



Questions de cours : (4 pts)

Voici la 1^{re} ligne de la définition d'une classe ImageViewer :

```
public class ImageViewer implements ActionListener
```

Dans la documentation, on trouve que ActionListener a une seule méthode, appelée actionPerformed().

- 1) Qu'est-ce que ActionListener ?
- 2) Que doit contenir la classe ImageViewer pour qu'il n'y ait pas d'erreur à la compilation ?
- 3) On trouve souvent que les types dont le nom se terminent par Listener ont un équivalent dont le nom se termine par Adapter. Exemples : MouseListener / MouseAdapter, KeyListener / KeyAdapter... Ce n'est pas le cas pour ActionListener, pourquoi ?

Exercice 1 : (9 pts)

Examinez attentivement le code Java suivant. Certaines lignes provoquent une erreur.

```
1.  class A
2.  {
3.      protected void affiche() {
4.          System.out.println(this) ;
5.      }
6.      public void a() {
7.          System.out.print("a de A dans ");
8.          affiche();
9.      }
10.     public void b() {
11.         System.out.print("b de A dans ");
12.         affiche() ;
13.     }
14. }
15. class B extends A
16. {
17.     public void b() {
18.         super();
19.         System.out.print("b de B dans ");
20.         affiche();
21.     }
22.     public void c() {
23.         super.c();
24.         System.out.print("c de B dans ");
25.         affiche();
26.     }
27. }
28. public class Exercice
29. {
30.     public static void main(String[] args) {
31.         A a1 = new A();
32.         A b1 = new B();
33.         B a2 = new A();
34.         B b2 = new B();
35.         a1.a();
36.         b1.a();
37.         a2.a();
38.         b2.a();
39.         a1.b();
40.         b1.b();
41.         a2.b();
42.         b2.b();
43.         a1.c();
44.         b1.c();
45.         a2.c();
46.         b2.c();
47.         ((B) a1).c();
48.         ((B) b1).c();
49.         ((B) a2).c();
50.         ((B) b2).c();
51.     }
52. }
```

1) Donnez les numéros des lignes qui provoquent une erreur et indiquez si l'erreur se produit à l'exécution ou à la compilation.

2) On supprime toutes les lignes qui provoquent une erreur dans ce programme. Donnez le résultat de son exécution en indiquant à côté de chaque ligne affichée le numéro de ligne de l'appel dans `main()` qui lui correspond.

(Attention : toute réponse sans ces indications ne sera pas corrigée !).

Exercice 2 : (7 pts)

Nous avons modifié la classe Pile faite en TP et qui utilisait un tableau pour en faire une version utilisant une liste chaînée. Voici le nouveau code obtenu, examinez-le attentivement :

```
public class Liste
{
    // Objet de type ElementPile contenu dans ce maillon de la liste
    private ElementPile element;
    // Référence du maillon suivant de la liste chaînée
    private Liste suivant;
    // Constructeur ajoutant un ElementPile à une liste existante
    public Liste(ElementPile e, Liste suivant){
        element = e;
        this.suivant = suivant;
    }
    public ElementPile tete(){
        return element;
    }
    // Supprime l'élément se trouvant au début de cette liste
    // et retourne la référence de la nouvelle tête de liste
    public Liste decapiter(){
        element = null;
        Liste nouvelleTete = suivant;
        suivant = null;
        return nouvelleTete;
    }
    // Affiche la liste dans l'ordre (la tête à la fin)
    public void afficher(){
        if (suivant != null){
            suivant.afficher();
            System.out.print(", ");
        }
        if (element != null)
            element.afficher();
        else
            System.out.print("null");
    }
}

public class Pile
{
    private Liste liste;
    public Pile(int taille){
        liste = null; // pas de taille initiale
    }
}

...

```

La classe ElementPile n'a pas changé, elle contient entre autre un constructeur qui prend en paramètre un entier.

1) Complétez la classe Pile avec les méthodes :

- `estVide()` qui retourne `true` si la pile est vide,
- `empiler()` qui prend en paramètre un `ElementPile` et l'empile s'il n'est pas `null`,
- `sommet()` qui renvoie la réf. du dernier élément empilé ou `null` si la pile est vide,
- `depiler()` qui agit comme `sommet` mais supprime le dernier élément empilé de la pile,
- `afficher()` qui affiche les éléments de la pile.

2) Surchargez `empiler()` avec une méthode qui prend en paramètre un entier.

Question Bonus : la méthode `afficher()` de la classe `Liste` est récursive, modifiez-la en proposant une solution itérative donnant le même résultat (dans le même ordre!).