



Herança e Polimorfismo

? EXERCÍCIO 1: CENTRO E CIRCULO

```
public class Centro
{
    private int x, y;
    public Centro()
    {
        x = 0;
        y = 0;
    }
    public Centro(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
    public void CoodCentro()
    {
        Console.WriteLine("Centro = ({0}, {1})", x, y);
    }
}

public class Circulo : Centro
{
    private double Raio;
    public Circulo()
    {
    }
    public Circulo(int x, int y, double R): base(x, y)
    {
        Raio = R;
    }
    public double Area()
    {
        return Math.Round(Math.PI * Math.Pow(Raio, 2), 2);
    }
}
```

? EXERCÍCIO 2: VETOR DE CÍRCULOS

```
using System;
namespace CHP02
{
    public class Centro
    {
        private int x, y;
        public Centro()
        {
            x = 0;
            y = 0;
        }
        public Centro(int x, int y)
        {
            this.x = x;
            this.y = y;
        }
        public void CoodCentro()
        {
            Console.WriteLine("Centro=({0}, {1})", x, y);
        }
    }

    public class Circulo : Centro
    {
        private double Raio;
        public Circulo()
        {
        }
    }
}
```



```

        public Circulo(int x, int y, double R): base(x, y)
        {Raio = R;}
        public double Area()
        {return Math.Round(Math.PI * Math.Pow(Raio, 2), 3); }}

public class FazerCirculos
{static void Main(string[] args)
    {int N=4;
      Circulo[] C = new Circulo[N];
      int X=0, Y=0;
      double R=0.5;
      for (int I = 0; I <C.Length; I++)
      {C[I] = new Circulo(X, Y, R);
       C[I].CoordCentro();
       Console.WriteLine("Área={0}", C[I].Area());
       X++;
       Y++;
       R*=2 ;}}}}

```

? EXERCÍCIO 3: CONSTRUTORES E DESTRUIDORES

```

using System;
namespace CO3
{public class Escola
    {private string Nome;
      private String Morada;
      public Escola(String N, string M)
      {Nome = N;
       Morada = M;
       Console.WriteLine("Instanciei escola:{0}", Nome);}
    ~Escola()
    {Console.WriteLine("Destruí escola: {0}", Nome);}
    public string EscNome
    {get
      {return Nome;}
     set
      {Nome = value;}}
    public string EscMorada
    {get
      {return Morada;}
     set
      {Morada = value;}}}

public class Aluno : Escola
    {private String Nome;
      private int Nota;

```

```
public Aluno(string Escola, string Local, string N, int T1, int
T2): base(Escola, Local)
{
    Nome = N;
    Nota = (int)((T1 + T2) / 2);
    Console.WriteLine("Instanciei aluno: {0}", Nome);
}

~Aluno()
{
    Console.WriteLine("Destruí aluno: {0}", Nome);
}

public override string ToString()
{
    return Nome+" da escola "+base.EscNome+"(" +
        base.EscMorada+")"+ " obteve "+ Nota +" Valores";
}

public class DestruicaoObjectos
{
    static void Main(string[] args)
    {
        string[,] Nomes= {{"Teresa", "ABC"}, {"Aníbal", "Rio Tinto"},
        {"Rui", "Francos"}, {"Vítor", "Bolhão"}, {"Joana", "Bom
        Sucesso"}};
        int[,] Testes= {{10,12}, {11, 10}, {12,12}, {8, 12}, { 14,
        15}};
        Aluno[] A = new Aluno[5];
        for (int i=0 ; i<=A.Length-1; i++)
        {
            A[i]=new Aluno(Nomes[i,1], "Porto", Nomes[i,0],
            Testes[i,0], Testes[i,1]);
            Console.WriteLine(A[i].ToString());
        }
        A=null;
        System.GC.Collect();
    }
}
```

? EXERCÍCIO 4: PESSOA, AMIGO E COLEGA

```
public class Pessoa
{
    private string Nome;
    private int Tel;
    public Pessoa()
    {
        Nome = "";
        Tel = 0;
    }
    public Pessoa(string n, int t)
    {
        Nome = n;
        Tel = t;
    }
    public string Pnome
    {
        get
        {
            return Nome;
        }
    }
    public int Ptel
    {
        get
        {
            return Tel;
        }
    }
}

public class Amigo : Pessoa
```

```
{private string Onde;
private string Quando;
public Amigo(string n, int t, string o, String q): base(n,t)
{Onde = o;
Quando = q;}
public string Ponde
{get
{return Onde;}}
public string Pquando
{get
{return Quando;}}}}

public class Colega : Pessoa
{private string Local;
private string Profissao;
public Colega(string n, int t, string o, string p): base(n,t)
{Local = "";
Profissao = p;}
public string Plocal
{get
{return Local;}}
public string Pprofissao
{get
{return Profissao;}}}}
```

? EXERCÍCIO 5: LISTA DE AMIGOS E COLEGAS

```
using System;
namespace HP05
{public class Pessoa
{private string Nome;
private int Tel;
public Pessoa()
{Nome = "";
Tel = 0;}
public Pessoa(string n, int t)
{Nome = n;
Tel = t;}
public string Pnome
{get
{return Nome; }}
public int Ptel
{get
{return Tel;}}}}

public class Amigo : Pessoa
```

```
{private string Onde;
private string Quando;
public Amigo(string n, int t, string o, String q): base(n,t)
{Onde = o;
Quando = q;}
public string Ponde
{get
{return Onde;}}
public string Pquando
{get
{return Quando;}}

public class Colega : Pessoa
{private string Local;
private string Profissao;
public Colega(string n, int t, string o, string p): base(n,t)
{Local = "";
Profissao = p;}
public string Plocal
{get
{return Local;}}
public string Pprofissao
{get
{return Profissao;}}

public class DuasListas
{static void Main(string[] args)
{string[] NomesA={"Teresa G.", "Rui O.", "Ida J."};
string[] NomesC={"Rui Alves", "J. Pereira"};
int[] TelefA={2134567877, 22368977, 071345677};
int[] TelefC={22345677, 2112345677};
string[] OndeA={"Lancs", "Kent", "Warsaw"};
string[] OndeC={"Porto", "Lisboa"};
string[] QuandoA={"2007", "2002", "1993"};
string[] ProfissaoC={"Economista", "Engenheiro"};
int TA=NomesA.Length;
int TC=NomesC.Length;
Amigo[] A = new Amigo[3];
Colega[] C = new Colega[3];
Console.WriteLine("Lista de amigos");
for (int I = 0; I < TA; I++)
A[I]=new Amigo(NomesA[I],TelefA[I],OndeA[I],QuandoA[I]);
for (int I = 0; I < TC; I++)
C[I]=new Colega(NomesC[I],TelefC[I],OndeC[I],ProfissaoC[I]);
for (int I = 0; I < TA; I++)
Console.WriteLine(A[I].Pnome.PadRight(15) +
A[I].Ponde.PadRight(20) );
Console.WriteLine("Lista de colegas");
for (int I = 0; I < TC; I++)
```

```
Console.WriteLine(C[I].Pnome.PadRight(15) +  
C[I].Pprofissao.PadRight(20));}}}
```

? EXERCÍCIO 6: TOTAL DE VENDAS

```
using System;  
namespace HP06  
{public class Vendas  
{private static double ValorTotal = 0;  
public static double ActualizarVendas(double x)  
{return ValorTotal += x; }  
public static double GetValorTotal()  
{return ValorTotal; }}  
  
public class VendasProdA : Vendas  
{private int Quant;  
private double ValorUnit;  
public VendasProdA(int q, double v)  
{Quant = q;  
ValorUnit = v;  
ActualizarVendas(v * q);}}  
  
public class VendasProdB : Vendas  
{private double ValorTrans;  
public VendasProdB(double v)  
{ValorTrans = v;  
ActualizarVendas(v);}}  
  
public class TotalVendas  
{static void Main(string[] args)  
  
{VendasProdA PA1 = new VendasProdA(4, 5);  
VendasProdB PB1 = new VendasProdB(300);  
VendasProdA PA2 = new VendasProdA(5, 7);  
VendasProdA PA3 = new VendasProdA(10, 4);  
VendasProdB PB2 = new VendasProdB(100);  
Console.WriteLine("Total das vendas={0}",  
Vendas.GetValorTotal());}}}
```

? EXERCÍCIO 7: MORADIAS E APARTAMENTOS

```
using System;  
namespace HP07  
{public class Propriedade
```

```
{private string Proprietario;
private int NFiscal;
public Propriedade(string N, int NF)
{Proprietario=N;
NFiscal=NF;}
public override string ToString ()
{return Proprietario + " ---"+ NFiscal+"---";}
public int RendaMinima(double Area)
{double Renda;
const double P1 = 0.20;
const double P2 = 0.30;
if (Area < 50)
Renda = 100;
else if (Area < 100)
Renda= 100 + P1 * Area;
else Renda = 100 + P2 * Area;
return (int)Math.Round(Renda, 0);}}

public class Moradia : Propriedade
{private string Local;
private char Categoria;
public Moradia(string N, int NF, string L): base(N, NF)
{Local = L;
CategMoradia();}
public void CategMoradia()
{switch (Local)
{case "Lisboa":
Categoria = 'A';
break;
case "Porto":
Categoria = 'B';
break;
case "Coimbra":
Categoria = 'C';
break;
case "Braga":
Categoria = 'D';
break;}}
public int RendaMinima()
{int Renda=0;
switch (Categoria)
{case 'A':
case 'B':
Renda=1000;
break;
case 'C':
Renda=600;
break;
case 'D':
```

```
        Renda= 450;
        break;}
    return Renda;}}

public class Apartamento : Propriedade
{private string Tipo;
 private double Area;
 public Apartamento(string N,int NF,string T,double A): base(N,
 NF)
 {Tipo = T;
  Area = A;}
 public double LerArea()
 {return Area;}}

public class PropriedadesParaArrendar
{static void Main(string[] args)
 {Moradia M = new Moradia("J & P", 50090050, "Lisboa");

  Apartamento Ap = new Apartamento("G & H",50080060,"T2",120);
  Console.WriteLine(M.ToString()+"Renda Mínima="+
    M.RendaMinima());
  Console.WriteLine(Ap.ToString() + "Renda Mínima="
    +Ap.RendaMinima(Ap.LerArea()));}}}
```

? EXERCÍCIO 8: DATAS DE NASCIMENTO

```
using System;
namespace HP08
{public class DataCronologica
 {private int Dia, Mes, Ano;
  public DataCronologica( int D, int M, int A)
  {Dia = D;
   Mes = M;
   Ano = A;}
 public int Anos
 {get
  {return Ano; }
 set
  {Ano = value; }}
 public int Meses
 {get
  {return Mes; }
 set
  {Mes = value; }}
 public int Dias
 {get
```



```
        {return Dia; }
    set
        {Dia = value;}}}}

public class Pessoa
{private string Nome;
    public Pessoa(string N)
        {Nome = N;}

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

public class Amigos : Pessoa
{private DataCronologica Nasc;
    public Amigos (string N, DataCronologica D): base (N)
        {Nasc=D;}
    public DataCronologica Nascimento
        {get
            {return Nasc;}
        set
            {Nasc = value;}}}}

public class AniversariosDoisAmigos
{static void Main(string[] args)
    {String N1 = "Sara Tavares";
    String N2 = "António Ribeiro";
    Amigos A1 = new Amigos(N1, new DataCronologica(15,5,1996));
    Amigos A2=new Amigos(N2, new DataCronologica(15, 12, 1986));
    Impressao(A1);
    Impressao(A2);}
private static void Impressao(Amigos A)
    {Console.WriteLine(A.Nomes);
    DataCronologica Ddata = A.Nascimento;
    Console.WriteLine (Ddata.Dias+"/"+Ddata.Meses+"/"+ +
        Ddata.Anos);}}}
```

? EXERCÍCIO 9: ANOS E DIAS

```
using System;
namespace HP09
{public class DataCronologica
    {private int Dia;
    private int Mes;
```

```
private int Ano;
public DataCronologica(int D, int M, int A)
{
    Dia = D;
    Mes = M;
    Ano = A;
}
public int Anos
{
    get
    {
        return Ano;
    }
    set
    {
        Ano = value;
    }
}
public int Meses
{
    get
    {
        return Mes;
    }
    set
    {
        Mes = value;
    }
}
public int Dias
{
    get
    {
        return Dia;
    }
    set
    {
        Dia = value;
    }
}
public int Idade(int C)
{
    return C - Ano;
}
public int Idade(DataCronologica C)
{
    int Inic = DiaDoAno(this);
    int Fim = DiaDoAno(C);
    int TD = 0;
    if (this.Ano == C.Ano)
        TD = Fim - Inic;
    else
    {
        TD = 365 - DiaDoAno(this);
        if (Bissextto(this.Ano) == true)
            TD++;
        for (int I = this.Ano + 1; I <= C.Ano - 1; I++)
        {
            TD += 365;
            if (Bissextto(I) == true)
                TD++;
        }
        TD += Fim;
    }
    return TD;
}
public static int DiaDoAno(DataCronologica X)
{
    int TD = 0;
    int[] DiaMeses = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (Bissextto(X.Ano) == true)
        DiaMeses[1] = DiaMeses[1] + 1;
    for (int M = 0; M < X.Mes - 1; M++)
        TD += DiaMeses[M];
    TD += X.Dia;
    return TD;
}
public static bool Bissextto(int A)
{
    return ((A % 4 == 0 && A % 100 != 0) || (A % 400 == 0));
}
```

```
        return true;
    else
        return false;}}

public class AnosEDias
{
    static void Main(string[] args)
    {
        DateTime Hoje = System.DateTime.Now;
        DataCronologica Nasc = new DataCronologica(15, 5, 1996);
        DataCronologica C = new DataCronologica(Hoje.Day,
        Hoje.Month, Hoje.Year);
        Console.WriteLine("Faz {0} anos. Nasceu em {1}",
        Nasc.Idade(C.Anos), Nasc.Dias + "/" + Nasc.Meses + "/" +
        Nasc.Anos);
        Console.WriteLine("Até hoje (não inclusive) {0} decorreram
        {1} dias ", C.Dias + "/" + C.Meses + "/" + C.Anos,
        Nasc.Idade(C));}}}
```

? EXERCÍCIO 10: IDADE DOS AMIGOS

```
using System;
namespace HP10
{
    public class Pessoa
    {
        private string Nome;
        public Pessoa(string N)
        {
            Nome = N;
        }
        public string Nomes
        {
            get
            {
                return Nome;
            }
            set
            {
                Nome = value;
            }
        }
    }

    public class Amigos : Pessoa
    {
        private DataCronologica Nasc;
        public Amigos(string N, DataCronologica D) : base(N)
        {
            Nasc = D;
        }
        public DataCronologica Nascimento
        {
            get
            {
                return Nasc;
            }
            set
            {
                Nasc = value;
            }
        }
    }

    public class DataCronologica
    {
        private int Dia;
        private int Mes;
        private int Ano;
        public DataCronologica(int D, int M, int A)
    }
}
```

```
{Dia = D;
Mes = M;
Ano = A;}

public int Anos
{get
{return Ano;}
set
{Ano = value;}}
public int Meses
{get
{return Mes;}
set
{Mes = value;}}
public int Dias
{get
{return Dia;}
set
{Dia = value;}}
public int Idade(int C)
{return C - Ano;}
public int Idade(DataCronologica C)
{int TD = 365-DiaDoAno(this);
if (Bissexto(this.Ano) == true)
TD ++;
for (int I = this.Ano + 1; I <= C.Ano - 1; I++)
{TD += 365;
if (Bissexto(I)==true)
TD++;}
TD += DiaDoAno(C);
return TD;}
public static int DiaDoAno(DataCronologica X)
{int Td=0;
int[] DiaMeses = {31,28,31,30,31,30,31,31,30,31,30,31};
if (Bissexto(X.Ano)==true)
DiaMeses[1]=DiaMeses[1]+1;
for (int M=0; M<X.Mes-1; M++)
Td+= DiaMeses[M];
Td += X.Dia;
return Td;}
public static bool Bissexto(int A)
{if ((A % 4 == 0 && A % 100 != 0) || (A % 400 == 0))
return true;
else
return false;}}

public class IdadeDosAmigos
{static void Main(string[] args)
{String N = "Sara Tavares";
```

```
DataCronologica D = new DataCronologica(15, 5, 1996);
Amigos A = new Amigos(N, D);
Impressao(A);}

private static void Impressao(Amigos A)
{DataCronologica Ddata = A.Nascimentos;
DateTime Hoje = System.DateTime.Now;
DataCronologica C = new DataCronologica(Hoje.Day,
Hoje.Month, Hoje.Year);
Console.WriteLine("Este ano {0} faz {1} anos ", A.Nomes,
Ddata.Idade(Hoje.Year));
Console.WriteLine("Até hoje já viveu {0} dias",
Ddata.Idade(C));}}
```

? EXERCÍCIO 11: QUADRADOS E RETÂNGULOS

```
using System;
namespace HP11
{public class Quadrado
{protected int Comprimento;
public Quadrado(int L)
{Comprimento = L; }
public virtual int Area()
{return Comprimento * Comprimento; }
public string Lado()
{return Comprimento.ToString();}}

public class Rectangulo : Quadrado
{private int Largura;
public Rectangulo(int C, int L): base(C)
{Largura=L;}
public override int Area()
{return Comprimento*Largura;}
public string Dimensoes()
{return Comprimento+"x" +Largura;}}
```

```
public class Areas
{static void Main(string[] args)
{Quadrado Q=new Quadrado(4);
Rectangulo R= new Rectangulo(4,3);
Console.WriteLine("Área do quadrado com lado {0}={1} ",
Q.Lado(), Q.Area());
Console.WriteLine("Área do retângulo com dimensões {0}={1}",
R.Dimensoes(), R.Area());}}
```

? EXERCÍCIO 12: DIVERSOS REGIMES DE AVALIAÇÃO

```
using System;
namespace HP12
{public class Geral
    {private string Nome;
      private double T1;
      protected double CFinal;
      public Geral(string N, double T)
      {Nome = N;
       T1 = T;}
      public string PNome
      {get
       {return Nome;}}
      public double NotaT1
      {get
       {return T1;}}
      public double PCFinal
      {get
       {return CFinal;}}
      public virtual void NotaFinal()
      {Console.WriteLine("Regime Geral:");
       CFinal = Math.Round(T1, 0, MidpointRounding.AwayFromZero);}
      public override string ToString()
      {NotaFinal();
       return Nome + " obteve " + CFinal + " valores"; }}

public class Informatica : Geral
    {private double T2;
      public Informatica(string N, double T, double Z):base(N, T)
      {T2 = Z;}
      public override void NotaFinal()
      {Console.WriteLine("Regime específico para Informática:");
       base.CFinal = Math.Round((NotaT1+T2)/2, 0,
        MidpointRounding.AwayFromZero);}}
public class NotasAlunos
    {static void Main(string[] args)
      {Geral G = new Geral("Sara Tavares", 12.56);
       Console.WriteLine(G.ToString());
       Informatica I = new Informatica("Serafim Sousa", 12.56, 10);
       Console.WriteLine(I.ToString());}}
```

? EXERCÍCIO 13: TESTE DO REGIME DE AVALIAÇÃO

```
using System;
namespace HP13
{public class Geral
    {private string Nome;
      private double T1;
      protected double CFinal;
      public Geral(string N, double T)
      {Nome = N;
        T1 = T;}
      public string PNome
      {get
        {return Nome;}}

      public double NotaT1
      {get
        {return T1;}}
      public double PCFinal
      {get
        {return CFinal;}}
      public virtual void NotaFinal()
      {Console.WriteLine("Regime Geral:");
        CFinal = Math.Round(T1, 0, MidpointRounding.AwayFromZero);}

      public override string ToString()
      {NotaFinal();
        return Nome + " obteve " + CFinal + " valores"; }}
    public class Informatica : Geral
    {private double T2;
      public Informatica(string N, double T, double Z): base(N, T)
      {T2 = Z;}
      public override void NotaFinal()
      {Console.WriteLine("Regime Específico para Informática:");
        base.CFinal = Math.Round((NotaT1 + T2) / 2, 0,
          MidpointRounding.AwayFromZero);}}

    public class RegimeAvaliacaoII
    {static void Main(string[] args)
      {Object [] O=new Object[5];
        O[0]= new Geral("Sara Tavares", 9.56);
        O[1] = new Informatica("Serafim Costa", 10, 15.5);
        O[2] = new Geral("Sofia Carneiro", 12);
        O[3] = new Informatica("Saul Silva", 13,14);
        O[4] = new Geral("Saul Silva", 13);
        for (int I = 0; I <= O.Length - 1; I++)
          {Console.WriteLine(O[I].ToString());}}}
```

? EXERCÍCIO 14: MÉDIAS AMOSTRAIS

```
using System;
using System.Threading;
namespace HP14
{public class Populacao
    {protected int[] Pop;
      protected static int IndSup;
      public Populacao(int[] P)
      {Pop = P;
       IndSup = P.Length - 1;}
      public double Media(int[] X)
      {double M = 0;
       for (int i = 0; i <= X.Length-1; i++)
         M+=X[i];
       M=M/(X.Length);
       return Math.Round(M, 2);}}

public class Amostra : Populacao
    {private int N;
      private int[] Am ;
      public Amostra(int n, int[] P) : base(P)
      {N = n;
       Am = new int[n];
       ExtracaoAleat();}
      public void ExtracaoAleat()
      {Random R = new Random();
       int IndAleat, j, i = 0;
       while (i <= N - 1)
       {IndAleat = R.Next(0, IndSup);
        j = 0;
        Am[i] = Pop[IndAleat];
        while (Am[j] != Pop[IndAleat])
          j++;
        if (j == i)
          i++;}}
      public override string ToString()
      {String Valores = "";
       Valores = "Amostra: ";
       for (int i = 0; i <= N - 1; i++)
         Valores += Am[i] + " ";
       Valores += " Média da amostra=" + Media(Am);
       return Valores;}}

public class MediaAmostrai
    {static void Main(string[] args)
      {int[] P = { 1, 7, 8, 6, 5, 4, 10 };
```



```

Amostra[] A = new Amostra[3];
for (int i = 0; i <= 2; i++)
    {A[i] = new Amostra(P.Length + i-3, P);
    Console.WriteLine(A[i].ToString());
    Thread.Sleep(1);}
Console.WriteLine("Média da População=" + A[0].Media(P));}}}

```

? EXERCÍCIO 15: RESERVA DE QUARTOS

```

using System;
namespace HP15
{public class Basica
    {private string Nome;
    private string Tipo;
    public Basica(string N, string T)
    {Nome = N;
    Tipo = T;}

    public override string ToString()
    {return Nome + " reservou um quarto " + Tipo;}}

public class Cliente: Basica
    {private int Andar;
    public Cliente(string N, string T, int A): base (N, T)
    {Andar=A;}
    public override string ToString()
    {return base.ToString()+" no " +Andar + " andar";}}

public class ClienteFreq : Cliente
    {private bool SQuarto;
    public ClienteFreq(string N,string T,int A,bool S): base(N,T,A)
    {SQuarto=S;}
    public override string ToString()
    {string s=base.ToString();
    if (SQuarto==true)
        s+=" com ";
        else s+=" sem ";
    s+="serviço de quarto";
    return s;}}

public class Reservas
    {static void Main(string[] args)
    {object[] R = new object[4];
    R[0]=new Basica("J. Pinto", "duplo");
    R[1]=new Cliente("A. Paiva", "singular", 2);
    R[2] = new Basica("T. Caseiro", "com 4 camas");

```

```
R[3]=new ClienteFreq("C. Teixeira", "com 3 camas", 4, true);
for (int i=0; i<=R.GetLength(0)-1; i++)
Console.WriteLine( R[i].ToString());}}
```

? EXERCÍCIO 16: BIBLIOGRAFIA

```
using System;
namespace HP16
{public class Economia
{private string Titulo;
private string Autor ;
private int Ano;
public Economia(string N, string A, int C)
{Titulo = N;
Autor = A;
Ano = C;}
public int PAno
{get
{return Ano;}}
public string PAutor
{get
{return Autor;}}
public override string ToString()
{string S = "";
S += Autor+"(" + Ano +") "+ Titulo+".";
return S;}
public virtual string DepoisDe(int A)
{string S = "";
if (PAno > A)
S =ToString() ;
return S;}}

public class Livro : Economia
{private string Editor;
public Livro(string T,string A,int C,string E): base(T, A, C)
{Editor = E; }
public override string DepoisDe(int A)
{string S = "";
if (base.PAno > A)
S = base.ToString() + Editor + ".";
return S;}}

public class Revista : Economia
{private string NomeRev;
private int NumRev;
public Revista(string T, string A,int C,string R,int N)
```

```
        : base(T, A, C)
    {NomeRev=R;
      NumRev=N;}
    public override string DepoisDe(int A)
    {string S="";
      if (base.PAno > A)
        S= base.ToString() + NomeRev+"(" + NumRev+" ) " + ".";
      return S;}}

public class Biblioteca
{static void Main(string[] args)
{Economia[] L = new Economia[3];
  L[0]= new Livro("A Riqueza das Nações", "Smith, A.",
    1900,"ABC");
  L[1] = new Livro("O Vinho do Porto na Economia Portuguesa" ,
    "Chapman, C.", 1995, "FCA");
  L[2] =new Revista("Moeda e Preços em Adam Smith","Silva, A."
    , 1997, "Teoria em prática", 40);
  foreach (Economia E in L)
    if (E.DepoisDe(1990).Length>0)
      Console.WriteLine (E.DepoisDe(1990));}}}
```