Adding Null-Value Constraints to the Java Programming Language

A presentation of recent work
by Keith Lea

Problems

- Many bugs in applications are due to unexpected data
  - Buffer overflows, runtime exceptions, seg. Faults
- In Java, null (empty) values may cause hard-to-diagnose problems
  - Values may be stored away for use later or in another part of the program where nulls are not expected
- Java provides no way to formally specify nullness constraints
  - Instead, requires repeated manual null checks
  - Nullness is not enforced at compile time – null value checking relies on runtime exceptions to find errors

Problems

- Null values are either allowed or disallowed as arguments to API methods, but this fact is often not documented

My Project

- Allows easy specification of nullness constraints declaratively in Java code
- Provides three levels of protection against null-related problems:
  - Edit time: potential nullness problems highlighted in Java editor
  - Compile time: potential problems are shown as warnings or errors during compilation
  - Run time: problems are caught early by rigorous runtime null-checks

Design

- Nullness specification serves two purposes
- Contract enforcement
  - Methods may specify nullness of parameters and of return types
- Programmer assumption enforcement
  - Programmer can specify nullness of local variables

Design

- Contract enforcement for API designers
  - API users are not allowed to pass possibly null values for non-null parameters
  - If non-null parameters are passed null arguments, exception is thrown at runtime
- Contract enforcement for API users
  - Method with non-null return type attempts to return null, an exception is thrown at runtime, and the method never returns
  - If method with non-null return type returns null, the caller throws an exception

Design

- Assumption enforcement
  - Local variable declared as non-null is guaranteed never to hold null value
  - Programmer is only allowed to place guaranteed non-null values in variable
  - If non-null variable is assigned null value at runtime, exception is thrown

Implementation

- Based on Soot Java Optimization Framework
  - Powerful code analysis API
- Implemented as plugin for IntelliJ IDEA, a Java editor
  - Provides easy manipulation of Java code tree
  - Allows edit-time nullness checks & highlighting

Determining Nullness

- Soot knows what is possibly null and what is not, when analyzing Java code
- My project adds to Soot's knowledge of what is *not* null
  - Modifies the Java code, based on nullness declarations, before Soot analyzes it
- Soot uses this information to determine which values may be null

Determining Nullness

- My project uses Soot's analysis results
  - Provides feedback to programmer
  - Inserts runtime checks before compilation to enforce specified behavior

Step 1: Modifying Code Before Analysis

- Add checks to top of method for any non-null parameters
- Add checks for return value of method invocation with non-null return value
- Add checks for assignment to non-null variable

Step 2: Analysing with Soot

- Pass modified code to Soot, execute nullness dataflow analysis
  - Soot records a list of all values used in the code which may contain a null value

Step 3: Look For Illegal Null Values

- Check for possibly-null values found by Soot, which are
  - assigned to non-null variables
  - passed to methods as values for non-null parameters
  - being returned from method declared with non-null return value
- For each possibly-null value in non-null context,
  - show error in Java editor, if editing
  - record compilation error, if compiling
  - insert runtime check for nullness, if compiling

Potential Problems

- Compatibility
  - There exists lots of code in the world which specifies and enforces nullness in other ways (Javadoc, User's Guides)
  - This may make my project tedious to use because of values which the programmer knows are not null, but my project does not know about
- Lock-in
  - My project requires IntelliJ IDEA for development of applications which take advantage of compile-time and edit-time nullness checks

Result

- My project
  - provides a comprehensive solution to the problem of nullness requirements
  - makes it easy to write well-defined code which is safer from nullness problems
  - makes it easier to learn API's and to diagnose problems with them
  - makes it easier to keep API's well-documented regarding nullness constraints