



## GS04: Tools and Environments

### Lab Session 3: Building a Static Analyzer

The Hibernate Object Relational Mapping (ORM) is able to persist properties of Java Beans in relational databases. A JavaBean property is implemented as a private field and a pair of public getter and setter methods. For the sake of simplicity we assume that all fields in the JavaBean implementation are properties (i.e. there are no transient data structures in the bean). For Hibernate to be able to persist these properties, the Java Beans need to follow certain coding conventions.

The aim of this lab session is to build a static analyzer on top of the JDT AST API that checks that all JavaBeans contained in a persistence package do in fact meet the following conventions:

- The only types declared in the source code are classes (interfaces do not need persisting).
- The field that implements the property is private.
- The two set and get methods exist
- The two set and get methods are declared as public.
- The name of the set and get methods are the same of the name of the field preceded by 'get' and 'set' respectively. If the property is of type boolean, the getter has the prefix 'is' rather than 'get'
- The get method has an empty parameter list and a return type that is the same as the type of the corresponding field.
- The set method has a void return type and exactly one parameter that is the same type as the field type.

#### Setup

For this lab session, we will need a number of abstract syntax trees as test data. To obtain them, create an Eclipse project called *gz04CW* using the source at <http://www.cs.ucl.ac.uk/staff/w.emmerich/lectures/GS04-0708>. Make sure that there are no compile errors.

#### Static Analyzer Implementation

Implement the above static analyses. You may find it easiest to first build up symbol tables of all the fields and methods that are declared in a class. Use a HashMap so that you can hash it on the field and method names. It is probably simplest to fill that hash map using the VisitorPattern.

#### Testing

Your analyzer should create some output similar to the one given to the right.

