

Reflexió en Java

Concepter bàsics

Metaobjectes

Serveixen per representar l'estat intern del programa, l'auto-representació del programa. Les instàncies de `Class` i `Method` serien metaobjectes. Els metaobjectes ens permeten accedir i canviar l'estructura d'un programa, el seu comportament o les seves dades p.ex. invocació dinàmica o interceptar crides a mètodes.

Objectes de base (base-level objects)

Objectes “normals” que el nostre programa fa servir per fer el que volem. Les parts **no** reflexives d'un programa formen el *programa base*.

Class

Tots els objectes Java és instància d'una classe que ve donada pel mètode `getClass()`. Podem trobar la classe de qualsevol objecte. Els *class literals* serveixen per especificar estàticament un objecte instància de `Class` (`Persona.class`). A la classe `Class` podem trobar els mètodes:

`Method getMethod(String name, Class[] parameterTypes)`: Retorna el `Method` public (declarat a la classe o heretat d'una superclasse o d'una interfície) de la classe on s'invoca, amb nom `name` i els tipus dels paràmetres corresponents a `parameterTypes`.

`Method[] getMethods()`: Retorna tots els mètodes de la classe.

`Method getDeclaredMethod(String name, Class[] parameterTypes)`: Retorna el `Method` public declarat a la classe de la classe on s'invoca, amb nom `name` i els tipus dels paràmetres corresponents a `parameterTypes`.

`Method[] getDeclaredMethods()`: Retorna tots els mètodes declarats a la classe.

`String getName()`: Retorna el nom de la classe.

`Class getComponentType()`: Si és una array (array amb `[]`, no `Array` o `ArrayList`), ens dona la classe del tipus que representa.

`boolean isArray()`: True si la classe és un `Array`.

`boolean isInterface()`: True si la classe representa una interfície.

`boolean isPrimitive()`: True si la classe és primitiva o `void`.

`Class[] getInterface()`: Llista d'interfícies que implementa.

`Class getSuperclass()`: Superclasse.

`boolean isAssignableFrom(Class cls)`: True si la classe és la mateixa, superclasse o interfície

`boolean isInstance(Object obj)`: True si l'objecte `obj` és instància de la classe.

`Field getField(String name)`: Retorna el `Field` public (declarat a la classe o heretat d'una superclasse) de la classe on s'invoca, amb nom `name`.

`Field[] getFields()`: Llista d'atributs de la classe (heretats també).

`Field getDeclaredField(String name)`: Retorna el `Field` public declarat a la classe de la classe on s'invoca, amb nom `name`.

`Field[] getDeclaredFields()`: Llista d'atributs només de la classe.

`Constructor getConstructor(Class[] parameterTypes)`: Retorna el `Constructor` (declarat a la classe o heretat d'una superclasse o d'una interficia) de la classe on s'invoca, amb els tipus dels paràmetres corresponents a `parameterTypes`.

`Constructor[] getConstructors()`: Retorna tots els constructors de la classe.

`Constructor getDeclaredConstructor(String name, Class[] parameterTypes)`: Retorna el `Constructor` declarat a la classe de la classe on s'invoca, amb els tipus dels paràmetres corresponents a `parameterTypes`.

`Constructor[] getDeclaredConstructors()`: Retorna tots els constructors declarats a la classe.

Method

`Class getDeclaringClass()`: Retorna la classe on es declara el mètode.

`Class[] getExceptionTypes()`: Dona la llista de classes de les excepcions que pot activar el mètode.

`int getModifiers()`: Dona els modificadors (`public`, `static`, `protected`, ...) codificats en un enter (*WTF Java?*).

`String getName()`: Retorna el nom del mètode.

`Class[] getParameterTypes()`: Retorna una llista de les classes dels paràmetres que se li passen.

`Class getReturnType()`: Retorna la classe del valor que retorna.

`Object invoke(Object obj, Object args[])`: Al aplicar això a un mètode **M**, executa **M** a **obj** amb els paràmetres **args** i dona el valor que retorna **M**.

Field

`Class getType()`: Retorna la classe de l'atribut.

`Class getDeclaringClass()`: Retorna la classe on està l'atribut.

`int getModifiers()`: Dona els modificadors (`public`, `static`, `protected`, ...) codificats en un enter (*WTF Java?*).

`String getName()`: Retorna el nom de l'atribut.

`Object get(Object obj)`: Retorna el valor de l'atribut en **obj**.

`boolean getBoolean(Object obj)`: Retorna el valor booleà de l'atribut en **obj**. També hi ha `getInt`, `getChar`, etc.

`void set(Object obj, Object value)`: Dona valor **value** a l'atribut en **obj**.

`void set(Object obj, boolean value)`: Dona valor **value** a l'atribut en **obj**. També hi ha `set(..., int)`, `set(..., char)`, etc.

Modifier

`static boolean isPublic(int mod)`: True si el modificador és/conté `public`.

`static boolean isStatic(int mod)`: True si el modificador és/conté `static`.

... Una per cada modificador (`public`, `static`, `native`, `volatile`, `protected`, `abstract`, `synchronized`, `strictfp`, `private`, `final`, `transient`).

AccessibleObject

Superclasse de Method i Field.

`void setAccessible(boolean flag)`: Fa accessible o no (`true` o `false`) al mètode.

`boolean isAccessible()`: True si és accessible.

`static void setAccessible(AccessibleObject[] array, boolean flag)`: Fa accessibles o no a la llista d'objectes.

Array

`Object newInstance(Class componentType, int lenght)`: Crea un nova instància del array de tipus `componentType` amb mida `length`.

`Object newInstance(Class componentType, int[] dimensions)`: Crea un nova instància del array de tipus `componentType` amb dimensions `dimensions` (2D, 3D, ...).

`int getLength(Object array)`: Retorna la mida de l'array.

`Object get(Object array, int index)`: Retorna l'element de l'array amb `index`. `array[index]`.

`boolean getBoolean(Object array, int index)`: Retorna l'element de l'array booleà amb `index`. També hi ha `getInt`, `getChar`, etc.

`void set(Object array, int index, Object value)`: Dona valor `value` al element `index` d'array.

`void setBoolean(Object array, int, index, boolean value)`: Dona valor `value` al element `index` d'array. També hi ha `set(..., ..., int)`, `set(..., ..., char)`, etc.

Constructor

`Class getDeclaringClass()`: Retorna la classe on es declara el constructor.

`Class[] getExceptionTypes()`: Dona la llista de classes de les excepcions que pot activar el constructor.

`int getModifiers()`: Dona els modificadors (`public`, `static`, `protected`, ...) codificats en un enter (*WTF Java?*).

`String getName()`: Retorna el nom del constructor.

`Class[] getParameterTypes()`: Retorna una llista de les classes dels paràmetres que se li passen.

`Object newInstance(Object[] initArgs)`: Crea un nova instància del l'objecte amb els arguments donats.