

Algoritmos e técnicas de Programação - Aula 13

Aluno: Jean Carlo Machado Amaral

1-

```
using System;
public class Program
{
    public static void preencheNotas(double[] notas)
    {
        Console.WriteLine("Informe as notas: ");
        for (int i = 0; i < notas.Length; i++)
        {
            Console.WriteLine("Valor: ");
            notas[i] = double.Parse(Console.ReadLine());
        }
    }

    public static double mediaNotas(double[] notas)
    {
        int soma = 0;
        Console.WriteLine("Notas digitadas: ");
        for (int i = 0; i < notas.Length; i++)
        {
            Console.Write(notas[i] + " ");
        }

        for (int j = 0; j < notas.Length; j++)
        {
            soma += j;
        }

        double media = soma / notas.Length;
        return media;
    }

    public static double acimaMedia(double[] notas, double media)
    {
        int contador = 0;
        for (int i = 0; i < notas.Length; i++)
        {
            if (notas[i] > media)
            {
                contador++;
            }
        }
    }
}
```

```

        return contador;
    }

    public static void Main(string[] args)
    {
        double[] notas = new double[10];
        preencheNotas(notas);
        double media = mediaNotas(notas);
        double numeroAcimaMedia = acimaMedia(notas, media);

        Console.WriteLine("\n Média: " + media);
        Console.WriteLine("Quantidade de alunos acima da média: " +
numeroAcimaMedia);
    }
}

```

2-

```

using System;
public class Exercicio2
{
    public static void preencheMatrizes(int[,] matrizA, int[,] matrizB)
    {
        Random rd = new Random();
        for (int i = 0; i < matrizA.GetLength(0); i++)
        {
            for (int j = 0; j < matrizA.GetLength(1); j++)
            {
                matrizA[i, j] = rd.Next(0,101);
            }
        }

        for (int i = 0; i < matrizB.GetLength(0); i++)
        {
            for (int j = 0; j < matrizB.GetLength(1); j++)
            {
                matrizB[i, j] = rd.Next(0,101);
            }
        }
    }
}

```

```

        Console.WriteLine("matriz A: ");
        for (int i = 0; i < matrizA.GetLength(0); i++)
        {
            for (int j = 0; j < matrizA.GetLength(1); j++)
            {
                Console.Write(matrizA[i, j] + " ");
            } Console.WriteLine();
        }
        Console.WriteLine();

        Console.WriteLine("matriz B: ");
        for (int i = 0; i < matrizB.GetLength(0); i++)
        {
            for (int j = 0; j < matrizB.GetLength(1); j++)
            {
                Console.Write(matrizB[i, j] + " ");
            } Console.WriteLine();
        }
    }

    public static int[,] calculaSoma(int[,] matrizA, int[,] matrizB)
    {
        int [,] matrizSoma = new int [matrizA.GetLength(0),
matrizA.GetLength(1)];
        for (int i = 0; i < matrizA.GetLength(0); i++)
        {
            for (int j = 0; j < matrizB.GetLength(1); j++)
            {
                matrizSoma[i, j] = matrizA[i, j] + matrizB[i, j];
            }
        }
        return matrizSoma;
    }

    public static int[,] calculaDiferenca(int[,] matrizA, int[,]
matrizB)
    {
        int [,] matrizDiferenca = new int [matrizA.GetLength(0),
matrizA.GetLength(1)];
        for (int i = 0; i < matrizA.GetLength(0); i++)
        {
            for (int j = 0; j < matrizB.GetLength(1); j++)
            {
                matrizDiferenca[i, j] = matrizA[i, j] - matrizB[i, j];
            }
        }
    }
}

```

```

    }

    }

    return matrizDiferenca;
}

public static void Main(string[] args)
{
    int[,] matrizA = new int[4, 6];
    int[,] matrizB = new int[4, 6];
    preencheMatrizes(matrizA, matrizB);

    int[,] MatrizSoma = calculaSoma(matrizA, matrizB);
    int[,] MatrizDiferenca = calculaDiferenca(matrizA, matrizB);

    Console.WriteLine("\n Matriz Soma: ");
    for (int i = 0; i < matrizA.GetLength(0); i++)
    {
        for (int j = 0; j < matrizB.GetLength(1); j++)
        {
            Console.Write(MatrizSoma[i, j] + " ");
        } Console.WriteLine();
    }

    Console.WriteLine("\n Matriz Diferença: ");
    for (int i = 0; i < matrizA.GetLength(0); i++)
    {
        for (int j = 0; j < matrizB.GetLength(1); j++)
        {
            Console.Write(MatrizDiferenca[i, j] + " ");
        } Console.WriteLine();
    }
}
}

```

3-

```

using System;

public class Exercicio3
{
    public static void preencheMatriz(int[,] matriz)
    {
        Random rd = new Random();
        for (int i = 0; i < matriz.GetLength(0); i++)
        {
            for (int j = 0; j < matriz.GetLength(1); j++)

```

```

        {
            matriz[i, j] = rd.Next(0, 11);
        }
    }
}

public static void escreverMatriz(int[,] matriz) {
    Console.WriteLine("Matriz:");

    for (int i = 0; i < matriz.GetLength(0); i++)
    {
        for (int j = 0; j < matriz.GetLength(1); j++)
        {
            Console.Write(matriz[i, j] + " ");
        }
        Console.WriteLine();
    }
}

public static void primeiraMudanca(int[,] matriz, int linha1, int
linha2)
{
    int colunas = matriz.GetLength(1); //1 = coluna
    for (int j = 0; j < colunas; j++)
    {
        int temp = matriz[linha1, linha2];
        matriz[linha1, j] = matriz[linha2, j];
        matriz[linha2, j] = temp;
    }
}

public static void segundaMudanca(int[,] matriz, int coluna1, int
coluna2)
{
    int linhas = matriz.GetLength(0); //0= linha
    for (int i = 0; i < linhas; i++)
    {
        int temp = matriz[coluna1, coluna2];
        matriz[coluna1, i] = matriz[coluna2, i];
        matriz[coluna2, i] = temp;
    }
}

public static void inverteDiagonais(int[,] matriz)
{

```

```

        int tamanho = matriz.GetLength(0);
        for (int i = 0; i < tamanho; i++)
        {
            int temp = matriz[i, i];
            matriz[i, i] = matriz[i, tamanho - 1 - i];
            matriz[i, tamanho - 1 - i] = temp;
        }
    }

    public static void trocarLinhaComColuna(int[,] matriz, int linha,
int coluna)
    {
        int tamanho = matriz.GetLength(0);
        for (int i = 0; i < tamanho; i++)
        {
            int temp = matriz[i, linha];
            matriz[i, linha] = matriz[i, coluna];
            matriz[i, coluna] = temp;
        }
    }

    public static void Main(String[] args)
    {
        int[,] matriz = new int[10, 10];
        preencheMatriz(matriz);
        escreverMatriz(matriz);

        //trocar linha 2 com a 8
        primeiraMudanca(matriz, 1, 7);
        escreverMatriz(matriz);

        //trocar coluna 4 com a 10
        segundaMudanca(matriz, 3, 9);
        escreverMatriz(matriz);

        //inverte diagonal principal com secundaria
        inverteDiagonais(matriz);
        escreverMatriz(matriz);

        //inverte linha 5 com coluna 10
        trocarLinhaComColuna(matriz, 4, 9);
    }
}

```

```
}  
}
```

4-

```
using System;  
public class Exercicio4  
{  
  
    public static void preencheVetores(int[] vetorA, int[] vetorB)  
    {  
        Random rd = new Random();  
        for (int i = 0; i < vetorA.Length; i++)  
        {  
            vetorA[i] = rd.Next(0, 11);  
            vetorB[i] = rd.Next(0, 11);  
        }  
    }  
  
    public static void printaVetores(int[] vetor)  
    {  
        Console.WriteLine("Conteúdo do vetor:");  
        for (int i = 0; i < vetor.Length; i++)  
        {  
            Console.Write(vetor[i] + " ");  
        }  
        Console.WriteLine();  
    }  
  
    public static int[] alternaVetores(int[] vetorA, int[] vetorB)  
    {  
        int[] vetorIntercalado = new int[vetorA.Length + vetorB.Length];  
        int indexA = 0;  
        int indexB = 1;  
        for (int i = 0; i < vetorA.Length; i++)  
        {  
            vetorIntercalado[indexA] = vetorA[i];  
            vetorIntercalado[indexB] = vetorB[i];  
            indexA += 2;  
            indexB += 2;  
        }  
        return vetorIntercalado;  
    }  
}
```

```

public static void Main(string[] args)
{
    int[] vetorA = new int[10];
    int[] vetorB = new int[10];
    preencheVetores(vetorA, vetorB);
    Console.WriteLine("Vetor A: ");
    printaVetores(vetorA);
    Console.WriteLine("Vetor B: ");
    printaVetores(vetorB);

    int[] vetorIntercalado = alternaVetores(vetorA, vetorB);
    Console.WriteLine("\n Vetor intercalado: ");
    printaVetores(vetorIntercalado);

}
}

```

5-

Classe Aviao:

```

using System;
public class Aviao
{
    private int numeroVoo;
    private DateTime data;
    private int[] vagasOcupadas;

    public Aviao(int numeroVoo, DateTime data)
    {
        this.numeroVoo = numeroVoo;
        this.data = data;
        this.vagasOcupadas = new int[100];
    }

    public int ProximoLivre()
    {
        for (int i = 0; i < vagasOcupadas.Length; i++)
        {
            if (vagasOcupadas[i] == 0)
            {

```



```

        return i + 1;
    }

}

//se estiver todos ocupados
return -1;
}

public bool Verifica(int cadeira)
{
    return vagasOcupadas[cadeira - 1] != 0;
}

public bool Ocupa(int cadeira)
{
    if (vagasOcupadas[cadeira - 1] != 0)
    {
        return false;
    }
    else
    {
        vagasOcupadas[cadeira - 1] = 1;
        return true;
    }
}

public int Vagas()
{
    int vagasDisponiveis = 0;
    for (int i = 0; i < vagasOcupadas.Length; i++)
    {
        if (vagasOcupadas[i] == 0)
        {
            vagasDisponiveis++;
        }
    }
    return vagasDisponiveis;
}

public int GetVoo()
{
    return numeroVoo;
}
}

```

Main

```
using System;
public class Exercicio5
{
    public static void Main(string[] args)
    {
        Aviao voo = new Aviao(777, DateTime.Now); // .Now usa a hora
        atual

        Console.WriteLine("Número do voo: " + voo.GetVoo());
        Console.WriteLine("Próxima cadeira livre: " +
        voo.ProximoLivre());

        int cadeira = 3;
        if (voo.Verifica(cadeira))
        {
            Console.WriteLine("A cadeira " + cadeira + " está
            ocupada.");
        }
        else
        {
            bool ocupado = voo.Ocupa(cadeira);
            if (ocupado)
            {
                Console.WriteLine("A cadeira " + cadeira + " foi
                ocupada com sucesso.");
            }
            else
            {
                Console.WriteLine("A cadeira " + cadeira + " já está
                ocupada.");
            }
        }

        Console.WriteLine("Número de vagas disponíveis: " +
        voo.Vagas());
    }
}
```