

AN APPROACH TO INTERNAL DOMAIN-SPECIFIC LANGUAGES IN JAVA

Domain-specific language or DSL is often defined as a language that targets specific problems within a certain domain without necessarily being applicable to other domains/ problems. The article discusses in particular how DSL can be used alongside Java programming and pros and cons of such a collaboration.

On the one hand, Java is itself a fairly verbose language and for a developer, it is not too hard to read/maintain it. However with the use of DSL one can make the code readable to the extent that even business users would have no problems following every step in the code. This can be achieved through e.g. Method chaining, where ever method returns the value "this". This allows for clear method names to be placed one after another and creating a sentence effect. The other method is called Static factory methods and import, this method uses objects/classes to simplify the way the code can be read which reduces the amount of side comments that need to be added and making the text overall more compact.

Both methods are great ways to help users/customers orient themselves with the code. However the problem arises when one wants to maintain the code on regular basis. This can get pretty troublesome if there are too many methods/classes to pay attention to so some API designers might not be happy with the method. Though depending on the type of program that needs to be developed and the type of maintenance/customer requirements, developers can choose to use DSL in Java as it is, to some extent, very useful.