Interpreter Pattern

An Interpreter Pattern says that **"to define a representation of grammar of a given language, along with an interpreter that uses this representation to interpret sentences in the language".**

Interpreter pattern provides a way to evaluate language grammar or expression. This type of pattern comes under behavioral pattern. This pattern involves implementing an expression interface which tells to interpret a particular context. This pattern is used in SQL parsing, symbol processing engine etc.

Basically the Interpreter pattern has limited area where it can be applied. We can discuss the Interpreter pattern only in terms of formal grammars but in this area there are better solutions that is why it is not frequently used.

This pattern can applied for parsing the expressions defined in simple grammars and sometimes in simple rule engines.

**SQL Parsing uses interpreter design pattern.**

Advantage of Interpreter Pattern

* It is easier to change and extend the grammar.
* Implementing the grammar is straightforward.

Usage of Interpreter pattern:

It is used:

* When the grammar of the language is not complicated.
* When the efficiency is not a priority.

Example of Interpreter Pattern

Let's understand the example of Interpreter Pattern by the above UML diagram.

UML for Interpreter Pattern:

