

EXAMEN OBJETIVO FINAL PROGRAMACIÓN [CICLO D.A.W.]

NOMBRE Y APELLIDOS: _____ GRUPO: _____

1.- (4 puntos) Programa que lee de un fichero de texto, estas sencillas operaciones de cálculo. Una línea tendrá una instrucción y la siguiente sus datos)...

acumula <un número real>

<... añade al conteo el número.

divide <un número real>

<... divide el conteo acumulado entre el número indicado.

El programa muestra **al terminar de procesar el fichero** el resultado final, **y después** las operaciones realizadas (una por línea).

Al finalizar, el programa mostrará el resultado final obtenido.

Y después, por cada operación correctamente realizada, **las operaciones realizadas y su resultado parcial**.

El cálculo **empieza en cero**.

El nombre del fichero será **"op.num"**.

- Si en alguna línea, tiene un comando incorrecto, o tras un comando correcto no se encontrase dato o no fuese del tipo esperado, el comando no se aplicará (no se tendrá en cuenta).

Nota.- *Los comandos podrán estar escritos indistintamente en mayúsculas, o minúsculas.*

2. (6 puntos) Heredando de la clase *Persona* y de la interface *Diponible*, ...

```
namespace recuperacion;

public class Persona {
    private String nombre;
    private String apellidos;
    private int edad;
    public Persona (String nombre, String apellidos, int edad) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.edad = edad;
    }
    protected String getNombre () {
        return apellidos + ", " + nombre;
    }
    protected int getEdad () {
        return edad;
    }
}
```

```
namespace recuperacion;

public interface Disponible {
    bool haceVuelosTransatlanticos();
}
```

...escribe la clase *AsistenteDeCabina* en el espacio de nombres '*herencia*' con la que instanciar objetos, recibiendo en su único constructor los datos propios de cualquier *Persona* y además un número entero con las horas de vuelo del asistente de cabina.

Esta clase deberá implementar las interfaces ***Disponible*** y ***Comparable***, atendiendo a su comportamiento:

- Un *AsistenteDeCabina*, deberá poder informar de si ***haceVuelosTransatlanticos()***, o no.
 - Para hacer vuelos transatlanticos, deberá tener entre 25 y 50 años (ambos extremos incluidos) y tener más de 1000 horas de vuelo.
- Un *AsistenteDeVuelo*, se podrá comparar con otro asistente con el método ***compareTo()***
 - ... entendiendo que es menor el que menos horas de vuelo tenga.

Además esta clase debe incorporar:

- dos propiedades públicas de sólo lectura 'nombre' y 'edad' para poder consultar cómodamente los atributos heredados. (tener presente que se heredan privados)
- un atributo estático de sólo lectura que mantenga el número total de horas de todos los objetos *AsistenteDeCabina* creados.
- método ***ToString()*** para mostrar todos los datos del objeto.

Incluye un breve ejemplo creando un programa que instancie al menos 3 objetos *AsistenteDeCabina* dentro de una list que tras ser ordenada (mediante ***Sort()***) muestre los objetos en orden.