

[Home](#)[Newsletter](#)[Java Specialist Club](#)[Java Training](#)[Conference Venue Hire](#)[Java Resources](#)[Contact](#)

## The Java Specialists' Newsletter

➤ Issue 087 ➤ 2004-04-09 ➤ Category: **Language** ➤ Java version:

0

✉ [Subscribe Free](#) [RSS Feed](#) [Print](#)

**sun.reflect.Reflection**

by Dr. Heinz M. Kabutz

Welcome to the 87th edition of **The Java(tm) Specialists' Newsletter**. We have a new translation into Greek, thanks to Panos Konstantinidis. The English have an expression: "It's Greek to me.", which means that they could not understand what was being said. The Greeks have a similar saying: "It's Chinese to me." When I was in Greece, my wife's cousin explained it like this: "Chinese is the most difficult language, because of these two sayings." When I was in China last year, I asked whether they had a similar expression. Imagine if the Chinese said: "That's English to me!" then we would have a loop in the logic.

This year we decided to not have our traditional "April Fools" Java newsletter. However, I thought that Sun Microsystems and Microsoft had gotten the date wrong, when they announced on the 2nd of April that MS had opened up their petty cash box and found some spare \$2,000,000,000 that they thought should go to a good cause. Time will tell what the implications are for us Java developers...

I would like to thank Matthew Schmidt and Rick Ross from **Java Lobby** for the **nice things they wrote about our newsletter**.

### Upcoming Java Master Courses:

Duesseldorf, Germany, Aug 22  
Chania, Crete, Sep 6  
Cape Town, South Africa, Sep 12

**In-house courses** if these dates or locations do not suit you. Note that the course in Crete may also be attended remotely via webinar.

### sun.reflect.Reflection

A few weeks ago, I was chatting to my friend Niko Brummer, who is an expert in speaker verification. Niko loves watersports and frequently applies his theoretical knowledge to his sports, much to my amusement. Niko is always available for a chat about his latest discoveries in Java (and in the ocean).

When we were chatting, Niko mentioned a class called "Reflection" that I had not heard of before. This class comes with Sun's JVM and sits in the sun.reflect.\* package. The most useful method in this class is: **Class** **getCallerClass(int i)**. This method tells you which classes are in our call stack.

Let's look at an example of some classes for making soup. We want the Potato to have a reference to the Soup in which it swims, and we want the Soup to know which Potatoes it contains. We have a one-to-many relationship and we want this to be maintained correctly. Please remember that this solution is just one of many, and is simply an example of how you could use this class.

```
import java.util.*;
public class Soup {
    private final List<Potato> potatoes = new ArrayList();

    public void add(Potato p) {
        potatoes.add(p);
        p.setSoup(this);
    }

    public String toString() {
        return "Soup {potatoes=" + potatoes + "}";
    }
}

import sun.reflect.Reflection;

public class Potato {
    private final int id;
    private Soup soup;

    public Potato(int id) {
        this.id = id;
    }

    public void setSoup(Soup soup) {
```

### Newsletter Links

- [Book Review](#)
- [Concurrency](#)
- [Exceptions](#)
- [GUI](#)
- [Inspirational](#)
- [Language](#)
- [Performance](#)
- [Software Engineering](#)
- [Tips and Tricks](#)

**Java Courses**

- [Java Master](#)
- [Java Foundation](#)
- [Java 5 Tiger](#)
- [Design Patterns](#)

➤ [find out more](#)

### What is the Java Specialists Club?

### Venue for Hire

Looking for a super conference room for your next company event?



Crete is your perfect destination.

[Click here for more details](#)

```
    this.soup = soup;
    if (Reflection.getCallerClass(2) != Soup.class) {
        soup.add(this);
    }
}

public Soup getSoup() {
    return soup;
}

public String toString() {
    return "Potato " + id;
}
}
```

The interesting method is `Potato.setSoup()`. It checks whether the calling class is `Soup`. The method `getCallerClass()` takes the stack depth as a parameter, in this case it would be 2.

We can try this out by making a `soup`, adding some potatoes, and then associating the `soup` with a `potato` that did not have an association.

```
public class SoupTest {
    public static void main(String[] args) {
        Soup soup = new Soup();
        soup.add(new Potato(1));
        soup.add(new Potato(2));
        soup.add(new Potato(3));
        Potato p4 = new Potato(4);
        soup.add(p4);
        p4.setSoup(soup); // redundant code
        Potato p5 = new Potato(5);
        p5.setSoup(soup);
        System.out.println("soup = " + soup);
    }
}
```

When I run this with `java -showversion SoupTest`:

```
java version "1.4.2_04"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_04-b05)
Java HotSpot(TM) Client VM (build 1.4.2_04-b05, mixed mode)
```

```
soup = Soup {potatos=[Potato 1, Potato 2, Potato 3, Potato 4, Potato 4, Potato 5]}
```

A handy little class supplied in the Sun JDK.

Kind regards

Heinz

[➤ Language Articles](#) [➤ Related Java Course](#) [➤ Discuss at The Java Specialist Club](#)