Introduction

Before I can give my impression of Java, I must first mention my background in programming languages. I first learned Java in a beginning course I took in 2010. This was not my first exposure though, to a high-level programming language; all students in my high school were required to learn Pascal. Throughout my years as an undergraduate biology major, the programming languages I studied were Matlab, Mathematica, and C.

After a few years of working in the biotechnology industry (mainly in the laboratory), I became interested in the bioinformatics field. However, most of the scientists within this field were programming in Perl, so I took it upon myself to learn this language. I was initially struck by how messy the code could get. However, I came to like it because of the speed at which programs could be written.

Positive reactions to Java

Having been writing in Perl for more than a decade, I have come to accept the fact that it can be a challenge to understand other people’s code. One of the first things that struck me about Java was how much more readable the code is, even when written by others. This is most likely due to Perl’s well-earned motto of “There’s more than one way to do it” (Schwartz, 1998), which can lead to messy code. Unlike Perl, Java is much more strongly typed, which I find to be mainly a good thing. I have found that programs in which strong typing is not enforced can be difficult to debug.

Negative reactions to Java

One of my negative reactions to Java was how much slower it takes to write a program, when compared to Perl. For example, to write the “Hello world” program, in Perl I could simply have one line:

print “Hello world\n”;

While in Java, this would be longer:

public class HelloWorldApp {

public static void main( String[] args) {

System.out.println(“Hello world”);

}

}

As I mentioned in my positive reactions section above, Java is much more strongly typed, which is mainly a good thing. However, this can slow down development time. For my work, I process a lot of text files (many of which are tab-delimited) “on the fly”. If I had an array element which contained the text “10 30 21”, I can easily split this up into 3 fields, and assign them to integer variables using Perl. In Java, I would have to first convert each on into an integer before assigning it to an integer variable.

Conclusion

Though one can write programs much faster in Perl than in Java, I have found that a good amount of time is spent on maintenance. When code is difficult to understand (as much of Perl code is), it adds even more challenges to the maintenance duties. As I become more skilled in Java, I will transition more to it at the workplace, and further away from Perl (except for rapid-prototyping tasks). Over the years, I have seen the bioinformatics field transitioning to one more based in computer science, so I must become more skilled in more formal languages such as Java, if I am to remain in this field.

Reference:

Schwartz, A. (1998) ‘Tutorial: Perl, A Psychologically Efficient Reformatting Language’, *Behavior Research Methods, Instruments, & Computers*, vol. 30, no. 4, pp. 605-609.