

# بررسی الگوهای طراحی - Interpreter

توسط : محمد حسین زارع

@zare88r: twitter

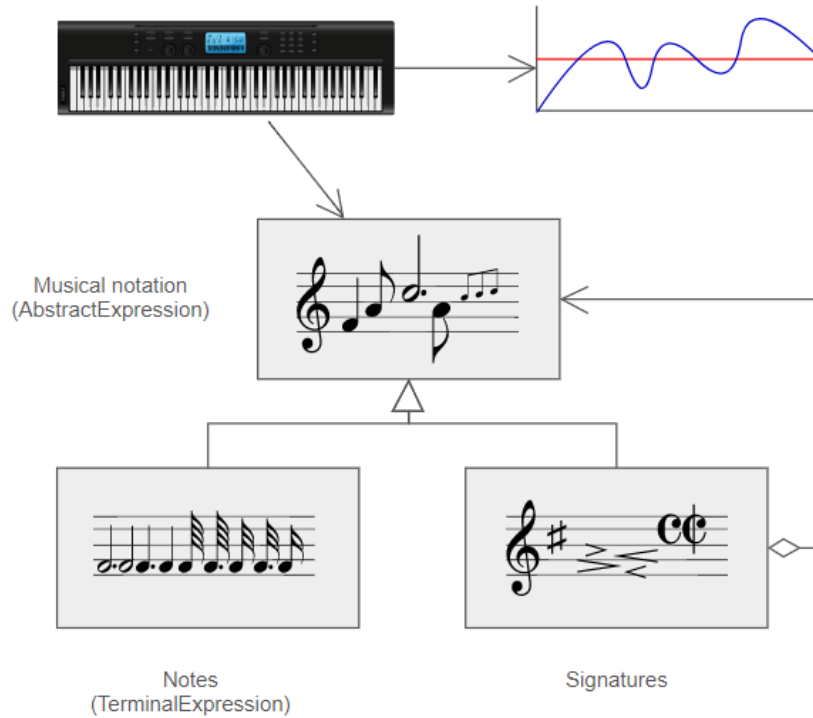
telegram

Join Us : @JavaLandCH

# مفاهیم

- پیاده سازی گرامر
- تفسیر یک عبارت
- نگاشت یک دامنه
- **AST (Abstract Syntax Tree)**
- مثال :
- **java.util.Pattern**
- **java.text.Format**

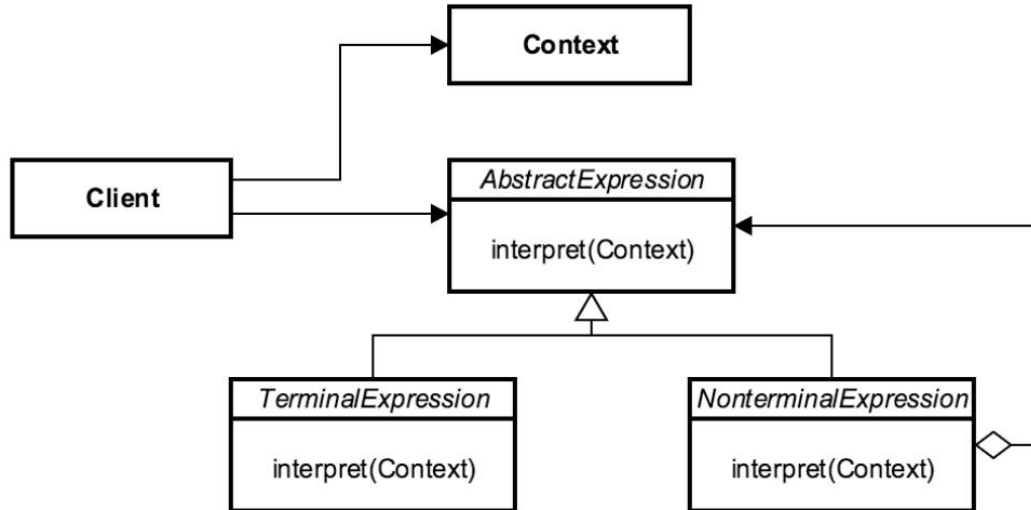
# مفاهيم



# طراحی

- **AbstractExpression**
- **TerminalExpression**
- **NonterminalExpression**
- **Context**
- **Client**

## طراحی



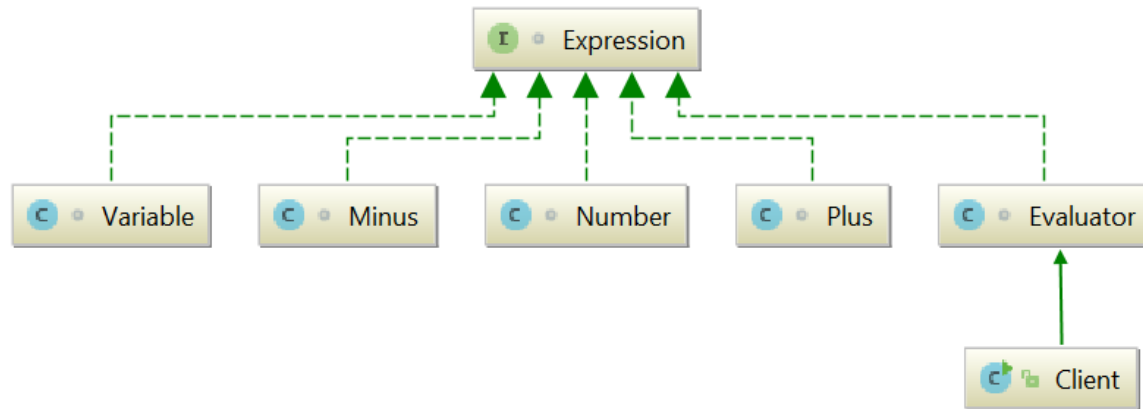
# مثال 1

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

Pattern pattern =
    Pattern.compile("<div>|<span>|<p>)(.*?)(</div>|</span>|</p>)");
Matcher matcher = pattern.matcher("<html><body><p>First Paragraph</p>" +
    "<article>not appeared</article>" +
    "<span>input text search</span>" +
    "</body><html>");

while (matcher.find()) {
    System.out.println(matcher.group());
}
```

# مثال 1



# یک نمونه

```
interface Expression {  
    int interpret(final Map<String, Num> variables);  
}
```

```
class Num implements Expression {  
    private int value;  
    public Num(final int value) {  
        this.value = value;  
    }  
    public int interpret(final Map<String, Num> variables) {  
        return value;  
    }  
}
```



# یک نمونه

```
public class Evaluator implements Expression {
    private Expression syntaxTree;
    public Evaluator(final String expression) {
        final Stack<Expression> expressionStack = new Stack<>();
        for (final String token : expression.split(" "))
            switch (token) {
                case "+":
                    final Expression addExpression =
                        new Plus(expressionStack.pop(), expressionStack.pop());
                    expressionStack.push(addExpression);
                    break;
                case "-":
                    final Expression right = expressionStack.pop();
                    final Expression left = expressionStack.pop();
                    final Expression subExpression = new Minus(left, right);
                    expressionStack.push(subExpression);
                    break;
                default:
                    expressionStack.push(new Variable(token));
                    break;
            }
        syntaxTree = expressionStack.pop();
    }

    public int interpret(final Map<String, Num> context) {
        return syntaxTree.interpret(context);
    }
}
```

# یک نمونه

```
public static void main(final String[] args) {  
    final String expression = "w x z - +";  
    final Evaluator sentence = new Evaluator(expression);  
    final Map<String, Num> variables = new HashMap<>();  
    variables.put("w", new Num(5));  
    variables.put("x", new Num(10));  
    variables.put("z", new Num(42));  
    final int result = sentence.interpret(variables);  
    System.out.println(result);  
}
```

# جمع بندی

- برای تعریف یک گرامر استفاده میشود
- الگویی برای موارد خاص میباشد
- پیاده سازی پیچیده ای دارد
- به ازای هر قانون باید یک کلاس پیاده سازی کنیم