

Aluno: **Jean Carlo Machado Amaral**

Aula 11 - Structs

### Questão 1.

```
using System;
using System.Diagnostics;

public struct Aluno
{
    public string Nome;
    public double NotaSemestre;
    public double NotaAvaliacaoEspecial;

    public double notaFinal()
    {
        return (NotaSemestre + NotaAvaliacaoEspecial)/2;
    }
}

public class Program1
{
    public static void Main(string[] args)
    {
        Aluno aluno1, aluno2;

        Console.WriteLine("Aluno 1");

        Console.WriteLine("Nome: ");
        aluno1.Nome = Console.ReadLine();

        Console.WriteLine("Nota do semestre: ");
        aluno1.NotaSemestre = double.Parse(Console.ReadLine());

        Console.WriteLine("Nota avaliação especial: ");
        aluno1.NotaAvaliacaoEspecial = double.Parse(Console.ReadLine());
        Console.WriteLine("\n" + "Nome: " + aluno1.Nome);
        Console.WriteLine("\n" + " Nota do semestre: " + aluno1.NotaSemestre);
        Console.WriteLine("\n" + "Nota da avaliação: " + aluno1.NotaAvaliacaoEspecial);
        Console.WriteLine("\n" + "Nota Final: " + aluno1.notaFinal());

        //Aluno 2

        Console.WriteLine("\n" + " Aluno 2");

        Console.WriteLine("Nome: ");
```

```

aluno2.Nome = Console.ReadLine();

Console.WriteLine("Nota do semestre: ");
aluno2.NotaSemestre = double.Parse(Console.ReadLine());

Console.WriteLine("Nota avaliação especial: ");
aluno2.NotaAvaliacaoEspecial = double.Parse(Console.ReadLine());
Console.WriteLine("\n" + "Nome: " + aluno2.Nome);
Console.WriteLine("\n" + " Nota do semestre: " + aluno2.NotaSemestre);
Console.WriteLine("\n" + "Nota da avaliação: " + aluno2.NotaAvaliacaoEspecial);
Console.WriteLine("\n" + "Nota Final: " + aluno2.notaFinal());
}
}

```

## Questao 2

```

using System;
public class Program2
{
    struct data
    {
        public string dia;
    }

    struct avaliacao
    {
        public string disciplina;
        public double valor;
        public double notaObtida;
    }

    public static void Main(string[] args)
    {
        data data1, data2;
        avaliacao prova1, prova2;

        //Aluno1
        Console.WriteLine("Data da avaliação: ");
        data1.dia = Console.ReadLine();
        Console.WriteLine("Informe a disciplina: ");
        prova1.disciplina = Console.ReadLine();
        Console.WriteLine("Quanto vale essa avaliação? ");
        prova1.valor = double.Parse(Console.ReadLine());
        Console.WriteLine("\n" + "Pontuação obtida: ");
        prova1.notaObtida = double.Parse(Console.ReadLine());

        Console.WriteLine("\n" + "Dia da avaliação: " + data1.dia);
    }
}

```

```

Console.WriteLine("\n" + "Disciplina da prova 1: " + prova1.disciplina);
Console.WriteLine("\n" + "Valor: " + prova1.valor);
Console.WriteLine("\n" + "Pontuação obtida: " + prova1.notaObtida);

```

```
//Aluno2
```

```

Console.Write("\n" + s"Segunda avaliação: ");
Console.WriteLine("Data da avaliação: ");
data2.dia = Console.ReadLine();
Console.WriteLine("Informe a disciplina: ");
prova2.disciplina = Console.ReadLine();
Console.WriteLine("Quanto vale essa avaliação? ");
prova2.valor = double.Parse(Console.ReadLine());
Console.WriteLine("\n" + "Pontuação obtida: ");
prova2.notaObtida = double.Parse(Console.ReadLine());

```

```

Console.WriteLine("\n" + "Dia da avaliação: " + data2.dia);
Console.WriteLine("\n" + "Disciplina da prova 1: " + prova2.disciplina);
Console.WriteLine("\n" + "Valor: " + prova2.valor);
Console.WriteLine("\n" + "Pontuação obtida: " + prova2.notaObtida);

```

```
}
```

```
}
```

### Questao 3

```
using System;
```

```
struct Contato
```

```

{
    public string nome;
    public string telefone;
    public string celular;
    public string email;
    public string dataAniversario;
}

```

```
public class AgendaTelefonica
```

```

{
    private Contato[] contatos;
    private int quantidadeContatos;
    private int tamanhoMaximo;

    public AgendaTelefonica(int tamanhoMaximo)
    {
        this.tamanhoMaximo = tamanhoMaximo;
    }
}

```

```

    contatos = new Contato[tamanhoMaximo];
    quantidadeContatos = 0;
}

public void CadastrarContato()
{
    if (quantidadeContatos < tamanhoMaximo)
    {
        Contato novoContato;

        Console.WriteLine("Nome do contato: ");
        novoContato.nome = Console.ReadLine();

        Console.WriteLine("Telefone: ");
        novoContato.telefone = Console.ReadLine();

        Console.WriteLine("Celular: ");
        novoContato.celular = Console.ReadLine();

        Console.WriteLine("E-mail:");
        novoContato.email = Console.ReadLine();

        Console.WriteLine("Aniversário: (DD/MM)");
        novoContato.dataAniversario = Console.ReadLine();

        contatos[quantidadeContatos] = novoContato;
        quantidadeContatos++;

        Console.WriteLine("Cadastro concluído");
    }
    else
    {
        Console.WriteLine("A Agenda está cheia, não é possível cadastrar!!");
    }
}

public void EditarContato(string nome)
{
    int indice = EncontrarContatoPorNome(nome);

    if (indice != -1)
    {
        Console.WriteLine("Digite o novo telefone do contato:");
        contatos[indice].telefone = Console.ReadLine();

        Console.WriteLine("Digite o novo celular do contato:");
        contatos[indice].celular = Console.ReadLine();
    }
}

```

```

        Console.WriteLine("Digite o novo e-mail do contato:");
        contatos[indice].email = Console.ReadLine();

        Console.WriteLine("Digite a nova data de aniversário do contato (DD/MM):");
        contatos[indice].dataAniversario = Console.ReadLine();

        Console.WriteLine("Contato atualizado com sucesso.");
    }
    else
    {
        Console.WriteLine("Contato não encontrado.");
    }
}

public void RemoverContato(string nome)
{
    int indice = EncontrarContatoPorNome(nome);

    if (indice != -1)
    {
        for (int i = indice; i < quantidadeContatos - 1; i++)
        {
            contatos[i] = contatos[i + 1];
        }

        quantidadeContatos--;

        Console.WriteLine("Contato removido com sucesso.");
    }
    else
    {
        Console.WriteLine("Contato não encontrado.");
    }
}

public void BuscarContatoPorNome(string nome)
{
    int indice = EncontrarContatoPorNome(nome);

    if (indice != -1)
    {
        ImprimirContato(contatos[indice]);
    }
    else
    {
        Console.WriteLine("Contato não encontrado.");
    }
}

```

```

public void BuscarContatoPorCelular(string celular)
{
    int indice = EncontrarContatoPorCelular(celular);

    if (indice != -1)
    {
        ImprimirContato(contatos[indice]);
    }
    else
    {
        Console.WriteLine("Contato não encontrado.");
    }
}

public void ImprimirTodosContatos()
{
    if (quantidadeContatos > 0)
    {
        Console.WriteLine("Lista de contatos:");

        for (int i = 0; i < quantidadeContatos; i++)
        {
            ImprimirContato(contatos[i]);
            Console.WriteLine();
        }
    }
    else
    {
        Console.WriteLine("A agenda está vazia. Não há contatos cadastrados.");
    }
}

private int EncontrarContatoPorNome(string nome)
{
    for (int i = 0; i < quantidadeContatos; i++)
    {
        if (contatos[i].nome.Equals(nome, StringComparison.OrdinalIgnoreCase))
        {
            return i;
        }
    }

    return -1;
}

private int EncontrarContatoPorCelular(string celular)
{

```

```

        for (int i = 0; i < quantidadeContatos; i++)
        {
            if (contatos[i].celular.Equals(celular))
            {
                return i;
            }
        }

        return -1;
    }

    private void ImprimirContato(Contato contato)
    {
        Console.WriteLine("Nome: " + contato.nome);
        Console.WriteLine("Telefone: " + contato.telefone);
        Console.WriteLine("Celular: " + contato.celular);
        Console.WriteLine("E-mail: " + contato.email);
        Console.WriteLine("Data de aniversário: " + contato.dataAniversario);
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        AgendaTelefonica agenda = new AgendaTelefonica(5);
        int opcao = 0;
        while (opcao != 6)
        {
            Console.WriteLine("Escolha uma opção:");
            Console.WriteLine("1 - Cadastrar contato");
            Console.WriteLine("2 - Editar contato");
            Console.WriteLine("3 - Remover contato");
            Console.WriteLine("4 - Buscar contato por nome");
            Console.WriteLine("5 - Buscar contato por celular");
            Console.WriteLine("6 - Sair");
            opcao = int.Parse(Console.ReadLine());

            switch (opcao)
            {
                case 1:
                    agenda.CadastrarContato();
                    break;
                case 2:
                    Console.WriteLine("Nome do contato a ser editado:");
                    string nomeEditar = Console.ReadLine();
                    agenda.EditarContato(nomeEditar);
                    break;
            }
        }
    }
}

```

```

        case 3:
            Console.WriteLine("Nome do contato a ser removido:");
            string nomeRemover = Console.ReadLine();
            agenda.RemoverContato(nomeRemover);
            break;
        case 4:
            Console.WriteLine("Nome do contato a ser buscado:");
            string nomeBuscar = Console.ReadLine();
            agenda.BuscarContatoPorNome(nomeBuscar);
            break;
        case 5:
            Console.WriteLine("Número do celular a ser buscado:");
            string celularBuscar = Console.ReadLine();
            agenda.BuscarContatoPorCelular(celularBuscar);
            break;
        case 6:
            Console.WriteLine("Saindo...");
            break;
        default:
            Console.WriteLine("Opção inválida");
            break;
    }

    Console.WriteLine();
}
}
}

```

#### Questão 4

```
using System;
```

```
struct Veiculo
```

```
{
    public string Placa;
    public TimeSpan TempoEntrada; // pesquisei e encontrei essa estrutura que permite fazer
operacoes com valores de tempo (TimeSpan)
    public TimeSpan TempoSaida;
}
```

```
public class Estacionamento
```

```
{
    private Veiculo[] veiculos;
    private int quantidadeVeiculos;
    private const double ValorPorMinuto = 0.10;

    public Estacionamento(int tamanhoMaximo)
    {

```



```

        veiculos = new Veiculo[tamanhoMaximo];
        quantidadeVeiculos = 0;
    }

    public void CadastrarEntrada()
    {
        if (quantidadeVeiculos < veiculos.Length)
        {
            Veiculo novoVeiculo = new Veiculo();

            Console.WriteLine("Placa do veículo: ");
            novoVeiculo.Placa = Console.ReadLine();

            Console.WriteLine("Tempo de entrada (horas minutos):");
            string[] tempoEntrada = Console.ReadLine().Split(' ');
            int horas = int.Parse(tempoEntrada[0]);
            int minutos = int.Parse(tempoEntrada[1]);
            novoVeiculo.TempoEntrada = new TimeSpan(horas, minutos, 0);

            veiculos[quantidadeVeiculos] = novoVeiculo;
            quantidadeVeiculos++;

            Console.WriteLine("Entrada cadastrada com sucesso.");
        }
        else
        {
            Console.WriteLine("O estacionamento está cheio. Não é possível cadastrar uma nova entrada.");
        }
    }

    public void RegistrarSaida()
    {
        Console.WriteLine("Placa do veículo a ser retirado:");
        string placa = Console.ReadLine();

        Veiculo veiculo = BuscarVeiculoPorPlaca(placa);

        if (veiculo.Placa != null)
        {
            Console.WriteLine("Tempo de entrada: " + veiculo.TempoEntrada);
            Console.WriteLine("Tempo de saída (horas minutos):");
            string[] tempoSaida = Console.ReadLine().Split(' ');
            int horas = int.Parse(tempoSaida[0]);
            int minutos = int.Parse(tempoSaida[1]);
            veiculo.TempoSaida = new TimeSpan(horas, minutos, 0);

            double valorPago = CalcularValorAPagar(veiculo);

```

```

        Console.WriteLine("Valor a ser pago: R$ " + valorPago.ToString("F2"));
    }
    else
    {
        Console.WriteLine("Veículo não cadastrado.");
    }
}

private Veiculo BuscarVeiculoPorPlaca(string placa)
{
    Veiculo veiculoNaoEncontrado = new Veiculo();

    for (int i = 0; i < quantidadeVeiculos; i++)
    {
        if (veiculos[i].Placa.Equals(placa, StringComparison.OrdinalIgnoreCase))
        {
            return veiculos[i];
        }
    }

    return veiculoNaoEncontrado;
}

private double CalcularValorAPagar(Veiculo veiculo)
{
    TimeSpan tempoEstacionado = veiculo.TempoSaida - veiculo.TempoEntrada;
    double minutosEstacionados = tempoEstacionado.TotalMinutes;
    double valorPago = minutosEstacionados * ValorPorMinuto;
    return valorPago;
}
}

public class Program
{
    public static void Main(string[] args)
    {
        Estacionamento estacionamento = new Estacionamento(10);
        int opcao = 0;

        while (opcao != 3)
        {
            Console.WriteLine("Escolha uma opção:");
            Console.WriteLine("1 - Cadastrar entrada de veículo");
            Console.WriteLine("2 - Registrar saída de veículo");
            Console.WriteLine("3 - Sair");
            opcao = int.Parse(Console.ReadLine());

            switch (opcao)

```

```
{
    case 1:
        estacionamento.CadastrarEntrada();
        break;
    case 2:
        estacionamento.RegistrarSaida();
        break;
    case 3:
        Console.WriteLine("Saindo...");
        break;
    default:
        Console.WriteLine("Opção inválida");
        break;
}

Console.WriteLine();
}
}
```