Object-Oriented Principles in Strictly-Typed vs. Dynamically-Typed Languages

CSCI 5448

Xinyu Jiang

Qiuyang Wang

Vladimir Zhdanov

For our graduate research project, we propose exploring the differences in object-oriented principles between strictly-typed and dynamically-typed languages. A strictly-typed language, such as Java or C++, is defined as a language where the type of each variable needs to be explicitly determined before compilation. A dynamically typed language, such as JavaScript or Python, is defined as a language where the type of each variable is not required to be specified, and each variable's type is associated with the runtime values of the program. We believe that it would be interesting to research how object-oriented principles differ between these two types of languages. We could analyze how polymorphism, inheritance, abstraction, and encapsulation are implemented within these languages, and any patterns or hindrances to each type of implementation.

We decided to choose this topic because all of our team members have had extensive experience with dynamically-typed languages including JavaScript and Python. We all appreciate the simplicity and ease of use of these languages. But, despite our experiences, we are not very aware of the object-oriented aspects of these languages. Thus, we would like to compare the pros and cons of using a dynamically-typed language over a strictly-typed language for object-oriented program design.