



Classes e Objetos

? EXERCÍCIO 1: CLASSE ESTUDANTE DE INFORMÁTICA

```
public class EstudanteInf
{private string Nome;
  private double Teste1;
  private double Teste2;
  public EstudanteInf(String N, double T1, double T2)
  {Nome = N;
    Teste1 = T1;
    Teste2 = T2;}
  public double ClassFinal()
  {return Math.Round((Teste1 + Teste2)/2,0,
    MidpointRounding.AwayFromZero);}}
```

? EXERCÍCIO 2: CLASSIFICAÇÃO FINAL DE UM ALUNO

```
using System;
namespace CO2
{public class EstudanteInf
  {private string Nome;
    private double Teste1;
    private double Teste2;
    public EstudanteInf(string N, double T1, double T2)
    {Nome = N;
      Teste1 = T1;
      Teste2 = T2;}
    public double ClassFinal()
    {return Math.Round((Teste1 + Teste2)/2,0,
      MidpointRounding.AwayFromZero);}}

public class ClassificacaoFinal
{static void Main(string[] args)
  {EstudanteInf A = new EstudanteInf("Joana", 13, 14);
    Console.WriteLine("Classificação final= "+ A.ClassFinal());}}
```

? EXERCÍCIO 3: LEITURA DO NOME DO ALUNO

```
using System;
namespace CO3
{public class EstudanteInf
    {private string Nome;
      private double Teste1;
      private double Teste2;
      public EstudanteInf(String N, double T1, double T2)
      {Nome = N;
        Teste1 = T1;
        Teste2 = T2;}
      public double ClassFinal()
      {return Math.Round((Teste1+Teste2)/2,0,
        MidpointRounding.AwayFromZero);}
      public string LerNome()
      {return Nome;}}

public class NomeENota
{static void Main(string[] args)
    {EstudanteInf A = new EstudanteInf("Joana Pinto", 13, 14);
      Console.WriteLine("{0,-20} {1,-10}", "Aluno", "Classificação");
      Console.WriteLine("{0,-20} {1,8}", A.LerNome(), A.ClassFinal());
    }}}
```

? EXERCÍCIO 4: OUTRO CONSTRUTOR E ACESSOR

```
using System;
namespace CO4
{public class EstudanteInf
    {private string Nome;
      private double Teste1;
      private double Teste2;
      public EstudanteInf()
      {Nome = "";
        Teste1 = 0;
        Teste2 = 0;}
      public EstudanteInf(String N, double T1, double T2)
      {Nome = N;
        Teste1 = T1;
        Teste2 = T2;}
      public double ClassFinal()
      {return Math.Round((Teste1 + Teste2)/2, 0,
        MidpointRounding.AwayFromZero); }
      public string LerNome()
```

```
        {return Nome; }
    public void EscreverNome(String N)
    {Nome = N; }
    public void EscreverTeste1(int T1)
    {Teste1 = T1; }
    public void EscreverTeste2(int T2)
    {Teste2 = T2; }}

public class AlunosSemNomeENotas
{static void Main(string[] args)
{EstudanteInf A = new EstudanteInf();
  A.EscreverNome("Abel Paiva");
  A.EscreverTeste1(12);
  A.EscreverTeste2(14);
  Console.WriteLine("{0,-20} {1,-10}", "Aluno", "Classificação");
  Console.WriteLine("{0,-20} {1,8}", A.LerNome(), A.ClassFinal());
}}}
```

? EXERCÍCIO 5: IMPRESSÃO DE VARIÁVEIS DE INSTANTE

```
using System;
namespace CO5
{public class EstudanteInf
{private string Nome;
  private double Teste1;
  private double Teste2;
  public EstudanteInf()
  {Nome = "";
   Teste1 = 0;
   Teste2 = 0;}
  public EstudanteInf(String N, double T1, double T2)
  {Nome = N;
   Teste1 = T1;
   Teste2 = T2;}
  public double ClassFinal()
  {return Math.Round((Teste1 + Teste2) / 2,
    0, MidpointRounding.AwayFromZero); }
  public string LerNome()
  {return Nome; }
  public void EscreverNome(String n)
  {Nome = n;}
  public void EscreverTeste1(int t1)
  {Teste1 = t1;}
  public void EscreverTeste2(int t2)
  {Teste2 = t2;}
  public void Impressao()
  {
  }
  }
```

```
{Console.WriteLine("{0,-20} {1,-8} {2, -8} {3, -10}",
    "Aluno", "Teste1", "Teste2", "Classificação");
Console.WriteLine("{0,-10}{1,12}{2, 8}{3, 14}", Nome, Teste1,
    Teste2, ClassFinal());}}
```

```
public class ImpressaoVarInstante
{static void Main(string[] args)
{EstudanteInf A = new EstudanteInf("Teresa Paiva",12, 13);
    A.Impressao();}}
```

? EXERCÍCIO 6: VETOR DE OBJETOS

```
using System;
namespace CO6
{public class EstudanteInf
{private string Nome;
    private double Teste1;
    private double Teste2;
    public EstudanteInf()
    {Nome = "";
        Teste1 = 0;
        Teste2 = 0;}
    public EstudanteInf(string N, double T1, double T2)
    {Nome = N;
        Teste1 = T1;
        Teste2 = T2;}
    public double ClassFinal()
    {return Math.Round((Teste1 + Teste2) / 2, 0,
        MidpointRounding.AwayFromZero); }
    public string LerNome()
    {return Nome; }
    public void AtribuirNome(string n)
    {Nome = n; }
    public void AtribuirTeste1(int t1)
    {Teste1 = t1; }
    public void AtribuirTeste2(int t2)
    {Teste2 = t2;}}
```

```
public class PautaDeInformatica
{static void Main(string[] args)
{Console.Write("Quantos alunos há?");
    int N= Convert.ToInt16(Console.ReadLine());
    EstudanteInf[] A = new EstudanteInf[N];
    for (int I = 0; I <= N - 1; I++)
    {Console.Write("Nome do aluno? ");
        A[I] = new EstudanteInf();
        A[I].AtribuirNome(Console.ReadLine());
    }
}
```

```
        Console.WriteLine("Nota do teste1? ");
        A[I].AtribuirTeste1(Convert.ToInt16(Console.ReadLine()));
        Console.WriteLine("Nota do teste2? ");
        A[I].AtribuirTeste2(Convert.ToInt16(Console.ReadLine()));}
Console.Clear();
Console.WriteLine("{0,-20} {1,-10}", "Aluno",
                  "Classificação");
for (int I = 0; I <= N - 1; I++)
    Console.WriteLine("{0,-20} {1,8}", A[I].LerNome(),
                        A[I].ClassFinal());}}
```

? EXERCÍCIO 7: LEITURA DE VARIÁVEIS DE INSTANTE

```
public class EstudanteInf
{private string Nome;
 private double Teste1;
 private double Teste2;
 public EstudanteInf()
 {Nome = "";
  Teste1 = 0;
  Teste2 = 0;}
 public EstudanteInf(String N, double T1, double T2)
 {Nome = N;
  Teste1 = T1;
  Teste2 = T2;}
 public double ClassFinal()
 {return Math.Round((Teste1 + Teste2)/2,0,
  MidpointRounding.AwayFromZero);}
 public string Pnome
 {get
 {return Nome;}
 set
 {Nome = value; }}
 public double Pteste1
 {get
 {return Teste1; }
 set
 {Teste1 = value; }}
 public double Pteste2
 {get
 {return Teste2; }
 set
 {Teste2 = value; }}}}
```



? EXERCÍCIO 8: NOVA PAUTA DE INFORMÁTICA

```
using System;
namespace CO8
{
    public class EstudanteInf
    {
        private string Nome;
        private double Teste1;
        private double Teste2;
        public EstudanteInf()
        {
            Nome = "";
            Teste1 = 0;
            Teste2 = 0;
        }
        public EstudanteInf(String N, double T1, double T2)
        {
            Nome = N;
            Teste1 = T1;
            Teste2 = T2;
        }
        public double ClassFinal()
        {
            return Math.Round((Teste1 + Teste2)/2,0,
                MidpointRounding.AwayFromZero);
        }
        public string Pnome
        {
            get
            {
                return Nome;
            }
            set
            {
                Nome = value;
            }
        }
        public double Pteste1
        {
            get
            {
                return Teste1;
            }
            set
            {
                Teste1 = value;
            }
        }
        public double Pteste2
        {
            get
            {
                return Teste2;
            }
            set
            {
                Teste2 = value;
            }
        }
    }

    public class PautaInformativaVersao2
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Quantos alunos há?");
            int N = Convert.ToInt16(Console.ReadLine());
            EstudanteInf[] A = new EstudanteInf[N];
            for (int I = 0; I <= N - 1; I++)
            {
                A[I] = new EstudanteInf();
                Console.WriteLine("Nome do aluno? ");
                A[I].Pnome=Console.ReadLine();
                Console.WriteLine("Nota do teste1? ");
                A[I].Pteste1=Convert.ToInt16(Console.ReadLine());
                Console.WriteLine("Nota do teste2? ");
            }
        }
    }
}
```

```
A[I].Pteste2=Convert.ToInt16(Console.ReadLine());}
Console.Clear();
Console.WriteLine("{0,-10} {1,-8} {2, -8} {3, -10}", "Aluno",
"Teste1", "Teste2", "Classificação");
for (int I = 0; I<=N - 1; I++)
    Console.WriteLine("{0,-10}{1,4}{2, 9}{3, 15}", A[I].Pnome,
    A[I].Pteste1, A[I].Pteste2, A[I].ClassFinal());}}
```

? EXERCÍCIO 9: PAUTA DE APROVADOS

```
using System;
namespace CO9
{public class EstudanteInf
{private string Nome;
private double Teste1;
private double Teste2;
public EstudanteInf()
{Nome = "";
Teste1 = 0;
Teste2 = 0;}
public EstudanteInf(String N, double T1, double T2)
{Nome = N;
Teste1 = T1;
Teste2 = T2;}
public string Pnome
{get
{return Nome;}
set
{Nome = value;}}
public double Pteste1
{get
{return Teste1;}
set
{Teste1 = value;}}
public double Pteste2
{get
{return Teste2;}
set
{Teste2 = value;}}
public double ClassFinal()
{return Math.Round((Teste1 + Teste2)/2,0,
MidpointRounding.AwayFromZero);}
public string SituacaoDoAluno()
{string Sit;
double CF = ClassFinal();
if (CF >= 10)
```

```
Sit = "Aprovado";
else
    if (CF >= 8)
        Sit = "Oral";
    else
        Sit = "Reprovado";
return Sit;}}

public class PautaAprovados
{static void Main(string[] args)
{Console.Write("Quantos alunos há? ");
int N = Convert.ToInt16(Console.ReadLine());
EstudanteInf[] A = new EstudanteInf[N];
for (int I = 0; I<=N - 1; I++)
{Console.Write("Nome do aluno? ");
A[I] = new EstudanteInf();
A[I].Pnome=Console.ReadLine();
Console.Write("Nota do testel? ");
A[I].Ptestel=Convert.ToInt16(Console.ReadLine());
Console.Write("Nota do teste2? ");
A[I].Pteste2=Convert.ToInt16(Console.ReadLine());}
Console.Clear();
Console.WriteLine("{0,-10}", "Aluno");
for (int I = 0; I<= N - 1; I++)
if (A[I].SituacaoDoAluno() == "Aprovado")
    Console.WriteLine("{0,-10} ", A[I].Pnome);}}}
```

? EXERCÍCIO 10: CLASSIFICAÇÃO MÉDIA DA TURMA

```
using System;
namespace CO10
{public class EstudanteInf
{private string Nome;
private double Testel;
private double Teste2;
public EstudanteInf()
{Nome = "";
Testel = 0;
Teste2 = 0;}
public EstudanteInf(String N, double T1, double T2)
{Nome = N;
Testel = T1;
Teste2 = T2;}
public double ClassFinal()
{return Math.Round((Testel + Teste2) / 2, 0,
System.MidpointRounding.AwayFromZero );}}
```



```
        public string Pnome
        {
            get
            {
                return Nome;
            }
            set
            {
                Nome = value;
            }
        }
        public double Ptest1
        {
            get
            {
                return Teste1;
            }
            set
            {
                Teste1 = value;
            }
        }
        public double Pteste2
        {
            get
            {
                return Teste2;
            }
            set
            {
                Teste2 = value;
            }
        }
        private static double SomatorioCL=0;
        private static int Nalunos=0;
        public void SomaNota()
        {
            SomatorioCL+=ClassFinal();
            Nalunos++;
        }
        public static double CalcMedia()
        {
            return Math.Round(SomatorioCL/Nalunos, 2,
                System.MidpointRounding.AwayFromZero);
        }
    }

    public class MediaDaTurma
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Quantos alunos há?");
            int N = Convert.ToInt16(Console.ReadLine());
            EstudanteInf[] A = new EstudanteInf[N];
            for (int I = 0; I <= N - 1; I++)
            {
                Console.WriteLine("Nome do aluno? ");
                A[I] = new EstudanteInf();
                A[I].Pnome=Console.ReadLine();
                Console.WriteLine("Nota do test1? ");
                A[I].Ptest1=Convert.ToInt16(Console.ReadLine());
                Console.WriteLine("Nota do teste2? ");
                A[I].Pteste2=Convert.ToInt16(Console.ReadLine());
                A[I].SomaNota();
            }
            Console.WriteLine("Classif. média da turma="+
                EstudanteInf.CalcMedia());
        }
    }
}
```

? EXERCÍCIO 11: CONTAGEM DAS CLASSIFICAÇÕES POSITIVAS

```
using System;
namespace CO11
{
    public class EstudanteInf
```

```
{private string Nome;
private double Teste1;
private double Teste2;
private static int Npositivas=0;
public EstudanteInf()
{Nome = "";
Teste1 = 0;
Teste2 = 0;}
public EstudanteInf(String N, double T1, double T2)
{Nome = N;
Teste1 = T1;
Teste2 = T2;}
public double ClassFinal()
{return Math.Round((Teste1+Teste2)/2,0,
MidpointRounding.AwayFromZero);}
public string Pnome
{get
{return Nome; }
set
{Nome = value; }}
public double Pteste1
{get
{return Teste1; }
set
{Teste1 = value; }}
public double Pteste2
{get
{return Teste2; }
set
{Teste2 = value; }}
static double SomatorioCL=0;
static int Nalunos=0;
public void SomaNotas()
{SomatorioCL+=ClassFinal();
Nalunos++;}
public static double CalcMedia()
{return Math.Round(SomatorioCL/Nalunos, 2);}
public void IncPositivas()
{if (Teste1>= 10 || Teste2>=10 )
Npositivas++;}
public static int LerPositivas()
{return Npositivas;}}
public class ContagemPositivas
{static void Main(string[] args)
{Console.Write("Quantos alunos há? ");
int N = Convert.ToInt16(Console.ReadLine());
EstudanteInf[] A = new EstudanteInf[N];
for (int I = 0; I <= N - 1; I++)
{Console.Write("Nome do aluno?");
```

```
A[I] = new EstudanteInf();
A[I].Pnome = Console.ReadLine();
Console.Write("Nota do teste1?");
A[I].Pteste1 = Convert.ToInt16(Console.ReadLine());
Console.Write("Nota do teste2?");
A[I].Pteste2 = Convert.ToInt16(Console.ReadLine());
A[I].IncPositivas();
Console.WriteLine("{0} alunos tiveram positiva pelo menos "+
" num dos testes", EstudanteInf.LerPositivas());}}
```

? EXERCÍCIO 12: CLASSIFICAÇÃO E CONTAGEM

```
using System;
namespace CO12
{public class Pessoa
{private string Nome;
private int Idade;
private char Genero;
public static int[] Conta = new int[4];
public Pessoa()
{Nome = "";
Idade = 0;
Genero = ' ';}
public Pessoa(String N, int Id, char G)
{Nome = N;
Idade = Id;
Genero = G;}
public void Classifica()
{if (Idade < 18 )
{if (Genero == 'F')
Conta[0]++;
else
Conta[1]++;}
else
{if (Genero == 'F')
Conta[2]++;
else
Conta[3]++;}}
public static void AfixaContadores()
{string[] G = { "Rapariga", "Rapaz", "Mulher", "Homem" };
Console.WriteLine();
for (int I = 0; I <= G.Length - 1; I++)
Console.WriteLine("{0} --- {1}", G[I], Conta[I]);}

public class ClassificaEconta
{static void Main(string[] args)
```

```
{String Nome;
int Idade;
char Genero;
Pessoa P;
Console.Write("Nome ou ZZZ para terminar? ");
Nome = Console.ReadLine().ToUpper();
while (Nome != "ZZZ")
{Console.Write("Idade? ");
Idade = Convert.ToInt16(Console.ReadLine());
Console.Write("Gênero F ou M? ");
Genero = Convert.ToChar(Console.ReadLine());
P = new Pessoa(Nome, Idade, Genero);
Console.Write("Nome ou ZZZ para terminar? ");
Nome = Console.ReadLine().ToUpper();
P.Classifica();}
Pessoa.AfixaContadores();}}
```

? EXERCÍCIO 13: ÁREA E AMPLIAÇÃO DE TRIÂNGULOS

```
using System;
namespace CO13
{public class Triangulo
{private double Base;
private double Altura;
public Triangulo()
{Base = 0;
Altura = 0;}
public Triangulo(double B, double H)
{Base = B;
Altura = H;}
public double Area()
{return Base * Altura / 2; }
public double[] Ampliacao(double X)
{double[] Medidas=new double[2];
Medidas[0]=Base*X;
Medidas[1]=Altura*X;
return Medidas;}}

public class AreaEAmpliacao
{static void Main(string[] args)
{Triangulo T = new Triangulo(4, 5);
Console.WriteLine("Area ={0}", T.Area());
double[] Med = new double[2];
Med = T.Ampliacao(3);
Console.WriteLine("Base ampliada={0}", Med[0]);
Console.WriteLine("Altura ampliada={0}", Med[1]);}}
```

? EXERCÍCIO 14: TEMPERATURAS EM CELSIUS E FAHRENHEIT

```
using System;
namespace CO14
{public class Temperatura
    {private double C;
     private double F;
     public Temperatura()
     {C = 0;
      F = 32;}
     public Temperatura(double C)
     {this.C = C;
      F = C * 1.8+ 32;}
     public void ImprimirTemp()
     {Console.WriteLine("{0} Celsius = {1} Fahrenheit", C, F);}}

public class Temperaturas
    {static void Main(string[] args)
     {Temperatura T = new Temperatura(20);
      T.ImprimirTemp();}}
```

? EXERCÍCIO 15: VIAGEM

```
using System;
namespace CO15
{public class Viagem
    {private double KmInicio;
     private double LInicio;
     private double KmFim;
     private double LFim;
     public Viagem()
     {KmInicio = 0;
      KmFim = 0;
      LInicio = 0;
      LFim = 0;}
     public Viagem(double KmIn, double LIn)
     {KmInicio = KmIn;
      LInicio = LIn;
      KmFim = 0;
      LFim = 0;}
     public void LeiturasFinais(double KmFi, double LFi)
     {KmFim = KmFi;
      LFim = LFi;}}
```

```
public double ConsumoLitros()
{return (LInicio-LFim)* 100/(KmFim-KmInicio);}
public double ConsumoValor(double Preco)
{return ConsumoLitros() * Preco; }}

public class Viagens
{static void Main(string[] args)
{Viagem V = new Viagem (10000, 45);
 V.LeiturasFinais(12000, 5);
 Console.WriteLine ("Consumo médio aos 100={0} litros",
 V.ConsumoLitros());
 Console.WriteLine ("Consumo médio aos 100={0} Euros",
 V.ConsumoValor(1.1));}}}
```

? EXERCÍCIO 16: CONSTRUTOR COM VALIDAÇÃO DE DADOS

```
using System;
namespace COL6
{public class Aluno
{private string Nome;
 private int Nota;
 private static int Inf;
 private static int Sup;
 public Aluno()
 {Nome = " ";
  Nota = 0;}
 public Aluno(string N)
 {Nome = N;
  int X=0;
  bool Valida = false;
  while (Valida == false)
  {try
  {Console.Write("Digite a nota do aluno "+ Nome+" ");
   X = Convert.ToInt16(Console.ReadLine());
   Valida = X >= Inf && X <= Sup;
   if (Valida == false)
    throw new Exception();}
  catch
  {Console.WriteLine("A nota é um número inteiro entre {0} e
   {1}", Inf, Sup);}}
  Nota = X;}
 public static void AtribuicaoLimites(int I, int S)
 {Inf=I;
  Sup=S;}
 public string NomeENota()
 {return Nome + " obteve a classificação " + Nota; }}
```

```
public class ClassificacaoValida
{
    static void Main(string[] args)
    {
        Aluno.AtribuicaoLimites(0, 20);
        Console.Write("Digite o nome do aluno ");
        String N = Console.ReadLine();
        Aluno A = new Aluno(N);
        Console.WriteLine(A.NomeENota());
    }
}
```

? EXERCÍCIO 17: NAIPE DE CARTAS

```
public class CartasDeJogar
{
    private int Naipe;
    private int Carta;
    public CartasDeJogar()
    {
        Naipe = 0;
        Carta = 0;
    }
    public CartasDeJogar(int NR, int VR)
    {
        Naipe = NR;
        Carta = VR;
    }
    static string[] Np={"Ouros", "Espadas", "Paus", "Copas"};
    static string[] Vp = {"2","3","4","5","6","7","Valete","Dama",
                          "Rei"};

    public string Pnaipe
    {
        get
        {
            return Np[Naipe];
        }
    }
    public string Pcarta
    {
        get
        {
            return Vp[Carta];
        }
    }
}
```

? EXERCÍCIO 18: EXTRAÇÃO DE DUAS CARTAS ALEATÓRIAS

```
using System;
namespace CO18
{
    public class CartasDeJogar
    {
        private int Naipe;
        private int Carta;
        public CartasDeJogar()
        {
            Naipe = 0;
            Carta = 0;
        }
        public CartasDeJogar(int NR, int VR)
        {
            Naipe = NR;
            Carta = VR;
        }
    }
}
```

```
static string[] Np={"Ouros", "Espadas", "Paus", "Copas"};
static string[] Vp = {"2","3","4","5","6","7","Valete","Dama",
                    "Rei"};

public string Pnaipe
{
    get
    {
        return Np[Naipe];
    }
}
public string Pcarta
{
    get
    {
        return Vp[Carta];
    }
}

public class CartasAleatorias
{
    static void Main(string[] args)
    {
        Random R = new Random();
        CartasDeJogar[] Extraidas=new CartasDeJogar[2];
        for (int i = 0; i <= 1; i++)
        {
            Extraidas[i] = new CartasDeJogar(R.Next(4), R.Next(9));
            Console.WriteLine(Extraidas[i].Pcarta + " de " +
                              Extraidas[i].Pnaipe);
        }
    }
}
```

? EXERCÍCIO 19: DEPÓSITOS E LEVANTAMENTOS – 1.ª VERSÃO

```
using System;
namespace CO19
{
    public class ContaCorrente
    {
        private double Valor;
        private char Transacao;
        private static double Saldo = 0;
        private static int NumTrans = 0;
        public ContaCorrente(double V, char T)
        {
            Valor = V;
            Transacao = T;
            NumTransESaldo();
        }
        public void NumTransESaldo()
        {
            NumTrans++;
            if (Transacao == 'D')
                Saldo += Valor;
            else
                Saldo -= Valor;
        }
        public void ImpressaoTransESaldo()
        {
            Console.WriteLine("Transação número{0}:", NumTrans);
            if (Transacao == 'D')
                Console.WriteLine("{0} de {1} euros", "Depósito", Valor);
            else
                Console.WriteLine("{0} de {1} euros", "Levantamento",
                                   Valor);
        }
    }
}
```



```
        Console.WriteLine("Saldo={0} euros", Saldo);}}  
public class Contas  
{static void Main(string[] args)  
{ContaCorrente C1 = new ContaCorrente(100, 'D');  
C1.ImpressaoTransESaldo();  
ContaCorrente C2 = new ContaCorrente(50, 'L');  
C2.ImpressaoTransESaldo();  
ContaCorrente C3 = new ContaCorrente(1000, 'D');  
C3.ImpressaoTransESaldo();  
ContaCorrente C4 = new ContaCorrente(500, 'L');  
C4.ImpressaoTransESaldo();}}}
```

? EXERCÍCIO 20: DEPÓSITOS E LEVANTAMENTOS – 2.ª VERSÃO

```
using System;  
namespace CO20  
{public class ContaCorrente2  
{private double Valor;  
private char Transacao;  
private bool Valida;  
private static double Saldo = 0;  
private static int NumTrans = 0;  
public ContaCorrente2(double V, char T)  
{if (Validar(V, T) == true)  
{Valida = true;  
NumTrans++;  
Valor = V;  
Transacao = T;  
NumTransESaldo();}}}  
public static bool Validar(double V, char T)  
{if (T == 'L' && V > Saldo)  
return false;  
else  
return true;}  
public void NumTransESaldo()  
{if (Transacao == 'D')  
Saldo += Valor;  
else  
Saldo -= Valor;}  
public void ImpressaoTransESaldo()  
{if (Valida == false)  
Console.WriteLine("Transação anulada", NumTrans);  
else  
{Console.WriteLine("Transação número {0}: ", NumTrans);  
if (Transacao == 'D')  
Console.WriteLine("{0} de {1} euros", "Depósito", Valor);  
if (Transacao == 'L')  
Console.WriteLine("{0} de {1} euros", "Levantamento", Valor);  
}}}
```

```
        else
            Console.WriteLine("{0} de {1} euros", "Levantamento",
                Valor);
        Console.WriteLine("Saldo={0} euros", Saldo);}}

public class Contas2Versao
{
    static void Main(string[] args)
    {
        ContaCorrente2 C1 = new ContaCorrente2(100, 'D');
        C1.ImpressaoTransESaldo();
        ContaCorrente2 C2 = new ContaCorrente2(1000, 'L');
        C2.ImpressaoTransESaldo();
        ContaCorrente2 C3 = new ContaCorrente2(200, 'D');
        C3.ImpressaoTransESaldo();
        ContaCorrente2 C4 = new ContaCorrente2(50, 'L');
        C4.ImpressaoTransESaldo();}}}
```

? EXERCÍCIO 21: SOBREPOSIÇÃO DE MÉTODOS

```
using System;
namespace C021
{
    public class Amigo
    {
        private string Nome;
        private string Apelido;
        public Amigo(string N, string A)
        {
            Nome = N;
            Apelido = A;
        }
        public override string ToString()
        {
            return Apelido + ", " + Nome;
        }
    }

    public class ApelidoENome
    {
        static void Main(string[] args)
        {
            Amigo A = new Amigo("Nelson", "Évora");
            Console.WriteLine(A.ToString());
        }
    }
}
```

? EXERCÍCIO 22: SOBRECARGA DE MÉTODOS

```
using System;
namespace C022
{
    public class RaizQuad
    {
        public RaizQuad()
        {
        }
        public int RaizQ (int x)
        {
            return (int) Math. Sqrt(x);
        }
    }
}
```

```
public double RaizQ(double x)
{return Math.Round(Math.Sqrt(x),2);}}

public class RaizesQuadradas
{static void Main(string[] args)
{RaizQuad R=new RaizQuad();
int x=49;
double y=22.50;
Console.WriteLine("Raiz quadrada de {0}={1}", x,R.RaizQ(x));
Console.WriteLine("Raiz quadrada de {0}={1}", y, R.RaizQ(y));
}}}
```

? EXERCÍCIO 23: COMPARAÇÃO DE OBJETOS

```
using System;
namespace CO23
{public class Carro
{private string Nomeprop;
private int Ano;
private double NumKm;
public Carro()
{ }
public Carro(string N, int A, int K)
{Nomeprop = N;
Ano = A;
NumKm = K;}
public string Comparacao(int AnoCorrente, Carro Outro)
{string MaisKm = "O carro de " +this.Nomeprop;
MaisKm += " tem maior quilometragem média" ;
double EsteMedKm = this.NumKm / (AnoCorrente - this.Ano);
double OutroMedKm=Outro.NumKm/(AnoCorrente-Outro.Ano);
if (EsteMedKm < OutroMedKm)
{MaisKm = "O carro de " +Outro.Nomeprop;
MaisKm+=" tem maior quilometragem média." ;}
else if (EsteMedKm==OutroMedKm )
MaisKm="Os dois carros têm a mesma quilometragem
média.";
return MaisKm;}}

public class ComparacaoDeCarros
{static void Main(string[] args)
{Carro A=new Carro ("Joana Costa", 2000, 10000);
Carro B=new Carro ("Aníbal Pinto", 1998, 15000);
Console.WriteLine(A.Comparacao(2010, B));}}}
```

? EXERCÍCIO 24: CLASSIFICAÇÃO MÉDIA

```
using System;
namespace CO24
{public class Disciplina
{private string Desig;
private static int SomaNotas;

private static int NumAlunos;
public Disciplina(string Disc, int N)
{Desig = Disc;
SomaNotas += N;
NumAlunos++;}
public static double MediaTotal()
{return (double)SomaNotas / NumAlunos;}}

public class MediaInfMat
{static void Main(string[] args)
{Console.Write("Digite uma nota de Informática(-1 para terminar)
");
int Nota= Convert.ToInt16(Console.ReadLine());
while (Nota >= 0)
{Disciplina I = new Disciplina("Informática", Nota);
Console.Write("Digite uma nota de Informática(-1 para
terminar) ");
Nota = Convert.ToInt16(Console.ReadLine());}
Console.Write("Digite uma nota de Matemática(-1 para terminar)
");
Nota = Convert.ToInt16(Console.ReadLine());
while (Nota >= 0)
{Disciplina M = new Disciplina("Matemática", Nota);
Console.Write("Digite uma nota de Matemática(-1 para terminar)
");
Nota = Convert.ToInt16(Console.ReadLine());}
Console.WriteLine("Classificação média das duas disciplinas "
+"{0, 2:F2}", Disciplina.MediaTotal());}}}
```

? EXERCÍCIO 25: FORNO E PRATO

```
using System;
namespace CO25
{public class Forno
{private double TempForno;
public Forno()
{}}
```

```
        public Forno(double TF)
        {TempForno =TF ;}
        public double TForno()
        {return TempForno;}}
public class Prato
{private double TempPrato;
 private int SegNoForno;
 public Prato()
 {}
 public Prato(double TP, int S)
 {TempPrato = TP;
  SegNoForno = S;}
 public double TFinalPrato(double TF)
 {return TempPrato+=(TF-TempPrato)/10*SegNoForno;}}
public class PratoNoForno
{static void Main(string[] args)
{Forno F = new Forno (250);
 Prato P = new Prato (15, 5);
 Console.WriteLine("Temperatura final do prato={0}C",
                   P.TFinalPrato(F.TForno()));}}}
```