

Pour les méthodes et les attributs

7.4 MODIFICATEUR STATIC

Exemple 7.4.1

Méthodes et modificateur `static`

- Pas besoin d'instance !

```
class SimpleMath{
    static int cube(int x){
        return x*x*x;
    }

    static int abs(int x){
        return x > 0 ? x : -x;
    }
}
...
int a = SimpleMath.cube(3);
int b = SimpleMath.abs(-34);
int c = SimpleMath.cube(SimpleMath.abs(-32));
```

Méthodes et modificateur `static` (2)

- Attention ! Erreur typique des méthodes `static` :
 - ▶ Pas de `this` utilisable
 - ▶ Pourquoi ?



Exemple complet static

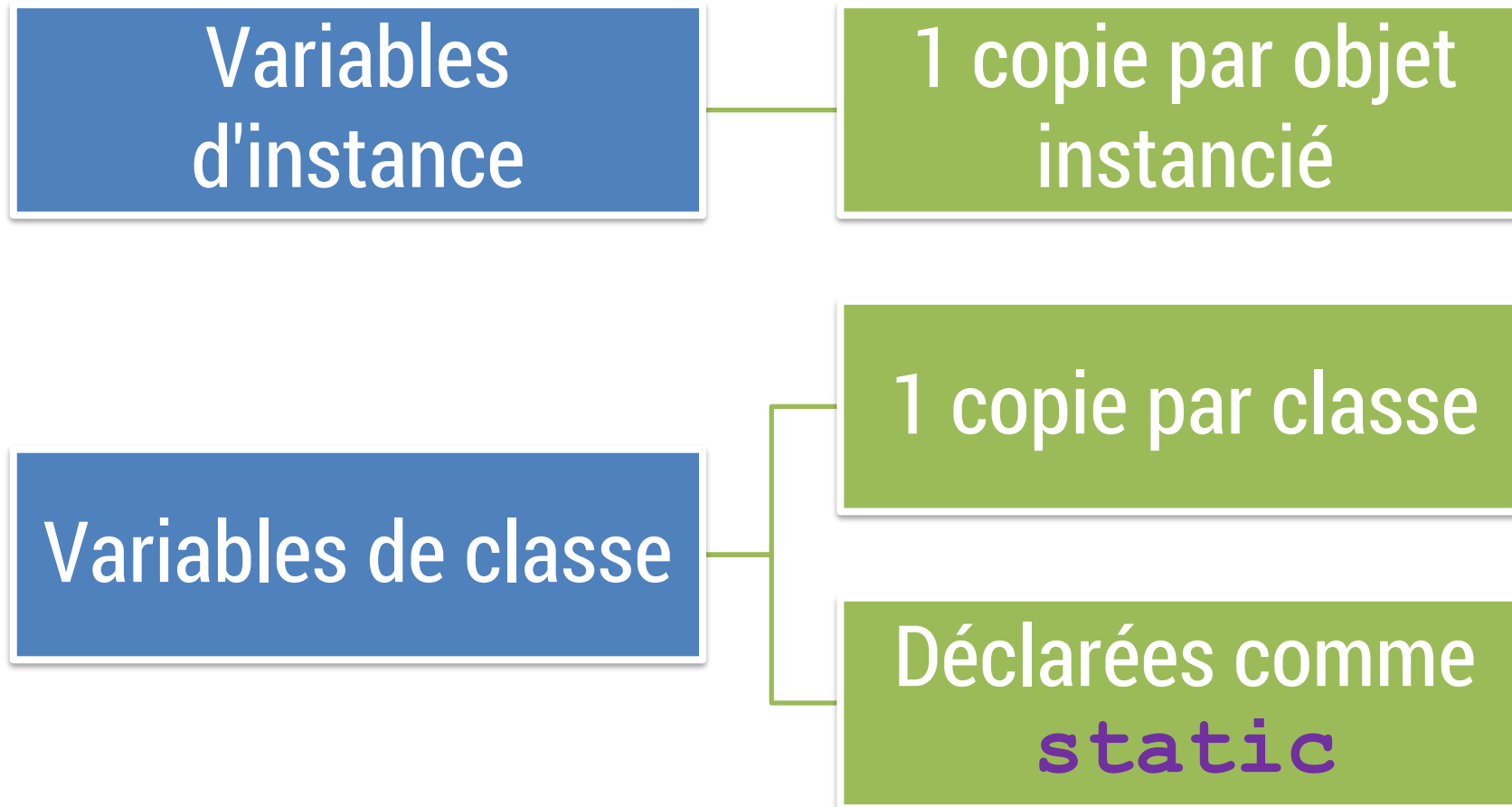
```
class Point2D {
    int x, y;

    public Point2D(int initx, int inity) {
        x = initx;
        y = inity;
    }

    double distanceTo(Point2D other) {
        return Math.sqrt(Math.pow(other.x - this.x, 2.0)
            + Math.pow(other.y - this.y, 2.0));
    }

    static double distanceBetween(Point2D a, Point2D b) {
        return Math.sqrt(Math.pow(b.x - a.x, 2.0)
            + Math.pow(b.y - a.y, 2.0));
    }
}
...
Point2D a, b; a = new Point2D(2, 1); b = new Point2D(3, 2);
double d1 = a.distanceTo(b);
double d2 = Point2D.distanceBetween(a, b);
```

Variables et modificateur **static**

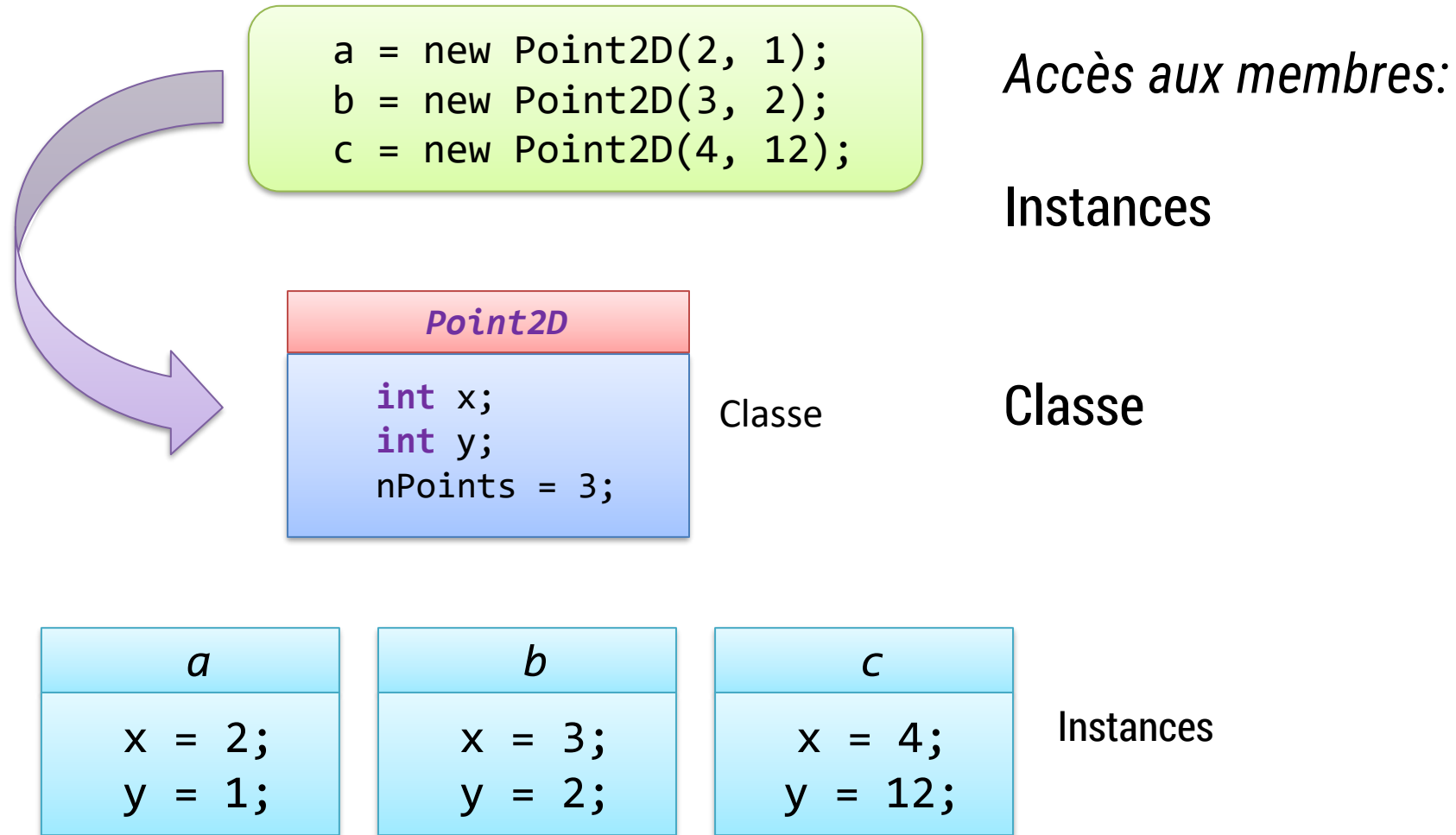


Exemple variable **static**

```
class Point2D{
    static int nPoints= 0;
    int x, int y;

    public Point2D(int initx, int inity){
        nPoints++;
        x = initx; y = inity;
        System.out.println("- Creation of a point: " + nPoints);
    }
}
...
Point2D a, b, c;
a = new Point2D(2, 1);
b = new Point2D(3, 2);
c = new Point2D(4, 12);
```

Exemple *static* (2)



En pratique static et main

```
class Foo {  
    int a;  
  
    Foo() {  
        // Here no longer static, please implement here  
    }  
  
    public static void main(String args[]) {  
        Foo f = new Foo();  
    }  
}
```