

# UE INF1601

## sujet de TP n<sup>o</sup>1

**Avertissement :** Tous les TP sont à faire sous Unix. Tous les programmes Java sont à développer avec Eclipse.

### Exercice 1 : Utilisation de dot en ligne

1. Lisez au moins les trois premières pages de la documentation de `dot` (cf. section *Documentations* sur l'ENT).
2. Utilisez `dot` pour créer un fichier *PDF* contenant le graphe de l'expression arithmétique  $\frac{34+22 \times 3}{15}$  sous la forme d'un arbre infixe.
3. Utilisez le programme `evince` (ou `okular`)<sup>1</sup> pour afficher le fichier créé par `dot` à la question précédente.

### Exercice 2 : Utilisation de dot dans un programme Java

1. Ecrivez un programme JAVA qui lit une suite de lignes de la forme `x -> y` et qui affiche le graphe correspondant à l'écran en utilisant `dot` puis `evince` par des appels "systèmes".

#### Contraintes à respecter :

- le fichier créé pour `dot` sera `/tmp/exp.dot` et celui pour `evince` sera `/tmp/exp.pdf`
- la lecture au clavier se fera avec un `BufferedReader.readLine()` et l'analyse de la ligne lue avec l'aide de la méthode `String.split()`
- l'écriture du fichier d'entrée de `dot` se fera avec un `BufferedWriter.append()`
- le lancement de `dot` et de `evince` se feront par la méthode `Runtime.getRuntime().exec()`
- avant de lancer le processus de visualisation avec `evince` on s'assurera que le processus précédent `dot` est terminé avec la méthode `Process.waitFor()`

---

1. Ces deux programmes prennent un peu de temps au démarrage la première fois qu'ils sont lancés.

2. Vérifiez que votre programme fonctionne en l'appliquant à l'expression arithmétique  $\frac{34+22 \times 3}{15}$  puis à l'arbre syntaxique suivant :

