

UEO15 – Programmation objet

Deuxième session

Durée 2H -Documents autorisés-

Exercice 1 : les exceptions (6 points)

On suppose définie une liste de figures : *LinkedList<Figure> figures* ;

On souhaite afficher la liste des noms des figures. Certaines figures possèdent un nom (cercleNommé, polygoneNommé, ...) et d'autres pas (point, ...). Les figures possédant un nom implémentent l'interface *Nomme* qui possède une méthode *public void String getNom()*.

1. En utilisant *instance of*, coder la méthode *public String Noms()* qui concatène les noms de toutes les figures qui en possède un.
2. En utilisant les exceptions (et sans utiliser *instance of*), coder la même méthode. On rappelle que lorsqu'un appel est fait sur une méthode qui n'est pas présente dans une classe, l'exception *NoSuchMethodException* est levée.

Exercice 2 : les interfaces graphiques (5 points)

1. Expliquer brièvement le fonctionnement de l'interaction entre un utilisateur et une interface graphique en java.
2. On souhaite coder dans un unique fichier une interface graphique possédant un bouton appelé *inverse* qui affiche alternativement « Bonjour » et « Au revoir » dans un *Jlabel* (pour changer le texte du *Jlabel* on utilise la méthode *setText()* prenant en paramètre une chaîne de caractère).

Exercice 3 : les structures de données (6 points)

1. Coder une méthode prenant en paramètre : une structure de donnée ne pouvant contenir que des *Figures* et une *Figure F*. La méthode renvoie le nombre de *Figures* de même type que *F* contenues dans la structure. La méthode *getClass()* est présente dans toutes les classes java et renvoie la classe de l'objet appelant la méthode (exemple : *Integer I* ; *I.getClass()* renvoie *Integer*).
2. On souhaite stocker les *Figures* dans une liste triée (*sortedSet*). Quelle interface doivent implémenter les figures pour que cela puisse fonctionner ?
3. On suppose codée la méthode *getPoids()* qui renvoie le poids de la figure (nombre de points de celle-ci par exemple). Coder la méthode présente dans l'interface précédente et permettant de faire la comparaison entre deux figure en utilisant le poids.

Exercice 4 : la sérialisation (3 points)

1. Expliquer le fonctionnement de la sérialisation.
2. Indiquer le contenu du fichier (en expliquant) après les opérations suivantes
 - a. *Point P= new Point(1,2)* ;
 - b. Ouverture d'un accès à un fichier *F*
 - c. Sérialisation de *P* dans le fichier *F*
 - d. *P.translater(1,1)*
 - e. Sérialisation de *P* dans le fichier *F*
 - f. Fermeture du fichier *F*