

surcharge

- On peut également surcharger une méthode. Cela peut dire qu'on peut définir, dans la même classe plusieurs méthodes qui ont le même nom et des signatures différentes;
- La signature d'une méthode désigne la liste des arguments avec leurs types.
- Un constructeur peut appeler un autre constructeur de la même classe en utilisant le mot `this()`.

Classe de type final

- Une classe de type final est une classesqui ne peut pas être dérivée.
- Autrement dit, on ne peut pas hériter d'une classe final.
- La classe Voiture peut être déclarée final en écrivant:

```
package metier;  
public final class Voiture implements MonInterface {  
    private String nom;  
    private float prix;  
    public Voiture(String nom, float prix) {  
        this.nom = nom;  
        this.prix = prix;  
    }  
    @Override  
    public void afficher() {  
        System.out.println("Nom :"+nom+" Prix:"+prix);  
    }  
}
```

Variables et méthodes final

- Une variable final est une variable dont la valeur ne peut pas changer. Autrement dit, c'est une constante,
exemple : `private final double pi=3.14;`
- Une méthodes final est une méthode qui ne peut pas être redéfinie dans les classes dérivées.
- La méthode afficher de la classe Voiture ne peut pas être redéfinie dans les classes dérivées car elle est déclarée final :

```
package metier;  
public final class Voiture {  
    private String nom;  
    private float prix;  
  
    public Voiture(String nom, float prix) {  
        this.nom = nom;  
        this.prix = prix;  
    }  
    public final void afficher() {  
        System.out.println("Nom :"+nom+" Prix:"+prix);  
    }  
}
```

Classe abstraite

- Une classe abstraite est une classe non instanciable.
- Une classe abstraite peut servir comme superclasse.

```
Document.java x
1 package metier;
2
3 @ public abstract class Document { 2 usages 2 inheritors
4     private int numEnreg; 2 usages
5     private String titre; 2 usages
6     private static int count=0; 2 usages
7
8     public Document(String titre){ 2 usages
9         //ici this est obligatoire: nom du paramètre est identique au nom de l'attribut
10        this.numEnreg=count;
11        this.titre=titre;
12        count++;
13    }
14
15 @> public String toString() { return ("Numéro d'enregistrement: "+numEnreg+" Titre: "+titre); }
18 }
19
```

Les méthodes abstraites

- Une méthode abstraite peut être déclarée à l'intérieur d'une classe abstraite.
- Une méthode abstraite est une méthode qui n'a pas de définition.
- Une méthode abstraite est une méthode qui doit être redéfinie dans les classes dérivées.

```
Document.java x
1 package metier;
2
3 @ public abstract class Document { 2 usages 2 inheritors 2 related problems
4     private int numEnreg; 1 usage
5     private String titre; 1 usage
6     private static int count=0; 2 usages
7
8     public Document(String titre){ 2 usages
9         //ici this est obligatoire: nom du paramètre est identique au nom de l'attribut
10        this.numEnreg=count;
11        this.titre=titre;
12        count++;
13    }
14    public abstract void afficher(); no usages
15
16
17 }
```

Les interfaces

- Une interface est une sorte de classe abstraite qui ne contient que des méthodes abstraites.
- Dans java une classe hérite d'une seule classe et peut hériter (implémenter) en même temps de plusieurs interfaces.
- Une interface peut hériter de plusieurs interfaces. Exemple d'interface:

```
package metier;  
public interface MonInterface {  
    void afficher();  
}
```

- Exemple d'une classe qui implémente l'interface « MonInterface » :

```
package metier;  
public class Voiture implements MonInterface {  
    private String nom;  
    private float prix;  
    public Voiture(String nom, float prix) {  
        this.nom = nom;  
        this.prix = prix;  
    }  
    @Override  
    public void afficher() {  
        System.out.println("Nom :"+nom+" Prix:"+prix);  
    }  
}
```