

using teoria4;

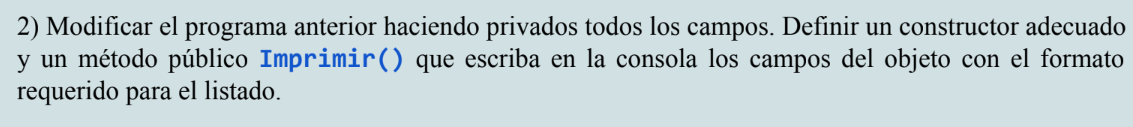
Persona p = new Persona ();

p.dni= 4;

p.Nombre = "a";

p.edad = 40;

Console.WriteLine(p.Nombre);



public class Persona

{

    private String? Nombre;

    private int edad,dni;

public Persona (String nombre, int edad,int dni){

    this.Nombre = nombre;

    this.edad = edad;

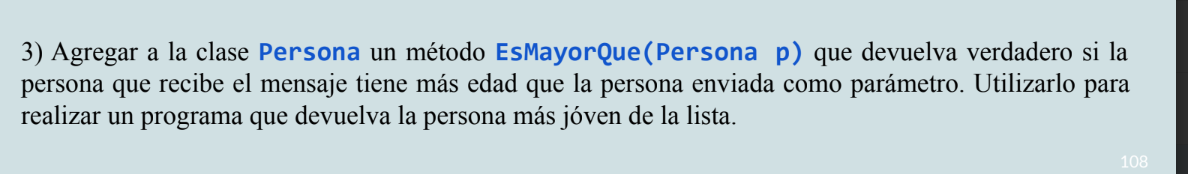
    this.dni = dni;

}

public void Imprimir (){

    Console.WriteLine("NOMBRE : " +Nombre + " EDAD : " + edad + "DNI" + dni);

}



using teoria4;

Persona p = new Persona ("PERRA DE FEDE",55,445);

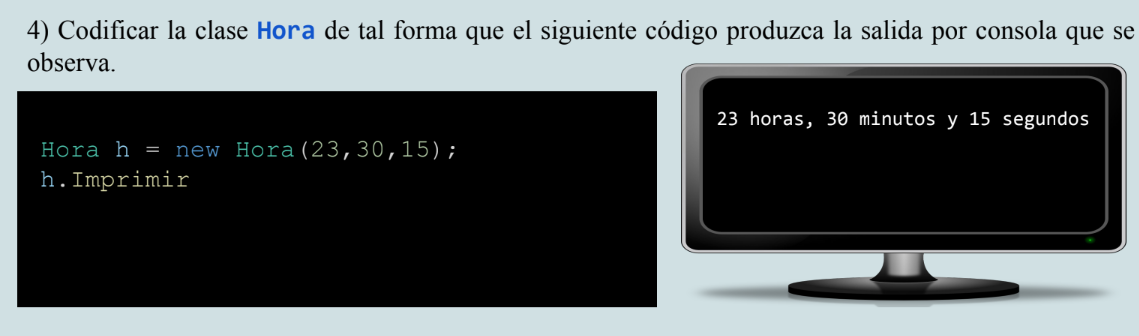
Persona p2=new Persona ("LEAN",18,34005);

if (p.esMayorQue(p2))

    Console.WriteLine("ES MAYOR");

else

    Console.WriteLine("NO ES MAYOR");



using System;

namespace teoria4;

public class Hora

{

    private int horas;

    private int minutos;

    private int segundos;

public Hora (int horas,int minutos,int segundos){

    this.horas = horas;

    this.minutos = minutos;

    this.segundos = segundos;

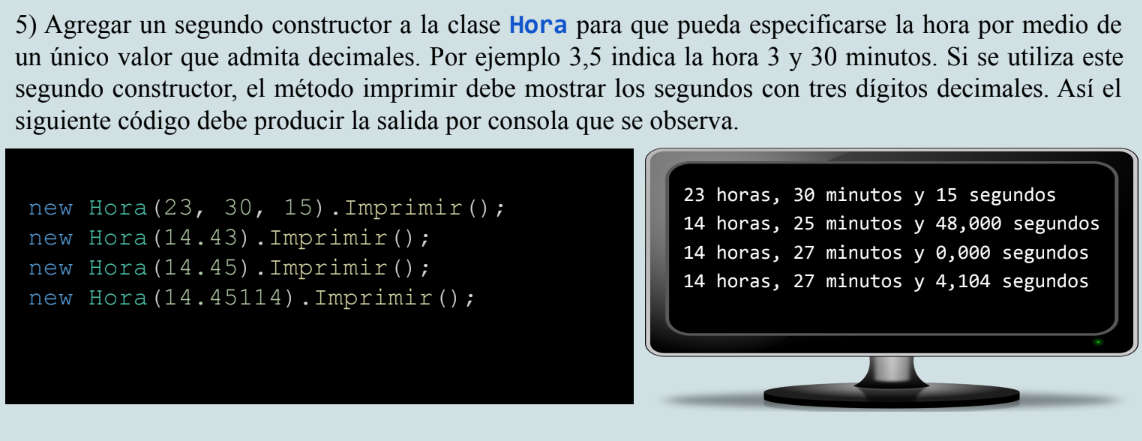
}

public void Imprimir (){

    Console.WriteLine(this.horas +" horas " + this.minutos + " minutos " + this.segundos+"segundos");

}

}



using System;

namespace teoria4;

public class Hora

{

    private int horas;

    private double minutos;

    private double segundos;

public Hora (int horas,int minutos,int segundos){

    this.horas = horas;

    this.minutos = minutos;

    this.segundos = segundos;

}

public Hora (double horas){   //14,45

    this.horas = (int)horas; //14

    double  parteDecimal = horas-this.horas; //14.45-14 = 0.45

    this.minutos=  Math.Round(parteDecimal\*60);  // 26

    this.segundos =(this.minutos-(parteDecimal\*60))\*60;

    this.segundos=  Math.Round((this.segundos\*100) /100);

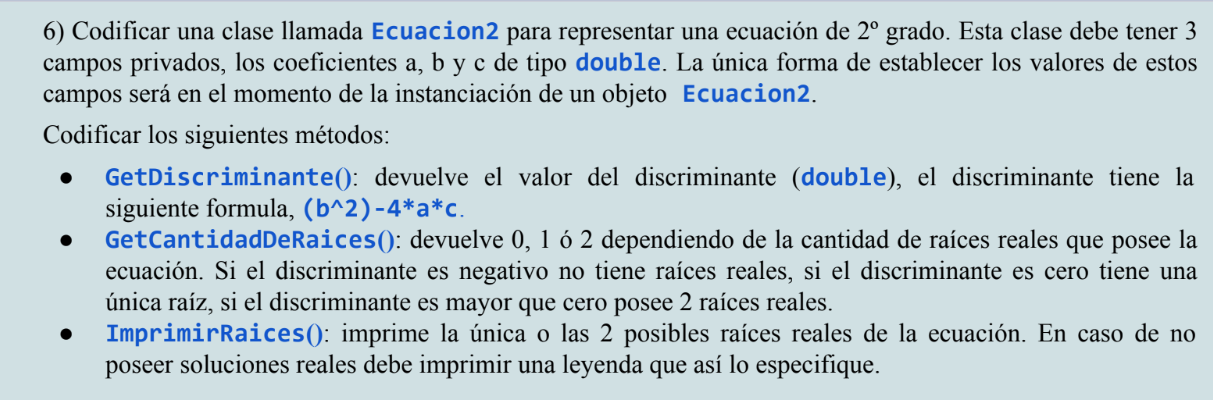
}

public void Imprimir (){

    Console.WriteLine(this.horas +" horas " + this.minutos + " minutos " + this.segundos+"segundos");

}

}



using System;

namespace teoria4;

public class Ecuacion2

{

    private Double \_a;

    private Double \_b;

    private Double \_c;

public Ecuacion2 (double a,double b,double c){

    this.\_a = a;

    this.\_b=b;

    this.\_c=c;

}

public double getDiscriminante(){

    return (Math.Pow(this.\_b,2))-(4\*this.\_a\*this.\_c);

}

public int getCantidadDeRaices(){

    return getDiscriminante() switch {>0 => 2, <0 => 0, \_ => 1};

    }

public void ImprimirRaices(){

    int aux = this.getCantidadDeRaices();

    if (aux == 2){

        double x1 = ((-this.\_b + Math.Sqrt(this.getDiscriminante()))/(2\*this.\_a));

        double x2 = ((-this.\_b - Math.Sqrt(this.getDiscriminante()))/(2\*this.\_a));

        Console.WriteLine(" X1: "+x1+"   X2: "+x2);

    }

    else if (aux == 1){

        double x = (-this.\_b)/(2\*this.\_a);

        Console.WriteLine("La raiz de la ecuacion es: " + x);

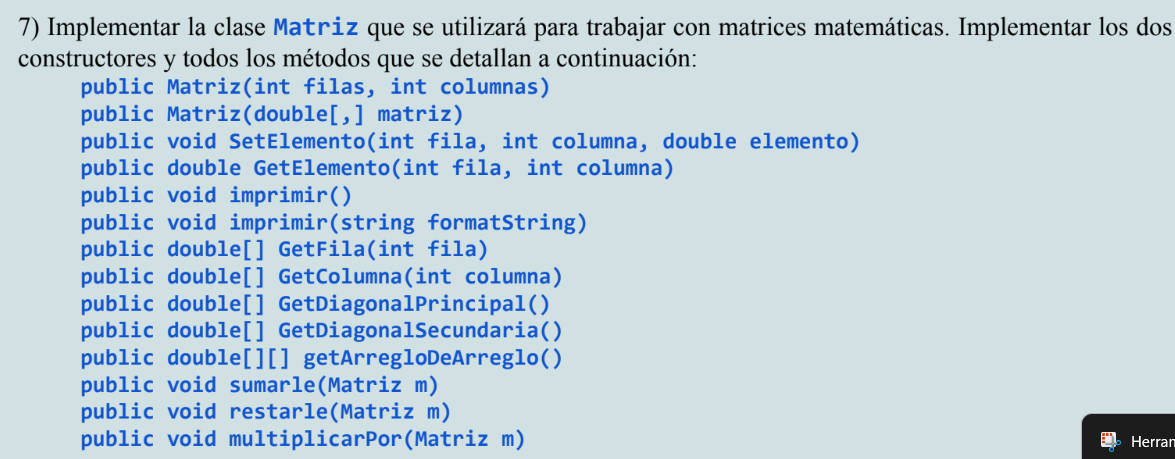
    }

    else

        Console.WriteLine("No posee soluciones reales");

}

}



using System;

using System.Dynamic;

using System.Runtime.CompilerServices;

namespace teoria4;

public class Matriz

{

    private double [,] matriz;

    public Matriz (int fila,int columnas){

        this.matriz = new double  [fila,columnas];

    }

    public Matriz (double [,] matriz){

        this.matriz = matriz;

    }

    public void SetElemento(int fila,int columna,double elemento){

        this.matriz[fila,columna] = elemento;

    }

    public double getElemento (int fila,int columna){

        return this.matriz [fila,columna];

    }

    public void imprimir(){

        for (int i=0;i<this.matriz.GetLength(0)\*this.matriz.GetLength(1);i++){

            Console.Write(this.matriz[i/this.matriz.GetLength(1),i%this.matriz.GetLength(1)]);

            Console.Write(i%this.matriz.GetLength(1)==this.matriz.GetLength(1)-1 ? "\n" : " ");

        }

    }

       public void imprimir(String stringFormat){

        for (int i=0;i<this.matriz.GetLength(0)\*this.matriz.GetLength(1);i++){

            Console.Write(this.matriz[i/this.matriz.GetLength(1),i%this.matriz.GetLength(1)].ToString(stringFormat));

            Console.Write(i%this.matriz.GetLength(1)==this.matriz.GetLength(1)-1 ? "\n" : " ");

            }

       }

        public double [] GetFila (int fila){

            double [] f = new double [this.matriz.GetLength(1)];

            for (int i=0;i<this.matriz.GetLength(1);i++){

                f[i] = this.matriz[fila,i];

            }

            return f;

        }

        public double [] GetColumnas (int columna){

            double [] c = new double [this.matriz.GetLength(0)];

            for (int i =0;i<this.matriz.GetLength(0);i++){

                c[i]=this.matriz[i,columna];

            }

            return c;

        }

        public double [] getDiagonalPrincipal(){

            int f = this.matriz.GetLength(1);

             double [] v = new double [f];

              for (int i=0;i<this.matriz.GetLength(1);i++){

                v[i] = matriz[i,i];

              }

              return v;

        }

          public double [] getDiagonalSecundaria(){

            int f = this.matriz.GetLength(1);

             double [] v = new double [f];

             int j=this.matriz.GetLength(1)-1;

              for (int i=0;i<this.matriz.GetLength(1);i++){

                v[i] = matriz[j--,i];

              }

              return v;

        }

            public void sumarle (Matriz m){

                for (int i=0;i<this.matriz.GetLength(0);i++){

                    for (int j=0;j<this.matriz.GetLength(1);j++){

                        this.matriz[i,j] += m.getElemento(i,j);

                    }

                }

            }

            public void restarle (Matriz m){

                 for (int i=0;i<this.matriz.GetLength(0);i++){

                    for (int j=0;j<this.matriz.GetLength(1);j++){

                        this.matriz[i,j] -= m.getElemento(i,j);

                    }

                }

            }

            public void multiplicar (Matriz m){

                if (this.matriz.GetLength(1) != this.matriz.GetLength(0))

                    Console.WriteLine("LA MATRIZ NO SE PUEDE MULTIPLICAR");

                else

                    {

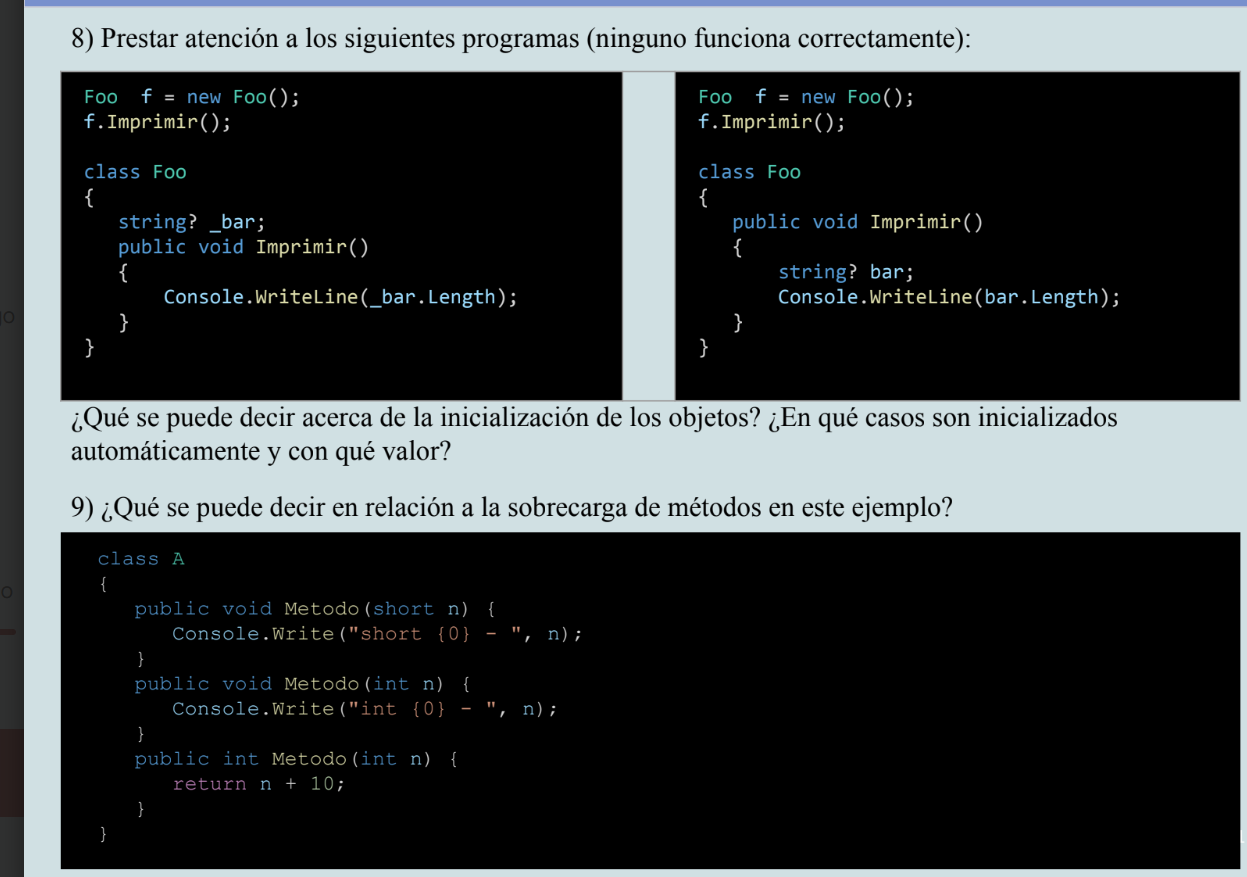
                        Matriz mat = new Matriz (this.matriz.GetLength(0),this.matriz.GetLength(0));

                        //F NO TENGO GANAS

                    }

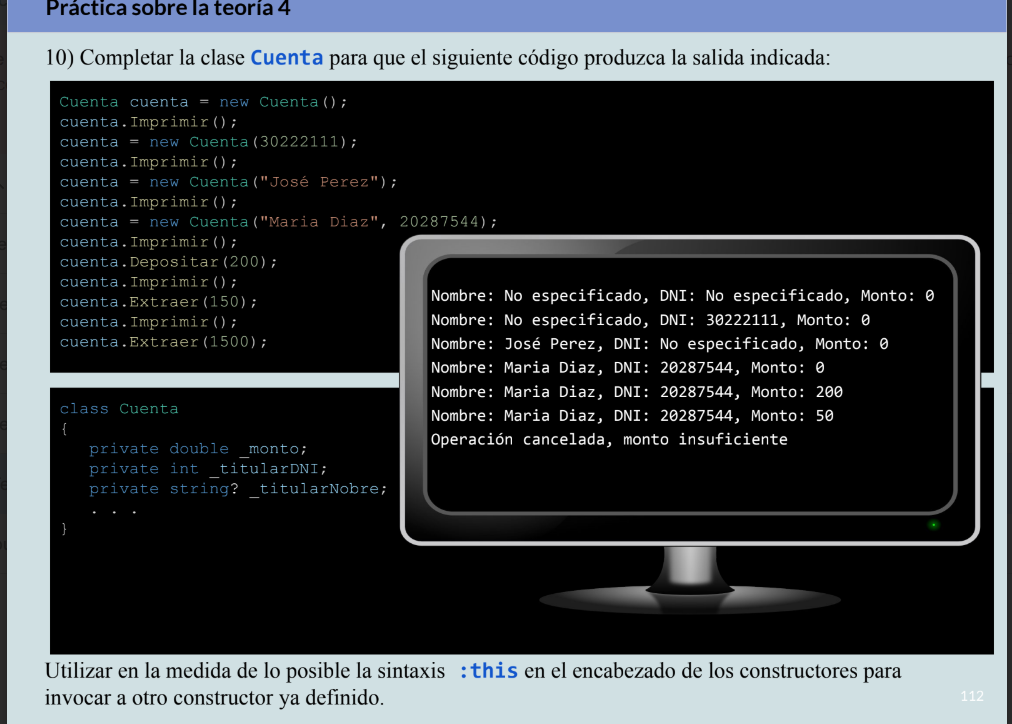
    }

}



8 ) se debería cargar bar con null y preguntar si el valor es null y devolver una exeption

9) El método esta clonado con mismo nombre y mismos parámetros por ende no funcionara.



using System;

using System.Runtime.InteropServices;

namespace teoria4;

public class Cuenta

{

    private double \_monto;

    private int \_titularDNI;

    private string? \_titularNobre;

    public Cuenta (){

        this.\_monto = 0;

        this.\_titularDNI=0;

        this.\_titularNobre="Sin Especificar";

    }

    public Cuenta (int dni):this(){

        \_titularDNI=dni;

    }

    public Cuenta(String nombre):this(){

        \_titularNobre = nombre;

    }

    public Cuenta (String nom , int dni):this(){

        this.\_titularDNI=dni;

        this.\_titularNobre=nom;

    }

    public void Imprimir (){

        Console.WriteLine ($"NOMBRE : {this.\_titularNobre}DNI : {this.\_titularDNI}MONTO : {this.\_monto}");

    }

    public void Depositar(double p){

        this.\_monto += p;

        Console.WriteLine("DEPOSITO REALIZADO...");

    }

    public void Extraer (double p){

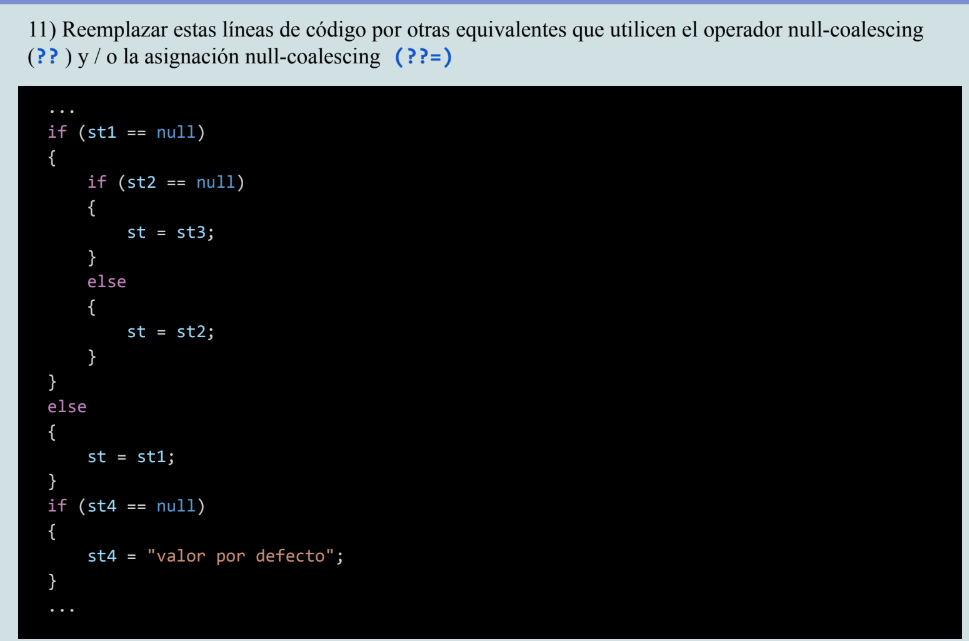
        if (this.\_monto >p) this.\_monto-=p;

        else

            Console.WriteLine("Fondos insuficientes...");

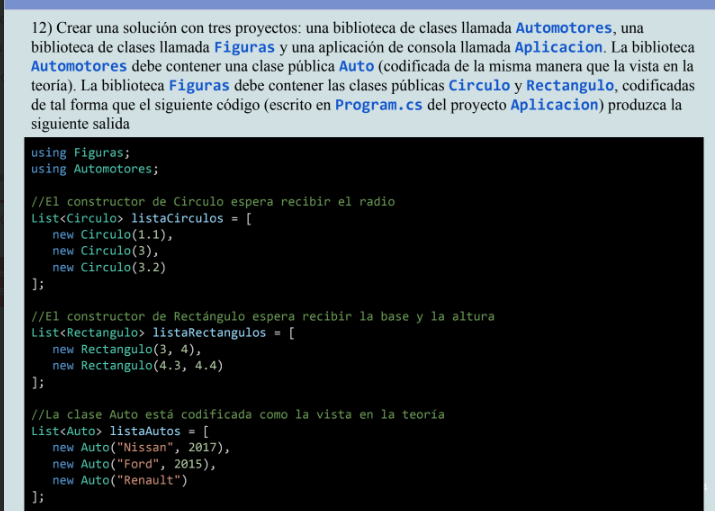
    }

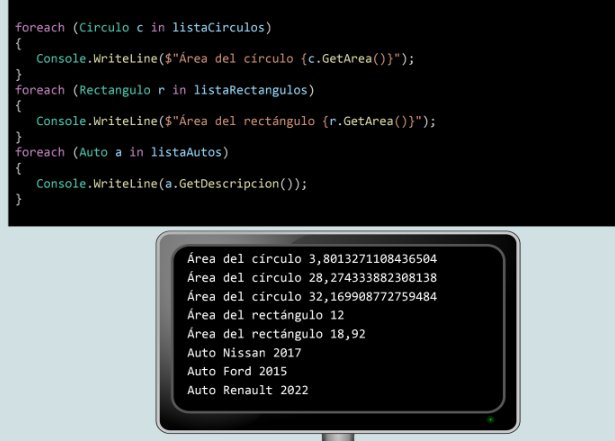
}



st = st1 ?? st2 ?? st3 ;   //si st1 es null va a st2 y pregunta si no es null y si no lo es va st3 y lo define

st4 ??= "ValorPorDefecto";      //pregunta si st4 no es null si lo es pone valorPorDefecto.





using System;

namespace teoria4;

public class Circulo

{

    private double \_radio;

    public Circulo (double r){

        this.\_radio = r;

    }

    public double GetArea(){

        return Math.PI\*(this.\_radio\*this.\_radio);

    }

}

using System;

namespace teoria4;

public class Rectangulo

{

    private double \_base;

    private double \_altura;

    public Rectangulo (double \_base, double \_altura) {

         this.\_base = \_base;

         this.\_altura = \_altura;

    }

    public double GetArea (){

        return this.\_base \* this.\_altura;

    }

}

using System;

namespace teoria4;

public class Auto

{

private string? \_marca;

private int \_modelo;

public Auto (String Marca){

    this.\_marca = Marca;

}

public Auto(string marca, int modelo) : this (marca)

{

\_modelo = modelo;

}

public string GetDescripcion() =>$"Auto {\_marca} {\_modelo}";

}