Enumerações e Estruturas

? EXERCÍCIO 1: ENUMERAÇÃO PARA SUBALFABETO

```
using System;
namespace EE1
{public class SubAlfabeto
 {enum Letras:int {B=66, P=80, Passo=2};
  static void Main(string[] args)
  {string Subalfabeto = "";
   for (int I=(int)Letras.B; I<=(int)Letras.P;
   I+=(int)Letras.Passo)
    {Subalfabeto += (char)I; }
     Console.WriteLine(Subalfabeto);}}}
```

? EXERCÍCIO 2: ENUMERAÇÃO PARA HORÁRIO SEMANAL

```
using System;
namespace EE2
{public class HorarioSemanal
 {enum Semana
  {Domingo = 1,
   Segunda,
   Terça,
   Quarta,
   Quinta,
   Sexta,
   Sábado};
 static void Main(string[] args)
  {for (int D = (int)Semana.Domingo; D<=(int)Semana.Sábado;D++)
   {if (D % 2 == 0)
      Console.WriteLine("{0} - {1}", (Semana)D, "Ginástica");
      else
      Console.WriteLine("{0} - {1}", (Semana) D, "Ioga");}}}}
```

? EXERCÍCIO 3: ENUMERAÇÃO PARA TOTAL MENSAL

```
using System;
namespace EE3
{public class TotaisMensais
 {enum Meses
```

```
  {Jan = 0,Fev,Mar,Abr,Mai,Jun,Jul,Ago,Set,Out,Nov,Dez};
 static void Main(string[] args)
  {double[] Vendas={1000,1200,500,300,400,600,700,800,900,1000,
                    1200,2000};
   double Soma = 0;
   for(int i=(int) Meses.Mai; i<=(int) Meses.Ago; i++)
    {Soma+=Vendas[i]; }
   Console.WriteLine("Total de Vendas de {0} a {1}={2}",
   Meses.Mai, Meses.Ago,Soma);}}}
```

? EXERCÍCIO 4: ENUMERAÇÃO PARA MENU DE OPÇÕES

```
using System;
namespace EE4
{public class MenuOpcoes
 {enum Bebidas
  {Chá = 1,
   Café = 2,
   Água = 3,
   Laranjada = 6,
   CocaCola = 7,
   Porto = 4,
   Champanhe = 5};
 static void Main(string[] args)
  {int Opcao = 0;
   Bebidas[] Menu = new Bebidas[4];
   for (int i = 0; i <= 3; i++)
    Menu[i] = (Bebidas) i+1;
   do
    {Console.Clear();
     foreach (Bebidas Op in Menu)
      Console.WriteLine((int)Op + ". " + Op.ToString());
     Console.Write("Qual a sua opção? ");
     Opcao = Convert.ToInt16(Console.ReadLine());
   } while (Opcao<1 || Opcao>Menu.Length);
   Console.WriteLine("Escolheu " + (Bebidas)Opcao);}}}
```

? EXERCÍCIO 5: ENUMERAÇÕES PARA DEPÓSITOS BANCÁRIOS

```
using System;
namespace EE5
 {public enum Limites
  {L1=2,
   L2=5}
```

```
 public enum Taxas
  {A = 20,
   B = 24,
   C = 30}
public class Deposito
{private double Capital;
 private int Duracao;
 private Taxas Tx;
 public Deposito(double C, int N)
  {Capital = C;
   Duracao = N;
   if (N < (int) Limites.L1)
   Tx = Taxas.A;
   else
    if (N < (int)Limites.L2)
    Tx = Taxas.B;
    else
    Tx = Taxas.C;}
 public Taxas PTx
  {get
   {return Tx; }}
 public double Acumulado()
  {double t = (double)(int)Tx / 1000;
   return Capital*Math.Pow((1+t), Duracao);}}

public class DepositosBancarios
 {static void Main(string[] args)
  {Console.Write("Duração do depósito em anos=");
   int N =Convert.ToInt16(Console.ReadLine());
   Deposito D = new Deposito(1000, N);
   Console.WriteLine("Escalão={0} e Taxa={1}% ", D.PTx,
                     ((double)(int)D.PTx/10));
   Console.WriteLine("Capital acumulado={0, 10:F2} euros",
                     D.Acumulado());}}}
```

? EXERCÍCIO 6: ESTRUTURA-VALOR E CLASSE-REFERÊNCIA

```
using System;
namespace EE6
 {public struct ColegasS
  {private string Nome;
   public string PNome
    {get
     {return Nome; }
```

```
     set
      {Nome = value;}}}

 public class ColegasC
  {private string Nome;
  public ColegasC(string N)
  {Nome = N; }
  public string PNome
   {get
     {return Nome;}
   set
     {Nome = value;}}

 public class ValorEReferencia
  {static void Main(string[] args)
   {ColegasS S = new ColegasS();
    S.PNome="Joana Silva";
    ColegasS S1 = S;
    S1.PNome = "Rui Alves";
    Console.WriteLine("Nomes de S e S1: {0} e {1} ", S.PNome,
    S1.PNome);
    ColegasC C = new ColegasC("Teresa Pinto");
    ColegasC C1 = C;
    C1.PNome = "Pedro Moita";
    Console.WriteLine("Nomes de C e C1: {0} e {1}", C.PNome,
                      C1.PNome);}}}}
```

? EXERCÍCIO 7: ESTRUTURA DE CONDÓMINOS

```
using System;
namespace EE7
 {public struct Condomino
  {private string Nome;
   private double Mensalidade;
   private bool Pagou;
   public Condomino(string N, double M)
    {Nome = N;
     Mensalidade = M;
     Pagou = false;}

   public Condomino(string N, double M, bool P)
     {Nome = N;
     Mensalidade = M;
     Pagou = P;}
   public string PNome
    {get
```

```
      {return Nome; }
     set
      {Nome = value; }}
   public double PMensal
    {get
     {return Mensalidade; }
    set
     {Mensalidade = value; }}
   public bool PPagou
    {get
     {return Pagou; }
    set
     {Pagou = value; }}
   public override string ToString()
    {string R = "";
     if (Pagou == false)
      R = String.Format("O condómino {0} tem {1} euros deste mês
      por pagar ", Nome, Mensalidade);
      else
      R = String.Format("O condómino {0} já pagou {1} euros ",
      Nome, Mensalidade);
    return R;}}

 public class Condominios
  {static void Main(string[] args)
   {Condomino C = new Condomino("A. Ruelas", 140);
    Console.WriteLine(C.ToString());
    C = new Condomino("B. Brochado", 120, true);
    Console.WriteLine(C.ToString());}}}
```

? EXERCÍCIO 8: VETOR DE ESTRUTURAS

```
using System;
namespace EE8
 {public struct Condomino
  {private string Nome;
   private double Mensalidade;
   private bool Pagou;
   public Condomino(string N, double M)
   {Nome = N;
    Mensalidade = M;
    Pagou = false;}
  public Condomino(string N, double M, bool P)
   {Nome = N;
    Mensalidade = M;
    Pagou = P;}
```

```
  public string PNome
   {get
    {return Nome;}
   set
    {Nome = value;}}
  public double PMensal
   {get
    {return Mensalidade;}
   set
    {Mensalidade = value;}}
  public bool PPagou
   {get
    {return Pagou; }
   set
    {Pagou = value;}}
  public override string ToString()
   {string R = "";
    if (Pagou == false)
     R = String.Format("O condómino {0} tem {1} deste mês por
     pagar ", Nome, Mensalidade);
     else
     R = String.Format("O condómino {0} já pagou {1} ", Nome,
     Mensalidade);
    return R;}}

public class PorCobrar
 {static void Main(string[] args)
  {string N; double M; bool P;
   Console.Write("Número de condóminos ");
   int Ncond = Convert.ToInt16(Console.ReadLine());
   Condomino[] C = new Condomino[Ncond];
   for (int I=0; I<=C.Length-1; I++)
    {Console.Write("Nome do condómino ");
     N = Console.ReadLine();
     Console.Write("Mensalidade ");
     M = Convert.ToDouble(Console.ReadLine());
     Console.Write("Pagou (True or False)");
     P = Convert.ToBoolean(Console.ReadLine());
     C[I] = new Condomino(N, M, P);}
   double Tot=0;
   for (int I = 0; I <= C.Length - 1; I++)
    if (C[I].PPagou == false)
     Tot += C[I].PMensal;
   Console.WriteLine("Montante a cobrar {0} euros ", Tot);}}}
```

? EXERCÍCIO 9: ESTRUTURA COM SOBREPOSIÇÃO DE MÉTODO

```
using System;
namespace EE9
 {public struct Turma
  {private char desig;
   private int nr;
   private int nrz;
   public Turma(char desig, int nr, int nrz)
    {this.desig = desig;
     this.nr = nr;
     this.nrz = nrz;}
   public int numraparigas
    {get
     {return nr;}}
   public int numrapazes
    {get
     {return nrz; }}
   public override string ToString()
    {return String.Format("A turma {0} tem {1} alunos",desig,
              nr + nrz);}}

 public class TotalAlunos
  {static void Main(string[] args)
   {Turma T= new Turma('A', 10, 20);
    Console.WriteLine(T.ToString());
    T = new Turma('B', 50, 10);
    Console.WriteLine(T.ToString());}}}
```

? EXERCÍCIO 10: VETOR DE ESTRUTURAS E ENUMERAÇÃO

```
using System;
namespace EE10
 {public struct Jogo
  {private int EQ1;
   private string EQ2;
   private char Resultado;
   public static int Njogos = 0;
   public Jogo(int EQ1, string EQ2, char R)
   {this.EQ1 = EQ1;
    this.EQ2 = EQ2;
    Resultado = R;
    Njogos++;}
   public int PEquipa
    {get
```

```
      {return EQ1;}}
    public char PResultado
     {get
      {return Resultado; }}}
public class PontuacaoEquipas
 {enum Equipas : int { FCP = 1, SLB = 2, SCP = 3 };
 static void Main(string[] args)
  {string Adv;
   char Result; int I = 0; Jogo[] J = new Jogo[10];
   for (int E = (int) Equipas.FCP; E <= (int )Equipas.SCP; E++)
   {Console.WriteLine("Jogos do {0}", (Equipas) E);
    Console.Write("Nome da equipa adversária (Fim para terminar)
    ");
    Adv = Console.ReadLine();
    while (Adv.ToUpper().CompareTo("FIM")!=0)
     {Console.Write("Resultado (V, E ou D)");
      Result = Convert.ToChar(Console.ReadLine());
      J[I] = new Jogo(E, Adv, Result);
      I++;
      Console.Write("Nome da equipa adversária (Fim para
      terminar) ");
      Adv = Console.ReadLine();}}
    Console.WriteLine("{0}\t{1}","Equipa", "Pontuação");
    int Tot, Pontos; I=0;
    for (int E = (int)Equipas.FCP; E <= (int)Equipas.SCP; E++)
     {Tot=0;
     while (J[I].PEquipa== E)
      {switch (J[I].PResultado)
       {case 'E':
        Pontos = 1;
        break;
       case 'V':
        Pontos=3;
        break;
       default:
        Pontos=0;
        break;}
      Tot+=Pontos;
      I++;}
     Console.WriteLine("{0}\t{1}",(Equipas) E, Tot); }
    Console.WriteLine("Número de jogos efetuados: {0}",
    Jogo.Njogos);}}}
```