

## Práctica 2

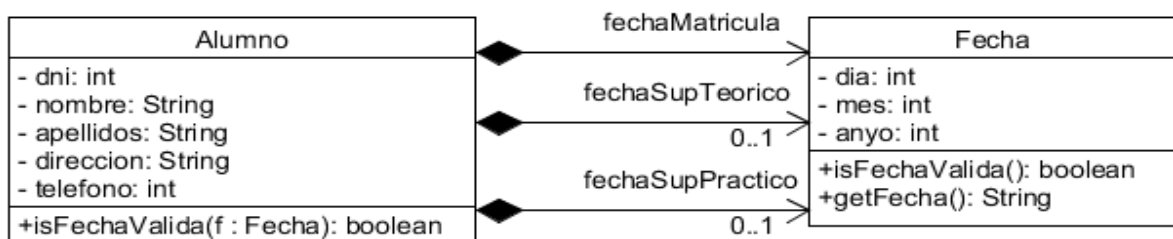
### Introducción al Diseño Orientado a Objetos

**Inicio:** Semana del 13 de febrero.  
**Duración:** 2 semanas.  
**Entrega:** Semana del 27 de febrero.  
**Peso de la práctica:** 15%

El objetivo de esta práctica es introducir los conceptos básicos de la programación orientada a objetos desde el punto de vista del diseño, así como de su correspondencia en Java, con particular énfasis en el diseño de clases con relaciones de herencia, asociación y composición mediante diagramas de clases en UML.

#### Apartado 1 (2.5 puntos):

Nos han encargado desarrollar en lenguaje Java las clases del siguiente diagrama de clases UML que se integrarán en una aplicación destinada a la gestión de una cadena de autoescuelas. La clase *Alumno* recoge la información que se necesita mantener de los alumnos, básicamente su DNI, nombre y apellidos, una dirección, un teléfono de contacto y tres fechas significativas: la fecha de matriculación, la fecha de superación del examen teórico y la fecha de superación del examen práctico. Para tratar con las fechas se empleará la clase *Fecha*, que permitirá comprobar si una fecha es válida. Por simplicidad, al crear un alumno solo se incluirá su fecha de matriculación, con los campos día, mes y año introducidos por separado.



Las operaciones que deben implementarse en las clases Java apropiadas son las siguientes:

- Comprobar si una fecha es válida. Para simplificar el cálculo se considerará que los días tienen un valor comprendido entre 1 y 31 y los meses un valor entre 1 y 12.
- Escribir la información de cada clase con el formato adecuado que puede verse en el tester que se muestra a continuación.

#### Se pide:

Desarrollar en Java las clases especificadas en el diagrama anterior de forma que tu implementación permita compilar y ejecutar el siguiente tester obteniendo la salida que se indica más abajo (**fíjate bien en los datos que se incluyen para crear el alumno. Esta información te será útil para escribir los constructores de las dos clases**)

#### Tester

```
package eps.uam.ads.p2.autoescuela.testers;

import eps.uam.ads.p2.autoescuela.alumnos.Alumno;

/**
 * Tester el primer apartado de la P2
 * @author Profesores ADS
 */
```

```

public class TesterAlumnos {

    public static void main(String[] args) {

        Alumno a1 = new Alumno(12345678, "Pedro", "García Gómez", "C/ Delicias 34",
644444243, 10, 2, 2016);
        Alumno a2 = new Alumno(22233444, "Lucía", "Pérez Rodríguez", "Avda. Mediterraneo
67", 655777372, 32, 5, 2016);
        Alumno a3 = new Alumno(33344666, "Beatriz", "Sánchez Ríos", "Paseo de los Olivos
54", 688990011, 4, 13, 2015);

        System.out.println("Datos del alumno " + a1);
        System.out.println();
        System.out.println("Datos del alumno " + a2);
        System.out.println();
        System.out.println("Datos del alumno " + a3);
        System.out.println();

        System.out.println("isValida <" + a1.getFechaMatricula().getFecha() + "> ? " +
a1.isFechaValida(a1.getFechaMatricula()));
        System.out.println("isValida <" + a2.getFechaMatricula().getFecha() + "> ? " +
a2.isFechaValida(a2.getFechaMatricula()));
        System.out.println("isValida <" + a3.getFechaMatricula().getFecha() + "> ? " +
a3.isFechaValida(a3.getFechaMatricula()));

        a1.setFechaSupTeorico(15, 3, 2016);
        a1.setFechaSupPractico(13, 5, 2016);

        System.out.println();
        System.out.println("El alumno " + a1.getNombreYApellidos() + " aprobó el examen
teórico el día " + a1.getFechaSupTeorico().getFecha());
        System.out.println("El alumno " + a1.getNombreYApellidos() + " aprobó el examen
práctico el día " + a1.getFechaSupPractico().getFecha());
    }
}

```

Siendo la salida esperada de la ejecución del programa la siguiente:

```

Datos del alumno <dni: 12345678>
Nombre y apellidos: Pedro García Gómez
Dirección: C/ Delicias 34
Teléfono: 644444243
Fecha de matriculación: 10/2/2016

Datos del alumno <dni: 22233444>
Nombre y apellidos: Lucía Pérez Rodríguez
Dirección: Avda. Mediterraneo 67
Teléfono: 655777372
Fecha de matriculación: 32/5/2016

Datos del alumno <dni: 33344666>
Nombre y apellidos: Beatriz Sánchez Ríos
Dirección: Paseo de los Olivos 54
Teléfono: 688990011
Fecha de matriculación: 4/13/2015

isValida <10/2/2016> ? true
isValida <32/5/2016> ? false
isValida <4/13/2015> ? false

El alumno Pedro García Gómez aprobó el examen teórico el día 15/3/2016
El alumno Pedro García Gómez aprobó el examen práctico el día 13/5/2016

```

## Apartado 2 (4.5 puntos):

El diagrama UML del apartado anterior contenía una pequeña parte del sistema de gestión de la cadena de autoescuelas. El objetivo de este apartado es diseñar el módulo completo teniendo en cuenta la siguiente información:

Las autoescuelas tienen un código interno, una dirección y un encargado. Además, disponen de una serie de aulas para impartir clases de teoría y una flota de coches para las clases de prácticas.

Los alumnos que se matriculan en las autoescuelas tienen las características que se mostraron en el apartado 1, por lo que esta parte del diseño ya estaría realizada. Sin embargo, debes relacionar esta parte con el resto de elementos del módulo. Por ejemplo, un alumno debería matricularse solo en una autoescuela.

Las clases que se imparten en cada autoescuela son de dos tipos: teóricas y prácticas. Todas las clases tienen una fecha y hora en la que se imparten, un precio básico (5 euros) y una duración que se mide en minutos. La fecha de cada clase debe tener un valor comprendido entre el 1-1-2015 y el 31-12-2017. Igual que se hizo con las fechas de interés respecto a los alumnos, para simplificar se considerará que el día tiene un valor entre 1 y 31 y el mes un valor entre 1 y 12. Las clases teóricas y las prácticas añaden al precio básico una cantidad extra diferente en cada caso. Además, las clases teóricas tienen un aula en el que se imparten y las clases prácticas un recorrido y un tipo que puede ser circuito o maniobras. El precio de la clase será simplemente la suma del precio básico y el precio de la clase teórica o práctica.

Cada clase es impartida por un único profesor de la autoescuela. La información de interés sobre los profesores incluye su DNI, su número de la Seguridad Social, su nombre y apellidos, el teléfono, la antigüedad en la autoescuela y el sueldo. Por norma general un profesor de la cadena de autoescuelas trabaja en una de ellas, pero si lo solicita puede cambiar a otra autoescuela de la cadena, de manera que interesa también saber la fecha de inicio y de fin de actividad del profesor en cada autoescuela. El salario que se paga al profesor está constituido de un sueldo base y de unos complementos que dependen del número de clases prácticas que ha impartido en cada mes. El sueldo base es de 900 euros y cada clase práctica se paga a 20 euros.

Para las clases prácticas, cada autoescuela dispone de una serie de coches, que tienen una matrícula, un número de bastidor, una marca y un modelo. El funcionamiento de la clase práctica permite que puedan ir un máximo de dos alumnos en el coche, de manera que se alternen en el recorrido. En cuanto a las clases de teoría, los alumnos pueden asistir a las que deseen sin mayores restricciones.

#### **Se pide:**

Realizar un diagrama de clases UML para el problema de la gestión de autoescuelas.

**Nota:** En el diagrama de clases en UML no es necesario incluir constructores, ni métodos *getters* y *setters*.

### **Apartado 3 (3 puntos):**

La cadena de autoescuelas para la que se ha realizado el diseño del apartado anterior ha llegado a un convenio con una compañía de talleres de reparación de vehículos, para que se encargue del mantenimiento de su flota de coches. A partir de ahora, cuando uno de los coches tenga un problema o deba pasar una revisión rutinaria se enviará a un taller de esta compañía, del que interesa saber su nombre comercial, dirección, encargado y teléfono. Los trabajos que se realicen en el coche serán por tanto de dos tipos: reparación o revisión. En el primer caso se debe obtener un presupuesto y una lista de los trabajos necesarios, así como las piezas que harían falta para reparar el vehículo. Además, en caso de que se acepte la reparación se deberá calcular el precio final, incluyendo un IVA del 21% y un descuento del 15% solo para las piezas en virtud del convenio entre ambas compañías. El caso de las revisiones es bastante más simple, ya que el acuerdo implica aplicar una cantidad fija, siempre que los trabajos sean los especificados de común acuerdo entre las partes: cambio de aceite, filtros y pastillas de freno.

#### **Se pide:**

Añadir las modificaciones necesarias al diseño UML realizado en el apartado anterior para representar la nueva funcionalidad de reparación de vehículos.

Sólo se deberá entregar el diagrama de clases UML modificado, no es necesario entregar la implementación *Java*.

### **Apartado 4 (Opcional, 1 punto):**

Implementa en Java las clases *Profesor* y *Autoescuela*, de manera que pueda obtenerse la información de los periodos de tiempo en que cada profesor ha impartido clases en una autoescuela cualquiera de la cadena. Para la implementación de Java utiliza la parte del diseño de tu diagrama UML que recoge esta funcionalidad, omitiendo aquellos aspectos que no sean necesarios e incluyendo otras clases que consideres indispensables. Por ejemplo, el programa debería ofrecer información como:

*El profesor Antonio Pérez García trabajó en la autoescuela Arenal desde el 2/4/2004 hasta el 4/10/2009.*

**Normas de Entrega:**

- Se deberán entregar los apartados 1, 2, 3 y, opcionalmente, el 4.
- La entrega la realizará uno de los alumnos de la pareja a través de Moodle.
- Si el ejercicio pide código Java, se entregará el código fuente, y la documentación generada con *JavaDoc*.
- Si el ejercicio pide un diagrama de diseño, se deberá entregar en PDF junto con una breve explicación (dos o tres párrafos a lo sumo).
- Se debe entregar un único fichero ZIP / RAR con todo lo solicitado, que deberá llamarse de la siguiente manera: GR<numero\_grupo>\_<nombre\_estudiantes>.zip. Por ejemplo Marisa y Pedro, del grupo 2261, entregarían el fichero: GR2261\_MarisaPedro.zip.
- La estructura de los ficheros entregados deberá estructurarse en directorios, uno por cada apartado.