Aprende programación:
15 Ejercicios resueltos en C#.

Matías Salom Avellà

Presentación:

Este libro de ejercicios pretende ser un complemento de otros libros más completos y con más fundamentos teóricos, pero que pueden necesitar ejercicios actualizados para asimilar mejor la teoría.

Es un libro de ejercicios indicado para una primera aproximación a la programación, pudiendo ser parte de la asignatura de "Fundamentos de la programación" o para una formación autodidacta.

Está en preparación otro libro para completar los temas referentes a la programación visual y el acceso a dates desde lenguajes de programación .NET.

Para cualquier consulta al autor, se puede acceder a la web www.ibserveis.com, y desde allí al apartado de Consulta.

Capítulo 0

Iniciación a la programación

Capítulo 0

Iniciación a la resolución de problemas estructuradamente con PSEUDOCODIGO

Es de gran importancia, antes de empezar a teclear con el ordenador, plantear correctamente la solución de nuestro proyecto.
Existen infinitas soluciones para cada uno del problemas que podremos programar. Siempre podemos encontrar la solución más adecuada, por ser la más práctica y fácil de implementar.

La filosofía de ***"la solución más simple, es probablemente la más acertada"*** (Navaja de Occam, wiki: Occam) y su actualización para proyectos informáticos: la filosofía KISS (Keep it Simple) , serán las filosofías que se considerarán prioritarias para resolver estos ejercicios.

Como ejemplo anecdótico (totalmente falso , pero ilustrativo):
Se comenta que la NASA invirtió muchos millones en un bolígrafo capaz de escribir boca arriba y en condiciones extremas de temperaturas, la URSS hizo servir lápices . (la historia real en google: sondas espaciales boli)

Un buen artículo respecto a mantener la simplicidad en el software (google: soitu keep simple)
http://www.soitu.es/soitu/2008/03/14/pieldigital/1205521516_335354.html

Y si alguien aún duda que la simplicidad es bella ,eficiente e indicada para la realización de proyectos, puede aplicar que el "tiempo es oro" si se tiene que desarrollar un proyecto para una empresa, con un presupuesto y tiempo limitado.

Estos ejercicios pueden ser ejecutados en un entorno de programación actual (2008), i gratuito: Microsoft Visual C# Express edition.

Por otra parte, para todos aquellos que no tienen intención de seguir el ritmo de las novedades informáticas, puede visitar esta web:
http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=devesb

http://www.adictosaltrabajo.com/detalle-noticia.php?noticia=49

google: adictos al trabajo tutoriales C#

INTENTAR SOLUCIONAR LOS EJERCICIOS ANTES DE VER LA SOLUCIÓ PROPUESTA.

Breve explicación de los enunciados del Capítulo 0:

Lo más importante de estos primeros ejercicios es la "ordenación de instrucciones".
Existen muchas soluciones para cada enunciado. Se propone una, lo más ordenada y estructurada posible para que los programas realizados más adelante también sean estructurados.

0.1) Pasos a seguir para conseguir cambiar una cuerda de guitarra

Para cambiar una cuerda de guitarra se hacen una serie de movimientos en el clavijero para conseguir la tensión necesaria en cada cuerda, estos pasos son los que describe la solución.

0.2) Resolución del algoritmo de compra de una camisa

En el día a día realizamos muchas acciones que podrían formar parte de un programa, es como si tuviéramos que dar las instrucciones , justas y necesarias, a un robot para escoger ropa.

Pensar en que instrucciones, paso a paso, seguimos para escoger ropa en una tienda.

0.3) Reparación de una fuente de agua que no funciona.

En todas las reparaciones, sean de ordenadores o de otras máquinas se sigue un algoritmo para conseguir que la reparación sea lo más rápida y eficiente posible. En el ejercicio es una fuente, pero podría ser perfectamente cualquier máquina, motor, o aparato electrónico.

0.4) PSEUDOCODIGO que calcula la suma de los números enteros entre 1 y 10 (1+2+3+4....)

Este es el primer ejercicio "matemático", las instrucciones para resolver el problema son más abstractas que los ejercicios anteriores, pero siguen teniendo una dificultad inicial baja.

¿Qué instrucciones daríamos a un niño de primaria para que sume los 10 primeros números?

0.5) Pseudocódigo para calcular el producto de números enteros del 10 al 20 10*12*14...

El mismo tipo de ejercicio anterior, pero con otra operación matemática.

0.1) Pasos a seguir para conseguir cambiar una cuerda de guitarra

```
Inicio
    Preparar material necesario
    Extraer cuerda rota
    Introducir cuerda nueva
    Mientras no tenga  tensión
        Girar clavijero
    volver

    ¿Esta afinada?
      NO : Mientras no esté  afinada
                ¿ Esta en un tono más alto?
                    Sí : Girar clavijero a la izquierda
                    No: Girar clavijero a la derecha
           volver
      Sí : esta afinada
Fi
```

0.2) Resolución del algoritmo de compra de una camisa

```
Inicio
    Entrar a la Tienda
    Mientras no vea nada que me gusta
        Buscar
    volver

    Mirar precio
    ¿Es un precio interesante?
      SÍ:  ¿Es de mi  talla?
            SÍ:  Voy al probador
                 ¿Me gusta como me queda?
                     SÍ: Compro la ropa
                     NO: volver a buscar

            NO: volver a buscar

      NO: volver a buscar

Fi
```

0.3) Reparación de una fuente de agua que no funciona.

El sistema a reparar consta de un motor dentro de una fuente que hace salir agua. En el exterior hay una caja de conexión eléctrica a la cual llega la corriente. Sale de esta caja, un cable de corriente que se conecta a un enchufe de 220V.

Inicio
Mientras Fuente NO Marcha **hacer**

¿Llega corriente al motor?
NO llega corriente al motor

¿Llega corriente a la caja?
NO llega corriente a la caja

¿Hay corriente en el enchufe?
NO hay corriente en el enchufe
Reparar enchufe
Fuente Sí marcha

SÍ hay corriente
Reparar cable enchufe-caja
Fuente Sí marcha

SÍ llega corriente a la caja
Reparar cable entre caja y motor de la fuente
Fuente Sí marcha

SÍ llega corriente al motor
Cambiar motor
Fuente Sí marcha

Fin Mientras

Avisar que ya funciona
Cobrar factura

Fin

0.4) PSEUDOCODIGO que calcula la suma de los números enteros entre 1 y 10 (1+2+3+4....)

```
inicio
        suma ← 0: conta ← 0

        Mientras conta menor que 10
                conta ← conta + 1
                suma ← suma + conta
        Fin Mientras

        Escribir "La suma es"; suma
fin
```

0.5) Pseudocódigo para calcular el producto de números enteros del 10 al 20 10*12*14...

```
inicio
        suma ← 0: conta ← 10: producto ← 1

        Mientras conta menor o igual que 20
                producto ← producto * conta
                conta ← conta + 2
        Fin Mientras

        Escribir "El producto es"; producto
fin
```

Capítulo 1

Programas ejecutables en Basic y C#

Capítulo 1: Programas ejecutables en Basic y C#

Breve explicación de los enunciados del Capítulo 1:

Hemos aprendido como plantear un algoritmo de resolución un problema, ahora necesitamos aprender el "idioma" del ordenador para que ejecute la solución diseñada.

Para que el ordenador entienda nuestras instrucciones, necesitaremos un entorno de trabajo (IDE) que nos permita escribir y corregir código, así como también ejecutarlo.

Por su proximidad al pseudocódigo y su facilidad de aprendizaje, se han resuelto unos cuantos ejercicios en lenguaje Basic . El entorno de trabajo escogido es el del **SmallBasic**, inicialmente destinado a máquinas portátiles y móviles, tiene versiones para muchos sistemas operativos:

- BASIC: http://smallbasic.sourceforge.net/ , version: smallbasic 0_9_7_ftlk

No es el objetivo de este libro enseñar a manejar entornos de programación. Existen tutoriales muy buenos en internet para empezar a programar en Microsoft Visual C#:

Google: Tutorial c#

www.devjoker.com : Tutorial C#

http://functionx.com/csharp/Lesson01.htm : FunctionX

Libro gratuito recomendado: www.josanguapo.com

1.1) 1.2) Comprobar con la calculadora del S.O. 4 digitos binarios son 1 digito hexadecimal.
La relación decimal-hexadecimal-binario puede haber perdido importancia en lenguajes de alto nivel, pero ha sido la base matemática de muchas operaciones en lenguaje C.

1.3) Programa que calcula la suma de los números enteros entre 1 y 10 (1+2+3+4....)
1.4) Programa producto números enteros del 10 al 20 10*12*14...
Podemos ver que los ejercicios 1.4 - 1.5 tienen fácil traslación al lenguaje de programación BASIC.

1.5 - 1.6) Programa: usuario introduce números y el ordenador cuenta cuantos son positivos.
Ya podemos comparar ambos lenguajes BASIC- C# , el mismo pseudocódigo escrito en cada uno de estos lenguajes**.** Nuestra "Piedra de Rosetta" particular (google: piedra rosetta)

1.7) Contabiliza personas de más de 180, entre 180 y 170, entre 170 y 160, y más bajas que 160cm.
El mismo pseudocódigo que el ejercicio 1.7 para comprobar que si hemos pensado correctamente la solución, es fácil implementarla con un lenguaje de programación. Solución con C# en el 2.14

1.8) Programa para poner notas: suspendido, aprobado, bien... con la nota numérica.
Solución del ejercicio 1.9 con lenguaje C#

1.13) Realizar programa en el que el ordenador "piensa" en un número al azar entre 1 y 50.
El usuario ha de adivinarlo en 5 oportunidades. El ordenador señalará si es mayor o menor.

1.1) Ejecutar el siguiente programa en Basic

```
REM Programa números decimal, binario , hexadecimal
10 x1=&h0001: x2=1 : x3=&b0001
20 print x1,x2,x3
30 y1=&h000A: y2=10: y3=&b00001010
40 print y1,y2,y3
50 z1=&h001B: z2=27: z3=&b00011011
60 print z1,z2,z3
70 end
```

1.2) Comprobar con la calculadora del S.O. (ver científica) que a cada 4 digitos binarios corresponden a 1 digito hexadecimal. Ejemplos: 1001_2 -> 9_{16} , 1100_2 ->C_{16} , $1000\ 1001_2$ ->89_{16}

1.3) Programa que calcula la suma de los números enteross entre 1 y 10 (1+2+3+4....)

```
REM Programa suma números enteros

suma =0: conta =0
WHILE conta <10
  conta = conta + 1
  suma = suma + conta
WEND

PRINT "La suma es"; suma
END
```

1.4) Programa producto números enteros del 10 al 20 10*12*14...

```
REM Programa producto números enteros

suma =0: conta = 10: producto =1
 WHILE conta <= 20
  producto = producto * conta
  conta = conta + 2
WEND

PRINT "El producto es"; producto
END
```

1.5) Programa : El usuario introduce números y el ordenador cuenta cuantos son positivos.

```
REM Programa conta número positivos

positius =0: conta =0
INPUT "Introduce número";numero

WHILE numero <>999
 conta = conta + 1
 if numero >0 then positivos = positivos +1

INPUT "Introduce número";numero
WEND

PRINT "Has introducido un total de "; conta
PRINT "y son positivos "; positivos
END
```

TUTORIAL INICIO C# : http://functionx.com/csharp/Lesson01.htm

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Proyecto1
{
    class Holamundo
    {
        public static void Main(String[] args)
        {
            Console.WriteLine("¡Hola {0}!", args[0]);
        }
    }
}
```

Compilar: csc Holamundo.cs
Ejecutar: Holamundo Maria

1.6) Programa : El usuario introduce números y el ordenador cuenta cuantos son positivos.

```
class Program
 {
      static void Main(string[] args)
      {
          int positivos = 0; int conta = 0; int numero;

          Console.WriteLine("Introduce número ");
          numero = Int32.Parse(Console.ReadLine());

          while (numero != 999)
          {
              conta = conta + 1;
              if (numero > 0) positivos = positivos + 1;

              Console.WriteLine("Introduce número ");
              numero = Int32.Parse(Console.ReadLine());
          }

          Console.WriteLine("Has introducido un total de {0}", conta);
          Console.WriteLine("y son positivos {0}", positivos);

      } //fin Main
 }
```

1.7) Contabiliza personas de más de 180, entre 180 y 170, entre 170 y 160, y más bajas que 160cm.

```
REM Programa estadísticas altura

 conta1 = 0:conta2=0:conta3 =0:conta4 =0
 INPUT "Introduce altura en cm"; altura
 WHILE altura <> 999
   if altura>=160 then
     if altura>=170 then
       if altura >=180 then
         conta1 = conta1 + 1
       else
         conta2 = conta2 + 1
       fi
     else
       conta3= conta3 + 1
     fi
   else
     conta4=conta4 + 1
   fi
INPUT "Introduce altura en cm"; altura
WEND

print "Más grandes de 180 " ;conta1
print "Entre 170 y 180 ";conta2
print "Entre 160 y 170 ";conta3
print "Más bajas que 160 ";conta4
END
```

1.8) Programa para poner notas: suspendido, aprobado, bien .. con la nota numérica.

```
class Program
{
   static void Main(string[] args)
   {
     double nota=0;
     string c_nota = "no asignada";

     bool repetir = true;
     while (repetir == true)
      {
        try
        {
            Console.WriteLine("Introduce nota");
            nota = Double.Parse(Console.ReadLine());
            repetir = false;
        }

        catch (FormatException)
        {
            Console.Clear();
            Console.WriteLine("Introducir sólo valores numéricos");
            repetir = true;
        }
      }

     if (nota >0 && nota<10)
     {
       if (nota>=5)
         {
          if (nota>=6)
            {
              if (nota >= 7)
               {
                   if (nota >= 9) c_nota = "sobresaliente";
                   else c_nota = "notable";
               }
              else c_nota = "bien";
            }
          else c_nota="aprobado";
          }
       else c_nota ="suspendido";
      }


     if (c_nota != "no asignada")
     {
        Console.WriteLine("La nota es de {0}", c_nota);
     }
     else Console.WriteLine("Nota no valida");

     Console.WriteLine();
     Console.WriteLine("Final del programa");

   } // fi Main
}   // fin Class
```

1.13) Realizar programa en el que el ordenador "piensa" en un número al azar entre 1 y 50. El usuario ha de adivinarlo en 5 oportunidades. El ordenador señalará si es mayor o menor.

```
rem randomize(timer)
label inicio
cls
conta =0: numero =0: azar =0:encertat =0
azar = int(rnd*50)+1

 WHILE conta < 5 AND acertado = 0
    PRINT "Oportunidades que quedan:  "; (5-conta);
    ?:?:?
    PRINT "Introduce numero "

    INPUT  numero

     if numero > azar then
        PRINT "El numero es menor"
     else
          if numero < azar then
             PRINT "El numero es mayor"
          else
             PRINT "Enhorabona, has acertado"
             acertado = 1
          endif
     endif

     conta = conta +1
 WEND

PRINT:PRINT "Fin del programa"
```

Capítulo 2

Bucles y Funciones

2.1) Programa en el que el ordenador lanza 50 veces un dado y cuenta cuantas veces sale el nº 1.
Ejemplo para el uso de 'bucles' *for* para repetir un número determinado de veces unas instrucciones. También muestra cómo conseguir números al azar.

2.2) Programa que muestra 15 líneas como estas: 1 12 123 1234
Ejemplo del uso de dos bucles anidados de tipo *for*

2.3) Programa que dibuja un Triangulo isósceles
Una aplicación más compleja i completa del uso de bucles anidados.

2.4) Programa que señala si es múltiplo del número 5
La primera función que aplicamos retorna un valor boolean (verdadero/falso) si el número que enviamos per analizar es o no múltiplo de 5.

2.5) Programa que muestra el día que será mañana. Ex: 31/12/08 -> 01/01/09
Clásico ejercicio donde se ponen en práctica los conocimientos de programación estructurada. Se deja como ejercicio pendiente la versión en C# (por otro lado, nada complicada de realizar)

2.6) Programa para calcular Potencias, Tensiones e Intensidades. P = V * I
Más prácticas con funciones simples, comparando diferentes lenguajes de programación.

2.1) Programa en el que el ordenador lanza 50 veces un dado y cuenta cuantas veces sale el nº 1.

```
contador =0
randomize timer
cls

for t = 1 to 50
  dado = int(rnd*9)+1
  print dado;
  if dado = 1 then contador = contador + 1
next t

print "Han salido ";contador ;" números  1 a la lista"
end
```

VERSIÓ C#

```
static void Main(string[] args)
    {
          int contador =0; int dado=0;
          Random numero = new Random();
          Console.Clear();

          for (int t = 0; t<=50; t=t + 1)
          {
              dado = numero.Next(1, 7);
              if (dado == 1)
              {
                  contador = contador + 1;
                  Console.ForegroundColor = ConsoleColor.Red;
              }
              else Console.ForegroundColor = ConsoleColor.Gray;

              Console.Write(" {0}", dado);
          }

          Console.WriteLine("-");
          Console.WriteLine("Ha salido el número1 {0} veces",contador);
    }
```

google: Consola C# , console C#
http://www.c-sharpcorner.com/UploadFile/scottlysle/ColorConsoleCS060820080557 47AM/ColorConsoleCS.aspx

2.2) Programa que muestra 15 líneas como estas: 1 12 123 1234

1
12
123 ....

```
static void Main(string[] args)
 {
            int i, j;
            for (i = 1; i <= 15; i++) // 15 lineas
            {
                for (j = 1; j <= i; j++)  // números a cada línea
                Console.Write(" - {0}", j);
                Console.WriteLine("  ") ;
            }
  }
```

2.3) Programa que dibuja un Triangulo isósceles

```
static void Main(string[] args)
 {
        Console.Clear();

    // dibujo de cada linea (bucle externo)
    for (int fila=1; fila <= 7; fila ++)
    {
        //dibuja espacios en blanco (1er bucle interno)
        for (int espacios = 7 - fila; espacios > 0; espacios--)
            Console.Write(" ");// espai en blanc

        // dibuja estrellas (2º bucle interno)
        for (int conta = 1; conta < (2 * fila); conta++)
            Console.Write("*");

        Console.WriteLine(" ");
    }
  }
```

2.4) Programa que señala si es múltiplo del número 5

```
class Exercici4
 {
       static void Main(string[] args)
       {
           int num = 1;
           bool respuesta;

           while (num <= 50)
           {
               Console.Write(" - {0}", num);
               respuesta = multiplo5(num);

               if (respuesta) Console.WriteLine(" Es múltiplo de 5");
               else Console.WriteLine(" No es múltiplo de 5");

               num++;
           }
       }


       public static bool multiplo5 (int n)
       {
           if((n % 5) !=0) return false;
           else return true;
       }

  }
```

2.5) Programa que muestra el día que será mañana. Ex: 31/12/08 -> 01/01/09

```
#include <iostream>
using namespace std;
int funcion_divisor(int numero, int divisor);

void main()
{
	int d,max,m,a,resto;

	printf("Introduce el dia: ");
	scanf("%d",&d);
	printf("\nIntroduce el mes: ");
	scanf("%d",&m);
	printf("\Introduce el anyo: ");
	scanf("%d",&a);
	printf("\nHoy es dia %d de %d del %d",d,m,a);

	if (m==4 || m==6 || m==9 || m==11) max=30;
	if (m==1 || m==3 || m==5 || m==7 || m==8 || m==10 || m==12) max=31;

	if (m==2)
	{
		resto = funcion_divisor(a,4);
		if (resto==0) max=29;
		else max=28;
	}

	d++;

	if (d>max)
	{
		d=1;
		m++;
		if (m>12) { m=1; a++; }
	}

	printf("\n\ny mañana será %d de %d del %d",d,m,a);
	printf("\n\n");
}


int funcion_divisor(int numero, int divisor)
{
	int resto = numero % divisor;
	return (resto);
}
```

2.6) Programa para calcular Potencias, Tensiones e Intensidades. P = V * I

```
label inicio
  cls
  Print "Bienvenid@ al programa" : ?:?
  Print "¿Qué quieres calcular?"
  input "1- Poténcia , 2-Intensidad 3-Tensión "; opcion
  if opcion ="1" then goto Calculo_Potencia
  if opcion ="2" then goto Calculo_Intensidad
  end

 label Calculo_Potencia
    print : print "Cálculo de poténcia "
    input "Introduce Tensión: ";tension
    input "Introduce Intensidad¨: ";intensidad
    print " - "
    potencia = tension * intensidad
    print "La poténcia es de ";potencia

    input "Pulsa (s) para volver a calcular"; otra
    if otra ="s" then goto inicio
 end

  label Calculo_Intensidad
    print : print "Cálculo de Intensidad "
    input "Introduce Potencia: ";potencia
    input "Introduce Tensión: ";tension
    print " - "
    Intensidad = potencia / tension
    print "La Intensidad es de ";intensidad ;: ? " ampers"

    input "Pulsa (s) para volver a calcular"; otra
    if otra ="s" then goto inicio
 end
```

VERSIO C#

```
using System;
using System.Collections.Generic;
using System.Text;

class Ejercicio_2_6
{
  static void Main(string[] args)
  {
   string n_resultado="";
   double resultado=0;
   string teclado ="null";
   while (teclado != "4")
   {
     Console.WriteLine("-");
     Console.WriteLine("1-Poténcia  2-Tensión  3-Intensidad  4-Sortir");
     teclado = Console.ReadLine();

     switch (teclado)
      {
        case "1":
            n_resultado = "Potencia";
            resultado = fcalcul(n_resultado,"Tension","Intensitat"); break;
        case "2":
            n_resultado = "Tension";
            resultado = fcalcul(n_resultado, "Potencia", "Intens."); break;
        case "3":
            n_resultado = "Intensidad";
            resultado = fcalcul(n_resultado, "Potencia", "Tension"); break;
        case "4":
             break;
        default: Console.WriteLine("Tecla equivocada") ; break;
      }

      Console.WriteLine("El resultado {0} es {1}", n_resultado, resultado);
   }
  } // fi Main


  public static double fcalcul(string n_calculo, string nom1, string nom2)
     {
            Console.WriteLine("Introduce {0}",nom1);
            string temp1 = Console.ReadLine();
            int var1 = Int32.Parse(temp1);

            Console.WriteLine("Introduce {0}", nom2);
            string temp2 = Console.ReadLine();
            int var2 = Int32.Parse(temp2);

            if (n_calculo=="Potencia") return (var1 * var2);
            else return (var1 / var2);
      }

   }   //fi class Exercici6
```

Capítulo 3

Clases

3.1) Creación objeto y asignación de variables propias del objeto.

Podemos ver como se crea un objeto en base a una clase, delgadoero en lenguaje C++, después en C# 2.0 (Visual Studio 2005) y finalmente con C# 3.0 propio del Visual Studio 2008.

3.2) Creación objetos y asignación de variables propias de los objetos.

Asignación de valores a variables privadas de la clase a través de funciones públicas (accesibles desde cualquier parte del código)

3.3) Asignación propiedades del objeto.

Variables privadas de la clase utilizadas a través de funciones get/set. **OJO!** con las mayúsculas

3.4) Variables STATIC

Variables que podemos emplear en todo el proyecto (si son public) sin necesidad de objetos.

3.5) Trabajando con namespaces. Organización de clases

Podemos agrupar diferentes clases en un solo namespace. Facilita la organización de proyectos.

3.6) A partir del namespace: Hospital , guardado en al archivo Clase_Hospital .cs , se crea la clase "Paciente" y se utiliza en otro archivo: Ejercicio.cs

Ejemplo de cómo utilizar diferentes archivos en el mismo proyecto.

3.7) Ejemplo de envío de un objeto como parámetro de una función.

Los objetos como unidad de información, una variable personalizada.

3.1) Creación objeto y asignación de variables propias del objeto.

Versión C++ , en dos archivos:

cabecera.h

```
#include <iostream>
#include <string>
using namespace std;

class FichaDatos

{
       string nombre;
       int edad;

 public:
 FichaDatos(string dato1, int dato2)
 {
       nombre=dato1;
       edad=dato2;
 }

 void Cargar()
 {
       nombre = "Pedro";
       edad = 29;
 }
 void Mostrar();

};
```

principal.cpp

```
#include <iostream>
#include <string>
using namespace std;
#include "cabecera1.h"

void FichaDatos::Mostrar()
{
       cout<<"El nombre es "<<nombre<<endl;
       cout<<"edad "<<edad;
}

void main (void)
{
       FichaDatos Paciente("Ana",29);

       Paciente.Mostrar();
       Paciente.Cargar();
       cout<<endl<<endl;
       Paciente.Mostrar();

}
```

Versión C# 2.0 : (Visual Studio 2005)

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Ejercicio_3_1
{
    public class Propiedades
    {
        public long identificador;
        public string propietario;
        public double precio;
    }

    public class Ejercicio1
    {
        static void Main()
        {
            Propiedades Casa = new Propiedades();
            Casa.identificador = 1001;
            Casa.propietario = "Joan Pera";
            Casa.precio = 999999;

            Console.WriteLine("Lista Propiedades"); ;
            Console.Write("Propiedad {0}:  ", Casa.identificador);
            Console.WriteLine("Propietario {0}: ", Casa.propietario);
            Console.Write(" precio:  ");Console.Write(Casa.precio);
            Console.WriteLine(); Console.WriteLine();
        }
    }
}
```

Versión C# 3.0 : (Visual Studio 2008)

```
using System;
namespace Ejercicio_3_1
{
    public class Propiedades
    {
        public long identificador;
        public string propietario;
        public double precio;
    }
    public class Exercici1
    {
        static void Main()
        {
            var Casa = new Propiedades();

            Casa.identificador = 1001;
            Casa.propietario = "Joan Mas";
            Casa.precio = 999999;

            Console.WriteLine("Lista Propiedades"); ;
            Console.Write("Propiedad {0}:  ", Casa.identificador);
            Console.WriteLine("Propietario {0}: ", Casa.propietario);
            Console.Write(" precio:  ");Console.Write(Casa.precio);
        }
    }
}
```

3.2) Creación objetos y asignación de variables propias de los objetos.

```
namespace Ejercicios_Cap3
{
    public class Propiedades
    {
        public long Identificador;
        private string propietario;
        public int habitaciones;
        double precio;

        public Propiedades (string nom, double precio)
        {
                propietario = nom;
                this.precio = precio;
        }

        public string Mostra_Propietario ()
        {
            return (this.propietario);
        }

        public double Mostra_precio()
        {
            return (this.precio);
        }
    }

    public class Exercicio_3_2
    {
        static void Main()
        {
            Propiedades Casa = new Propiedades("desconocido",999999);

            Casa.Identificador = 1001;
            Casa.habitaciones = 4;

            Propiedades Casa2 = new Propiedades("Maria Puyol", 300000);
            Casa2.Identificador = 1002;
            Casa2.habitaciones = 3;

            Console.WriteLine("Llistat Propiedades"); ;
            Console.Write("Propiedad {0}:  ", Casa.Identificador);
            Console.Write("Propietario:  ");
            string veure = Casa.Mostra_Propietario();
            Console.Write(veure);
            Console.Write("\n habitaciones:       ");
            Console.Write(Casa.habitaciones);
            Console.Write(" precio:  ");
            double veure_precio = Casa.Mostra_precio();
            Console.WriteLine(veure_precio.ToString());

            Console.Write("\n\n");

            Console.Write("Propiedad {0}:  ", Casa2.Identificador);
            Console.Write("Propietario:  ");
            veure = Casa2.Mostra_Propietario();
            Console.Write(veure);
            Console.Write("\n habitaciones:       ");
            Console.Write(Casa2.habitaciones);
            Console.Write(" precio:  ");
            veure_precio = Casa2.Mostra_precio();
            Console.WriteLine(veure_precio.ToString());

        }
    }
}
```

3.3) Asignación propiedades del objeto.

```
namespace Ejercicio_3_3
{
    public class Vehicles
    {
        private string marca;
        private double precio;
        private double potencia;

        public string Marca
        {
            get { return marca; }
            set { marca = value; }
        }


        public double FuncionPrecio
        {
            get
            {
                if (precio <= 0) return 0;
                else return precio;
            }
            set { precio = value; }
        }


        public double Potencia
        {
            get
            {
                if (potencia > 140)
                {
                    Console.WriteLine("Impuesto por alta poténcia");
                }
                return potencia;
            }
            set { potencia = value; }
        }
    }


    class Program
    {
        static void Main(string[] args)
        {
            Vehicles Familiar1 = new Vehicles();
            Familiar1.Marca = "Seat";
            Familiar1.Potencia = 150;
            Familiar1.FuncionPrecio = 999999;

            Console.WriteLine("Vehiculos en la tienda");
            Console.WriteLine("Marca: {0} ", Familiar1.Marca);
            Console.WriteLine("Potencia: {0} ", Familiar1.Potencia);
            Console.WriteLine("Precio: {0} ", Familiar1.FuncionPrecio);
        }
    }

}
```

3.4) Variables STATIC

```
using System;
using System.Collections.Generic;
using System.Text;

public class Libro
{
    public static string Titulo;
    public string Autor;
    double precio;

    public void asignar_precio(double numero)
    {
        precio = numero;
    }

    public void mostrar_precio(double numero)
    {
        Console.WriteLine("Precio: {0}", precio);
    }

}


public class Ejercicio4
{
    static void Main()
    {

        Libro Delgadoer = new Libro();
        Libro.Titulo = "Lenguaje C para todos";
        Delgadoer.Autor = "Macia Salavela";
        Delgadoer.asignar_precio(999);

        Console.WriteLine("Libro Delgadoero");
        Console.Write("Titulo:  ");
        Console.WriteLine(Libro.Titulo);
        Console.Write("Autor: ");
        Console.WriteLine(Delgadoer.Autor);
        Console.WriteLine();

        Libro Segundo = new Libro();
        Libro.Titulo = "Lenguaje D avanzado";
        Segundo.Autor = "Andrea Tejeiro";
        Segundo.asignar_precio(555);

        Console.WriteLine("Libro Segundo");
        Console.Write("Titulo:  ");
        Console.WriteLine(Libro.Titulo);
        Console.Write("Autor: ");
        Console.WriteLine(Segundo.Autor);

        Console.ReadLine();
    }
}
```

3.5) Trabajando con namespaces. Organización de clases

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Empresa
{
   public class Producto
   {
            public string identificador;
            public decimal precio;
   }

   namespace Empleado
   {
     public class Personal
     {
         public string nom;
         public string telefon;
     }

     public class FichaNomina
     {
        private double Nomina;
     }
   }

}


public class Programa
{
 static void Main()
   {
    Empresa.Producto Casa = new Empresa.Producto();

    Casa.identificador = "ref_pisMTH";
    Casa.precio = 999999;

    Console.WriteLine("Inmobiliaria CompraMolt");
    Console.Write("Propiedad :   ");
    Console.WriteLine(Casa.identificador);
    Console.Write("Precio: ");
    Console.WriteLine(Casa.precio);
    Console.WriteLine();

    Empresa.Empleado.Personal NuevaFicha = new Empresa.Empleado.Personal();
    NuevaFicha.nom = "Aina Costa";

    Console.Write("Nombre personal: ");
    Console.WriteLine(NuevaFicha.nom);
   }
}
```

3.6) A partir del namespace: Hospital , guardado en al archivo Clase_Hospital .cs , se crea la clase "Paciente" y se utiliza en otro archivo: Program.cs

Archivo: Clase_Hospital.cs

```
using System;

namespace Hospital
    {
        public class Paciente
        {
            public string nombre;
            public int edad;
            public decimal dias;

            public Paciente()
            {
                nombre = "sin asignar";
                edad = 99;
                dias = 0;
            }
        }
    }
```

Archivo: Program.cs

```
using System;
using Hospital;

public class Program
{
   static void Main()
      {

       Paciente nuevo = new Paciente();

       Console.WriteLine("Nombre pacient:  {0}",nuevo.nombre);
       Console.WriteLine("Edad:  {0}",nuevo.edad);
       Console.WriteLine("Dias hospitalizado:     {0}",nuevo.dias);
       Console.WriteLine("");

       }
}
```

3.7) Ejemplo de envío de un objeto como parámetro de una función.

```
using System;

namespace Hospital_Central
{
    public class Paciente
    {
        public string nombre;
        internal int dias;
    }


    public class Program
    {
     static void Main()
      {
          Paciente nuevo = new Paciente();

          Console.Write("Introducir nombre Paciente: ");
          nuevo.nombre = Console.ReadLine();

          Console.Write("Introducir dias: ");
          nuevo.dias = int.Parse(Console.ReadLine());

          Mostrar(nuevo);
      }


      private static void Mostrar(Paciente obj)
       {
        Console.WriteLine("Pacientes");
        Console.WriteLine("Nombre: {0} ", obj.nombre);
        Console.WriteLine("Dias hospitalizado: {0} ", obj.dias);
       }
     }

}
```

ÍNDICE