CALIFORNIA STATE SCIENCE FAIR
2015 PROJECT SUMMARY

Name(s)
Abe N. Jellinek

Project Number
S1410

Project Title
Lingua: A Programming Language for the Uninitiated

Abstract

Objectives/Goals
The purpose of this computer science engineering project was to create a programming language usable by anyone, from a total beginner to an advanced programmer. Therefore, it was designed to be approachable and simple, yet encompassing of a wide variety of programming paradigms and concepts. Its syntax was based on languages such as Java and C, but simplified and refined in many ways in order to be approachable by new students as well as experienced programmers. I called it Lingua, after the Latin word for language.

Methods/Materials
Materials:
- Java 8 - a cross-platform programming language
- Google Guava - Google#s standard library of collections for Java
- JUnit 3 - an automated unit test framework

I researched parsing algorithms and settled on Pratt parsing, a variant of recursive descent parsing. It makes operator precedence quite easy, and that was a huge help.
Next, I developed a simple parser for mathematical expressions, which then gained support for variables, functions, data types, looping, and many other essential parts of modern programming languages.
Finally, I wrote a tree-walk interpreter for the language, because, while not the fastest method, it#s quick to prototype. It works by evaluating each top-level expression, which then evaluates each of its child expressions, and so on, until it reaches the #leaves# of the tree.

Results
The entire implementation came out to nearly 10,000 lines of code in 93 classes. Much of that is the lexer/parser (the part of the implementation that cuts text into tokens, and then joins tokens into expressions), which is relatively complex (not hard to understand, but large). The interpreter is surprisingly simple, and performance is much, much better than you would expect from a tree-walk interpreter. The Java VM is very smart with its optimization, and successive runs get significantly faster as it optimizes and JIT-compiles the code.

Conclusions/Discussion
I accomplished my goal in its entirety. The language is approachable by beginners, has a fairly large feature set, and is quite flexible. It is fast and complete enough that I actually believe that it could be used as a teaching tool, and all it needs to really succeed is a larger standard library and more documentation.

Summary Statement
A programming language with a focus on usability by beginners and a broad feature-set.

Help Received
I received very small amounts of help with my board from my family.