

TP 2

Design Pattern "Factory"

(ou appelé aussi "Factory Method")

Nous allons, dans ce TP, essayer de réaliser un système pour créer deux types d'ordinateurs, soit un pc, soit un serveur, en utilisant le design pattern Factory.

Etape 1 :

Créez une classe abstraite **Computer** qui correspond à une classe mère et qui définit

- trois signatures (abstraites) de type de retour String :
 - o getRAM : qui permet de renvoyer la taille de la RAM.
 - o getHDD : qui permet de renvoyer la taille du Disque Dur.
 - o getCPU : qui permet de renvoyer la cadence du CPU.
- la méthode toString()

Etape 2 :

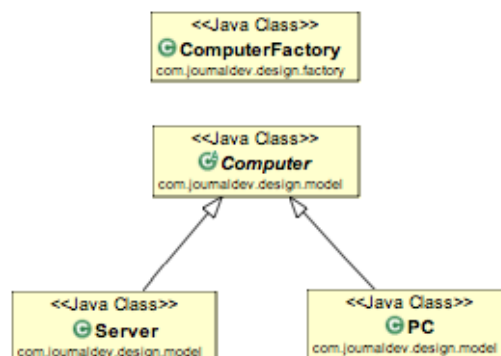
Créez deux classes filles **PC** et **Server** qui héritent de la classe **Computer** avec pour chacune :

- trois propriétés : ram, hdd, cpu.
- un constructeur avec tous les paramètres
- les getters uniquement.

Etape 3 :

Créez une classe **ComputerFactory** qui contient les méthodes nécessaires à la création d'un **Computer** (ici une seule méthode **getComputer(...)** statique qui reçoit en paramètres, en plus des propriétés d'un computer, le type de computer).

Voici à quoi ressemble le diagramme de classes final :



Etape 4 :

Créez une classe de test appelée **Main** qui utilise l'implémentation du modèle de conception réalisé ci-dessus en insérant le code suivant :

```
Computer pc = ComputerFactory.getComputer("pc", "2GB", "500GB", "2.4GHz");
Computer server = ComputerFactory.getComputer("server", "16GB", "1TB", "2.9GHz");
System.out.println("Factory PC Configuration : " + pc);
System.out.println("Factory Server Configuration : " + server);
```

Exécutez et observez le résultat. Que peut-on déduire ?