

Clases y Objetos

Clases

- La implementación de una clase Java debe ir en un fichero en formato texto con la extensión *.java y nombre idéntico a la clase implementada.
- La declaración de una clase Java se realiza
- mediante la keyword: class seguida de su nombre.
- La keyword siempre va precedida por un modificador de acceso: public, protected, private o default (nada).
- La implementación de la clase irá contenida en un bloque { } justo después de la declaración.
- Declaración de una clase:

[crayon-6a9d57824721e257829436/]

- Ejemplo:

[crayon-6a9d578247227790076255/]

Atributos y métodos

- La implementación de una clase consiste en una serie de:
- Declaración de un atributo:
- Ejemplo:

[crayon-6a9d578247229659871158/]

- Declaración de un método:

[crayon-6a9d57824722c923177027/]

- La implementación del método irá contenida en un bloque { } justo después de la declaración.
- Ejemplo:

[crayon-6a9d57824722f123360851/]

- Se permite definir un número indefinido de parámetros del mismo tipo mediante: ...
- Lo que recibimos es un array del tipo definido.
- Ejemplo:

[crayon-6a9d578247231048055333/]

- Hay que tener en cuenta que podemos recibir, cero, uno o varios valores en dicho parámetro.
- Tiene algunas restricciones:
- Ejemplo:

[crayon-6a9d578247234769793536/]

Constructores

- Existe un tipo de método especial en Java llamado constructor.
- Sirve para la construcción (instanciación) de objetos (instancias) a partir de esa clase.
- En su implementación se suele dar valores a los atributos para ese objeto.
- Su declaración es idéntica a la de los métodos convencionales con dos salvedades:
- Declaración de un constructor:

[crayon-6a9d578247236489937448/]

- Ejemplo:

[crayon-6a9d578247238073255296/]

- Si nuestra clase no tiene constructores, el compilador añade por defecto uno sin parámetros.

Sobrecarga de métodos

- Se dice que un método está sobrecargado cuando existen dos métodos con el mismo nombre y tipo de retorno pero con parámetros distintos.
- De esta manera podemos tener en una clase varios constructores.
- Ejemplo:

[crayon-6a9d57824723b429866481/]

- Se permite la sobrecarga de métodos cambiando también el tipo de retorno, pero siempre que:
- El método que se está sobrecargando sea de una clase padre (de la que heredamos directa o indirectamente).
- El nuevo tipo de retorno sea hijo del tipo de retorno del método original (es decir, que herede de él directa o indirectamente).
- Por tanto, no es válido para tipos primitivos.

Convenciones en Java

- El nombre de las clases comenzará con mayúsculas:
- El nombre de los atributos comenzará con minúsculas:
- El nombre de los métodos comenzará con minúsculas (a excepción de los constructores):

Objetos

- Los objetos en Java no son mas que variables de tipo complejo, frente a las de tipo primitivo.
- El tipo de un objeto es la clase de la que se ha instanciado.
- La declaración de un objeto es idéntica a la declaración de una variable de tipo primitivo:
- El valor por defecto de un objeto sin inicializar es: null
- La inicialización de un objeto si que es algo distinta a la inicialización de las variables de tipo primitivo:
- Es decir:

Variables primitivas vs. complejas

- Una variable de tipo primitivo contiene el dato directamente:
- Una variable de tipo complejo contiene una referencia (puntero) a la zona de memoria donde está el objeto:
- Clases y Objetos

Manejo de objetos

- El trabajo con un objeto consiste en acceder:
- En ambos casos utilizaremos el operador . (punto).
- Acceso a un atributo:
- Acceso a un método (lo que en Orientación a Objetos se denominaba mensaje):
- La posibilidad de acceso a un atributo o a un método de un objeto dependerá del modificador de acceso que exista en su definición.

- Las llamadas a métodos se pueden encadenar:
- Equivaldría a:

El método main

- Existe un método especial en Java llamado main:
- Es el método donde comienza la ejecución de un programa Java.
- Las clases representaban entidades que participaban en la resolución de un problema. ¿En qué entidad tiene sentido incluir el método main?
- En ninguna. Por eso crearemos siempre una clase a parte, que solo tenga el método main.

Destruyores

- Los destructores son unos métodos encargados de eliminar los objetos de memoria.
- En Java no existe este tipo de métodos.
- En Java lo que existe es un proceso que se ejecuta en la JVM a la vez que nuestra aplicación y que se encarga de buscar todos aquellos objetos en memoria no utilizados y limpiarlos.
- Este proceso se llama Garbage Collector.

Garbage Collector

- ¿Cómo sabe el Garbage Collector que un objeto ya no está siendo utilizado por la aplicación y que por tanto puede ser eliminado?
- Porque no está referenciado por ninguna variable.
- Existen tres motivos por los que una variable deja de referenciar a un objeto: