

Chap 5: Classes et Méthodes Utilitaires

Mehdi Ellouze

Agenda

- Introduction
- Méthode Statiques
- Attributs Statiques
- Méthodes et Attributs de la classes Math
- Méthodes de la classes String

Introduction

- Pour développer des méthodes nous avons besoin de quelques fonctions prédéfinies: valeur absolue, racine carré, conversion, etc
- Il existe en Java des classes qui proposent de tels services
- Pour invoquer ces méthodes nous n'avons pas besoin d'instancier des classes
- Les méthodes sont invoqué directement à partir des classes, on les appelle des méthodes de classes ou des méthodes statiques

Méthode Statiques

- Parfois le comportement d'une méthode ne dépend de l'instance d'objet créée
- Quelque soit les attributs de la classe la sortie de la méthode ne change pas
- Ces méthodes on les appels des méthodes statique et on les déclare comme suit

```
public class Calculatrice {  
    public static int somme( int a,int b)  
    {  
        return a+b;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        int c= Calculatrice.somme(3, 2);  
  
        System.out.print(c);  
  
    }  
}
```

Invoker la méthode sur la classe directement!!!!

Attributs Statiques

- Parfois il existe des attributs communs à toutes les instances
- Exemple:
 - Un paramètre de connexion à la base de données les objets de la classe et exister indépendamment à tout objet de la classe
 - Adresse IP d'un serveur
- Ces Attributs on les déclare static et on les appelle attributs de classe
- L'accès à ces attributs ne se fait pas via les instances mais via la classe directement
- La modification de la valeur de ces attributs dans une instance entrainera sa modification dans toutes les instances

Attributs Statiques

```
public class Test {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Calculatrice C1=new Calculatrice();  
        double s= C1.calculerSurface(2);  
        System.out.println(s);  
  
        Calculatrice.PI=3;  
  
        Calculatrice C2=new Calculatrice();  
        s= C2.calculerSurface(2);  
        System.out.println(s);  
    }  
}
```

```
public class Calculatrice {  
  
    public static double PI= 3.14;  
  
    public static int somme( int a,int b)  
    {  
        return a+b;  
    }  
  
    public double calculerSurface(double r)  
    {  
        return PI*Math.pow(r,2);  
    }  
  
}
```

Méthode et Attributs de la classe Math

- **Constantes**

- **E (double):** 2.7182818284590452354
- **PI (double):** 3.14159265358979323846

- **Méthodes**

- **public static number abs(number):** retourne la valeur absolue
- **public static number max(number, number):** retourne le maximum entre 2 nombres
- **public static number min(number, number):** retourne le minimum entre 2 nombres
- **public static number pow(number, number):** retourne le minimum entre 2 nombres

Méthodes et Constantes de la classe Math

- **Méthodes d'arondi**
 - **public static double ceil(double a):** retourne le double le plus grand
 - **public static double floor(double a):** retourne le double le plus grand
 - **public static long round(double a):** retourne l'entier long le plus proche
 - **public static int round(float a):** retourne l'entier le plus proche
- **cosinus, sinus, ...:**
 - **public static double cos(double a)**
 - **public static double sin(double a)**
 - **public static double tan(double a)**
 - **public static double acos(double a)**
 - **public static double asin(double a)**
 - **public static double atan(double a)**

Méthode et Attributs de la classe Math

- **Autres méthodes**

- **public static double log(double a):** retourne le log d'un double
- **public static double exp(double a):** retourne l'exponentiel
- **public static double pow(double a, double b):** retourne a exposant b
- **public static double random():** retourne un double aléatoire entre 0.0 et 1.0
- **public static double sqrt(double a):** retourne la racine carré arrondie d'un double

Méthodes de la classe String

- `public static String valueOf(boolean b)` : Retourne la représentation en chaîne du booléen
- `public static String valueOf(char c)`: Retourne la représentation en chaîne du caractère
- `public static String valueOf(char[] data)` : Retourne la représentation en chaîne du tableau de caractères
- `public static String valueOf(char[] data, int offset, int count)`: Retourne la représentation en chaîne du tableau de caractères (partie)
- `public static String valueOf(double d)` : Retourne la représentation en chaîne du double
- `public static String valueOf(float f)`: Retourne la représentation en chaîne du float
- `public static String valueOf(int i)` : Retourne la représentation en chaîne du int