LABORATORIO: 01 JAVA: INTRODUCCIÓN A LA POO

Introducción a la Programación Orientada a Objetos

La programación orientada a objetos es una nueva forma de entender la creación de programas. Esta filosofía de programación se basa en el concepto de objeto.

Objeto

Un objeto se define como un elemento que tiene unas **propiedades** y **métodos**.

Un objeto posee unas características (ancho, alto, color, etc…) A las características de un objeto se les llama **propiedades**.

Un objeto es un elemento "inteligente". A un objeto se le puede dar órdenes y él obedecerá. A estas órdenes se les llama **métodos**. Con los métodos podemos cambiar las características del objeto, pedirle información, o hacer que el objeto reaccione de alguna forma.

En Java todo son objetos. Veamos algunos ejemplos de uso de objetos en Java:

Ejemplo 1

Supongamos que tenemos una etiqueta llamada *etiTexto*. Esta etiqueta **es un objeto**.

Como objeto que es, la etiqueta *etiTexto* tiene una serie de características, como por ejemplo: el color de fondo, el tamaño, la posición que ocupa en la ventana, el ser opaca o no, el ser invisible o no, etc… Son las *propiedades* de la etiqueta.

A una etiqueta le podemos dar órdenes, a través de *métodos*.

A través de los métodos podemos por ejemplo cambiar las características del objeto. Por ejemplo, se puede cambiar el tamaño y posición de la etiqueta usando el método *setBounds*:

```
etiTexto.setBounds(10,20,100,20);
```

Normalmente, los métodos que permiten cambiar las características del objeto son métodos cuyo nombre empieza por *set*.

Los métodos también permiten pedirle al objeto que me de información. Por ejemplo, podríamos usar el conocido método *getText* para recoger el texto que contenga la etiqueta y almacenarlo en una variable:

```
String texto;
texto = etiTexto.getText();
```

Los métodos que le piden información al objeto suelen tener un nombre que empieza por *get*.

Los métodos también sirven para ordenarle al objeto que haga cosas. Por ejemplo, podemos ordenar a la etiqueta *etiTexto* que se vuelva a pintar en la ventana usando el método *repaint*:

```
etiTexto.repaint();
```

Ejemplo 2

Supongamos que tenemos un cuadro de texto llamado *txtCuadro*. Como todo en Java, un cuadro de texto es un **objeto**.

Un objeto tiene **propiedades**, es decir, características. Nuestro cuadro de texto *txtCuadro* tiene características propias: un color de fondo, un ancho, un alto, una posición en la ventana, el estar activado o no, el estar visible o no, etc…

A un objeto se le puede dar órdenes, llamadas **métodos**. Estas órdenes nos permiten cambiar las características del objeto, pedirle información, o simplemente pedirle al objeto que haga algo.

Por ejemplo, podemos cambiar el color de fondo del cuadro de texto *txtCuadro* usando el método llamado *setBackground*:

```
txtCuadro.setBackground(Color.RED);
```

Otros métodos que permiten cambiar las propiedades del objeto *txtCuadro* son:

setVisible - permite poner visible / invisible el cuadro de texto
setEnabled - permite activar / desactivar el cuadro de texto
setEditable - permite hacer que se pueda escribir o no en el cuadro de texto
setText - permite introducir un texto en el cuadro de texto
setBounds - permite cambiar el tamaño y posición del objeto
setToolTipText - permite asociar un texto de ayuda al cuadro de texto
etc…

Un objeto nos da información sobre él. Para pedirle información a un objeto usaremos métodos del tipo *get*. Por ejemplo, para pedirle al cuadro de texto el texto que contiene, usaremos el método *getText*:

```
String cadena = txtCuadro.getText();
```

Otros métodos que le piden información al cuadro de texto son:

getWidth - te dice la anchura que tiene el cuadro de texto
getHeight - te dice el alto que tiene el cuadro de texto
getSelectedText - te devuelve el texto que está seleccionado dentro del cuadro de texto
getToolTipText - te dice el texto de ayuda que tiene asociado el cuadro de texto
etc…

También se le puede dar al objeto simplemente órdenes para que haga algo. Por ejemplo, podemos ordenar al cuadro de texto *txtCuadro* que seleccione todo el texto que contiene en su interior a través del método *selectAll*:

```
txtCuadro.selectAll();
```

Otros métodos que ordenan al cuadro de texto son:

repaint - le ordena al cuadro de texto que se vuelva a pintar
copy - le ordena al cuadro de texto que copie el texto que tenga seleccionado
cut - le ordena al cuadro de texto que corte el texto que tenga seleccionado
paste - le ordena al cuadro que pegue el texto que se hubiera copiado o cortado
etc…

Clase

Todo objeto es de una “clase” determinada, o dicho de otra forma, tiene un “tipo de datos” determinado.

Por ejemplo, las etiquetas que se usan en las ventanas son objetos que pertenecen a la clase *JLabel*. Los cuadros de texto en cambio son objetos de la clase *JTextField*.

Para poder usar un objeto hay que *declararlo* y *construirlo*.

Declarar un Objeto

La declaración de un objeto es algo similar a la declaración de una variable. Es en este momento cuando se le da un nombre al objeto. Para declarar un objeto se sigue la siguiente sintaxis:

```
Clase nombreobjeto;
```

Por ejemplo, para declarar la etiqueta del ejemplo 1, se usaría el siguiente código:

```
JLabel etiTexto;
```

Para declarar, en cambio, el cuadro de texto del ejemplo 2, se usaría el siguiente código:

```
JTextField txtCuadro;
```

Construir un Objeto

En el momento de la “construcción” de un objeto, se le asignan a este una serie de propiedades iniciales. Es decir, unas características por defecto. Se puede decir que es el momento en que “nace” el objeto, y este nace ya con una forma predeterminada, que luego el programador podrá cambiar usando los métodos del objeto.

Es necesario construir el objeto para poder usarlo. La construcción del objeto se hace a través del siguiente código general:

```
nombreobjeto = new Clase();
```

Por ejemplo, para construir la etiqueta del ejemplo 1, se haría lo siguiente:

```
etiTexto = new JLabel();
```

Para construir el cuadro de texto del ejemplo 2, se haría lo siguiente:

```
txtCuadro = new JTextField();
```

NOTA. En algunos casos, la sintaxis de la declaración y la construcción se une en una sola línea. Por ejemplo, supongamos que queremos declarar la etiqueta *etiTexto* y construirla todo en una línea, entonces se puede hacer lo siguiente:

```
JLabel etiTexto = new JLabel();
```

En general, para declarar y construir un objeto en una sola línea se sigue la siguiente sintaxis:

```
Clase nombreobjeto = new Clase();
```

La Clase como generadora de objetos

Conociendo la Clase, se pueden crear tantos objetos de dicha Clase como se quiera. Es decir, la Clase de un objeto funciona como si fuera una plantilla a partir de la cual fabricamos objetos iguales. Todos los objetos creados a partir de una clase son iguales en un primer momento (cuando se construyen) aunque luego el programador puede dar forma a cada objeto cambiando sus propiedades.

Por ejemplo, la Clase JLabel define etiquetas. Esto quiere decir que podemos crear muchas etiquetas usando esta clase:

```
JLabel etiTexto = new JLabel();
JLabel etiResultado = new JLabel();
JLabel etiDato = new JLabel();
```

En el ejemplo se han declarado y construido tres etiquetas llamadas *etiTexto, etiResultado* y *etiDato*. Las tres etiquetas en este momento son iguales, pero a través de los distintos métodos podemos dar forma a cada una:

```
etiTexto.setBackground(Color.RED);
etiTexto.setText("Hola");
etiResultado.setBackground(Color.GREEN);
etiResultado.setText("Error");
etiDato.setBackground(Color.BLUE);
etiDato.setText("Cadena");
```

En el ejemplo se le ha dado, usando el método *setBackground*, un color a cada etiqueta. Y se ha cambiado el texto de cada una. Se le da forma a cada etiqueta.

EJERCICIO

Hasta ahora ha usado objetos aunque no tenga mucha conciencia de ello. Por ejemplo ha usado botones. Como ejercicio se propone lo siguiente:

- ¿Cuál es el nombre de la clase de los botones normales que usa en sus ventanas?
- ¿Cómo declararía un botón llamado *btnAceptar*, y otro llamado *btnCancelar*?
- ¿Cómo construiría dichos botones?
- Indique algunos métodos para cambiar propiedades de dichos botones (métodos set)
- Indique algunos métodos para pedirle información a dichos botones (métodos get)
- Indique algún método para dar órdenes a dichos botones (algún método que no sea ni set ni get)

CONCLUSIÓN

Un Objeto es un elemento que tiene una serie de características llamadas PROPIEDADES.

Por otro lado, al objeto se le pueden dar órdenes que cumplirá de inmediato. A dichas órdenes se les denomina MÉTODOS.

Los métodos se pueden dividir básicamente en tres tipos:

- **Para cambiar las propiedades del objeto (Métodos set)**
- **Para pedir información al objeto (Métodos get)**
- **Para dar órdenes al objeto.**

Todo objeto pertenece a una CLASE.

La CLASE nos permite declarar objetos y construirlos:

Declaración:

```
CLASE nombreobjeto;
```

Construcción:

```
nombreobjeto = new CLASE();
```

Declaración y Construcción en una misma línea

```
CLASE nombreobjeto = new CLASE(),
```

En la construcción de un objeto se asignan unas propiedades (características) por defecto al objeto que se construye, aunque luego, estas características pueden ser cambiadas por el programador.