

Visiteur

Exercice 1 - Visiteur

Voici un exemple d'un petit évaluateur et simplificateur d'expression arithmétique `td5eval.zip` (`td5eval.zip`). On cherche à ajouter un mécanisme de visiteur au code existant.

Rappeler ce qu'est un visiteur, à quoi il sert et dans quel cas on l'utilise.

On souhaite écrire un visiteur `SimplifyVisitor` effectuant la même opération que la méthode `simplify` de l'interface `Expr`.

```
public class SimplifyVisitor {
    public Expr visit(Variable variable) {
        ...
    }
    public Expr visit(Constant constant) {
        ...
    }
    public Expr visit(Plus plus) {
        ...
    }
    public Expr visit(Minus minus) {
        ...
    }
    public Expr visit(Star star) {
        ...
    }
    public Expr visit(Slash slash) {
        ...
    }
}
```

Le visiteur sera appelé sur chaque noeud de l'arbre afin de simplifier l'expression courante et renvoyer l'expression simplifiée

Dans le `main` on utilisera le visiteur de cette façon :

```
Expr expr2=expr.accept(new SimplifyVisitor());
```

pour demander la simplification de l'expression.

Écrire le code des méthodes `accept`.

Écrire le code de la classe `SimplifyVisitor`.

On souhaite maintenant utiliser le même mécanisme de visiteur pour écrire un visiteur `ToStringVisitor` effectuant l'équivalent de la méthode `toString()`.

Pour cela, nous allons créer un interface `Visitor` que les classes `SimplifyVisitor` et `ToStringVisitor` implanteront.

Pensez au `generics` !

Écrire le code de l'interface `Visitor`, changer le code des méthodes `accept` et de la classe `SimplifyVisitor` en conséquence.

Écrire le code de la classe `ToStringVisitor`.

On souhaite enfin écrire le visiteur correspondant à l'opération `eval`. Écrire le code de la classe `EvalVisitor` et modifier le code en conséquence.

Attention à la déclaration des exceptions !