**JAVA TD**

**Lê Nguyễn Minh Hương**

**Exercice 1 : Classe, Hiérachie**

1. **Repérez des classes, indiquez si elle hérite d’une autre :**  
   - la classe Cercle  
   - la classe VisualiseurDeFormes hérite de la classe JFrame  
   - la classe ZoneDeDessin hérite de la classe JPanel
2. **Pourquoi Forme est-elle une interface et non une classe :**  
   - l’entête de la fonction redessiner( Graphics g ) peut être utilisée par plusieurs classes 🡪 cette fonction sera redéfinie par d’autres classes de façon différente.  
   - Forme est presque une classe abstraite : il s’agit d’une conception abstraite qui n’a pas de précis attributs ( une Forme peut être un Cercle, un Rectangle, un Triangle… ) et dont les sous-classes se comportent différemment.
3. **Quelle est la classe qui représente la fenêtre de l’application ? Changez donc le titre affiché sur la fenêtre :**- c’est la classe VisualiseurDeFormes, car :   
    + le programme principal ( public static void main ) s’écrit dans le corps de cette classe  
    + dans son constructeur, il y a des instructions qui servent à définie une application ( déterminer la taille, gérer des événements de boutons cliqués, de la fin de l’application… )  
     
   - pour changer le titre affiché sur la fenêtre ( « un très joli visualiseur » ) : on modifie l’instruction super(« … » ) dans le constructeur.
4. **La classe qui se charge de dessiner les cercles ?** 🡪 la classe ZoneDeDessin  
   **La méthode òu l’affichage des formes/cercles est réalisée** 🡪 public void paintComponent( Graphics g )

**Exercice 2 : Objets**

Tous les objets instanciés dans le constructeur de VisualiseurDeFormes  et leur utilité :

* m\_formes = new **Vector**( ) 🡪 un conteneur sert à stocket les formes
* m\_random = new **Random**( ) 🡪 créer un nouveau générateur de nombres aléatoires
* new **Dimension**( 400, 300 ) 🡪 construire une dimension en initialisant sa longueur ( 400 ) et largeur ( 300 )
* new **BorderLayout**( ) 🡪 définir une disposition des composants
* new **ZoneDeDessin**( ) 🡪 instancier la classe qui charge de dessiner les formes
* new **JPanel**( ) 🡪 définir un nouvel objet de type JPanel
* new **JLabel**( "Visualiseur de Formes" ) 🡪 créer une instance de la classe JLabel avec le texte spécifié entre les « … »
* new **BoutonCercleAction**( this ) 🡪créer une instance de la classe BoutonCercleAction qui modélise la création d’un cercle suite à un clic sur le bouton
* new **JButton**( "Cercle" ) 🡪 créer un nouveau bouton avec le texte spécifié entre les «… »

**Exercice 3 : Programme principal**

* La méthode main se trouve dans la classe VisualiseurDeFormes.   
  Elle crée la fenêtre graphique en faisant l’allocation VisualiseurDeFormes visu = new VisualiseurDeFormes( ) ;
* Faire 2 instances de VisualiseurDeFormes pour créer 2 fenêtres d’application.
* Pour que le titre des fenêtres soit différent, on ajouter le paramètre String title dans le constructeur de la classe VisualiseurDeFormes / modifier par la fonction setTitle( string ) dans le constructeur de VisualiseurDeFormes.

**Exercice 4 : Polymorphisme : d’autres formes**

* **Les cercles et rectangles sont stockés dans un même objet (un vecteur d’Objet) et pourtant leur affichage est bien différencié. Comment cela est-il possible ?**  
  🡪 Parce que ces classes héritent d’une même interface Forme. Il s’agit bien d’un Vecteur contenant les pointeurs de type Forme, et chaque pointeur prend l’adresse d’un objet plus précis des sous-classes.
* **Que se passe-t-il si vous enlevez dans la méthode ZoneDeDessin.paintComponent le terme (Forme) devant m\_formes.get( i ) ?**  
  🡪 Le compilateur signale des erreurs : *Type mismatch ( cannot convert from Object to Forme )*🡪 Il faut un transtypage pour transformer l’objet donné par m\_formes.get( i ) au type plus précis ( Forme )
* **m\_formes.add( new String( "Texte" ) );  
  Le programme JAVA compile-t-il ? Si oui, que se passe-t-il à l’exécution ?**  
  🡪 Oui, il compile, mais une exception est lancée :   
  *Exception in thread "AWT-EventQueue-0" java.lang.ClassCastException: java.lang.****String cannot be cast to Forme at ZoneDeDessin.paintComponent****(ZoneDeDessin.java:36)*

**Exercice 5 : Bouton de création de rectangles**

**Exercice 6 : Bouton « reset »**

**Exercice 7 : Bouton « aléatoire »**

**Exercice 8 : Classe abstraite FormeColoree**

- Cette classe doit être abstraite car il s’agit d’une conception générale de l’objet FORME, il n’a pas d’attributs précis, et la méthode redessiner( Graphics g ) se comporte différemment selon différents sous classe

**Exercice 9 : Classe imbriquée**

**Exercice 10 : IHM**