OBJECT ORIENTED PROGRAMMING

Lab 6: OOP inheritance (1)

### Exercise 1:

What results does this program provide? Explain the results.

**class A**

**{**

**public A (int nn)**

**{**

**System.out.println ("Entree Constr A - n=" + n + " p=" + p) ;**

**n = nn ;**

**System.out.println ("Sortie Constr A - n=" + n + " p=" + p) ;**

**}**

**public int n ;**

**public int p=10 ;**

**}**

**class B extends A**

**{**

**public B (int n, int pp)**

**{**

**super (n) ;**

**System.out.println ("Entree Constr B - n=" + n + " p=" + p + "**

**q=" + q) ;**

**p = pp ;**

**q = 2\*n ;**

**System.out.println ("Sortie Constr B - n=" + n + " p=" + p + "q=" + q) ;**

**}**

**public int q=25 ;**

**}**

**public class TstInit**

**{**

**public static void main (String args[])**

**{**

**A a = new A(5) ;**

**B b = new B(5, 3) ;**

**}**

**}**

Answer：

Entree Constr A - n=0 p=10

Sortie Constr A - n=5 p=10

Entree Constr A - n=0 p=10

Sortie Constr A - n=5 p=10

Entree Constr B - n=5 p=10q=25

Sortie Constr B - n=5 p=3q=10

### Exercise 2:

We have the following class:

**class Point**

**{**

**public void initialise (int x, int y)**

**{**

**this.x = x ; this.y = y ;**

**}**

**public void deplace (int dx, int dy)**

**{**

**x += dx ; y += dy ;**

**}**

**public int getX() { return x ; }**

**public int getY() { return y ; }**

**private int x, y ;**

**}**

Create a **PointA** class, derived from Point with a method **affiche** that display (in console window) the coordinates of a point. Write a small test program that uses the two classes **Point** and **PointA**.

What if the Point class did not have **getX** and **getY** methods?

### Exercise 3:

We have the following class:

**class Point**

**{**

**public void setPoint (int x, int y)**

**{**

**this.x = x ; this.y = y ;**

**}**

**public void deplace (int dx, int dy)**

**{**

**x += dx ;**

**y += dy ;**

**}**

**public void affCoord ()**

**{**

**System.out.println ("Coordonnees : " + x + " " + y) ;**

**}**

**private int x, y ;**

**}**

Create a **PointNom** class, derived from **Point**, allowing to handle points defined by two coordinates (**int**) and a name (**character**). The following methods will be provided for:

* *setPointNom* to define the coordinates and name of a **PointNom** type object,
* *setName* to define only the name of such an object,
* *affCoordNom* to display the coordinates and name of a **PointNom** type object.

Write a small program using the **PointName** class.

### Exercise 4:

We have the following class (this time with a manufacturer):

**class Point**

**{**

**public Point (int x, int y) {**

**this.x = x ; this.y = y ;**

**}**

**public void affCoord(){**

**System.out.println ("Coordonnees : " + x + " " + y) ;**

**}**

**private int x, y ;**

**}**

Create a **PointName** class, derived from **Point**, allowing you to handle points defined by their coordinates (**integers**) and a name (**character**). The following methods will be provided for:

* *constructor* to define the coordinates and the name of a **PointNom** type object,
* *affCoordNom* to display the coordinates and name of a **PointNom** type object.

Write a small program using the **PointName** class..

### Exercise 5:

We have the following class:

**class Point**

**{**

**public Point (int x, int y) {**

**this.x = x ; this.y = y ;**

**}**

**public void affiche(){**

**System.out.println ("Coordonnees : " + x + " " + y) ;**

**}**

**private int x, y ;**

**}**

Create a **PointNom** class, derived from **Point**, allowing you to handle points defined by their coordinates and a name (**character**). The following methods will be provided for:

* *constructor* to define the coordinates and the name of a **PointNom** type object,
* *affiche* to display the coordinates and the name of a **PointNom** type object.