

Examen de Prácticas de Programación I. 11-09-2007

Se desea escribir un programa Java que gestione las calificaciones los alumnos de una asignatura. Para ello se dispone de una clase ya realizada llamada **Alumno** que posee la siguiente interfaz.

Constructor Summary	
<code>Alumno</code>	<code>(java.lang.String nombre)</code> Constructor que crea un alumno con el nombre que se le da como argumento

Method Summary	
boolean	<code>aprobado()</code> Determina si el alumno está aprobado: más de 5 en nota final, más de 5 en ejercicios y prácticas
java.lang.String	<code>nombre()</code> Retorna el nombre del alumno
double	<code>notaEjercicios()</code> Retorna la nota de los ejercicios
double	<code>notaExamen()</code> Retorna la nota del examen
double	<code>notaFinal()</code> Retorna la nota final calculada con un peso del 60% para el examen, 30% para las prácticas y 10% para los ejercicios
double	<code>notaPracticas()</code> Retorna la nota de prácticas
void	<code>ponNotaEjercicios(double nota)</code> Pone la nota de ejercicios
void	<code>ponNotaExamen(double nota)</code> Pone la nota del examen
void	<code>ponNotaPracticas(double nota)</code> Pone la nota de prácticas

Todas las notas son un valor numérico entre 0 y 10. La calificación final de un alumno se obtiene mediante la media ponderada de las calificaciones obtenidas en el examen, en las prácticas y en los ejercicios, tal y como se especifica en el método `notaFinal()`.

Se dispone también de una clase ya realizada denominada `Asignatura` con los siguientes atributos:

```
private int num; // numero de alumnos matriculados

private Alumno asig[]=new Alumno[99]; //tabla de alumnos
```

y la siguiente interfaz:

Constructor Summary	
<code>Asignatura</code> ()	Constructor que pone a cero el número de alumnos de la asignatura

Method Summary	
boolean	<code>añadir</code> (Alumno alumno) Si el alumno no está en la tabla y la tabla no está llena lo añade en la posición num, incrementando el contador num y retorna true; si el alumno está o la tabla está llena retorna false
int	<code>buscar</code> (java.lang.String nombreAlumno) si el alumno está en la tabla retorna la posición de la tabla entre 0..num-1 en la que el alumno se encuentra, si el alumno no está en la tabla retorna num
boolean	<code>llena</code> () indica si la asignatura está llena
int	<code>numeroMatriculados</code> () retorna el número de matriculados

Se pide:

1.- (1 punto) Añadir a la clase `Alumno` un método `leeDatos()` que lea los datos de un alumno (nombre, nota de examen, nota de

ejercicios, nota de prácticas) desde el teclado en un objeto de la clase [Lectura](#).

2. (1 punto) Añadir a la clase [Alumno](#) un método [muestraDatos\(\)](#) que muestre los datos de un alumno (nombre, nota final y si está o no aprobado) en un objeto de la clase [Escritura](#).

3. (2 puntos) Añadir a la clase [Asignatura](#) un método [ordena\(\)](#) que ordene la tabla de alumnos por nota final (de menor a mayor).

4. (2 punto) Añadir a la clase [Asignatura](#) un método [listaAprobados\(\)](#) que muestre por la pantalla los nombres de los alumnos aprobados siguiendo el orden de menor a mayor nota final.

5. (4 puntos) Diseñar y codificar un programa principal con un menú que se ejecute en un bucle hasta que el usuario escoja la opción salir que permita realizar las operaciones siguientes.

- a. Añadir un alumno leyendo sus datos desde el teclado.
- b. Mostrar los datos (nombre, nota final y si está o no aprobado) de un alumno cuyo nombre se lee desde el teclado.
- c. Mostrar por pantalla la lista de nombres de alumnos aprobados en orden de menor a mayor nota final.
- d. salir.