

clone()

Cloneable

La classe implémente l'interface Cloneable (c'est une interface de *tag*).

```
public class MyObject implements Cloneable {  
    //...  
}
```

Overriding

Réécriture de la méthode clone
La méthode devient **public**
La méthode ne retourne plus Object mais "MyObject"

Gestion de l'exception CloneNotSupportedException

```
import java.lang.CloneNotSupportedException ;  
  
public class MyObject implements Cloneable {  
  
    public MyObject clone() {  
        MyObject mo = null;  
        try {  
            mo = (MyObject) super.clone() ;  
        } catch (CloneNotSupportedException ce) {  
            // Traiter l'exception  
            ce.printStackTrace();  
        }  
        // Si j'ai des attributs mutables, je les clone ici  
        mo.oo = oo.clone();  
        return mo;  
    }  
}
```

Shallow copy (Copie superficielle)

Cette méthode permet de faire une **copie superficielle** de l'objet
Convient pour des attributs

- de type **primitif** et/ou
- des attributs **immuables** (immuables)

... ou pour une copie superficielle.

Deep copy (Copie profonde)

Que faire lorsque l'on veut faire une copie en profondeur d'un objet ?
Après le *clone*, prévoir une copie (via la méthode *clone*) de chaque attribut mutable de l'objet

- Attributs de type primitif, *inutile*
- Attributs de type référence *cloneable*: faire un appel à la méthode clone
- Attributs de type références **non cloneable**: revoir ces objets

```
import java.lang.CloneNotSupportedException ;  
  
public class MyObject implements Cloneable {  
    private int i ;  
    private OtherObject oo;  
  
    public MyObject clone() {  
        MyObject mo = null;  
        try {  
            mo = (MyObject) super.clone() ;  
        } catch (CloneNotSupportedException ce) {  
            // Traiter l'exception  
            ce.printStackTrace();  
        }  
        // Si j'ai des attributs mutables, je les clone ici  
        mo.oo = oo.clone();  
        return mo;  
    }  
}
```

Arrays (Cas particulier des tableaux)

Cas particulier des tableaux (arrays)

Un tableau est cloneable

```
import java.lang.CloneNotSupportedException ;  
  
public class MyObject implements Cloneable {  
    private int i ;  
    private OtherObject oo;  
    private OtherObject[] os;  
  
    public MyObject clone() {  
        MyObject mo = null;  
        try {  
            mo = (MyObject) super.clone() ;  
        } catch (CloneNotSupportedException ce) {  
            // Traiter l'exception  
            ce.printStackTrace();  
        }  
        // Si j'ai des attributs mutables, je les clone ici  
        mo.oo = oo.clone();  
        mo.os = os.clone();  
        return mo;  
    }  
}
```

Dans le cas d'un tableau à plusieurs dimensions (un tableau de tableau) seule la **première dimension**(le premier tableau) est cloné ...

Cas particulier des tableaux ... de tableaux

```
import java.lang.CloneNotSupportedException ;  
  
public class MyObject implements Cloneable {  
    private int i ;  
    private String s ;  
    private OtherObject oo;  
    private OtherObject[] os;  
    private OtherObject[][] oss;  
  
    public MyObject clone() {  
        MyObject mo = null;  
        try {  
            mo = (MyObject) super.clone() ;  
        } catch (CloneNotSupportedException ce) {  
            // Traiter l'exception  
            ce.printStackTrace();  
        }  
        // Si j'ai des attributs mutables, je les clone ici  
        mo.oo = oo.clone();  
        mo.os = os.clone();  
        mo.oss = oss.clone();  
        for (int i=0; i < mo.oss.length; i++) {  
            mo.oss[i] = oss[i].clone();  
        }  
        return mo;  
    }  
}
```