TD6 : Visitor

Exercice 1 - Visiteur

Voici un exemple d'un petit évaluateur et simplificateur d'expression arithmétique visitor0.zip (http://igm.univ-mlv.fr/ens/IR/IR2/2011-2012/PrOO/visitor0.zip). On cherche à ajouter un mécanisme de visiteur au code existant.

- Rappelez 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 de renvoyer l'expression simplifiée.
Dans le `main` on demandera alors la simplification de l'expression de cette façon :

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

- Écrivez le code des méthodes `accept`.
- Écrivez 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 une interface `Visitor` que les classes `SimplifyVisitor` et `ToStringVisitor` implanteront.
Pensez au generics !

- Écrivez le code de l'interface `Visitor`, changez le code des méthodes `accept` et de la classe `SimplifyVisitor` en conséquence.

- Écrivez le code de la classe `ToStringVisitor`.
- On souhaite enfin écrire le visiteur correspondant à l'opération `eval`. Écrivez le code de la classe `EvalVisitor` et modifiez le code en conséquence.
  Attention à la déclaration des exceptions !