Peer Review of Object-Oriented Principles in Strictly-Typed vs. Dynamically-Typed Languages

Reviewers

Andy Fan
Lincoln Schafer
Alex Orlowski

Comments

Strict and dynamically typed languages are both in major use which makes this an interesting comparison. Looking at the OOP principles will be a great way to better understand how these languages were developed originally. It could potentially be worth opening up the languages that will be included in the review. The outline mostly includes java and python, however the title suggests that more languages might be reviewed. Looking at many languages or just looking at Java and Python are both fine approaches, and being explicit about the approach will make the research more valuable. The trade off of those options would be depth of analysis versus breadth of analysis. Looking at more languages would also enable some historical looks at how older languages might implement OOP principles which could also be an interesting read. Finally, there are many developments around static type checking and compilation for dynamically typed languages, which is probably important practical information for readers.

References to Consider

1. The Closure Compiler from Google is an interesting approach to using compilation and type checking practices with javascript - https://developers.google.com/closure/compiler
2. Python also has static type checking which is another lens to look at python through - https://realpython.com/python-type-checking/
3. Here's a research paper reviewing some older dynalically typed languages against java, c and c++ https://page.mi.fu-berlin.de/prechelt/Biblio/jccpprt2_advances2003.pdf
4. In java 10 the var keyword was implemented allowing for implicit typing. Another interesting angle to look at this comparison through - https://developers.redhat.com/blog/2018/05/25/simplify-local-variable-type-definition-using-the-java-10-var-keyword/
5. Typescript is another language to add to this mix if you go with the multi-language approach - https://www.typescriptlang.org/docs/handbook/release-notes/overview.html