Elegant Objects: Developing a Programming Language for Large-Scale Applications

Keywords:***Elegant Objects,ϕ-calculus,EO,Eolang,compiler,ANTLR4,OSS,Java,Maven***

The EO programming language is an object-oriented language that is being developed as an R&D solution, the purpose of which is to show that industrial programming in the pure OOP paradigm is possible.
The language is based on the philosophy “Elegant Objects” and a fundamentally new formal model of ϕ-calculus, which defines basic operations on objects, positioned as necessary and sufficient to achieve object-oriented properties of the language.

Egor Bugaenko, the author of the “Elegant Objects” concept underlying the EO language, identifies the following problems of existing object-oriented programming languages:
1.Mutability of data generates side effects, which leads to unpredictable behavior of the program.
2.Using null references to denote the absence of a value.
3. Applying reflection breaks the encapsulation of classes, allowing the developer to use classes in unpredictable scenarios.
4. Inheritance allows you to get rid of visible duplication of code in similar classes, however, making any changes to parent classes only becomes more complicated because of the need for consistency of changes with all inherited classes.

The unique features of EO are listed below:

- *Lack of static entities (classes, methods).*
- *Rejection of the concept of classes.*
- *Rejection of inheritance as a composition operation.*
- *Immutability of objects and their fields.*
- *Avoiding null references.*
- *Rejection of global variables and procedures.*
- *Refusal of reflection.*
- *Refusal of explicit type conversion.*
- *Lack of primitive data types.*
- *Refusal from annotating code.*
- *No unchecked exceptions.*
- *Avoiding syntactic sugar, the DSL model (domain-specific language), which allows programmers to extend the language to suit their needs.*
- *Rejection of operators and control structures at the language level.*

This work aims to develop and contribute an EO-lang compiler that will be accessible at Github repository (OSS, MIT), which can build java classes from EO source code. Compiler

Maven plugin wrapper to start compiler at build phase to generate class files, which will be packaged by Maven later.
Contribution of such a solution will bring us to answer the question: Is it possible to rewrite Java Cloud web-module with EOlang.

The plan for the work is the following:

1. Translation of EO program code into ".java" files in the Java programming language.
2. Programming languages used in the implementation of the source code of the translator, objects of the standard library, as well as the test base (including languages used to supplement the existing code base) - Java, EO.
3. The parser (as well as the lexer) must be implemented (and augmented) as part of the ANTLR4 solution.
4. The preprocessing of the object structure after the parsing process must be done through XSLT transformations of XML documents.
5. The final solution (translator) should be delivered in an easy-to-use form: a Maven plugin in the Maven Central repository.

References:

1. An introduction to functional programming through lambda calculus, url=https://www.amazon.in/Introduction-Functional-Programming-Calculus-Mathematics-ebook/dp/B00CWR4USM, journal=Amazon.in: KindleStore, isbn=0486478831, author=Gregory Michaelson,year=2011, publisher=Dover Publications Inc.
2. Marten Abadi and Luca Cardelli. 2013. A Theory of Objects. Springer-Verlag New York.
3. Yegor Bugayenko. [n.d.]. Elegant Objects. https://www.elegantobjects.org/
4. J. Paquet and Serguei A. Mokhov. 2010. Comparative Studies of Programming Languages; Course Lecture Notes. ArXiv abs/1007.2123 (2010).
5. [n.d.]. Object-Oriented Programming with Java: AnIntroduction.y6s https://www.amazon.com/Object-Oriented-Programming-Java-David-Barnes/dp/0130869007
6. Robert V. Sebesta. 2015. Concepts of programming languages. https://www.amazon.com/Concepts-Programming-Languages-Robert-Sebesta/dp/013394302X