

ACTIVIDAD POO (PROGRAMACIÓN ORIENTADA A OBJETOS, TEMAS CLASES, OBJETOS, CONSTRUCTORES Y HERENCIA)

Objetivo: repasar y afianzar los temas vistos sobre Programación Orientada a Objetos en Java.

EJERCICIO 1

Construya una clase llamada Asignatura que represente el nombre de una asignatura y la nota correspondiente obtenida. Se debe implementar:

- Constructor que acepte como parámetros el nombre de la asignatura y la nota obtenida.
- Métodos para modificar la nota (setNota()) y para consultar la nota (getNota()).
- Método que nos devuelva "Aprobado" si la nota es mayor o igual a 3.5 o "Reprobado" si la nota es menor que 3.5.
- Método para consultar el nombre de la asignatura.

Ahora implemente la clase Alumno que incluya referencias a máximo cuatro Asignaturas a las que el alumno ha asistido. Además de incluir los atributos nombre y edad. Las operaciones disponibles sobre el alumno son:

- Constructor que acepte como parámetro el nombre del alumno y edad.
- Métodos para modificar el nombre (setNombre) y para consultarlo (getNombre).
- Métodos para modificar y consultar la edad.
- Método que nos devuelva el promedio del alumno.
- Método para agregar una Asignatura a su plan de estudio; verificar que la asignatura no exista previamente las asignaturas de ese Alumno.
- Implemente la clase Aplicación para hacer uso de las clases Alumno y Asignatura.

Crear 3 alumnos (tres instancias de la clase Alumno) con sus respectivos nombres y edad. Para cada alumno establecer sus asignaturas y la nota obtenida.

Imprimir en pantalla:

- Nombre del alumno.
- Edad.
- Asignaturas que cursó
- Nombre de la asignatura.
- Nota obtenida.
- Si es una asignatura aprobada o no.
- Promedio del alumno

EJERCICIO 2: Interfaces en Java



El ornitorrinco es un animal muy especial que ha tenido pensando a los biólogos, pues implementa las capacidades de otros animales, lo que le permite ser mamífero, colocar huevos, respirar bajo el agua y hasta envenenar.

Lo anterior lo convierte en un candidato ideal para que apliquemos el tema de las **interfaces en Java**. De modo que deberás crear interfaces para:

- **Anfibio**
- **Oviparo**
- **Veneno**

Construye una clase abstracta llamada **Mamifero.java** y una llamada **Ornitorrinco.java** que herede de **Mamifero**. La clase **Ornitorrinco** deberá implementar las interfaces de **Oviparo, Anfibio y venenoso**; de este modo se deben sobrescribir los métodos de dichas interfaces. Finalmente deberás crear una clase llamada **main.java** en la que se creen objetos de la clase Ornitorrinco, que hagan uso de **todos** los métodos sobreescritos de la clases padre y de las interfaces. La clase ornitorrinco deberá incluir un método constructor que modifique el atributo **huevos**, de modo que, cada vez que se cree una instancia de ornitorrinco se modifique dicho valor. Así las cosas, la ejecución del programa permita salidas similares a esta:

```
Respirando por branquias ✓
He puesto: 2 huevitos ✓
He puesto: 3 huevitos ✓

Process finished with exit code 0
```