

## Contrôle n°2

Durée : 1 heure. Aucun document autorisé.

**Exercice 1 : (3 points)** Qu'affiche le programme suivant ?

```
class B extends A {
    B(int y) {
        super(3);
        System.out.println("[B : " + y + "]");
        x = super.truc(y);
    }
    int truc(int v) { System.out.println("B : " + v); return v + x; }
}
class A {
    protected int x;
    A(int v) {
        x = 12 + v;
        System.out.println("[A : " + v + "]");
    }
    int truc(int v) { System.out.println("A : " + (v + x)); return v + x; }
}
class C extends D {
    int truc(int v) { System.out.println("C : " + v); return x + super.truc(x); }
}
abstract class D extends A {
    D() {
        super(9);
        System.out.println("[D]");
    }
}
class CC2HeritageTer {
    public static void main(String[] args) {
        A a = new B(13);
        D d = new C();
        System.out.println("(" + a.truc(8) + ", " + d.truc(1) + ")");
    }
}
```

**Exercice 2 : (7 points)**

On souhaite développer une application de dessin vectoriel. La partie qui nous intéresse concerne la représentation des figures géométriques.

On distingue trois éléments géométriques : un point est défini par deux coordonnées ; un segment est défini par deux points ; un polygone est défini par une suite ordonnée de points (de 10 points maximum). Tous les éléments géométriques ont une couleur.

1. Définir les classes `Element`, `Point`, `Segment` et `Polygone` pour représenter les éléments décrits ci-dessus. Mettre dans chaque classe une méthode `public String toString()` et un constructeur prenant en paramètre toutes les informations, sauf pour la classe `Polygone` dont le constructeur prendra en paramètre les trois premiers points du polygone.

2. Définir une interface `Figure` regroupant les segments et les polygones. Y mettre une méthode `public float longueur()`. En supposant l'existence d'une méthode extérieure `public static void calcul(Point p1, Point p2)` dans une classe `Distance`, implanter la méthode `longueur` dans les sous-types de `Figure`.
3. Ajouter dans la classe `Element` une méthode abstraite `public Figure ajouterPoint(Point p)`. Le comportement de cette méthode est le suivant : ajouter un point à un autre point renvoie un segment composé des deux points (dans l'ordre) ; ajouter un point à un segment renvoie un polygone composé des trois points (dans l'ordre) ; ajouter un point à un polygone modifie le polygone et le renvoie.
4. Ajouter dans la classe `Element` une méthode abstraite `public Element supprimerDernierPoint()`. Son comportement est le suivant : supprimer un point d'une figure de plus de trois points retire le dernier et renvoie le polygone modifié ; supprimer un point d'un polygone de trois points renvoie le segment restant ; supprimer un point d'un segment renvoie le premier point ; supprimer un point d'un point renvoie le point lui-même.