
Digital
House >
Coding School

Parte A

1. Realizar un diagrama de clase que modele a la clase Alumno.
   En principio, un alumno posee un nombre **(String)**, un apellido **(String)** y un código de alumno **(Integer)**.

2. Implementar la clase creando los atributos necesarios.

3. Crear un constructor para el alumno que tome por parámetro un nombre, un apellido y un código de alumno.

4. Crear los getter y setters necesarios.

Parte B

1. Realizar un diagrama de clase que modele a la clase Curso.
   En principio, un curso posee un nombre **(String)** y un código de curso **(Integer)**.

2. Implementar la clase creando los atributos necesarios.

3. Crear los getter y setters necesarios para los atributos anteriores.

Parte C

1. Realizar un diagrama de clase que modele a la clase Profesor.
   En principio, un profesor posee un nombre **(String)**, un apellido **(String)**, una antigüedad **(Integer)** y un código de profesor **(Integer)**.

2. Implementar la clase creando los atributos necesarios.

3. Crear los getter y setters necesarios para los atributos anteriores.

Parte D

Se quiere agregar al modelo anterior dos nuevas categorías de profesores.

Los profesores titulares y los profesores adjuntos. Un profesor titular tiene una especialidad **(String)** y un profesor adjunto tiene una cantidad de horas que dedica para consultas **(Integer)**

1. ¿Cómo extendemos el diagrama de clases que realizó anteriormente?

2. Modificar la implementación contemplando los nuevos cambios. Crear las clases que sean necesarias.

3. Crear los getters y setters necesarios para los nuevos atributos.

Parte E

Un curso además de tener un nombre y un código de curso, posee un profesor titular **(ProfesorTitular)**, un profesor adjunto **(ProfesorAdjunto)**, un cupo máximo de alumnos **(Integer)** y una lista de alumnos inscriptos **(Alumnos)**.

1. ¿Cómo modificaría el diagrama de clase de Curso que realizó anteriormente?
2. Modificar la implementación contemplando los nuevos cambios.
3. Crear un método que permita listar los alumnos.
4. Crear un método que permita obtener a los profesores del curso.

Parte F

1. Realizar un diagrama de clase que modele a la clase DigitalHouseManager.
   En principio, DigitalHouseManager tiene una lista de alumnos **(Alumnos)**, una lista de profesores **(Profesores)** y una lista de cursos **(Cursos)**.

2. Implementar la clase creando los atributos necesarios.

Parte G

1. Crear un método en la clase **Curso** que permita agregar un alumno a la lista. El método devolverá **true** si el alumno puede agregarse o **false** en caso de que no haya cupo disponible.

    - **public function agregarUnAlumno(Alumno $unAlumno)**

Parte H

1. Crear un método en la clase **DigitalHouseManager** que permita dar de alta a un curso. El método recibe como parámetros el nombre del curso, el código y el cupo máximo de alumnos que se admite. El método debe crear un curso con los datos correspondientes y agregarlo a la lista de cursos.
    - **public function altaCurso($nombre, $codigoCurso, $cupoMaximoDealumnos)**

2. Crear un método en la clase **DigitalHouseManager** que permita dar de alta a un profesor adjunto. El método recibe como parámetros el nombre del profesor, el apellido, el código y la cantidad de horas disponibles para consulta. La antigüedad inicial del profesor será cero. El método debe crear un profesor adjunto con los datos correspondientes y agregarlo a la lista de profesores.
    - **public function altaProfesorAdjunto($nombre, $apellido, $codigoProfesor, $cantidadDeHoras)**

3. Crear un método en la clase **DigitalHouseManager** que permita dar de alta a un profesor titular. El método recibe como parámetros el nombre del profesor, el apellido, el código y la especialidad. La antigüedad inicial del profesor será cero. El método debe crear un profesor titular con los datos correspondientes y agregarlo a la lista de profesores.
    - **public function altaProfesorTitular($nombre, $apellido, $codigoProfesor, $especialidad)**

4. Cear un método en la clase **DigitalHouseManager** que permita dar de alta a un alumno. El método recibe como parámetros el nombre, el apellido y el código del alumno. El método debe crear un alumno con los datos correspondientes y agregarlo a la lista de alumnos.
    - **public function altaAlumno($nombre, $apellido, $codigoAlumno)**

Parte I

5. Crear un método en la clase **DigitalHouseManager** que permita inscribir un alumno a un curso. El método recibe como parámetros el código del alumno y código del curso al que se inscribe.
    - **public function inscribirAlumno($codigoAlumno, $codigoCurso)**

    El método debe:

    - Tomar el curso al que se quiere inscribir de la lista de cursos.
    - Tomar al alumno al que se quiere inscribir de a lista de alumnos.
    - Inscribir al alumno si hay cupo disponible.
        - Informar por pantalla la inscripción.
    - Si no hay cupo disponible:
        - Informar por pantalla que no se pudo inscribir porque no hay cupo

6. Crear un método en la clase **DigitalHouseManager** que permita asignar a un curso sus profesores. El método recibe como parámetros el código del curso, el código del profesor titular y el código del profesor adjunto

    - **public function asignarProfesores($codigoCurso, $codigoProfesorTitular, $codigoProfesorAdjunto)**

    El método debe:

    - Buscar al profesor titular en la lista de profesores.
    - Buscar al profesor adjunto en la lista de profesores.
    - Asignarle al curso ambos profesores.

Parte J

1. Crear un archivo DH.php e incluir todas las clases.

2. Dar de alta dos profesores titulares y dos profesores adjuntos. (Inventar todos sus valores)

3. Dar de alta dos cursos.

    - **Nombre del curso:** **Full Stack**
      Código del curso: **20001**
      Cupo máximo: **3**

    - **Nombre del curso:** **Android**
      Código del curso: **20002**
      Cupo máximo: **2**

4. Asignarle un profesor titular y un adjunto a cada curso.

5. Dar de alta a tres alumnos. (Inventar todos sus valores).

6. Inscribir a dos alumnos en el curso de Full Stack.

7. Inscribir a dos alumnos en el curso de Android.

8. Inscribir a tres alumnos más en el curso de Android.

Adicionales

1. ¿Cómo modificaría el diagrama de clases para que se le pueda consultar a un alumno a qué cursos está inscripto?

2. Crear un método en la clase **Curso** que permita eliminar un alumno de la lista de alumnos del curso.

    - **public eliminarAlumno(Alumno unAlumno)**

3. Crear un método en la clase **DigitalHouseManager** que permita dar de baja un curso. El método recibe como parámetro el código del curso. El método debe utilizar el código del curso para buscarlo en la lista de cursos y eliminarlo de la lista.

    - **public function bajaCurso($codigoCurso)**

4. Crear un método en la clase **DigitalHouseManager** que permita dar de baja a un profesor. El método recibe como parámetro el código del profesor. El método debe utilizar el código del profesor para buscarlo en la lista de profesores y eliminarlo de la lista.

    - **public function bajaProfesor($codigoProfesor)**