

## TD1 : Le Langage JAVA - Environnement

*Ce TD a pour objectif de vous familiariser avec les outils que nous allons utiliser et de vous permettre de vous rafraîchir la mémoire sur le langage Java.*

### Les outils utilisés dans le cadre de ce cours

---

La politique d'administration des PC des laboratoires fait que les outils que nous allons utiliser sont pré installés sur les machines. Décrivons les brièvement :

- **Java SE 6** : L'édition standard de Java en est à la version 6<sup>1</sup>. Le produit associé est le Java 2 SDK 1.6 (Software Development Kit). Il contient, le compilateur Java, la machine virtuelle, les librairies standards, ... . Ne plus utiliser la version 1.4 qui ne possède pas les dernières nouveautés du langage.
- **Java 2 SDK Documentation** : la documentation des API standards.
- **Netbeans** est un IDE (Integrated Development Environment), un environnement de développement intégré qui facilite la tâche du programmeur. Un bouton pour compiler, un autre pour exécuter, des facilités d'édition, de génération automatique de code, de « refactoring », ... La version actuelle est la 6.1.

#### Où les trouver ?

Ils sont intégralement (et gratuitement) téléchargeables aux adresses indiquées.

- Pour les produits SUN : <http://java.sun.com/>
- Pour Netbeans : <http://www.netbeans.org/>

### Utilisation de Netbeans

---

Apprenons à exploiter au mieux les possibilités de cet environnement de développement.

Le plus simple est de suivre un des nombreux tutoriaux qui accompagnent le logiciel. Allez au menu « Help/Tutorials/Quick Start Guide » et choisissez « General Java Applications or libraries » « Help/OnLine Docs and Support » et choisissez « NetBeans IDE Quick Start Guide » ensuite, suivez « Introduction to Developing General Java Applications ».

---

1 A comparer à Java 2 EE (la version entreprise pour les applications distribuées) et Java 2 ME (la version micro pour les applications embarquées)

## Se rafraîchir la mémoire (en Java)

---

### Le triangle de Pascal

Écrivez une fonction qui crée un **triangle de Pascal** de **n** lignes (un paramètre). La fonction génère une **IllegalArgumentException** si le paramètre est négatif ou nul. Veillez à n'allouer que l'espace mémoire nécessaire.

Pour rappel, le triangle de Pascal est donné par la récurrence.

$$P(i,1) = P(i,i) = 1$$

$$P(i,j) = P(i-1,j-1) + P(i-1,j) \quad \text{si } 1 < j < i$$

Écrivez également une fonction qui affiche un triangle de Pascal. On vous demande d'utiliser au mieux le « **foreach** » introduit dans la version 5 de Java. Si vous désirez un bel affichage avec des nombres alignés, vous pouvez regarder la méthode **printf()**.

Écrivez enfin une fonction principale pour tester le tout. Le paramètre **n** sera récupéré comme argument de la ligne de commande. Le programme affiche un petit message de type « usage : ... » si il est appelé avec un mauvais nombre d'argument, un argument non entier ou un argument refusé par la méthode de création du Triangle.

Écrivez et produisez la Javadoc.