

## Télécom 2ème année

### MOCI - TP Design Pattern

#### Objectif 1 : comparer les DP Interpréteur et Visiteur.

Un extrait de l'application **AgenceDeVoyages** vous est fourni : les classes **Media**, **Album**, et **Photo**, ainsi qu'une fonction **main** rudimentaire qui crée un album (ignorez la classe **SupportDeComm**, dans un premier temps). Ces classes sont très pauvres et nous allons les compléter petit à petit, en cherchant à chaque étape à prendre du recul par rapport au code pour estimer la qualité de ce que l'on fait et le temps que cela prend. L'objectif n'est pas de programmer une application complète, mais de comprendre et de comparer les patterns **Interpreteur** et **Visiteur**. Pour cela, il est indispensable de répondre à chaque question dans l'ordre et de ne pas chercher à écrire du code qui n'est pas demandé.

1. Quels design patterns sont appliqués dans le code fourni ?
2. Créez un nouveau projet IntelliJ avec deux répertoires **interpreteur** et **visiteur**. Placez une copie de toutes les classes dans les deux répertoires. Vérifiez que l'exécution des deux **Main** se passe sans problème (sans oublier de mettre l'option **-ea** dans les options de la machine virtuelle, dans les configurations d'exécution).
3. Dans le package **interpreteur**, complétez les classes **Album** et **Photo** avec une définition de la fonction **int getNombreDePhotos()** ; complétez la fonction **main** pour tester votre code.
4. Complétez le package **visiteur** avec une interface **Visiteur** et une sous-classe **VisitNbPhotos**. Ce visiteur est capable de visiter **Album** et **Photo** pour compter le nombre de photos. Complétez la classe **Media** avec la fonction **accept**, implémentée dans chaque sous-classe. Complétez la fonction **main** pour tester le code.
5. Un moment de pause : êtes-vous sûr d'avoir assimilé la différence entre les deux solutions ?
6. Il est temps maintenant d'ajouter de nouveaux supports de communication : **Video** (avec un nom de fichier et un nom d'auteur) et **Article** (avec un texte et un nom d'auteur). Complétez les deux packages, en respectant le design pattern correspondant. Complétez les classes de test. Comment estimez-vous l'effort à fournir pour intégrer cette évolution, dans chacun des deux cas ?
7. L'évolution de notre application se poursuit. On souhaite maintenant ajouter deux nouvelles fonctionnalités : *Consulter les textes attribués à un auteur donné* et *Consulter les vidéos au format mp4*. Complétez les packages, en respectant le DP correspondant. Complétez les classes de test. Comment estimez-vous l'effort à fournir dans chacun des deux cas ?
8. L'ajout de la nouvelle fonctionnalité *Supprimer les vidéos d'un auteur donné* pose-t-elle un problème dans l'une ou l'autre des deux implantations ? Comment le résoudre ?
9. Les DP **Interpreteur** et **Visiteur** ne devraient plus avoir de secrets pour vous. Faites la synthèse de cet exercice en dessinant, pour les deux solutions, les diagrammes de classe et de séquence des différents cas étudiés. Faites la synthèse de tous les DP utilisés dans cet exemple.