

LP3 – Exercícios

Aula 9 – Classes de Coleções

Instruções para entrega das listas de exercícios:

Meio de Entrega: As resoluções das listas de exercícios devem ser entregues exclusivamente por meio do ambiente Moodle (<http://eadcampus.spo.ifsp.edu.br>). Usar o mesmo usuário e senha do Sistema SUAP.

Forma de Entrega: Para exercícios com implementação de programas console, devem ser entregues os arquivos das classes (extensão CS). Para exercícios com implementação de programas Windows Forms, devem ser entregues as pastas dos projetos que contêm as aplicações desenvolvidas. Para exercícios com banco de dados, também devem ser entregues as instruções SQL usadas (extensão SQL ou TXT). Todos os arquivos da lista devem ser compactados em um único arquivo (extensão RAR ou ZIP), cujo nome deverá conter a aula, o nome e um sobrenome do aluno. Por exemplo: Aula2_JoaSilva.zip.

Prazo de Entrega: O prazo de entrega está definido na própria página de exercícios do Moodle, lembrando que o sistema bloqueia o envio de arquivos após a data e horário indicados.

Obs.: A resolução deste(s) exercício(s) deve ser feita de forma INDIVIDUAL. Listas de exercícios com uma ou mais respostas idênticas serão desconsideradas integralmente para efeitos de nota de participação.

1. **[Para Entrega]** Faça um programa Windows Forms em C# que apresente ao usuário as opções apresentadas na tela a seguir. Ao selecionar uma opção, seus respectivos campos de entrada (Elemento, Posição, etc) devem ser apresentados ao usuário. Ao clicar no botão **Executar**, o programa deverá efetuar a operação escolhida, com base nos dados informados nos campos de entrada (quando houver). A cada operação realizada, apresente a lista de elementos atualizada no ListBox da parte inferior da tela. Use a classe List<string> para armazenar os elementos. Os métodos referentes a cada operação devem ser implementados em uma classe chamada **Lista**.

Exercício 1 - List

Operações

- ☒ Adicionar elemento
- ☐ Adicionar elemento em posição específica
- ☐ Remover elemento (pelo valor)
- ☐ Remover elemento (pela posição)
- ☐ Ordenar lista

Elemento

Executar

Exercício 1 - List

Operações

- ☐ Adicionar elemento
- ☒ Adicionar elemento em posição específica
- ☐ Remover elemento (pelo valor)
- ☐ Remover elemento (pela posição)
- ☐ Ordenar lista

Elemento

Posição

Executar

Exercício 1 - List

Operações

- ☐ Adicionar elemento
- ☐ Adicionar elemento em posição específica
- ☒ Remover elemento (pelo valor)
- ☐ Remover elemento (pela posição)
- ☐ Ordenar lista

Valor

Executar

Exercício 1 - List

Operações

- ☐ Adicionar elemento
- ☐ Adicionar elemento em posição específica
- ☐ Remover elemento (pelo valor)
- ☒ Remover elemento (pela posição)
- ☐ Ordenar lista

Posição

Executar

Exercício 1 - List

Operações

- ☐ Adicionar elemento
- ☐ Adicionar elemento em posição específica
- ☐ Remover elemento (pelo valor)
- ☐ Remover elemento (pela posição)
- ☒ Ordenar lista

Executar

2. Faça um programa Windows Forms em C# que apresente ao usuário as opções apresentadas na tela a seguir. Ao selecionar a opção **Adicionar Elemento**, seu respectivo campo de entrada deve ser apresentado ao usuário. Ao clicar no botão **Executar**, o programa deverá efetuar a operação escolhida. A cada operação realizada, apresente a fila de elementos atualizada no ListBox da parte inferior da tela. Use a classe `Queue<string>` para armazenar os elementos. Os métodos referentes a cada operação devem ser implementados em uma classe chamada **Fila**.

The screenshot shows a Windows Forms application titled "Exercício 2 - Queue". It features a group box labeled "Operações" containing two radio buttons: "Adicionar elemento" (which is selected) and "Remover elemento". Below this group box is a text box labeled "Elemento" and a button labeled "Executar". At the bottom of the window is a large empty rectangular area, likely intended for a ListBox.

This screenshot shows the same "Exercício 2 - Queue" application window, but with the "Remover elemento" radio button selected instead of "Adicionar elemento". The "Elemento" text box and the "Executar" button are still present, as is the empty area at the bottom.

3. Faça um programa Windows Forms em C# que apresente ao usuário as opções apresentadas na tela a seguir. Ao selecionar a opção **Adicionar Elemento**, seu respectivo campo de entrada deve ser apresentado ao usuário. Ao clicar no botão **Executar**, o programa deverá efetuar a operação escolhida. A cada operação realizada, apresente a pilha de elementos atualizada no ListBox da parte inferior da tela. Use a classe `Stack<string>` para armazenar os elementos. Os métodos referentes a cada operação devem ser implementados em uma classe chamada **Pilha**.

The screenshot shows a Windows Forms application titled "Exercício 3 - Stack". It has a group box labeled "Operações" with two radio buttons: "Adicionar elemento" (selected) and "Remover elemento". Below the group box is a text box labeled "Elemento" and a button labeled "Executar". The bottom of the window features a large empty rectangular area for a ListBox.

This screenshot shows the "Exercício 3 - Stack" application window with the "Remover elemento" radio button selected. The layout, including the "Elemento" text box, the "Executar" button, and the empty area at the bottom, remains the same as in the previous screenshot.