

## CLASES Y OBJETOS

Un programa utiliza una serie de objetos y, normalmente, muchos son iguales. La descripción (modelo) de un tipo de dichos objetos es una clase. Una aplicación se compone de una serie de clases, produciendo objetos que interactúan entre sí.

El ejemplo conductor a lo largo del libro será el desarrollo de una aplicación que dé soporte informático a la matriculación universitaria de alumnos, en el control de las notas obtenidas en las diferentes asignaturas y en la generación de actas (listados de notas por asignatura). Un programa capaz de dar dicho soporte necesita modelar lo que es un "Alumno", una "Asignatura", un "Grupo", etc. Posteriormente, en el proceso de matriculación, muchos alumnos distintos se irán matriculando, habrá un conjunto de asignaturas que cursar, se podrá elegir entre diferentes grupos de clase y habrá horario de mañana y tarde. Es importante la diferencia entre el modelo de "Alumno" de los diferentes alumnos concretos que se matriculan. En POO (Programación Orientada a Objetos) se denomina "clase" al modelo o descripción de "Alumno" y se llama "objeto" o "instancia de una clase" a los alumnos concretos que maneja un programa. En definitiva, el programador define las clases que considera necesarias y luego crea objetos de las clases definidas.

En su forma más sencilla, una clase se define con la palabra reservada `class`, seguida del nombre elegido para la clase y un bloque de definición, que se delimita por llaves `{}`. Dentro de las llaves irán los detalles descriptivos de la clase.

```
class Alumno { ... detalles ... }
```

Una vez que la clase está disponible, el programador puede instanciar (es decir, crear) objetos de dicha clase. Por ello, los términos "objeto" e "instancia de clase" o, simplemente, "instancia" se usan como sinónimos.

Como ejemplo se van a usar dos instancias de la clase `Alumno` para obtener un par de alumnos concretos, `alumno1` y `alumno2`. Evidentemente, en la Universidad se matricularán muchos alumnos desconocidos *a priori*, pero en posteriores capítulos (6 y 9) se resolverá este punto.

La definición consiste en el nombre de la clase seguido del nombre elegido para el objeto.

```
Alumno alumno1;
Alumno alumno2;
```

Con esto hemos definido dos objetos, de nombre `alumno1` y `alumno2`, que usaremos para modelar un par de alumnos. Pero esta definición es sólo una declaración de intenciones: "este programa dispondrá de dos objetos, llamados `alumno1` y `alumno2`, de la clase `Alumno`". Para crear los objetos en sí, se usa la construcción `new` seguida del nombre de la clase de la cual se quiere crear los objetos, seguidos de una pareja abre-paréntesis, cierra-paréntesis.

```
alumno1= new Alumno();
alumno2= new Alumno();
```

Puede parecer redundante volver a poner el nombre de la clase después de `new`. Al final del capítulo veremos que hay otras posibilidades. Lo importante que ahora tenemos dos objetos de la clase `Alumno` y que el programa cono

dichos objetos por su nombre o referencia (`alumno1`, `alumno2`). Con estas tres líneas, hemos obtenido tres cosas: la clase (incompleta) `Alumno` (`class Alumno`), dos objetos de dicha clase (`new Alumno`) y dos referencias para manejar dichos objetos (`alumno1` y `alumno2`).

## ¿En qué consiste una clase?

Una clase es una descripción de un tipo de objetos. Para ello se disponen de dos tipos de componentes de los objetos:

1. Por una parte, se debe guardar la información del objeto. En el ejemplo anterior, de un alumno se necesita saber, por ejemplo, la siguiente información: el nombre, los apellidos, el curso en que está matriculado, horario de mañana o tarde, etc. Estos datos se almacenan en unos campos o atributos que se declaran poniendo el tipo del valor del atributo seguido del nombre que se les quiere dar:

```
enum Horario { MAÑANA, TARDE } // posibles horarios

class Alumno {
    String nombre;
    String apellidos;
    int añoDeNacimiento;
    int NP; // Número Personal: identificador único
    String grupo;
    Horario horario;
}
```

**Ejemplo 2.1.** Definición de atributos.

En este ejemplo, se ha declarado que la clase `Alumno` contiene 6 atributos, nombre, apellidos, añoDeNacimiento, NP, grupo y horario. El `Horario` se ha definido como un enumerado con dos posibles turnos, `MAÑANA` y `TARDE`.

2. Por otra parte, se debe poder definir el comportamiento del objeto en sus interacciones con otros objetos. Este comportamiento se define mediante los métodos que posee la clase. Siguiendo con el ejemplo, de un (objeto) `Alumno` se puede pedir su nombre, asignarle grupo, etc.

```
enum Horario { MAÑANA, TARDE } // posibles horarios

class Alumno {
    String nombre;
    String apellidos;
    int añoDeNacimiento;
    int NP; // Número Personal: identificador único
    String grupo;
    Horario horario;

    public String dameGrupo() { ... }
    public void ponGrupo(String nuevoGrupo) { ... }
    ...
}
```

**Ejemplo 2.2.** Declaración de atributos y métodos.