Cours 11

Variables de classe
Méthodes de classe

Variables de classe

- Une variable de classe décrit une propriété de la classe toute entière et non une propriété d'une instance particulière
  - Une variable de classe est une variable commune à toutes les instances de la classe
  - Elle est partagée par toutes les instances de la classe
  - Elle n'est pas liée à une instance particulière mais à la classe elle-même

Variable de classe

- Une variable de classe existe en un seul exemplaire en mémoire
  - a un seul espace mémoire réservé, une fois la classe chargée
- Alors que chaque instance a ses propres variables d'instance
  - Une variable d'instance = une zone mémoire spécifique pour chaque instance
  - Créée dès qu'on fait new...

Déclaration d'une variable de classe

- **Pour déclarer une variable de classe quand on définit une classe**
  - faire précéder la déclaration de la variable par le mot-clé **static**

```
public class Math {
    public static final double PI;
    ...
```

  - Par convention, une variable de classe est tout en majuscule

Utilisation d'une variable de classe

- **Pour utiliser une variable de classe dans un programme client (si elle est déclarée en public)**, on la fait précéder du nom de la classe qui la contient

```
public void static main(String[] args)  {

    System.out.println ("valeur de pi : "+  Math.PI)

}
```

Un exemple de la bibliothèque package java.awt

```
public class Color    {

public static final Color ORANGE;
public static final Color RED;
public static final Color BLACK;
public static final Color YELLOW;
....
```

accès à la couleur orange :

Analyse d'un exemple déjà vu System.out

```
public class System    {

public static final InputStream in ;
public static final PrintStream out ;
public static final PrintStream err ;
....
}
```

Méthode de classe

- Une méthode de classe est une méthode :
  - **commune** à toutes les instances de la classe
  - que l'on adresse à la classe et non à une instance de la classe
  - qui n'est liée à aucune instance particulière

Exemple déjà rencontré
Conversion d'une chaîne de caractères en entier

```
String s = new String("234" );
int n = Integer.parseInt(s) ;
```

- parseInt est une **méthode de classe** définie dans la classe Integer
- Sa signature : public static int parseInt(String s)
  - Elle retourne l'entier correspondant à la chaîne de caractères référencée par *s*
- Cette méthode n'est pas liée à une instance particulière d'Integer. Elle rend un service lié à la classe Integer

Déclaration et utilisation d'une méthode de classe

- Pour **déclarer** une méthode de classe
  - faire précéder la déclaration de la méthode par le mot-clé **static**

```
public static int parseInt(String s);
```

- Pour **utiliser** une méthode de classe, on l'adresse à la classe qui la contient

```
int i = Integer.parseInt("234") ;
```

Nom de la classe

Nom de la méthode de classe

A savoir (1)

- Dans les classes fournies par Java, il y a de nombreuses méthodes de classe
- ==> si le mot clé static est présent dans la signature d'une méthode :
  -- c'est une méthode de classe
  -- pour l'utiliser, il faut adresser la méthode **à la classe**

A savoir (2)

- Certaines classes ne possèdent même que des méthodes de classe
- On parle de **classes “outils”**
- **Elles ne possèdent pas de constructeur)**
  - Exemple : la classe Math ne possède que des
    - constantes de classes
    - des méthodes de classe rendant les services des maths
      - random, floor, sqrt, sin...
      - float f=Math.sqrt(4);

Le bloc static

un bloc static est défini par:

```
static {
      ….
      }
```

Un tel bloc est exécuté une seule fois lors du chargement de la classe avant sa première utilisation

Il peut servir à initialiser les variables de classes de la classe

Le mot-clé final pour une classe

- Devant la définition d'une classe :
  - Signifie « ne peut pas être » dérivée
  - Si une classe est déclarée final, on ne peut pas en définir une sous-classe
    - Exemple : les classes System, String
      - public final class System
      - public class MonSystem extends System

        Générera l'erreur :
        ⇨ *cannot inherit from final java.lang.System*

Le mot-clé final pour une variable statique

- Devant une variable de classe :
  - Signifie « ne peut pas être » modifiée
  - Donc sert à définir une constante
    - Exemple : public static final double Math.PI ;

```
public class Test {
   public void static main(String[] args)   {
                    Math.PI = 8;
}
```

Générera une erreur : ⇨ *cannot assign a value to final variable PI*