrayList() { 1, 2, "AED", new Queue(), "teste", 3.14 };
```

12 – Faça um programa que preencha um ArrayList com os números entre 1 e 25. Pede-se:

- Imprima todos os elementos
- Imprima todos os elementos em ordem invertida
- Imprima todos os elementos em posições ímpares (os elementos da posição 1, 3, 5, ...)
- Imprima todos os elementos ímpares
- Imprima apenas os elementos da primeira metade do vetor (posição 0 a 12).

OBS: você deve fazer esse programa 2 vezes. Primeiro usando o comando FOR e depois usando o comando FOREACH.

13 - Faça um programa que gere uma coleção com **n** números inteiros aleatórios (o valor de **n** deve ser informado pelo usuário no início da execução do programa. Imprima os elementos da coleção.

Exemplo de geração de números aleatórios.

```
Random r = new Random();
```

```
int x = r.Next(); // Retorna um número aleatório positivo
```

```
int y = r.Next(100); // Retorna um número aleatório entre 0 e 99
```

```
int z = r.Next(50, 100); // Retorna um número aleatório entre 50 e 99
```

```
double w = r.NextDouble(); // Retorna um ponto-flutuante entre 0.0 e 1.0
```

14 – Crie uma função para inverter os dados da coleção recebida como parâmetro. Obs1: use qualquer outra estrutura que julgar necessária. **Obs2: não utilize o método reverse da classe ArrayList.**

15 – Crie uma função que receba a coleção como parâmetro e retorne a soma de seus elementos. Obs: considere que todos seus dados são do tipo int.

16 – Crie uma função que calcule o número de elementos positivos de uma coleção passada como parâmetro.

17 – Crie uma função que calcule o número de ocorrências em uma coleção de um elemento passado como parâmetro.