



21002  
FINAL

Vincent Guigue



Pour sécuriser le code, interdisons les modifications de certaines valeurs (notamment les constantes).

*Exemple* : `Math.PI`, sécurisation = impossibilité de modifier

*Note* : une constante est indépendante des instances ⇒ `static`

*Usage* : une constante est définie **en majuscule**

Pour sécuriser le code, interdisons les modifications de certaines valeurs (notamment les constantes).

*Exemple* : `Math.PI`, sécurisation = impossibilité de modifier

*Note* : une constante est indépendante des instances ⇒ `static`

*Usage* : une constante est définie **en majuscule**

```
1 public class MaClasse{  
2     public final static int MACONSTANTE = 10;  
3     ...  
}
```

Usage:

- Constantes *universelles* (`Color.RED`, `Color.YELLOW`, `Math.PI`, `Double.POSITIVE_INFINITY`...)
- Typologie (type de codage d'un pixel, organisation du `BorderLayout`)...
- Bornes algorithmiques (`NB_ITER_MAX`, `TAILLE_MAX`...)

Idee : protéger ses objets... Et ses programmes

Initialiser les valeurs des attributs sans pouvoir les modifier ensuite

*Exemple* : String

```
1 public class Point{
2     public final double x,y;
3     public Point(double x, double y){
4         this.x = x; this.y = y;
5     }
6     // interdiction de modifier x, y dans la suite
7     // (et chez le client)
8 }
```

- Interdiction de modifier x et y dans les méthodes (pas de setter, pas de translation...)
- Modification d'un Point = création d'une nouvelle instance
- Possibilité de laisser les attributs public... Puisque non modifiable
- Sécurité lorsqu'un objet est passé en argument de méthode

- Méthode final : qui ne peut pas être redéfinie dans les classes filles
- Classe final : qui ne peut pas être étendue (String, Integer, Double...)

```
1 public class Point{
2     public final double getX(){ ... }
3 }
4
5 public class PointNomme extends Point{
6     ...
7     // Compilation impossible
8     public double getX(){ ... }
9 }
```

- Méthode final : qui ne peut pas être redéfinie dans les classes filles
- Classe final : qui ne peut pas être étendue (String, Integer, Double...)

```
1 public class Point{
2     public final double getX(){ ... }
3 }
4
5 public class PointNomme extends Point{
6     ...
7     // Compilation impossible
8     public double getX(){ ... }
9 }
```

```
1 public final class Point{
2     ...
3 }
4
5 // Compilation impossible
6 public class PointNomme extends Point{
7 }
```