

ALGORITHMIQUE III.9

LOGIQUE ET ARBRES

Autres opérateurs

En cours, nous avons commencé à définir une hiérarchie de classes pour la représentation de formules logiques, ce qui serait un premier pas vers la création d'un langage de programmation. Pour ce faire, nous avons établi une interface `Formule`, qui spécifie qu'une formule logique est une classe qui peut être évaluée de manière à obtenir un résultat booléen.

Les documents correspondants à ce TD, que vous trouverez à l'emplacement habituel,

Exercice 1. Dans la classe `FormuleFactory`, implantez la méthode `xor`, pour représenter le "ou exclusif" sans ajouter de nouvelle classe. Rappelons que la table de vérité du "ou exclusif" est

a	b	a XOR b
V	V	F
V	F	V
F	V	V
F	F	F

Testez cela à l'aide de la classe `Test`.

Exercice 2. Dans chacune des classes `Vrai`, `Non` et `Ou`, implantez la méthode

```
/**
 * Renvoie la représentation de la Formule sous la forme
 * d'une chaîne de caractères.
 */
public String toString()
```

Ainsi,

- pour $\neg V$, on obtiendra `Non (Vrai)`
- pour $V \vee F$, on obtiendra `(Vrai) ET (Non (Vrai))`
- etc.

Exercice 3. Retravaillez les méthodes `toString` précédentes de manière à supprimer les parenthèses inutiles. Vous aurez besoin d'utiliser `instanceof` pour raffiner vos choix.