

Day: Wednesday

DP Assignment - 5

N.V.S.K. Kalyani

Date: 23-04-2020

17UNIA0584

INTERPRETER PATTERN

3rd B.tech CSE-C

1. How Interpreter pattern solves the problem by defining the representation of its grammar?

A. The Interpreter pattern discusses defining a domain language as a simple language grammar, representing domain rules as language sentences and interpreting these sentences to solve the problem. The pattern uses a class to represent each grammar rule. And since grammars are usually hierarchical in structure, an inheritance hierarchy of rule classes maps nicely.

2. What is the difference between terminal and non terminal expressions?

Terminal Expression

1. These are the leaf nodes of the tree.
2. They don't contain other expressions.
3. Terminal expressions in our postfix mathematical expression. Ex: operands 2, 1 & 5.
4. It implements the interpreter operation for terminal expression.

Non-Terminal Expression

1. These are non-leaf nodes of the tree.
2. They contain other expressions.
3. The + and * operations are non-terminal expressions.
4. Implement operations the interpreter operates for all non-terminal expressions.

3. Define Abstract Syntax tree?

A. An abstract syntax tree is a tree representation of abstract syntactic structure of source code written in the programming language. It declares an abstract interpreter operation that is common to all nodes in the abstract syntax tree.

4. Why complex grammars are hard to maintain?

A. Because the pattern uses classes to represent grammar rules, you can use inheritance to change or extend the grammar. Complex grammars are hard to maintain. The interpreter pattern defines at least one class for every rule in the grammar. Hence the grammars containing many rules can be hard to manage & maintain. Other design patterns can be applied to mitigate the problem.

5. How the related patterns support this Interpreter pattern?

A. Composite:- The abstract syntax tree is an instance of the composite pattern.

Flyweight:- It shows how to share terminal symbols within the abstract syntax tree.

Iterator:- The interpreter can use a iterator to traverse the structure.

Visitor:- It can be used to maintain the behaviour in each node in the abstract syntax tree in one class.