Programación Orientada a Objetos
Curso 2023/2024
Ejercicio: Cursos

Queremos desarrollar una aplicación para la gestión de cursos organizados por una academia de formación.

1. Funcionalidad básica

La clase **Alumno** representa la persona que puede realizar un curso. Un alumno se caracteriza por tener las siguientes propiedades:

- *Nombre*: cadena de texto que identifica al alumno. Su valor no puede cambiar una vez inicializado en la construcción.
- *Dni*: es una cadena de texto que no puede cambiar una vez establecida en la construcción.
- *Crédito:* representa el dinero disponible para el pago de los cursos.
- *Cursos matriculados:* conjunto de cursos en los que el alumno se ha matriculado. El tipo de datos Curso se describe más adelante.

Para la construcción se requiere obligatoriamente el nombre y dni del alumno. El crédito disponible puede establecerse de manera opcional en el constructor. Por defecto, si no se establece, el valor inicial será de 100 euros.

La funcionalidad que ofrece la clase, además de la consulta de sus propiedades es:

- Incrementar el crédito en una cantidad establecida como parámetro.
- Decrementar el crédito en la una cantidad establecida como parámetro.
- Añadir curso. Métodos para añadir un curso al conjunto de cursos matriculados.

La clase **Curso** representa los cursos que pueden realizar los alumnos. La clase se caracteriza por las siguientes propiedades:

- *título* del curso.
- fecha de *inicio* del curso.
- fecha de *finalización* del curso.
- *precio* de matrícula. Número real que representa el dinero que hay que pagar para matricularse del curso.
- *alumnos matriculados*: conjunto de alumnos que se han matriculado del curso.
- *número de alumnos* matriculados.

El título, las fechas de inicio y finalización y el precio se establecen en el constructor. La colección de alumnos matriculados inicialmente estará vacía.

Para implementar parte de la funcionalidad de la clase Curso se tiene que hacer uso de los conceptos de **clase abstracta** y **método plantilla** dado que en la clase curso se implementan los algoritmos generales de comportamiento que dependen de funcionalidad propia de los tipos de cursos (los tipos de cursos se describen más adelante). La funcionalidad que ofrece la clase curso es la siguiente:

- Consultar si un curso ha terminado. Se considerará que el curso ha terminado si la fecha actual del sistema es posterior a la fecha de finalización del curso.
- *Matricular*. Esta operación recibe como parámetro el alumno que se quiere matricular del curso y devuelve un valor booleano para indicar si la operación se ha realizado con éxito. Un alumno podrá matricularse en un curso si cumple los requisitos establecidos para ello.

Por regla general, un alumno se podrá matricular en un curso si: 1) el curso todavía no ha empezado, 2) si tiene crédito suficiente para afrontar el precio de la matrícula y 3) si cumple los *requisitos particulares* del curso del que se quiere matricular. Los requisitos particulares dependen de cada tipo de curso. En el caso de que el alumno cumpla los requisitos (tanto los generales como los particulares de cada curso), dicho alumno se añadirá al conjunto de alumnos matriculados, el crédito del alumno se decrementa según el precio de la matrícula y el curso quedará registrado en el conjunto de cursos del alumno. El método debe devolver un valor booleano para informar si el alumno ha podido matricularse o no.

2. Tipos de cursos

Un **curso online** es un tipo de curso que se caracteriza por establecer como requisito para la matriculación que el alumno haya realizado un conjunto de cursos con anterioridad. Estos cursos previos se establecen en la construcción y no pueden cambiar (argumento variable). Por tanto, el requisito particular de matriculación de un curso online es que el alumno contenga en su conjunto de cursos matriculados los cursos previos establecidos.

Un **curso presencial** es un tipo de curso que, como su nombre indica, se imparte presencialmente. Las propiedades que caracterizan este tipo de cursos son:

- cupo: número máximo de alumnos que pueden matricularse.
- plazas libres: número de alumnos que aún pueden matricularse en el curso, esto es, el cupo menos el número de alumnos matriculados.

El cupo se establece en la construcción y no pueden cambiar.

Los requisitos particulares para la matriculación en un curso presencial es que queden plazas libres, esto es, que no se haya alcanzado el cupo.

3. Programa

Implementa el siguiente programa para probar la funcionalidad:

- Declara una variable que referencie a un Alumno con DNI “34678904” y nombre “Pepe”.
- Declara una variable que referencie a un Alumno con DNI “17679456” y nombre “Andrea” con crédito inicial de 50€.
- Declara una variable que referencie a un curso presencial de título “Diseño de Bases de Datos” con fecha de inicio y fin dentro de un mes. El precio del curso es de 55€ y el un cupo de 20 alumnos.
- Declara una variable que referencie a un curso online de título “Administración de Bases de Datos” con fecha de inicio mañana y fin dentro de una semana. El precio del curso es de 25€ y como requisito que se haya realizado previamente el curso presencial de “Diseño de Bases de Datos”.
- Crea una colección de cursos y añade los cursos creados anteriormente.
- Recorre la colección de cursos y para cada curso
    - matricula a todos los alumnos.
    - Muestra la información del curso en la consola. Para ello debes implementar el método `toString` en las clases implementadas. De los alumnos se debe mostrar el nombre, dni, crédito y el número de cursos en los que está matriculado.

NOTA: Sólo debe estar matriculado Pepe en los dos cursos porque Andrea no tenía crédito suficiente para matricularse del curso presencial que es requisito para el curso online.