The Trait Extractor: A Tool for Generating Traits in Java
Emerson Murphy-Hill and Andrew P. Black
Portland State University

Purpose
Create a tool to assist trait-building in Java within the Eclipse IDE.

Procedure
- Build tool as a plugin to the Eclipse IDE
- Leverage built-in Java source manipulation library
- Integrate with pre-existing trait plugin
- Model functionality after Smalltalk version of tool

Objective
Create a tool that facilitates turning a class into a trait.

What’s a Trait?
- A collection of pure methods
- Like a class, but without state or a superclass
- Reusable building blocks for “use” in classes
- Alternative solutions: duplication, multiple inheritance, and mixins

A Need for the Tool
Traits are usually created from pre-existing code in classes, rather than from scratch.

Findings
Traits in Java must support features not present in the Smalltalk implementation, including constructors, initializers, and inner classes.

Future Directions
- Use tool in context of major software project
- Allow complete refactoring: integrate trait into original class
- Allow explicit super calls in traits, as in super.method()

What the User Sees

What The Tool Does

The Alternative
Without this tool, users would have to copy and modify the class manually, at the cost of possible compilation errors and unanticipated runtime bugs.

Performance
The entire operation generally takes up to 10 seconds on a typical development machine, but performance is not the primary concern.

The Trait Extractor: A Tool for Generating Traits in Java
Emerson Murphy-Hill and Andrew P. Black
Portland State University