IF26 - développement d'applications mobiles

TP01 - révisions Java

Préparations

➢ Objectifs

- L'objectif de ce premier TP est de se familiariser avec un IDE pour la conception de programmes en Java. Pour les deux premiers TP, nous utiliserons l'IDE Netbeans.

➢ Connaissances et compétences

- Principaux concepts d'un langage Objet (classe, objet, héritage, polymorphisme, classe abstraite, interface, évènements, etc.

➢ Ressources

- Cours de Marc Lemercier sur elearning

Exercice 1 : Mes premières classes en Java

1. Lancez l'IDE Netbeans et créez un nouveau projet Java ayant pour nom la concaténation de vos noms de famille (exemple DupontDurant).
2. Rédigez une classe Module appartenant au package if26.tp01.solution1 et comportant les attributs privés suivants : sigle, categorie, parcours (TCBR, FCBR, TC, UTT) de type chaîne de caractères et credit de type entier.
3. Quel doit être le nom des méthodes constructrices d'une classe Java ?
4. Ajoutez un constructeur par défaut qui initialise les attributs avec des données par défaut.
5. Générez avec Netbeans un nouveau constructeur comprenant tous les paramètres possibles (liste de tous les attributs).
6. Générez automatiquement toutes les méthodes get() et set().
7. Générez aussi la méthode toString() qui permet de construire une vue textuelle d'un objet de la classe Module.
8. Rédigez une méthode public affiche() qui permet l'affichage à l'écran des informations d'un document.
9. Créez une nouvelle classe Main contenant une seule méthode public static void main(String [ ] args). Cette classe sera donc la classe principale de cet exercice.
10. Dans la méthode main(), créez un objet module labélisé lo07 à l'aide du constructeur par défaut et un module if26 avec l'autre constructeur en proposant des valeurs pour les paramètres.
11. Ajoutez les instructions nécessaires pour afficher les deux modules.
12. Lancez l'exécution de la classe Main et valider votre travail par la visualisation des résultats.

Exercice 2 : Collection d'objets

1. Maintenant, nous souhaitons conserver un ensemble d'objets Module dans une structure de données appelée Cursus. Rédigez une nouvelle classe Cursus dans le package if26.tp01.solution1 possédant un unique attribut profil de type ArrayList.
2. Ajoutez un constructeur par défaut.

3. Rédigez une méthode ajoute() permettant l'ajout dans la structure profil d'un objet de la classe Module passé en paramètre.
4. Rédigez une méthode affiche1() permettant de visualiser l'ensemble des modules présents en utilisant un objet de la classe Iterateur.
5. Complétez la méthode main() de la classe Main avec l'ajout d'un objet cursus de la classe Cursus contenant les deux modules déjà créés.
6. Ajoutez les instructions nécessaires pour afficher les informations de l'objet cursus.
7. Lancez l'exécution de la classe Main et valider votre travail par la visualisation des résultats.
8. Créez une nouvelle classe Stage comprenant les attributs suivants : sigle, categorie, credit et entreprise.
9. Générez automatiquement un constructeur, les méthodes get() et set() et toString().
10. Ajoutez la méthode affiche().
11. Dans la la méthode main(), ajoutez un objet de la classe Stage à l'objet cursus et lancez l'exécution de votre programme.
12. Expliquez l'erreur obtenue et proposez une solution dans une méthode affiche2().
13. Validez votre proposition par une exécution du programme.
14. Ajoutez des commentaires Javadoc (/**) dans le code de la classe Module et générez la Javadoc. Vérifiez que vos commentaires sont bien intégrés (niveau classe, niveau méthode, ...).

Exercice 3 : Comprendre l'héritage et le polymorphisme

1. L'exercice 1 a montrer les limites de la conception d'un projet Java construit avec des classes indépendantes. L'exercice 3 va permettre de clarifier et de simplifier la conception à l'aide des propriétés associées à l'héritage. Maintenant un objet de la classe Cursus ne pourra contenir que des objets de la classe Element. Les classes Module et Stage vont hériter de la classe Element. Cependant, un objet élément n'a pas d'existence réelle pour un cursus. Donc la classe Element sera une classe abstraite afin d'interdire la création d'objet de cette classe.
2. Dans un nouveau package if26.tp01.solution2, rédigez la classe abstraite Element ayant les attributs sigle, categorie et crédit. Générez ou rédigez les méthodes constructeurs, set(), get(), toString() et affiche().
3. Recopiez dans ce package la classe Module de l'exercice 1 et effectuez les modifications demandées.
4. Même chose avec la classe Stage.
5. Même chose avec la classe Cursus. La classe Cursus contiendra une seule méthode affiche().
6. Même chose avec la classe Main.
7. Validez votre exercice en lançant l'exécution de la classe Main du package if26.tp01.solution2.
8. Dans la méthode main(), essayez de construire un objet de la classe Element. Bilan.