

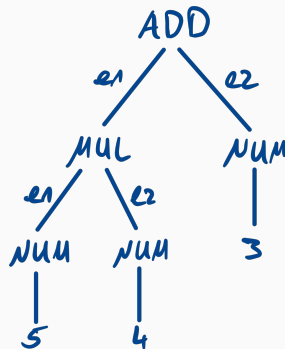
Visitor-Pattern

Carsten Gips (HSBI)

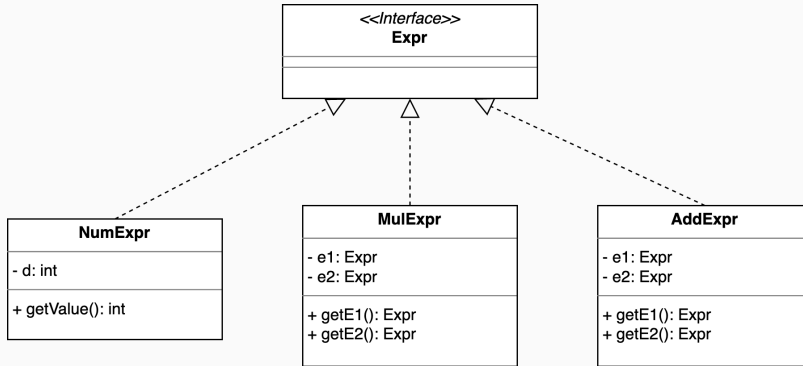
Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

Motivation: Parsen von "5*4+3"

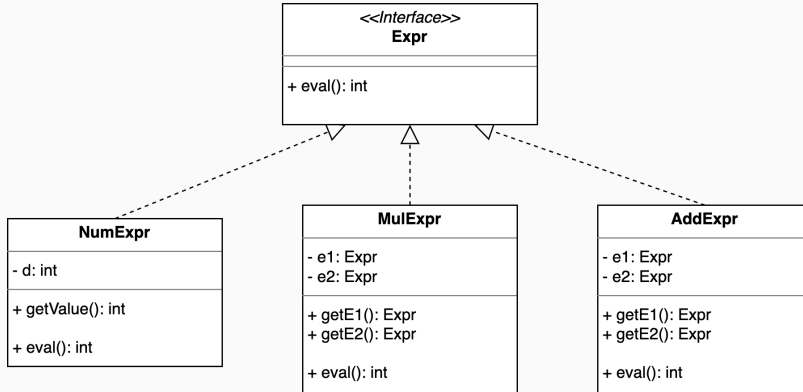
```
expr : e1=expr '*' e2=expr      # MUL  
      | e1=expr '+' e2=expr      # ADD  
      | INT                      # NUM  
      ;
```



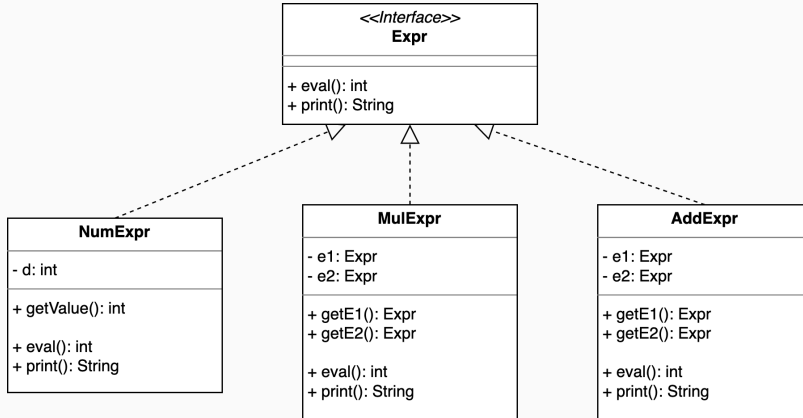
Strukturen für den Parsetree



Ergänzung I: Ausrechnen des Ausdrucks

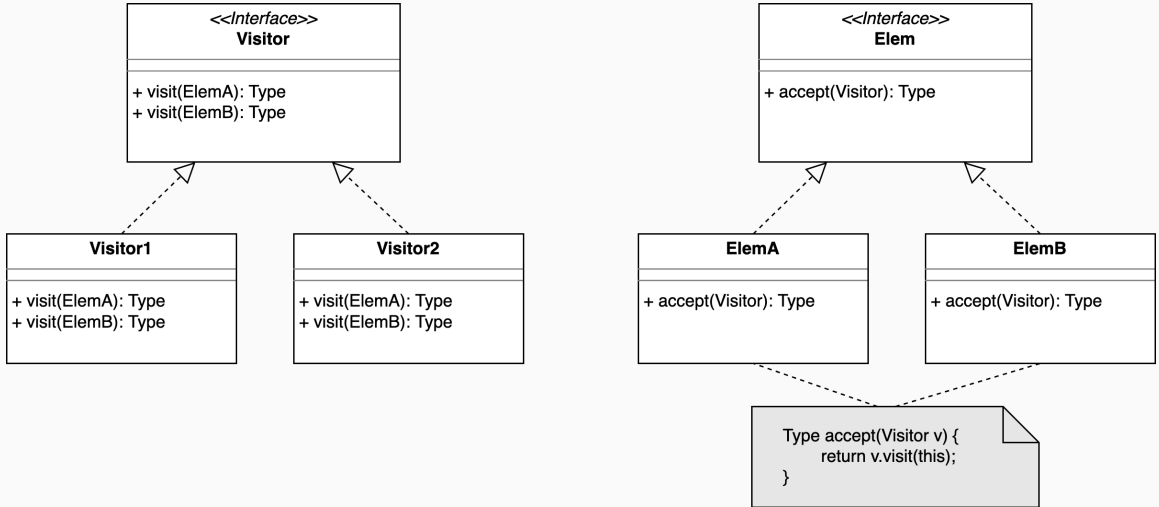


Ergänzung II: Pretty-Print des Ausdrucks

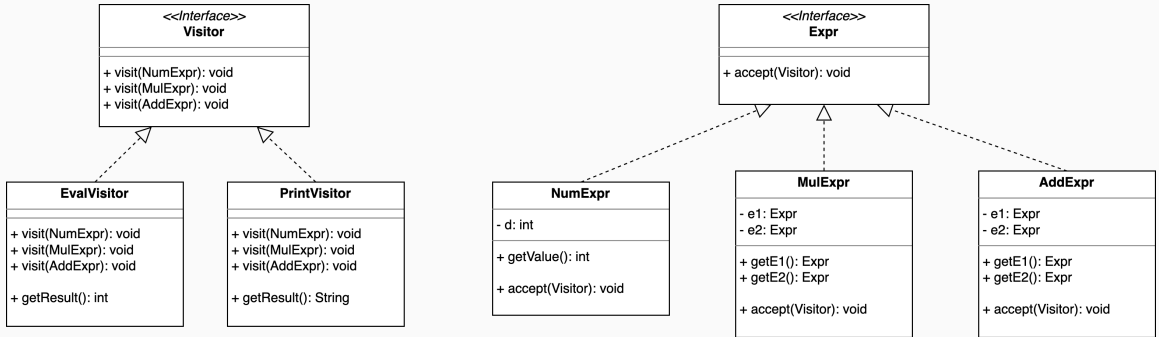


Das geht besser.

Visitor-Pattern (Besucher-Entwurfsmuster)



Ausrechnen des Ausdrucks mit einem Visitor



Demo: `visitor.visit.extrav.DemoExpr`

Visitor-Pattern: Auslagern der Traversierung in eigene Klassenstruktur

- Klassen der Datenstruktur
 - bekommen eine `accept()`-Methode für einen Visitor
 - rufen den Visitor mit sich selbst als Argument auf
- Visitor
 - hat für jede Klasse eine Überladung der `visit()`-Methode
 - Rückgabewerte schwierig: Intern halten oder per `return`
- (Double-) Dispatch: Zur Laufzeit wird in `accept()` der Typ des Visitors und in `visit()` der Typ der zu besuchenden Klasse aufgelöst

LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.