fundación esplai
ciudadanía comprometida

T10 - POO

*Reto 1

Crearemos una clase llamada **Serie** con las siguientes características:

- Sus atributos son **titulo, numero de temporadas**, **entregado, genero y creador.**
- Por defecto, el numero de temporadas es de 3 temporadas y entregado **false**. El resto de atributos serán valores por defecto según el tipo del atributo.

Los constructores que se implementaran serán:

- ✓ Un constructor con el titulo y creador. El resto por defecto.

Los métodos que se implementara serán:

- Métodos get de todos los atributos, excepto de entregado.
- Métodos set de todos los atributos, excepto de entregado.
- Sobrescribe los métodos toString.

Crearemos una clase **Videojuego** con las siguientes características:

- Sus atributos son **titulo, horas estimadas, entregado, genero y compañia**.
- Por defecto, las horas estimadas serán de 10 horas y entregado false. El resto de atributos serán valores por defecto según el tipo del atributo.

Los constructores que se implementaran serán:

- ✓ Un constructor con todos los atributos, excepto de entregado.

Los métodos que se implementara serán:

- Métodos get de todos los atributos, excepto de entregado.
- Métodos set de todos los atributos, excepto de entregado.
- Sobrescribe los métodos toString.

Como vemos, en principio, las clases anteriores no son padre-hija, pero si tienen en común, por eso vamos a hacer una interfaz llamada **Entregable** con los siguientes métodos:

- **entregar()**: cambia el atributo prestado a true.
- **devolver()**: cambia el atributo prestado a false.
- **isEntregado()**: devuelve el estado del atributo prestado.
- Método **compareTo (Object a)**, compara las horas estimadas en los videojuegos y en las series el numero de temporadas. Como parámetro que tenga un objeto, no es necesario que implementes la interfaz Comparable. Recuerda el uso de los casting de objetos.

Implementa los anteriores métodos en las clases Videojuego y Serie. Ahora crea una aplicación ejecutable y realiza lo siguiente:

- Crea dos arrays, uno de **Series** y otro de **Videojuegos**, de 5 posiciones cada uno.
- Crea un objeto en cada posición del array, con los valores que desees, puedes usar distintos constructores.
- Entrega algunos **Videojuegos** y **Series** con el método **entregar()**.
- Cuenta cuantos **Series** y **Videojuegos** hay entregados. Al contarlos, devuélvelos.
- Por último, indica el **Videojuego** tiene más horas estimadas y la serie con mas temporadas. Muestralos en pantalla con toda su información (usa el método toString()).

fundación esplai
ciudadanía comprometida

*Reto 2

Crear una clase Libro que contenga los siguientes atributos:

- ISBN
- Titulo
- Autor
- Número de páginas

Crear sus respectivos métodos get y set correspondientes para cada atributo.

Crear el método toString() para mostrar la información relativa al libro con el siguiente formato: “El libro con ISBN creado por el autor tiene páginas”

En el fichero main, crear 2 objetos Libro (los valores que se quieran) y mostrarlos por pantalla.

Por último, indicar cuál de los 2 tiene más páginas.

fundación esplai
ciudadanía comprometida

*Reto 3

Vamos a realizar una clase llamada Raices, donde representaremos los valores de una ecuación de 2º grado.
Tendremos los 3 coeficientes como atributos, llamémosles a, b y c.
Hay que insertar estos 3 valores para construir el objeto.
Las operaciones que se podrán hacer son las siguientes:

- getDiscriminante(): devuelve el valor del discriminante (double), el discriminante tiene la siguiente formula, (b^2)-4*a*c
- tieneRaices(): devuelve un booleano indicando si tiene dos soluciones, para que esto ocurra, el discriminante debe ser mayor o igual que 0.
- tieneRaiz(): devuelve un booleano indicando si tiene una única solución, para que esto ocurra, el discriminante debe ser igual que 0.
- calcular(): mostrara por consola las posibles soluciones que tiene nuestra ecuación, en caso de no existir solución, mostrarlo también.
- obtenerRaices(): imprime las 2 posibles soluciones
- obtenerRaiz(): imprime única raíz, que será cuando solo tenga una solución posible.

Formula ecuación 2º grado: (-b±√((b^2)-(4*a*c)))/(2*a)
Solo varia el signo delante de -b

fundación esplai
ciudadanía comprometida

*Reto 4

Queremos representar con programación orientada a objetos, un aula con estudiantes y un profesor.

Tanto de los estudiantes como de los profesores necesitamos saber su nombre, edad y sexo. De los estudiantes, queremos saber también su calificación actual (entre 0 y 10) y del profesor que materia da.

Las materias disponibles son matemáticas, filosofía y física.

Los estudiantes tendrán un 50% de hacer novillos, por lo que si hacen novillos no van a clase pero aunque no vayan quedara registrado en el aula (como que cada uno tiene su sitio).

El profesor tiene un 20% de no encontrarse disponible (reuniones, baja, etc.)

Las dos operaciones anteriores deben llamarse igual en Estudiante y Profesor (polimorfismo).

El aula debe tener un identificador numérico, el número máximo de estudiantes y para que esta destinada (matemáticas, filosofía o física). Piensa que más atributos necesita.

Un aula para que se pueda dar clase necesita que el profesor esté disponible, que el profesor de la materia correspondiente en el aula correspondiente (un profesor de filosofía no puede dar en un aula de matemáticas) y que haya más del 50% de alumnos.

El objetivo es crear un aula de alumnos y un profesor y determinar si puede darse clase, teniendo en cuenta las condiciones antes dichas.

Si se puede dar clase mostrar cuantos alumnos y alumnas (por separado) están aprobados de momento (imaginad que les están entregando las notas).

NOTA: Los datos pueden ser aleatorios (nombres, edad, calificaciones, etc.) siempre y cuando tengan sentido (edad no puede ser 80 en un estudiante o calificación ser 12).

fundación esplai
ciudadanía comprometida

*Reto 5

Nos piden hacer un programa orientado a objetos sobre un cine (solo de una sala) tiene un conjunto de asientos (8 filas por 9 columnas, por ejemplo).

Del cine nos interesa conocer la película que se está reproduciendo y el precio de la entrada en el cine.

De las películas nos interesa saber el título, duración, edad mínima y director.

Del espectador, nos interesa saber su nombre, edad y el dinero que tiene.

Los asientos son etiquetados por una letra (columna) y un número (fila), la fila 1 empieza al final de la matriz como se muestra en la tabla. También deberemos saber si está ocupado o no el asiento.

8 A 8 B 8 C 8 D 8 E 8 F 8 G 8 H 8 I
7 A 7 B 7 C 7 D 7 E 7 F 7 G 7 H 7 I
6 A 6 B 6 C 6 D 6 E 6 F 6 G 6 H 6 I
5 A 5 B 5 C 5 D 5 E 5 F 5 G 5 H 5 I
4 A 4 B 4 C 4 D 4 E 4 F 4 G 4 H 4 I
3 A 3 B 3 C 3 D 3 E 3 F 3 G 3 H 3 I
2 A 2 B 2 C 2 D 2 E 2 F 2 G 2 H 2 I
1 A 1 B 1 C 1 D 1 E 1 F 1 G 1 H 1 I

Realizaremos una pequeña simulación, en el que generaremos muchos espectadores y los sentaremos aleatoriamente (no podemos donde ya este ocupado).

En esta versión sentaremos a los espectadores de uno en uno.

Solo se podrá sentar si tienen el suficiente dinero, hay espacio libre y tiene edad para ver la película, en caso de que el asiento este ocupado le buscamos uno libre.

Los datos del espectador y la película pueden ser totalmente aleatorios.

fundaciónesplai
ciudadanía comprometida

Jose Marín – FullStack Developer

fundacionesplai.org

josé-marín-20013460