

## **Chapitre 3**

# **Communiquer une information**

Dans un programme, on est souvent amené à :

- **Saisir des données**

= opérations d'entrée (in)

- **Afficher des résultats**

= opérations de sortie (out)

Nous allons voir les fonctionnalités proposées par Java pour gérer les opérations d'entrée-sortie avec la bibliothèque *System*.

# 1. La bibliothèque *System*

Elle contient des fonctionnalités permettant de réaliser des opérations d'entrées-sorties.

Par exemple, l'affichage de valeurs ou de texte est réalisé par l'utilisation de `System.out.print()`

Exemple :

```
System.out.print("Un peu de texte");
```

## Pourquoi un nom si compliqué ?

- Java est composé de bibliothèques contenant des outils de bases pour la programmation.
- Les noms de ces bibliothèques renseignent l'utilisateur sur leurs fonctionnalités.
  - *Math* contient les fonctions mathématiques
  - *System* contient les éléments de gestion des commandes bas niveaux (clavier, écran, etc.)

La notation point (.) est une écriture courante en programmation objet. On retrouvera cette notation dans la suite de ce cours.

Nous trouvons **System.in**, **System.out** et **System.err**.

**System.err** permet d'afficher les erreurs éventuelles d'un programme sur la sortie standard des erreurs.

## 2. L'affichage de données

Le message à afficher est placé comme **paramètre** de la fonction `system.out.print()` (c-à-d à l'intérieur des parenthèses).

Plusieurs types d'affichages existent :

## 2.1 Affichage de la valeur d'une variable

```
int valeur = 22;  
System.out.print(valeur) ;
```

A l'écran s'affiche :



22

## 2.2 Affichage de texte et de variables

Il est possible d'ajouter du texte entre guillemets avant ou après l'affichage d'une variable :

```
System.out.print("Le montant s'élève à: " + valeur);
```

OU :

```
System.out.print(valeur + " est le montant total");
```

Concaténation de textes et de variables avec le signe '+'.



## 2.3 Affichage de plusieurs variables

```
int var1=5, var2=10;
```

```
System.out.print("la premiere variable vaut " + var1  
                + " et la seconde vaut " + var2 );
```

Affiche :

```
la premiere variable vaut 5 et la seconde vaut 10
```

## 2.4 Affichage de la valeur d'une expression arithmétique

On peut afficher directement le résultat d'une expression arithmétique.

```
int a=10, b=5;
```

```
System.out.print(a + "fois" + b +  
                  "est égal à" + (a*b));
```

Affiche :

```
10 fois 5 est égal à 50
```

## 2.5 Affichage d'un texte.

```
System.out.print ("Il fait beau et chaud");
```

→ permet d'afficher tout le texte en paramètre sur la même ligne. Équivalent :

```
System.out.print("Il fait beau");  
System.out.print(" et chaud");
```

→ permet d'afficher :

```
Il fait beau et chaud
```

```
System.out.println("Il fait beau");
```

```
System.out.println("et chaud");
```

→ permet d'afficher les résultats sur deux lignes successives :

```
Il fait beau  
et chaud
```

### 3. La saisie de données

Pour saisir une valeur au clavier dans un environnement non graphique, on peut utiliser la classe **Scanner** du package **java.util**

Cette classe dispose de méthodes de lecture pour chaque type simple :

```
Scanner lectureClavier = new Scanner(System.in);  
byte octet = lectureClavier.nextByte();  
int entier = lectureClavier.nextInt();  
float réel = lectureClavier.nextFloat();  
String caractère = lectureClavier.next().charAt(0);  
String mot = lectureClavier.next();  
String phrase = lectureClavier.nextLine();
```

## Exemple

```
import java.util.*;
public class Saisie
{
    public static void main (String [] arg)
    {
        int    entier;
        String phrase;
        Scanner lectureClavier = new Scanner(System.in) ;

        System.out.println("Entrez un entier : ") ;
        entier = lectureClavier.nextInt() ;

        lectureClavier.nextLine() ;
        System.out.println("Entrez une phrase : ") ;
        phrase = lectureClavier.nextLine() ;
    }
}
```

# **Les commentaires**

Il est possible (et même vivement recommandé!) de commenter le code source.

- Documente le programme
- Améliore la lisibilité
- Facilite la maintenance

Les commentaires sont ignorés par le compilateur.

## Commentaire de style C

Commence par la paire de symboles `/*` et se termine par `*/`

```
/* Voici un commentaire sur plusieurs lignes.  
   Très pratique pour désactiver des portions  
   de code.
```

```
   int valeur1;  
   int valeur2;  
*/  
   int valeur3;
```

Permet de faire des commentaires sur plusieurs lignes.



## Commentaire de style C++

Commence par la paire de symboles `//` et se termine à la fin de la ligne.

```
int num;    // Le numérateur  
int den;    // Le dénominateur
```

Utile pour annoter les instructions.