

Re: Does one "call a method" when polymorphism is used?

Re: Does one "call a method" when polymorphism is used?

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2008-01/msg00564.html>

- *From:* dagon@xxxxxxxxxx (Mark Rafn)
 - *Date:* Fri, 4 Jan 2008 14:45:30 -0800
-

Stefan Ram <ram@xxxxxxxxxxxxxxxxxxxxxx> wrote:

I wonder what the correct wording for the following situation is:
void print(final java.lang.Object object)
{ java.lang.System.out.println(object.toString()); }
Without further thinking, one might say "he method
"toString()" is being called here."

Yup, that's what I call it, and what everyone I talk to calls it. Sometimes, we say "invoke" rather than "call", but the object (of the sentence) is still "the method toString on the Object referenced by object".

However, according to the JLS, a method is a specific member of a specific class.
"toString" above does not denote such an entity, because for each invocation the method activated will be determined only at run-time.

Different context. Defining a method and invoking a method are not the same, and "method" can mean different things in different situations.

To call this to call a method might be misleading, because it hides the indirection via late-binding and gives the wrong

Yup. Welcome to English. Almost no simple description is exactly correct. And a lot of these ambiguities are far too ingrained to change, combined with not actually harmful very often.

When you say "invoking a method starts with the VM determining the correct method to invoke", it's obvious with a little thought that the two uses of "method" in the sentence refer to different things.

So, is there a more appropriate and precise wording?

Re: Does one "call a method" when polymorphism is used?

Re: Does one "call a method" when polymorphism is used?

Not in common use. Where precision is needed, I'd recommend "declared method" and "dispatched method". This is close to the distinction between `getMethods()` and `getDeclaredMethods()` in `java.lang.Class`. Most of the time, though, you can just say "method" and the listener will have to know what you mean.

The first two ideas that come to my mind would be to say »to call a signature« or »to call an identifier«.

identifier is wrong – there's only one type you can call, and it's a method. signature isn't bad, but I doubt it'll catch on.

while there might be multiple possible signatures depending on minor details for an identifier (like in »print("a")« and »print(2)«).

Those details aren't minor – they can cause different code to be run as much as any other signature change.

—
Mark Rafn dagon@xxxxxxxxx <<http://www.dagon.net/>>

.