

3.C. Trabajando con objetos.

Marcar como hecha

1. Utilización de objetos.

1.2. Declaración.

Para la creación de un objeto hay que seguir los siguientes pasos:

- **Declaración:** Definir el tipo de objeto.
- **Instanciación:** Creación del objeto utilizando el operador **new**.

Pero ¿en qué consisten estos pasos a nivel de programación en Java? Veamos primero cómo declarar un objeto. Para la definición del tipo de objeto debemos emplear la siguiente instrucción:

```
<tipo> nombre_objeto;
```

Donde:

- **tipo** es la clase a partir de la cual se va a crear el objeto, y

- **nombre_objeto** es el nombre de la variable referencia con la cual nos referiremos al objeto.

Los tipos referenciados o referencias se utilizan para guardar la dirección de los datos en la memoria del ordenador.

Nada más crear una referencia, ésta se encuentra vacía. Cuando una referencia a un objeto no contiene ninguna instancia se dice que es una referencia nula, es decir, que contiene el valor **null**. Esto quiere decir que la referencia está creada pero que el objeto no está instanciado todavía, por eso la referencia apunta a un objeto inexistente llamado "nulo".

Para entender mejor la declaración de objetos veamos un ejemplo. Cuando veíamos los tipos de datos en la Unidad 2, decíamos que Java proporciona un tipo de dato especial para los textos o cadenas de caracteres que era el tipo de dato **String**. Veíamos que realmente este tipo de dato es un **tipo referenciado** y creábamos una variable mensaje de ese tipo de dato de la siguiente forma:

```
String mensaje;
```

Los nombres de la clase empiezan con mayúscula, como **String**, y los nombres de los objetos con minúscula, como **mensaje**, así sabemos qué tipo de elemento utilizando.

Pues bien, **String** es realmente la clase a partir de la cual creamos nuestro objeto llamado **mensaje**.

Si observas poco se diferencia esta declaración de las declaraciones de variables que hacíamos para los tipos primitivos. Antes decíamos que **mensaje** era una variable del tipo de dato **String**. Ahora realmente vemos que **mensaje** es un objeto de la clase **String**. Pero **mensaje** aún no contiene el objeto porque no ha sido instanciado, veamos cómo hacerlo.

Por tanto, cuando creamos un objeto estamos haciendo uso de una variable que almacena la dirección de ese objeto en memoria. Esa variable es una **referencia** o un **tipo de datos referenciado**, porque no contiene el dato si no la posición del dato en la memoria del ordenador.

```
String saludo = new String ("Bienvenido a Java");
```

```
String s; //s vale null
```

```
s = saludo; //asignación de referencias
```

En las instrucciones anteriores, las variables **s** y **saludo** apuntan al mismo objeto de la clase **String**. Esto implica que cualquier modificación en el objeto saludo modifica también el objeto al que hace referencia la variable **s**, ya que realmente son el mismo.

◀ 3.B. Objetos y clases.

Ir a...

3.D. Métodos. ▶