Subscript: A Functional Programming Language for Scripting in Data-Driven JVM Applications

Krzeminski, Elliott (School: Red Mountain High School)

Most large applications eventually require user-defined behavior to remain adaptable to different users. Common approaches to user-defined behavior are configurations (for simple behavior) and scripting (for more complex behavior). In Terra, a world generation platform for Minecraft, user-defined behavior is required to generate structures in the world. Configurations would become bulky and bloated and make implementing custom logic tedious, so a scripting language is needed. Existing scripting languages for the JVM are either lacking in features, deprecated or slow, so Subscript was created to meet the exact requirements of Terra. Subscript is a purely functional language, intentionally lacking imperative constructs such as variables (rather, values are used to ensure immutability) and iteration (recursion is used instead). Subscript is a compiled language, compiling to JVM bytecode with an optimizing compiler designed to eliminate dead code and precompute expressions whenever possible. This allows it to perform equally to compiled Java code, drastically exceeding the performance of interpreted languages. Subscript was tested iteratively as language features were developed, with test scripts being written for specific language features. Available for easy inclusion in any project as a Maven/Gradle dependency, Subscript met all requirements for Terra, and is also useful to any other JVM application which requires complex user-defined behavior. It is also potentially useful as a teaching tool; functional programming is generally not taught due to its complexity, Subscript is a simple purely functional programming language that could be used as a stepping stone.

Awards Won:
Second Award of $2,000
Oracle Academy: Award of $5,000 for outstanding project in the systems software category.
Association for Computing Machinery: Fourth Award of $500