

Algoritmos y Programación

Práctica 4

- 1) Codifique la clase Hora de tal forma que al ejecutar el siguiente programa de aplicación (Main):

```
class Program
{
    public static void Main(string[] args)
    {
        Hora h=new Hora(23,30,15);
        h.imprimir();
        Console.ReadKey(true);
    }
}
```

se imprima por consola: 23 HORAS, 30 MINUTOS Y 15 SEGUNDOS

- 2) Agregue un segundo constructor a la clase Hora para que pueda recibir la hora en un solo parámetro de clase string. Así el siguiente código:

```
class Program
{
    public static void Main(string[] args)
    {
        Hora h1=new Hora(23,30,15);
        Hora h2=new Hora("14:25:47");
        h1.imprimir();
        h2.imprimir();
        Console.ReadKey(true);
    }
}
```

produce la siguiente salida por consola:

23 HORAS, 30 MINUTOS Y 15 SEGUNDOS

14 HORAS, 25 MINUTOS Y 47 SEGUNDOS

- 3) Defina una clase Persona con 3 campos: Nombre, Edad y DNI. En el constructor utilice parámetros cuyos nombres coincidan con los campos privados del objeto (usar this). Definir un método que permita imprimir los datos de la persona con el siguiente formato: Nombre (Edad) <TAB> DNI.

Ejemplo: Juan Perez (40) 2098745

- 4) Modifique el programa anterior agregando un campo privado FechaNacimiento. Defina un nuevo constructor que reciba como parámetro el nombre, la fecha de nacimiento (un DateTime) y el dni. En base a la fecha de nacimiento y la fecha actual debe calcular y guardar la edad. Utilice ambos constructores para resolver el ejercicio anterior, permitiendo ahora que el usuario tipee las entradas como: "Nombre<TAB>Documento<TAB>Edad<ENTER>"

o bien:

"Nombre<TAB>Documento<TAB>fecha de nacimiento<ENTER>".

- 5) Agregue a la clase Persona un método esMayorQue(Persona p) que devuelva verdadero si la persona que recibe el mensaje es más grande que la persona recibida como parámetro. Utilícelo para realizar un programa de aplicación (Main) que al recibir una lista de personas imprima los datos de la persona de mayor edad del grupo, con el formato estipulado en el ejercicio anterior.
- 6) Cree una clase llamada Operación con tres variables de instancia: operando1, operando2 y operador. Definir el constructor, todas las propiedades y el método **evaluar()** que será encargado de efectuar la operación solicitada según el tipo de operador ('+': sumar, '-': restar, '*': multiplicar o '/': dividir, operando 1 y operando 2). El programa de aplicación deberá imprimir todos los resultados solicitados y calculados.

Por ejemplo, si se ejecuta:

```
Operación op=new Operación(3,4,"+");
```

```
Console.WriteLine(op.evaluar()); //el programa imprime 7 según el ejemplo
```