

# PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Disciplina: Algoritmos e Estruturas de Dados

Obs: Cópias serão desconsideradas, ou seja, a nota será igual a 0 (zero).

## Lista 6 – Tipos Abstratos de Dados Lineares

**Exemplo - Analise, entenda e execute o código a seguir:**

```
using System;
class Aluno
{
    private string nome;
    private int matricula;
    private char sexo;

    public Aluno(string nome, int matricula, char sexo)
    {
        this.nome = nome;
        this.matricula = matricula;
        this.sexo = sexo;
    }

    public string Nome
    {
        get { return nome; }
        set { nome = value; }
    }

    public int Matricula
    {
        get { return matricula; }
        set { matricula = value; }
    }

    public char Sexo
    {
        get { return sexo; }
        set { sexo = value; }
    }

    public override String ToString()
    {
        return "[nome=" + nome + ", matricula=" + matricula + ", sexo=" + sexo + "];"
    }
}
```

```
using System;
class Fila
{
    private Aluno[] array;
    private int primeiro;
    private int ultimo;

    /**
     * Construtor da classe.
     */
    public Fila(int tamanho)
    {
        array = new Aluno[tamanho + 1];
        primeiro = ultimo = 0;
    }
}
```

```

/**
 * Insere um elemento na ultima posicao da fila.
 */
public void Inserir(Aluno x)
{
    if (((ultimo + 1) % array.Length) == primeiro)
    {
        throw new Exception("Erro");
    }

    array[ultimo] = x;
    ultimo = (ultimo + 1) % array.Length;
}

/**
 * Remove um elemento da primeira posicao da fila e movimenta
 * os demais elementos para o primeiro da mesma.
 */
public Aluno Remover()
{
    if (primeiro == ultimo)
    {
        throw new Exception("Erro");
    }

    Aluno resp = array[primeiro];
    primeiro = (primeiro + 1) % array.Length;
    return resp;
}

/**
 * Mostra os elementos da fila
 */
public void Mostrar()
{
    int i = primeiro;
    Console.WriteLine("Fila: ");

    while (i != ultimo)
    {
        Console.WriteLine(array[i]);
        i = (i + 1) % array.Length;
    }
    Console.WriteLine();
}

/**
 * Pesquisa um elemento na fila
 */
public bool Pesquisar(int matricula)
{
    bool achou = false;
    for (int i = primeiro; i != ultimo && !achou; i = ((i + 1) % array.Length))
    {
        if (array[i].Matricula == matricula)
        {
            achou = true;
        }
    }
    return achou;
}
}

```

```

using System;

class Program
{
    static void Main(string[] args)
    {
        Fila filaEspera = new Fila(60);

        Aluno a1 = new Aluno("Maria", 2023100, 'f');
        Aluno a2 = new Aluno("Jose", 2023101, 'm');
        Aluno a3 = new Aluno("Joao", 2023102, 'm');

        filaEspera.Inserir(a1);
        filaEspera.Mostrar();

        filaEspera.Inserir(a2);
        filaEspera.Mostrar();

        filaEspera.Inserir(a3);
        filaEspera.Mostrar();

        bool result = filaEspera.Pesquisar(2023102);
        if(result)
        {
            Console.WriteLine("O aluno está na fila de espera.\n");
        }
        else
        {
            Console.WriteLine("O aluno NÃO está na fila de espera.\n");
        }

        Aluno temp = filaEspera.Remove();
        Console.WriteLine("Aluno removido:" + temp.Nome);
        filaEspera.Mostrar();

        temp = filaEspera.Remove();
        Console.WriteLine("Aluno removido:" + temp.Nome);
        filaEspera.Mostrar();
        Console.ReadLine();
    }
}

```

## Implemente as classes Lista, Fila e Pilha (usando arrays) e resolva os exercícios a seguir:

### Lista

1) Considere que um corredor profissional precisa de um programa para gerenciar os tempos (em horas) das maratonas que ele participou. O programa deve apresentar o seguinte menu para o usuário:

Menu:

- 1) Inserir um tempo no início da lista
- 2) Inserir um tempo no final da lista
- 3) Inserir um tempo numa posição específica da lista
- 4) Remover o primeiro tempo da lista (Imprimir o tempo removido)
- 5) Remover o último tempo da lista (Imprimir o tempo removido)
- 6) Remover um tempo de uma posição específica na lista (O usuário deve informar a posição do tempo a ser removido. Imprimir o tempo removido)
- 7) Remover um tempo específico da lista (O usuário deve informar o tempo a ser removido) **Adicione na classe Lista um método para remover um elemento específico: int RemoverItem(int x)**
- 8) Pesquisar quantas vezes um determinado tempo consta na lista (O usuário deve informar o tempo a ser pesquisado) **Adicione na classe Lista um método que retorne o número de elementos com o valor especificado: int Contar(int x)**

- 9) Mostrar todos os tempos da lista
- 10) Encerrar o programa

O programa deverá ler a opção informada pelo usuário e executar a operação selecionada. Em seguida o programa deverá apresentar novamente o menu, depois ler e executar a operação selecionada. Esse processo deverá ser repetido até que o usuário digite a opção de encerrar o programa

2) Implemente as seguintes classes:

- Crie uma classe Produto com os atributos exibidos abaixo:

```
class Produto
{
    private string nome;
    private int quant;
    private double preco;
}
```

- Implemente a classe Lista. Deverá ser implementada uma lista de Produtos.
- Implemente a classe Teste contendo o método Main. No método Main deve ser apresentado o seguinte menu para os usuários:

1. Inserir um produto no final da lista
2. Remover um produto específico da lista. **Adicione na classe Lista um método para remover um elemento específico. O método deverá receber o nome do produto e deverá retornar o objeto Produto, ou null (caso o produto não conste na lista): Produto RemoveItem(string nome)**
3. Listar os dados de todos os produtos
4. Pesquisar se um produto já consta na Lista. **Adicione na classe Lista um método para pesquisar um elemento específico. O método deverá receber o nome do produto: bool Pesquisar(string nome)**
4. Sair

O programa deve apresentar um menu para que o usuário informe a opção que deseja executar. O programa deve ser executado até que a opção de encerramento do programa seja escolhida pelo usuário.

## Pilha

**Adicione na classe Pilha um método para verificar se a pilha está vazia: bool isVazia()**

3) Escreva um programa que leia uma sequência (string) de parênteses e colchetes e verifique se essa sequência está bem-formada, ou seja, se os parênteses e colchetes são fechados na ordem inversa àquela em que foram abertos. Utilize uma pilha para auxiliar nessa verificação.

Exemplos:

- ( ( ) [ ( ) ] ) a sequência está bem-formada
- ( [ ] ] a sequência está malformada
- ( ) ] a sequência está malformada
- ( ) [ a sequência está malformada

Dica: os abre parênteses e abre colchetes devem ser empilhados. Quando vier um fecha parênteses deve ser desempilhado um caractere da pilha.

## Fila

4) Escreva um programa que simule o controle de uma pista de decolagem de aviões em um aeroporto. Neste programa, o usuário deve ser capaz de realizar as seguintes tarefas:

1. Listar a quantidade de aviões que estão aguardando na fila de decolagem **Adicione na classe Fila um método que retorna o número de elementos contidos na Fila: int ObterTamanho( )**
2. Autorizar a decolagem do primeiro avião da fila de decolagem (isto é, retirar da fila de colagem)
3. Adicionar um avião na fila de decolagem
4. Listar todos os aviões que estão na fila de decolagem
5. Exibir o primeiro avião da fila de decolagem **Adicione na classe Fila um método que retorne o primeiro elemento da Fila: int ObterPrimeiro( )**
6. Sair

Obs: A fila deve armazenar o identificador de cada avião (string).

5) Faça um programa que simule uma fila de impressão. Deverá ser apresentado para o usuário um menu com 4 opções: 1 = Inserir arquivo na fila de impressão, 2 = Executar impressão, 3 = Exibir fila de impressão e 4 = Sair. Quando o usuário selecionar a opção 1, ele deve fornecer o nome e o número de páginas do arquivo que deseja imprimir e este arquivo será inserido na fila de impressão, e ficará lá até que seja executada a sua impressão. Quando o usuário selecionar a opção 2, o primeiro arquivo da fila de impressão será “encaminhado” para a impressão, ou seja, será retirado da fila (você não precisa imprimir nada, apenas retire o pedido da fila de impressão). Se o usuário selecionar a opção 3, deve ser exibido na tela as informações de todos os arquivos que estão na fila de impressão. A opção 4 encerra a execução do programa.

Para desenvolvimento desse programa, crie as classes Arquivo, Fila e Teste.