

La Programmation Orientée Objet (POO) avec Java

Module 1 – Les bases du Java



Objectifs

- Maîtriser les bases du langage Java avant d'attaquer la Programmation Orientée Objet

Le fameux HelloWorld !

```
package fr.eni.ecole.basesDuLangage;
```

```
public class HelloWorld {
```

```
    public static void main(String[] args) {  
        System.out.println("Hello World !");  
    }
```

```
}
```

Affichage d'un message
sur la console

L'affichage de messages sur la console

- Sortie standard
 - `System.out.println()`
 - `System.out.print()`
- Sortie des messages d'erreur
 - `System.err.println()`
 - `System.err.print()`

Le formatage des messages

- Fonctions permettant le formatage
 - `System.out.printf()`
 - `System.err.printf()`
 - `String.format()`
- Paramètres
 - Chaîne de caractères contenant des « trous » indiqués par des indicateurs
 - Les valeurs à utiliser pour compléter les « trous »

Le formatage des messages

- Principaux indicateurs

- %d : nombre entier en base 10
- %f : nombre à virgule
- %s : chaîne de caractères
- %n : retour à la ligne

- Exemple

```
System.out.printf("%s a été créée en %d et est située à %.1f km du centre de Nantes.%n",  
                  "ENI École Informatique", 1981, 6.07);
```

Plus de détails dans la Javadoc :
<https://docs.oracle.com/javase/10/docs/api/java/util/Formatter.html#syntax>



ENI École Informatique a été créée en 1981
et est située à 6,1 km du centre de Nantes.

Les variables et constantes

```
public class Taxe {
```

```
    public static final double TVA = 20.0/100;
```

Déclaration
d'une constante

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

```
        double prixHT = 17.85;
```

Déclaration d'une variable

```
        System.out.printf("Prix TTC de l'article : %.2f€\n", prixHT*(1+TVA));
```

```
    }  
}
```



Prix TTC de l'article : 21,42€

Les tableaux

- Déclaration

```
float[] valeurs = new float[100];
```

- Écriture dans une case

```
valeurs[7] = 1792;
```

- Lecture d'une valeur

```
System.out.println(valeurs[7])
```



Les indices des éléments d'un tableau commencent à zéro

un tableau est un type référence.
il est donc instancié sur le tas

La récupération des saisies de l'utilisateur

- Utilisation de Scanner

- `nextLine()` : Lecture d'une chaîne de caractères
- `nextInt()` : Lecture d'un nombre entier
- `nextFloat()` : Lecture d'un nombre à virgule
- ...

- Exemple :

```
Scanner s = new Scanner(System.in);
System.out.println("Quel est votre prénom ?");
// saisir la chaîne de caractères entrée par l'utilisateur
String prenom = s.nextLine();
s.close();
```



Les fonctions `next*()` ne consomment que les caractères nécessaires, il faut souvent les faire suivre d'un appel à `nextLine()`

Les conditionnelles

La condition doit être une valeur booléenne

- L'instruction if (simple)

```
if(prenom.length()==0)
    System.err.println("Le prénom est obligatoire !");
```

- L'instruction if (double)

```
if(prenom.length()==0)
    System.err.println("Le prénom est obligatoire !");
else
    System.out.printf("Bonjour %s !%n", prenom);
```

Accolades si plusieurs instructions

```
if(prenom.length()==0) {
    System.err.println("Le prénom est obligatoire !");
    System.exit(1);
}
```

Les conditionnelles

- L'instruction switch

```
switch (aJeter) {  
    case "papier":  
    case "carton":  
    case "boite de conserve":  
        System.out.println("Recyclage");  
        break; <<  
    case "végétaux":  
    case "épluchures":  
        System.out.println("Composte");  
        break; <<  
    default:  
        System.out.println("Poubelle");  
        break;  
}
```

Les types supportés sont les types de base, String et les énumérations

Ne pas oublier les **break** sinon les instructions du cas suivant sont exécutées

Les boucles

- For

```
float[] valeurs = {10.5f, 14.2f, 15,78f, 20, 19.2f};
for (int i = 0; i < valeurs.length; i++) {
    System.out.format("%.2f%n", valeurs[i]);
}
```

- Foreach

```
float[] valeurs = {10.5f, 14.2f, 15,78f, 20, 19.2f};
for (float f : valeurs) {
    System.out.format("%.2f%n", f);
}
```

Les boucles

- While

```
Scanner s = new Scanner(System.in);  
float total = 0;  
System.out.println("Saisissez une valeur ou rien pour terminer la saisie");  
String saisie = s.nextLine();  
while (!saisie.isEmpty()) {  
    float valeur = Float.parseFloat(saisie);  
    total += valeur;  
    System.out.println("Saisissez une note ou rien pour terminer la saisie");  
    saisie = s.nextLine();  
}  
System.out.printf("Total des valeurs saisies : %f%n" ", total);
```

Les boucles

- Do while

```
int min = 1;
int max = 10;
System.out.println("Entrez un nombre", message);
Scanner s = new Scanner(System.in);
int val = 0;
boolean ok;
do {
    val = s.nextInt();
    s.nextLine();
    ok = val >= min && val <= max;
    if (!ok)
        System.err.printf("La valeur doit être un entier compris entre %d et %d\nRessaisissez... ",
                           min, max);
} while (!ok);
```

Les énumérations

```
public enum Groupe {
    GARE,
    SERVICE_PUBLIC,
    MAUVE,
    BLEU,
    VIOLET,
    ORANGE,
    ROUGE,
    JAUNE,
    VERT,
    MARINE;
}
```

Parcours des différentes valeurs de l'énumération

```
for (Groupe groupe : Groupe.values()) {
    ...
}
```

«enumeration» Groupe
GARE
SERVICE_PUBLIC
MAUVE
BLEU
VIOLET
ORANGE
ROUGE
JAUNE
VERT
MARINE

utilisation d'une des valeur de l'énumération

```
Monopoly.plateau.ajouter(new Terrain("Rue Lecourbe", 60, c2, Groupe.MAUVE, 50));
```

Les fonctions et procédures

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

```
private static void salutation() {  
    String prenom = saisirPrenom();  
    afficherSalutation(prenom);  
}
```

```
private static void afficherSalutation(String prenom) {  
    System.out.println("Bonjour "+prenom+" !");  
}
```

```
private static String saisirPrenom() {  
    Scanner s = new Scanner(System.in);  
    System.out.println("Quel est votre prénom ?");  
    String prenom = s.nextLine();  
    s.close();  
    return prenom;  
}
```


Les exceptions



```
public static int saisie(String message, int min, int max) {  
    System.out.printf("%s ", message);  
    int val = 0; boolean ok;  
    do {  
        try {  
            val = s.nextInt();  
            ok = val >= min && val <= max;  
        } catch (InputMismatchException e) {  
            ok = false;  
        } finally {  
            s.nextLine();  
        }  
        if (!ok)  
            System.err.printf("Saisissez un entier compris entre %d et %d\n", min, max);  
    } while (!ok);  
    return val;  
}
```

Les exceptions

Le mot clef **throws** indique que cette méthode est susceptible de générer une exception

```
private static void verifNbFaces(int nbFaces) throws Exception {  
    if (nbFaces <= 1)  
        throw new Exception("Un dé doit avoir au moins deux faces");  
}
```

Le mot clef **throw** lève une exception

L'écriture de données dans un fichier

```
double[] valeurs = { 17.99, 22.41, 6.0, 39.02, 78.45, 44.7854 };
try {
    FileWriter f = null;
    try {
        f = new FileWriter("./valeurs.txt");
        for (int i = 0; i < valeurs.length; i++)
            f.write(String.format("%.2f%n", valeurs[i]));
        System.out.println("Enregistrement terminé avec succès");
    } finally {
        if (f != null)
            f.close();
    }
} catch (IOException e) {
    System.err.println("Écriture impossible");
}
```

L'écriture de données dans un fichier avec gestion automatique des ressources

```
double[] valeurs = { 17.99, 22.41, 6.0, 39.02, 78.45, 44.7854 };
try (FileWriter f = new FileWriter("./valeurs.txt")) {
    for (int i = 0; i < valeurs.length; i++)
        f.write(String.format("%.2f%n", valeurs[i]));
    System.out.println("Enregistrement terminé avec succès");
} catch (IOException e) {
    System.err.println("Écriture impossible");
}
```

Lecture de données d'un fichier

```
try {
    FileInputStream fichier = null;
    Scanner s = null;
    try {
        fichier = new FileInputStream("./valeurs.txt");
        s = new Scanner(fichier);
        while (s.hasNextLine())
            System.out.println(s.nextLine());
        System.out.println("Fichier lu intégralement avec succès");
    } finally {
        if (s != null)
            s.close();
        if (fichier != null)
            fichier.close();
    }
} catch (IOException e) {
    System.out.println("Lecture impossible");
}
```

Lecture de données d'un fichier avec gestion automatique des ressources

```
try (FileInputStream fichier = new FileInputStream("./valeurs.txt");
    Scanner s = new Scanner(fichier)) {
    while (s.hasNextLine())
        System.out.println(s.nextLine());
    System.out.println("Fichier lu intégralement avec succès");
} catch (IOException e) {
    System.out.println("Lecture impossible");
}
```

2 ressources sont générées
automatiquement

TP