Paradigma
Orientado a Objetos

Módulo 09:
Herramientas de instanciación

por Fernando Dodino
revisado por Lucas Spigariol
Versión 1.1
Julio 2019

Indice

1 Objetos que no se instancian explícitamente

Los objetos anónimos se creaban en forma programática:

```
const alumno = object {
	method estudia() = true
}
```

Mientras que los objetos nombrados (*wko*, como pepita) se instanciaban solos en el momento en que uno ejecuta la consola REPL, o bien un test, o un programa.

También están los literales de Wollok: objetos que tienen una sintaxis propia para construirse:

```
2          // permite referenciar al número 2
"hola"     // construye un String cuyo valor es "hola"
[7]        // construye una lista con un único elemento: el número 7
```

2 Instanciación sin parámetros

Para poder crear una instancia de una clase, necesitamos hacer new y todas las referencias de la clase tienen que tener una valor inicial.

Entonces, si tenemos esta definición

```
class Ave {
	var energia = 0

	method volar(kms) { energia = energia - (kms * 5) }
}
```

sabemos que para crear un ave en la consola podemos hacerlo de esta manera:

```
>>> const cormoran = new Ave()
```

¿Cómo queda inicializado el cormorán? En este caso hay una sola referencia, energía, que tiene como valor 0. Podemos enviar el mensaje volar sin problemas, el objeto está correctamente construido.

3 Instanciación con parámetros

Como vimos en el ejemplo del Ave, cuando el objeto es lo suficientemente simple, nos alcanza con tener variables previamente inicializadas en su definición. Pero muchas veces esto es insuficiente o no es posible de realizar.

Wollok permite inicializar las referencias al construir el objeto, lo que ayuda a darle más claridad y precisión a la instanciación y garantizar que quede bien construido. Por ejemplo, queremos modelar la presentación de un recital, que define

- en qué lugar se realizará (puede cambiar)
- qué músicos tocarán
- el valor de la entrada general
- y cuántas entradas se vendieron hasta el momento

```
class Presentacion {
	var property lugar
	const property musicos
	var property valorEntrada
	var property entradasVendidas
	...
}
```

La ocasión para darle un valor coherente a cada variable es en el momento de la instanciación, y lo hacemos de esta manera:

```
const seruEnLuna = new Presentacion(
	lugar = lunaPark,
	musicos = [seruGiran],
	valorEntrada = 2000,
	entradasVendidas = 180)
```

Estos parámetros revelan claramente cuáles son las referencias a inicializar. Además, se puede hacer en cualquier orden:

```
const seruEnLuna = new Presentacion(
	entradasVendidas = 180,
	lugar = lunaPark,
	musicos = [seruGiran],
	valorEntrada = 2000)
```

Por otra parte

- Dado que es una sintaxis un tanto verbosa el IDE ayuda proponiendo las referencias a inicializar.
- Buena parte de la responsabilidad de inicializar bien depende de quien instancia a la presentación.

Para aquellas referencias, ya sean variables o constantes, que tengan definido un valor inicial, es opcional enviar un valor por parámetro.

- En caso que se no se le envíe, mantiene en valor con que se inicializó.
- En caso que se envíe un valor por parámetro, se asigna este en vez del definido en la inicialización.

Si se define a la presentación de esta manera

```
class Presentacion {
	var property lugar
	const property musicos = []
	var property valorEntrada
	var property entradasVendidas = 0
}
```

Y si se crea el objeto seruEnLuna como se lo hizo anteriormente, con todos los parámetros, el objeto queda instanciado de igual manera, pero ahora se puede instanciar sin necesidad de pasar por parámetros todos los valores.

```
const vamosAVerQuienEnLuna = new Presentacion(
	lugar = lunaPark,
	valorEntrada = 2000)
```

Queda instanciada una presentación con todos los datos consistentes, donde aún no se han vendido entradas y la lista de músicos está vacía. Todos los mensajes deberían funcionar correctamente.

En el caso particular en que todas las referencias están inicializadas, se puede hacer new sin parámetros.

La instanciación del objeto se concibe como un solo proceso, que auna la inicializacion de variables y su seteo mediante los parámetros. De esta manera, en el caso de una referencia constante que tiene un valor inicial y a la vez recibe un valor por parámetro, no se considera este seteo una modificación de la referencia, sino que ambas asignaciones se asumen como la asignación inicial.

En general, el formato para utilizar los parámetros nombrados es:

```
new ClaseAInstanciar(
      referencia1 = valor1,
      referencia2 = valor2,
      ...,
      referenciaN = valorN)
```

Las referencias no necesariamente deben ser propiedades (ni es necesario generarles *setters*), sí deben estar definidas como atributos válidos y pueden tener o no valores por defecto.

La buena noticia es que el IDE de Wollok nos alerta cuando hay variables que no se inicializan ni se pasa el valor por parámetro.

4 Resumen

A lo largo de este capítulo hemos visto como instanciar un objeto e inicializar sus referencias para permitir un correcto uso del objeto. Las alternativas para inicializar son:

- En la declaración de las referencias de la clase se le puede asignar un valor inicial, que actúa como valor por defecto.
- Se le puede dar un valor a cada referencia haciendo una instanciación con parámetros con el new. Se especifica cada referencia con su correspondiente valor.

Las referencias que no se inicializan por parámetro, asumen por defecto su valor asignado en la declaración. Si una referencia fue inicializada en su declaración y luego recibe por parámetro un nuevo valor, queda con este último. Esta combinación de opciones permite flexibilidad, mantiene claridad en el código y garantiza que el objeto quede bien construído.