EJERCICIO GUIADO. JAVA: POO. CLASES PROPIAS

Objetos propios del lenguaje

Hasta el momento, todos los objetos que ha usado a la hora de programar en Java, han sido objetos incluidos en el propio lenguaje, que se encuentran disponibles para que el programador los use en sus programas.

> *Estos objetos son: las etiquetas (JLabel), botones (JButton), los cuadros de texto (JTextField), cuadros de verificación (JCheckBox), botones de opción (JRadioButton), colores (Color), imágenes de icono (ImageIcon), modelos de lista (DefaultListModel), etc, etc.*

Todos estos objetos, por supuesto, pertenecen cada uno de ellos a una Clase:

> *Las etiquetas son objetos de la clase JLabel, los botones son objetos de la clase JButton, etc.*

Todos estos objetos tienen propiedades que pueden ser cambiadas en la ventana de diseño:

> *Ancho, alto, color, alineación, etc.*

Aunque también poseen métodos que nos permiten cambiar estas propiedades durante la ejecución del programa:

> *setText cambia el texto del objeto, setBackground cambia el color de fondo del objeto, setVisible hace visible o invisible al objeto, setBounds cambia el tamaño y la posición del objeto, etc.*

En cualquier momento le podemos pedir a un objeto que nos de información sobre sí mismo usando los métodos get:

> *getText obtenemos el texto que tenga el objeto, getWidth obtenemos la anchura del objeto, getHeight obtenemos la altura del objeto, etc.*

Los objetos son como “pequeños robots” a los que se les puede dar órdenes, usando los métodos que tienen disponible.

> *Por ejemplo, le podemos decir a un objeto que se pinte de nuevo usando el método repaint, podemos ordenarle a un cuadro de texto que coja el cursor, con el método requestFocus, etc.*

Todos estos objetos, con sus propiedades y métodos, nos facilitan la creación de nuestros programas. Pero a medida que nos vamos introduciendo en el mundo de la programación y nos especializamos en un tipo de programa en concreto, puede ser necesaria la creación de objetos propios, programados por nosotros mismos, de forma que puedan ser usados en nuestros futuros programas.

Objetos propios

A la hora de diseñar un objeto de creación propia, tendremos que pensar qué propiedades debe tener dicho objeto, y métodos serán necesarios para poder trabajar con él. Dicho de otra forma, debemos pensar en qué características debe tener el objeto y qué órdenes le podré dar.

Para crear objetos propios hay que programar la Clase del objeto. Una vez programada la Clase, ya podremos generar objetos de dicha clase, declarándolos y construyéndolos como si de cualquier otro objeto se tratara.

A continuación se propondrá un caso práctico de creación de objetos propios, con el que trabajaremos en las próximas hojas.

Lo que viene a continuación es un planteamiento teórico de diseño de una Clase de Objetos.

CASO PRÁCTICO: MULTICINES AVENIDA

Planteamiento

Los Multicines Avenida nos encargan un programa para facilitar las distintas gestiones que se realizan en dichos multicines.

El multicine cuenta con varias salas, y cada una de ellas genera una serie de información. Para facilitar el control de la información de cada sala programaremos una Clase de objeto a la que llamaremos **SalaCine**.

La Clase *SalaCine*

La Clase *SalaCine* definirá características de una sala de cine, y permitirá crear objetos que representen salas de cine. Cuando la Clase *SalaCine* esté programada, se podrán hacer cosas como las que sigue:

Los Multicines Avenida tienen una sala central donde se proyectan normalmente los estrenos. Se podría crear un objeto llamado *central* de la clase *SalaCine* de la siguiente forma:

```
SalaCine central;
```

Por supuesto, este objeto puede ser construido como cualquier otro:

```
central = new SalaCine();
```

El objeto *central* representará a la sala de cine central de los Multicines Avenida.

Otro ejemplo. Los Multicines Avenida tienen una sala donde proyectan versiones originales. Se podría crear un objeto llamado *salaVO* de la clase *SalaCine* de la siguiente forma:

```
SalaCine salaVO;  //declaración

salaVO = new SalaCine(); //construcción
```

Propiedades de los objetos de la clase *SalaCine*

A la hora de decidir las propiedades de un objeto de creación propia, tenemos que preguntarnos, ¿qué información me interesa almacenar del objeto? Trasladando esta idea a nuestro caso práctico, ¿qué información me interesaría tener de cada sala de cine?

Para este ejemplo supondremos que de cada sala de cine nos interesa tener conocimiento de las siguientes características (propiedades):

- **Aforo**: define el número de butacas de la sala (un número entero).
- **Ocupadas**: define el número de butacas ocupadas (un número entero).
- **Película**: define la película que se está proyectando en el momento en la sala (una cadena de texto)
- **Entrada**: define el precio de la entrada (un número double)

Valores por defecto de los objetos *SalaCine*

Cuando se construye un objeto, se asignan unos valores por defecto a sus propiedades. Por ejemplo, cuando se construye una etiqueta (Clase JLabel), esta tiene un tamaño inicial definido, un color, etc.

Lo mismo se tiene que hacer con los objetos propios que definimos. Es necesario decidir qué valores tendrá las propiedades del objeto al construirse.

En nuestro caso, las características de un objeto *SalaCine* en el momento de construirse serán las siguientes:

Aforo: 100
Ocupadas: 0
Película: “” *(la cadena vacía)*
Entrada: 5,00

Observa este código, en él construimos el objeto correspondiente a la sala central del multicine:

```
SalaCine central;

central = new SalaCine();
```

En este momento (en el que el objeto *central* está recién construido) este objeto tiene asignado un aforo de 100, el número de butacas ocupadas es 0, la película que se proyecta en la sala central es “” y la entrada para esta sala es de 5 euros.

Los valores por defecto que se asignan a los objetos de una clase son valores arbitrarios que el programador decidirá según su conveniencia.

Métodos de los objetos *SalaCine*

Para comunicarnos con los objetos de la Clase *SalaCine* que construyamos, tendremos que disponer de un conjunto de métodos que nos permitan asignar valores a las propiedades de los objetos, recoger información de dichos objetos y que le den órdenes al objeto.

Será el programador el que decida qué métodos le interesará que posea los objetos de la Clase que está programando.

Para nuestro caso particular, supondremos que los objetos de la Clase *SalaCine* deberán tener los siguientes métodos:

Métodos de cambio de propiedades (Métodos *set*)

setAforo - asignará un aforo a la sala de cine
setOcupadas - asignará una cantidad de butacas ocupadas a la sala de cine
setLibres - asignará una cantidad de butacas libres a la sala de cine
setPelicula - asignará un título de película a la sala de cine
setEntrada - fijará el precio de las entradas a la sala de cine

Gracias a estos métodos podemos dar forma a un objeto *SalaCine* recién creado.

Por ejemplo, supongamos que queremos crear el objeto que representa la sala de versiones originales. Resulta que esta sala tiene de aforo 50 localidades, que se está proyectando la película “Metrópolis” y que la entrada para ver la película es de 3 euros. La sala está vacía de momento.

Para crear el objeto, se usaría el siguiente código:

```
//Se construye el objeto
SalaCine salaVO;
salaVO = new SalaCine();

//Se le asignan características
salaVO.setAforo(50);                //aforo 50
salaVO.setPelicula("Metrópolis");  //la película que se proyecta
salaVO.setEntrada(3);               //entrada a 3 euros
```

Al construir el objeto *salaVO* tiene por defecto los valores siguientes en sus propiedades:

Aforo: 100
Ocupadas: 0
Película: ""
Entrada: 5,00

Gracias a los métodos disponibles hemos asignados estos nuevos valores:

Aforo: 50
Ocupadas: 0
Película: "Metrópolis"
Entrada: 3,00

Métodos para pedirle información al objeto (Métodos get)

Se programarán los siguientes métodos get en la clase *SalaCine*:

getAforo - devuelve el aforo que tiene el objeto
getOcupadas - devuelve el número de butacas ocupadas que tiene el objeto
getLibres - devuelve el número de butacas que tiene libres el objeto
getPorcentaje - devolverá el porcentaje de ocupación de la sala
getIngresos - devolverá los ingresos producidos por la sala (entradas vendidas por precio)
getPelicula - devolverá el título de la película que se está proyectando
getEntrada - devolverá el precio de la entrada asignado al objeto

Estos métodos nos permitirán obtener información de un objeto del tipo *SalaCine*. Por ejemplo, supongamos que tenemos el objeto llamado *central* (correspondiente a la sala principal del multicine), para obtener la película que se está proyectando en dicha sala solo habría que usar este código:

```
String peli;      //una variable de cadena
peli = central.getPelicula();
```

O, por ejemplo, para saber los ingresos producidos por la sala central…

```
double ingresos;
ingresos = central.getIngresos();
```

Métodos para dar órdenes al objeto

Se programarán los siguientes métodos para dar órdenes a los objetos de la clase *SalaCine*.

vaciar

- Este método pondrá el número de plazas ocupadas a cero y le asignará a la película la cadena vacía.

entraUno

- Este método le dice al objeto que ha entrado una nueva persona en la sala. (Esto debe producir que el número de plazas ocupadas aumente en uno)

RESUMEN SALACINE

He aquí un resumen de la Clase de objetos *SalaCine*, la cual se programará en la próxima hoja:

Clase de objetos: ***SalaCine***

Propiedades de los objetos *SalaCine*:

Aforo - número entero (int)
Ocupadas - número entero (int)
Película - cadena (String)
Entrada - número decimal (double)

Valores por defecto de los objetos del tipo *SalaCine*:

Aforo: **100**
Ocupadas: **0**
Película: **(cadena vacía)**
Entrada: **5**

Métodos de los objetos del tipo *SalaCine*:

Métodos de asignación de propiedades (set)

setAforo - modifica la propiedad Aforo
setOcupadas - modifica la propiedad Ocupadas
setLibres - modifica la propiedad Ocupadas también
setPelicula - modifica la propiedad Película
setEntrada - modifica la propiedad Entrada

Métodos de petición de información (get)

getAforo - devuelve el valor de la propiedad Aforo
getOcupadas - devuelve el valor de la propiedad Ocupadas
getLibres - devuelve el número de butacas libres
getPorcentaje - devuelve el porcentaje de ocupación de la sala
getIngresos - devuelve los ingresos obtenidos por la venta de entradas
getPelicula - devuelve el valor de la propiedad Película
getEntrada - devuelve el valor de la propiedad Entrada

Métodos de orden

Vaciar - vacía la ocupación de la sala y borra la película
entraUno - le indica al objeto que ha entrado una persona más en la sala

EJERCICIO PRÁCTICO

Supongamos que programamos una Clase de objetos llamada *Rectangulo*, la cual permitirá construir objetos que representen a rectángulos.

1. ¿Cómo declararía y construiría un objeto llamado *suelo* del tipo *Rectangulo*?

2. Supongamos que las propiedades de los objetos de la clase *Rectangulo* sean las siguientes:

    a. Base (double)
    b. Altura (double)

    ¿Qué métodos de tipo *set* programaría para cambiar las propiedades de los objetos del tipo *Rectangulo*?

3. Como ejemplo de los métodos anteriores, suponga que quiere asignar al objeto *suelo* una base de 30 y una altura de 50. ¿Qué código usaría?

4. Teniendo en cuenta que nos puede interesar conocer el área de un objeto *Rectangulo* y su perímetro, y teniendo en cuenta que los objetos *Rectangulo* tienen dos propiedades (Base y Altura), ¿qué cuatro métodos *get* serían interesantes de programar para los objetos del tipo *Rectangulo*?

5. Teniendo en cuenta los métodos del punto 4. Supongamos que quiero almacenar en una variable double llamada *area* el área del objeto *suelo*. ¿Qué código usaría? Y si quiero almacenar el perímetro del objeto *suelo* en una variable llamada *peri*?

CONCLUSIÓN

Para crear un objeto propio, necesita tener claro lo siguiente:

Nombre de la Clase del objeto.

Propiedades que tendrán los objetos de dicha clase.

Valores iniciales que tendrán las propiedades cuando se construya cada objeto.

Métodos que serán interesantes de programar:

Métodos de ajuste de propiedades (set)
Métodos de petición de información (get)
Métodos de orden

Una vez que se tenga claro lo anterior, se podrá empezar la programación de la Clase de objetos.