



Questions de cours : (4 pts)

1) `ActionListener` est une interface. (1 pt)

2) Pour qu'il n'y ait pas d'erreur à la compilation la classe `ImageViewer` doit contenir la (re)définition de la méthode `actionPerformed()` héritée de `ActionListener`. (1 pt)

3) Il n'y a pas de classe `ActionAdapter` dans l'API de Java car `ActionListener` n'a qu'une seule méthode. (2 pts)

Exercice 1 : (9 pts)

1) Il y a 6 erreurs :

Ligne 18 : erreur à la compilation.

Ligne 23 : erreur à la compilation.

Ligne 33 : erreur à la compilation.

37 (<= 33)

41 (<= 33)

Ligne 43 : erreur à la compilation.

Ligne 44 : erreur à la compilation.

45 (<= 33)

Ligne 47 : erreur à l'exécution.

49 (<= 33)

2) Après avoir supprimé les lignes 18, 23, 33, 43, 44, 47 qui contiennent les erreurs et les lignes 37, 41, 45 et 49 qui contiennent la variable `a2` qui n'existe plus (elle était déclarée en ligne 33), l'exécution du programme donne :

a de A dans A@<adr. de a1> ← ligne 35

a de A dans B@<adr. de b1> ← ligne 36

a de A dans B@<adr. de b2> ← ligne 38

b de A dans A@<adr. de a1> ← ligne 39

b de B dans B@<adr. de b1> ← ligne 40

b de B dans B@<adr. de b2> ← ligne 42

c de B dans B@<adr. de b2> ← ligne 46

c de B dans B@<adr. de b1> ← ligne 48

c de B dans B@<adr. de b2> ← ligne 50

Exercice 2 : (7 pts)

1) et 2) classe Pile :

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

    public boolean estVide(){
        return (liste == null);
    }

    public void empiler(ElementPile element){
        if (element == null)
            return;
        liste = new Liste(element, liste);
    }

    public ElementPile sommet(){
        if (estVide())
            return null;
        return liste.tete();
    }

    public ElementPile depiler(){
        if (estVide())
            return null;
        ElementPile sommet = sommet();
        liste = liste.decapiter();
        return sommet;
    }

    public void afficher(){
        if (estVide()) {
            System.out.println("Pile vide.");
            return;
        }
        System.out.print("(");
        liste.afficher();
        System.out.println(")");
    }

    public void empiler(int valeur){
        ElementPile element = new ElementPile(valeur);
        empiler(element);
    }
}
```

Question Bonus : méthode afficher() itérative :

```
public void afficher() {
    // construction d'une copie de cette liste chaînée
    // en commençant par la tête de liste (==> ordre inverse)
    Liste inv = null;
    for(Liste p=this ; p!=null ; p=p.suivant){
        inv = new Liste(p.element, inv);
    }
    // parcours et affichage des éléments de la nouvelle liste
    for(Liste p=inv ; p!=null ; p=p.suivant){
        if (p.element != null){
            p.element.afficher();
            if(p.suivant != null)
                System.out.print(", ");
        } else
            System.out.print("null");
    }
}
```