Problem: A Simple Java Debugger

Specification Overview

A simple, non-magical Java debugger for teaching. The debugger should be started from the command line, have an intuitive graphical user interface and support the traditional operations of a debugger (start, step, breakpoints...). Ideally it should also graphically display the state of the program.

Minimum Requirements

- Must run in the Labs
- Start from the command line
- Supporting all the command line switches (-ea etc)
- Taking command line arguments (from the gui)
- Users must be able to see stdin, stdout, stderr
- Users must be able to see the source code
- Users must be able to inspect the current state of the program somehow
- Run/Pause/Step into/Step Over
- Multiple Files
- Breakpoints
- Supports Java static methods, Objects, Arrays, Control flows, Generics, Enumerations
- Minimal Documentation (--help, README, small user guide, etc)

Extensions

0) Lecture Mode
1) Display of state is graphical
2) Pictures of previous state and current state for comparison
2.1) Conditional breakpoints
2.2) Watch points
3) Multi-platform (OSX, Linux, Windows)
4) Maximal Documentation
5) Support all of Java (Threads etc)

Stakeholders

Project Owner - Tristan
First Year Students (who have used an imperative language before)
First Year Students (who have not used an imperative language before)
Other Lab Helpers
Who deploys software - CSG and Teaching Fellows

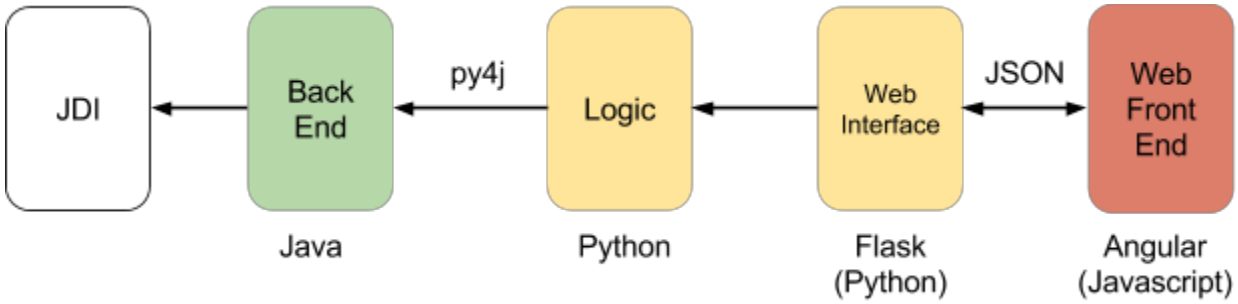
Solution: Deeva

Timeline

Week 1: Assigned Project
Week 2: Initial Meeting and Creating a Plan
Week 3: End-to-end Tech Demo
Week 4: MVP
Week 5: Add feature: Breakpoints
Week 6: Add feature: Graphical Display State
Week 7: Minimum Spec Completed + Demo Deeva to First Years
Week 8: Iterate on Feedback
Week 9: Polish / Sliptime / Extensions
Week 10: Revision (no project work)
Week 11: Exams (no project work)
January: Final Report & Presentation

Note: lines highlighted in red indicate no project work being done in those weeks.

Architecture



To leverage our existing knowledge (and to avoid the horrible bitrot that affects all Java GUI software) we are building our application like a web application with a front end built with HTML5/Javascript and using Python to glue it to our custom wrapper over the Java Debugger Interface.

Process

Source control: Github (https://github.com/chromy/Deeva)
Story tracking: Trello (https://trello.com/b/YC46o0Dc/deeva)
Questions: Facebook group
Evaluation cycle: Every Wednesday at 4:00 demo current state to Tristan and get feedback
Feature cycle: Every Monday estimate stories