Program #5 Review

Written by Alexander DuPree

Java Discussion

Program 5 was a continuation of program 4 and as such was written in Java. After working with Java for the past few weeks now I am starting to become a believer in Java over C++. C++ is a great language and gives the programmer absolute control over memory, algorithms and how your applications operate. However, this is also the pitfall of C++. Too much power in unskilled hands can result in a lot of undefined and unexpected behavior. Java takes away the low level details of memory management and pointers and a lot of other constructs that make the programmer powerful but also prone to error. After a few weeks adjusting to the differences of primitive types and reference types, as well as recursion in Java I think that Java is overall a much easier language to program in. This is not to say that Java is a better language, the best language is whatever language fits the requirements of your program.

Coming into program 5 I thought it would be impossible to implement a 2-3-4 tree, flex array, and a inheritance hierarchy all while learning a new language. But because Java takes away the low level details and shares similar syntax with C++, writing program 5 actually turned into an enjoyable learning experience. The error messages in Java are especially helpful, sharing the exact line number where the error occured as well as trace of the stack frames. This combined with the other features and huge community online supporting Java made learning Java a breeze and an enjoyable experience.

IDE Discussion

Java wasn't the only change required for program 5, we were also allowed the use of IDE's to implement our programs. VIM will always be my preferred development environment when working in a terminal, however, the power, accessibility, and quality of life improvements when working with an IDE made writing/testing the program much easier. I used the intelliJ IDEA for my development environment and after spending some time customizing and learning the features of the IDE I can say it will be hard to go back to the bare bones text editing of VIM. With the intelliJ IDE I could spin up a suite of unit tests within seconds, run those tests and debug them with a few clicks, and even perform a lengthy refactor through a guided contextual menu. The combined ability of all these features along with autocomplete, file navigation, syntax highlighting, and much more made writing code an enjoyable and simple process.