'Rusty' Iterators in Java

Andrew McDonald

February 14, 2024

Proposed Paper/Topic: I plan to base my project on the Rust standard library source code for an iterator [1]. This project will be a Java implementation of iterators written in a 'Rusty' way.

One of the Rust language's strongest aspects is its iterator implementation. It boasts on par performance with classical iteration while being able to be used more expressively. The basis for the iterator pattern is one of strict functional paradigms. The Rust implementation has a strong focus on performance so plenty of room has been made for non-functional paradigms to improve the experience and speed of using these iterators. Iterators have been proven and used extensively even at the lowest level of computing.

The idea behind this project was to port the Rust implementation of iterators over to a language that could use them desparately. I choose Java due to having a decent level of experience with the language and the knowledge that iterators would have made several past projects easier and clearer. I do not plan for a 1-1 implementation parity with the Rust alternative due to the massive scale and lacking feature set of Java. Instead, I plan on getting a similar front-end interaction with iterators as a proof of concept.

Expected Features

- Near feature parity with Rust's basic iterators.
- Simplistic implementation of Rust's double-ended iterators.
- Test suite proving the correct implementations.
- Properly packaged for use in other projects.
- *Tentative:* Example program showing a real-world use case.

References

[1] Iterator implementation in the Rust standard library.