Compatibility Criteria for Java Packages

Yannick Welsch
welsch@cs.uni-kl.de
Software Technology Group
University of Kaiserslautern

14.04.2009

In this talk, I analyze the interface evolution of Java packages regarding compatibility. I want to determine the criteria for a Java package implementation to be substitutable for another one in all possible contexts, leading to a program satisfying the typing rules and context conditions.

Although the Java Language Specification defines properties for binary compatibility, this notion is not as strong as source compatibility. In fact, source compatibility implies binary compatibility, but not vice versa. For example, introducing a new field with the same name as an existing field, in a subclass of the class containing the existing field declaration, does not break binary compatibility with preexisting binaries. However, at the source code level, this may lead to source incompatibility (type error). A new declaration is added, changing the meaning of a name in an unchanged part of the source code, while the preexisting binary for that unchanged part of the source code retains the fully-qualified, previous meaning of the name.

The two following topics can be addressed by giving criteria for source compatibility. Consider a setting where a library implementer develops a new version of the library. By respecting the compatibility criteria, he can ensure that the new version of his library will still compile with existing client code. Another far more complex topic is the following one. To define a notion of behavioral equivalence of two package implementations, a notion of compatibility is needed as only compatible package implementations can lead to identical behavior.

In an attempt to specify the criteria for package compatibility, I have formalized a Java subset in the spirit of Classic Java, enhanced by packages and access modifiers. The compatibility criteria I give are directly derived from the typing rules and context conditions.

As byproduct of my research on compatibility criteria, I was able to find a bug in the Eclipse Java compiler regarding overriding of methods across package boundaries.