



Fundamentos de Programación Grado en Ingeniería Informática

- Febrero 2018 -

*Dpto. Tecnologías de la Información
Escuela Técnica Superior de Ingeniería - Universidad de Huelva*

NOMBRE y APELLIDOS:

NOTA

La puntuación máxima de este examen es de 5 puntos

Problema 1.-

2.5 puntos

La empresa de construcción y puesta en órbita de satélites Hispasat quiere desarrollar un software para la captación de datos meteorológicos que será instalado en sus satélites. Para ello desean implementar el siguiente software de prueba:

Para almacenar los datos captados utilizaremos la siguiente estructura:

```
struct Toma
{
    int dia, mes;                //Fecha de captación
    float Temperatura, Presion; //Datos captados
    bool Correcta;               //Datos captados correctamente.
};
```

En esta estructura se almacena el **día** y **mes** que fue realizada dicha toma, la **temperatura** y la **presión** medida y un campo **Correcta** que indica si la toma fue realizada correctamente o no por el satélite.

Para almacenar todos los datos tomados por un satélite disponemos de la siguiente clase que define los atributos y métodos básicos para realizar una funcionalidad básica.

```
typedef char Cadena[50];
class Satelite
{
    Toma Datos[10000]; //Tomas captadas por el satélite
    int NTomas;         //Nº de tomas realizadas.
    Cadena Nombre;      //Nombre del satélite
    bool Activo;        //Si es true, el satélite está activo.
public:
    Satelite();
    bool Activar(Cadena pNombre, Cadena pPassword);
    bool Esta_Activo();
    int Dame_NTomas();
    Toma Dame_Toma(int pPos);
    void Actualiza_Toma(int pPos, Toma pData);
    void Dame_Nombre(Cadena pNombre);
    float TemperaturaMedia(int pMes)
    int AnularDatos(int pMes);
};
```

Los métodos de esta clase son:

Satelite(): Constructor de la clase que inicializa los atributos de la clase.

bool Activar(Cadena pNombre, Cadena pPassword): método que activa el satélite siempre y cuando la contraseña sea exactamente a "123qF349.". Si la contraseña es correcta se actualiza el nombre del satélite con el nombre pasado por parámetro y su atributo de estado (Activo). Además, devolverá si se ha activado o no.

bool Esta_Activo(): Devuelve el estado del satélite.

int Dame_NTomas(): Devuelve el número de tomas realizadas por el satélite.

Toma Dame_Toma(int pPos): Devuelve la toma realizada por el satélite que esté en la posición indicada por el parámetro **pPos**.

void Actualiza_Toma(int pPos, Toma pData): Actualiza la toma del satélite de la posición **pPos** con la toma pasada por parámetro.

void Dame_Nombre(Cadena pNombre): Devuelve el nombre del satélite en el parámetro **pNombre**.

float TemperaturaMedia(int pMes): Método que devuelve la temperatura media de las tomas realizadas por el satélite en el mes indicado por el parámetro **pMes**. Si una toma es incorrecta no se tendrá en cuenta para obtener la media. Si el satélite está inactivo la media siempre será **0**.

int AnularDatos(int pMes): Método que anula (pone a un estado incorrecto) todas las tomas del satélite que se hayan realizado en el mes indicado por el parámetro **pMes**. Una toma no puede ser anulada si previamente ya estaba en un estado incorrecto. El método devolverá el número de tomas que se han anulado por el método. En el caso de que el satélite esté inactivo, no se anulará ninguna toma y el método devolverá **-1**.

**** Código a implementar ****

De la clase **Satélite** se han de implementar los métodos **Activar**, **Dame Nombre** y uno de los métodos **TemperaturaMedia** y **AnularDatos** que el alumno puede elegir.

Además, se pide implementar la siguiente función:

int Buscar(Satelite pLista[50], **int** pNSatelites, Cadena pNombre)

que dado un vector de satélites (**pLista**) y el número de satélites utilizados (**pNSatelites**), encuentre la posición del satélite que verifique que está activo, que su nombre coincide con el pasado por parámetro (**pNombre**) y que el número de tomas que ha realizado supere las **1000**, devolverá -1 si no se cumplen estas condiciones.

Problema 2.-

2.5 puntos

Dadas las clases alumno y asignatura supuestamente ya implementadas al completo:

```
typedef char cad[20];
class alumno {
    /* Datos de un alumno*/
    cad nombre, direc;
    int edad;
public:
    void nombre(cad nom);
    /*Devuelve en el parámetro nom el nombre del alumno */
    int leeredad();
    /* Devuelve los años que tiene el alumno */
    void direccion (cad dir);
    /*Devuelve en el parámetro dir la dirección donde vive el alumno*/
};

class asignatura {
    /* Alumnos matriculados en una asignatura*/
    cad nomasig;          /*Nombre de la asignatura */
    alumno  talumnos[200]; /*alumnos matriculados */
    int nalumnos; /*Numero de alumnos matriculados desde la posición 0 hasta la
    posición nalumnos-1 */
public:
    int leern();
    /* Devuelve el número de alumnos matriculados en esa asignatura, si hay cinco estarán en las
    cinco primeras posiciones de talumnos */
    void leeralumno (int i, alumno &a);
    /* Dada la posición i devuelve en a el alumno que se encuentra en esa posición*/
    void leernomasig (cad n);
    /*Devuelve en n el nombre de la asignatura*/
};
```

Diseña el método void grado::asignaturasalumno (cad nomalum);

```
class grado {
    /* Asignaturas que forman parte de ese grado*/
    asignatura  tasignaturas[60];
    int nasignaturas; /* Si hay 5 asignaturas estarán desde la posición 0 hasta la posición 4 de
    la tabla tasignaturas */
public:
    void asignaturasalumno (cad nomalum);
    /* Dado el nombre de un alumno mostrará por pantalla el nombre de las asignaturas en las que
    está matriculado*/
};
```