



PROGRAMACIÓN ORIENTADA A OBJETOS

Recuperatorio Primer Parcial 2022



HOJA 1 / 4

Desarrollar en base a la siguiente descripción y diagrama de clases.

Se requiere desarrollar un sistema que maneje la información académica de los alumnos. Para eso se necesita contar con información de ellos, su historial académico, materias y profesores.

De un alumno se necesita conocer nombre, apellido, dni y su historial académico.

Para el historial académico se contará con una clase que contendrá un hashmap de materias como clave y una nota de tipo integer.

De los profesores se necesita conocer su nombre, apellido, dni y número de legajo.

Por otro lado, para las materias es necesario contar con nombre, código, año, un profesor y el listado de los alumnos que estén inscriptos a ella.

Una materia debe poder recibir inscripciones de alumnos. Para cada una debe validar si el alumno ya está inscripto, en caso negativo lo añadirá a la lista de alumnos y de lo contrario imprimirá un mensaje en consola: "El alumno ya se encuentra inscripto en la materia". **Este proceso no añade la materia al historial académico del alumno. Dicha acción se realizará después.**

Una materia debe poseer un método para imprimir el listado de alumnos inscriptos en ella.

Una materia debe poseer un método toString que retorne sus datos, excepto el listado de alumnos y el profesor.

Un alumno, al finalizar la cursada, debe poder recibir una materia y su nota. Al hacerlo deberá poder añadir esa nota y la materia al historial académico.

Un alumno debe poseer un método para imprimir su historial académico, solicitando dicha tarea a su historial previa impresión de sus propios datos personales, esta impresión deberá realizarla el mismo método.

Un historial académico de un alumno debe poder recibir una materia y una nota para asignarla a su hashmap. Además, debe poseer un método para imprimir todo el historial académico del alumno.



PROGRAMACIÓN ORIENTADA A OBJETOS

Recuperatorio Primer Parcial 2022



HOJA 2 / 4

Pruebas y resultados esperados

Una vez resuelto el modelo solicitado deberán efectuarse las siguientes acciones para poner a prueba el diseño:

Datos de prueba

- ~~Crear 30 alumnos.~~
- Crear 5 materias.
- Crear 5 profesores.
- Cada materia debe tener un profesor.
- Cada materia debe tener entre 5 y 10 alumnos elegidos de forma aleatoria.

Consideraciones

- Un alumno puede estar en más de una materia
- Un alumno puede no estar en ninguna materia.
- Un alumno solo puede estar una vez en cada materia.
- Un profesor solo puede estar en una materia.

Simulación

- Generar para toda materia y todos sus alumnos una nota aleatoria entre 1 y 10, debe ser asignada al historial académico del alumno.

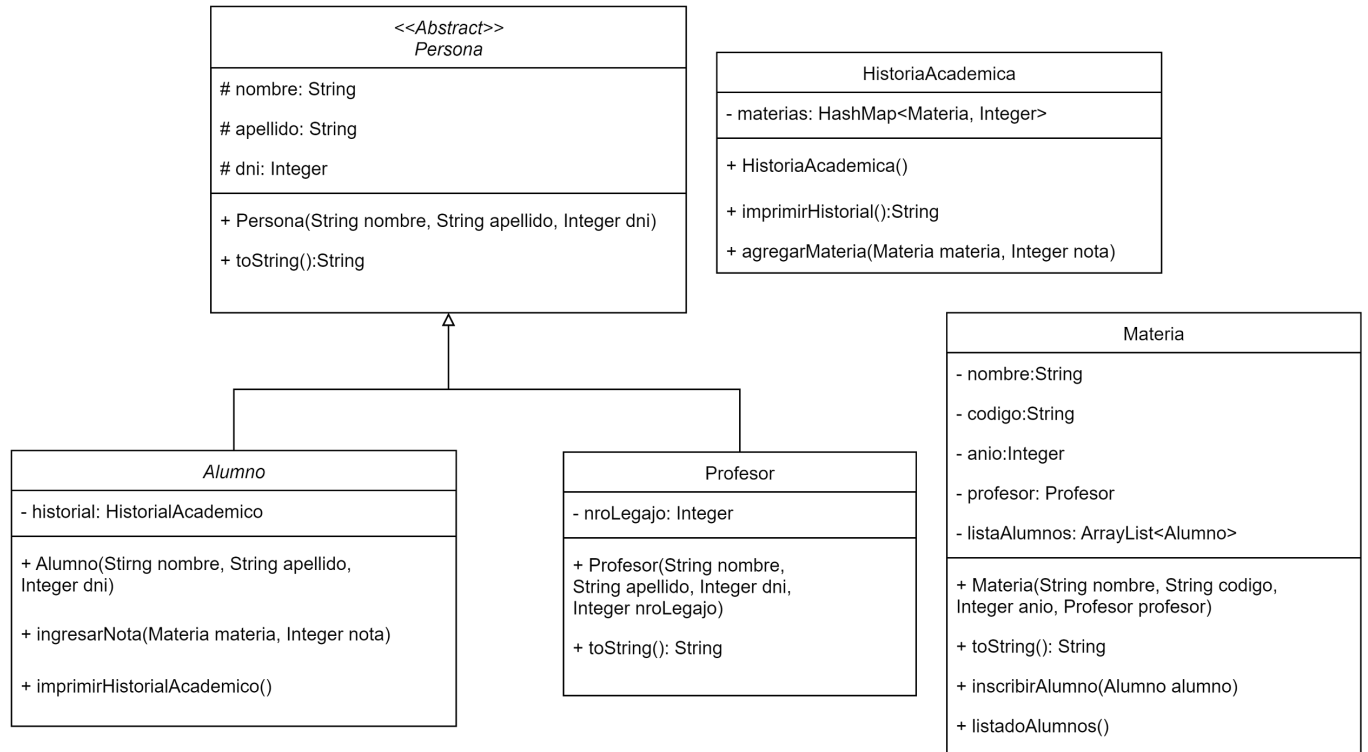
Salidas en consola

- 1) Los datos de cada materia, el profesor asignado y el listado de alumnos inscriptos.
- 2) Los datos de 10 alumnos aleatorios y su historial académico.

Extra(No exigido)

- Agregar a la impresión del historial académico el promedio del alumno.
- Lanzar, propagar y controlar una excepción para cuando el alumno ya se encuentra inscripto en una materia.
- Añadir un método para eliminar la inscripción de un alumno a una materia. Debe incluir un caso de prueba para validarlo.

Diagrama de Clase:



Aclaración en las clases utilice los getters y setters que crean necesario para resolver el enunciado.

Ejercicio 2

Dada las siguientes tres clases y un main que hace uso de ellas, responder a lo siguiente:

A- ¿Cómo podría abstraer el comportamiento de sacar fotos de las clases para que en main se pueda programar pensando en objetos que pueden sacar fotos?

Plantee el código faltante y modifique el existente para hacer uso de polimorfismo.

B- ¿Cómo agregaría polimorfismo estático al método sacarFoto si ahora también es posible añadir un delay al momento de la captura? Ejemplifique con una clase.

```

3 public class Telefono {
4
5     public void sacarFoto() {
6         System.out.println("saco foto desde un telefono celular.");
7     }
8
9 }

```



```
3 public class Satelite {
4
5     public void sacarFoto() {
6         System.out.println("saco foto desde un satelite.");
7     }
8
9 }

3 public class CamaraSeguridad {
4
5     public void sacarFoto() {
6         System.out.println("saco foto desde una camara de seguridad.");
7     }
8
9 }

3 public class Main {
4     public static void main(String[] args) {
5
6         Telefono telefono1 = new Telefono();
7         Telefono telefono2 = new Telefono();
8         Telefono telefono3 = new Telefono();
9         Satelite satelite1 = new Satelite();
10        Satelite satelite2 = new Satelite();
11        Satelite satelite3 = new Satelite();
12        CamaraSeguridad camara1 = new CamaraSeguridad();
13        CamaraSeguridad camara2 = new CamaraSeguridad();
14        CamaraSeguridad camara3 = new CamaraSeguridad();
15
16        telefono1.sacarFoto();
17        telefono2.sacarFoto();
18        telefono3.sacarFoto();
19
20        satelite1.sacarFoto();
21        satelite2.sacarFoto();
22        satelite3.sacarFoto();
23
24        camara1.sacarFoto();
25        camara2.sacarFoto();
26        camara3.sacarFoto();
27    }
28 }
```