

JAVA – TD 4

Exercice n° 1

Ecrire une classe *Entreprise*. Une entreprise a un nombre d'employés, un capital, un nom, une mission, et une méthode public String *mission()* qui renvoie la mission de l'entreprise et qui déclare le lancement de l'exception *SecretMissionException*. On aura également une méthode public *int capital()* qui renvoie le capital et qui déclare le lancement de l'exception *NonProfitException*.

- Ecrire une classe *EntrepriseSecrete* qui hérite d'*Entreprise* et dont la méthode *mission* lance l'exception *SecretMissionException*.
- Ecrire une classe *EntrepriseSansProfit* qui hérite d'*Entreprise* et dont la méthode *capital* lance l'exception *NonProfitException*.
- Ecrire une méthode qui prend en entrée un tableau d'entreprises et affiche la mission et le capital de toutes les entreprises (quand cela est possible).
- Tester la méthode sur les entreprises "Ford", "Shell", "Yahoo", "IBM", "Microsoft", "Alpha".

Exercice n° 2 [Utilisation des exceptions dans les constructeurs]

Toutou est une classe avec deux propriétés privées String *nom* et int *nombrePuces*. Ecrire un constructeur public Toutou (String n, int np) qui propage des exceptions de type *IllegalArgumentException* lorsque le nom n est null ou lorsque le nombre de puces np est négatif. Utiliser ce constructeur dans une méthode main pour contrôler les appels new Toutou ("milou", 4) et new Toutou ("medor", -11) et afficher les erreurs éventuelles lors de l'exécution des constructeurs

Exercice n° 3

1. Construisez une classe abstraite *TabTrie* qui correspond à un tableau trié d'objets. Cette classe doit notamment contenir :

- un tableau d'Object, tab, initialisé avec une capacité définie par défaut (il faut penser aussi à stocker le nombre d'Object contenus dans le tableau), et différentes méthodes qui peuvent être implantées ou abstraites :
- une méthode *plusGrand* qui compare deux objects et renvoie true si le premier est plus grand que le deuxième,
- une méthode *ajouter* qui insère un objet dans le tableau en respectant l'ordre croissant,
- une méthode *toString* qui renvoie une chaîne de caractères représentant le tableau. Lorsque la capacité du tableau est atteinte, l'insertion d'un nouveau élément lancera une exception *TabPlein*.

2. Construisez la classe *TabTriCouple* qui hérite de *TabTri* et ordonne des objets de type Couple lexicographiquement