Interpreter Pattern

Interpreter pattern

- Intention
    - Etant donné un langage/une grammaire/une expression, définit une représentation de celui/celle-çi.
- Motivation
    - Dans des projets d’envergures, il arrive que certaines classes n'interagissent pas comme il faut entre elles. Ce patron permet de vérifier directement ces interactions et de comparer les résultats de celles-çi avec ceux attendus.

Interpreter pattern

Avantages :

+ Pratique pour la grammaire
+ Plus facile pour changer et étendre une grammaire
+ Implémentation de la grammaire directe

Inconvénient :

- Les grammaires complexes sont plus difficiles à maintenir
- Réduit l'efficacité

Interpreter pattern

```
public interface Expression {
   public boolean interpret(String context);
}
public class TerminalExpression implements Expression {

   private String data;

   public TerminalExpression(String data){
      this.data = data;
   }

   @Override
   public boolean interpret(String context) {

      if(context.contains(data)){
         return true;
      }
      return false;
   }
}
```

Interpreter pattern

```
public class InterpreterPatternDemo {

   //Rule: Robert and John are male
   public static Expression getMaleExpression(){
      Expression robert = new TerminalExpression("Robert");
      Expression john = new TerminalExpression("John");
      return new OrExpression(robert, john);
   }

public static void main(String[] args) {
   Expression isMale = getMaleExpression();
   System.out.println("John is male? " + isMale.interpret("John"));
 }
```

Diagramme UML