

## IF26 - développement d'applications mobiles

### TP02 - Java et XML

#### Exercice 1 : Enumeration et Interface

1. Recopiez tous les fichiers du package `if26.tp01.solution2` dans un nouveau package `if26.tp02.exo1`. Faire attention aux imports au début de chaque fichier.
2. Recréez une énumération `Resultat` (enum Java) acceptant les valeurs suivantes : A(18), B(16), C(14), D(12), E(10), FX(-2), F(-5), ABS(0), R(0). Le paramètre correspond à une note sur 20. Générez le constructeur et les méthodes `set()` et `get()`.
3. Ajoutez un nouvel attribut `resultat` de type `Resultat` à la classe `Element` et modifiez les méthodes des classes `Element`, `Module` et `Stage` pour conserver la cohérence entre les classes.
4. Validez vos modifications en lançant l'exécution de la classe `Main` du package `if26.tp02.exo1`.
5. Créez maintenant une interface `Jury` contenant les méthodes suivantes :
  - a. `public String excellence();`
  - b. `public String triParCredit();`
  - c. `public String triParResultat();`
6. Modifiez la classe `Cursus` pour qu'elle implémente l'interface `Jury`. Que faut-il faire maintenant pour respecter les contraintes ainsi définies ? Réalisez automatiquement les modifications avec l'aide de Netbeans.
7. La méthode `excellence()` permet de calculer un indicateur sur les cursus des étudiants. La valeur correspond à la somme des notes (sur 20) associées à chaque élément.
8. La méthode `triParCredit()` doit réaliser un tri des éléments par rapport aux valeurs de l'attribut crédit. La solution consiste à utiliser la méthode `Collections.sort(...)` en indiquant la collection à trier (qui correspond à profil) et une classe de tri.
9. Rédigez la classe `ElementTriCredit` qui implémente l'interface `Comparator` possédant une méthode `compare()`.
10. Faire la même chose avec la méthode `triParResultat()`.
11. Ajoutez les instructions nécessaires dans la méthode `main()` de la classe `Main` pour valider la méthode `excellence()` et les deux méthodes de tri.

#### Exercice 2 : Communication événementielle

1. Recopiez tous les fichiers du package `if26.tp02.exo1` dans un nouveau package `if26.tp02.exo2`. Faire attention aux imports au début de chaque fichier.
2. Ajoutez un attribut `label` à la classe `Cursus`. Ajoutez les méthodes `set()` et `get()`, modifiez le constructeur de `Cursus` et la méthode `main`.
3. Nous souhaitons mettre en place un système permettant d'informer des professeurs quand le cursus d'un étudiant change (par ajout d'un module). Pour cela il faut utiliser le couple `Observer/Observable`. Modifiez la classe `Cursus` pour qu'elle hérite de la classe `Observable`.
4. Ajoutez dans la méthode `ajoute()` les méthodes `setChanged()` et `notifyObservers()`.
5. Rédigez une classe `Prof` ayant un attribut `nom` et implémentant l'interface `Observer`.
6. Ajoutez à la classe `Prof` un constructeur et les méthodes `set()` et `get()`.
7. Ajoutez à la classe `Prof` la méthode `update()` qui affichera le `nom` du prof et le `label` du cursus.

8. Modifiez la méthode `main()` en construisant deux `cursus` (Sandra et Lucie) contenant des `éléments` et deux `profs` (Michael et Marc).
9. Ajoutez les instructions nécessaires (méthode `addObserver()`) pour que Michael soit informé des modifications sur le cursus de Lucie et que Marc le soit pour les deux cursus.
10. Ajoutez un `élément` dans chaque `cursus` et vérifiez que les deux profs ont bien reçu les notifications.

### Exercice 3 : Application Android et fichiers XML (android-AppShortcuts)

1. Téléchargez le zip de l'application `android-AppShortcuts` disponible sur le site [IF26](#) (elearning).
2. Dans quel répertoire se trouve le fichier `AndroidManifest.xml` ?
3. Ce fichier est un document XML donc il respecte les contraintes des documents XML présentées dans le cours XML disponible sur le site [IF26](#) (elearning).
  - a. Quelle est l'élément racine du document XML ?
  - b. Que signifie `xmlns:android = " ..."` ?
  - c. Quelles sont les principales balises du document ? A quoi correspondent-elles ?
  - d. Quels sont les attributs des balises `<uses-permission>` et `<application>` ?
4. Recensez les informations disponibles dans ce fichier `AndroidManifest` :
  - a. Nom du package de l'application ?
  - b. La version minimale du SDK acceptée ?
  - c. Quels sont les droits demandés par l'application ?
  - d. Combien d'activités sont définies dans ce document ?
5. Dans quel répertoire sont localisées les fichiers `Java` de l'application ?
6. Quel est le rôle des fichiers `XML` localisés dans le répertoire `layout` ?
7. Que peut-on dire sur l'interface décrite dans le fichier `main.xml` ?
8. Même chose avec les autres fichiers `XML` du répertoire `layout` ?
9. Quel est le rôle du fichier `XML strings.xml` localisé dans le répertoire `values` ?
10. Que contient ce fichier `strings.xml` ?

### Exercice 4 : Application Android et fichiers XML (android-AutofillFramework)

1. Téléchargez le zip de l'application `android-AutofillFramework` disponible sur le site [IF26](#) (elearning).
2. Mêmes questions que pour l'exercice 3.
3. Commentaires ?
4. Bilan.

### Exercice final : devoir

1. Pour chaque étudiant, proposez soit une analyse critique, soit une question supplémentaire à l'un des exercices de ce TD/TP, soit un QCM ou soit une ressource disponible sur Internet.
2. Vous transmettez votre proposition via le devoir moodle sur le site e-learning du module.