

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



Lancer Eclipse

- Lancer Eclipse
- Fermer la welcome page
- Aller dans windows -> preferences
- Aller dans General -> workspace pour mettre le text encoding en ISO
- Faites "Apply and Close"
- Create a Java project
- Nommez le projet (EniCoursPOO)
- Décocher "create module.info"
- Cliquez sur "finish"

Le fameux HelloWorld !

```
package fr.eni.ecole.module1;
```

```
public class HelloWorld {
```

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

La notion de package ?

Le fichier doit avoir le même nom que la classe



Affichage d'un
message
sur la console

Exercice :

- Créer un nouveau package (clic droit sur SRC)
- Nommer le module "module1"
- Clic droit sur le package -> new -> Class
- Rentrer un nom de classe (HelloWorld) et cocher la case pour créer un main()
- Afficher un « Hello World »

L'affichage de messages sur la console

- Sortie standard
 - `System.out.println()` : *retour à la ligne*
 - `System.out.print()` : *sans retour à la ligne*
- Sortie des messages d'erreur
 - `System.err.println()`
 - `System.err.print()`

Les variables et constantes

```
public class Taxe {  
    public static final double TVA = 20.0/100;  
    public static void main(String[] args) {  
        double prixHT = 17.85;  
        System.out.println("Prix TTC de l'article : " + prixHT*(1+TVA));  
    }  
}
```

Déclaration
d'une constante

Déclaration d'une
variable

Exercice :

Réaliser l'exemple avec trois chiffres après la virgule



Prix TTC de l'article : 21,42

Les tableaux

- Déclaration

```
int[] valeurs = {45,12,22,66,98};
```

- Écriture dans une case

```
valeurs[2] = 1792;
```

- Lecture d'une valeur

```
System.out.println(valeurs[2])
```

Exercice :

Réaliser l'exemple.

Vous devez afficher « 1792 » à la fin.



Les indices des
éléments d'un tableau
commencent à zéro

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();
```

Exercice :

Réaliser l'exemple.

Egalement, je veux que le prénom saisi par l'utilisateur s'affiche à la fin.

Les conditionnelles

- 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.println("Bonjour " + prenom);
```

Accolades si plusieurs instructions

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

Exercice :

Concevoir un programme qui :

- Demande son prénom à l'utilisateur
- Si aucun prénom n'est tapé : affiche un message d'erreur
- Si un prénom est tapé : « Ok, c'est toi prénom » (remplacer 'prénom' par le prénom préalablement saisie)

Les conditionnelles

- L'instruction switch

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

Exercice :

Concevoir un programme qui, à l'aide d'un switch :

- Déclare une variable meteo, égale à « froid »
- Selon la valeur de meteo :
 - Affiche « Sortez les maillots » si meteo est égale à chaud
 - Affiche « Préparez les bonnets » si meteo est égale à froid
 - Sinon, affiche : « Aucune idée du temps qu'il fera »

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.println(valeurs[i]);  
}
```

- Foreach

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

Exercice :

Concevoir un programme qui :

- Déclare un tableau avec 2 valeurs : 11.5 et 12.5
- Afficher les 2 valeurs à l'aide d'un foreach

Les boucles

- While

```
int i = 1;
while (i < 3){
    System.out.println("Compteur : " + i);
    i++;
}
```

Exercice :

- Tester le code

Les boucles

- Do while

```
do {  
  
    instruction  
  
} while (condition);
```

Exercice :

- Faire un code qui demande le prénom de l'utilisateur tant que celui-ci n'en a pas entré un
- Si le prénom est entré, affichez le

Exercice :

- Refaire l'exercice avec un while() simple

Les fonctions et procédures

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

Exercice :

Concevoir un programme qui :

- Déclare une variable « age » égale à 30
- Passe la variable a une fonction privée « vieillir() »

La variable age doit vieillir d'un an après son passage par la fonction. Affichez la à la fin du programme.

Les exceptions

```
// .....  
  
try {  
    val = s.nextInt();  
} catch (Exception e){  
    system.err.println("Veuillez  
    entrer un entier")  
}
```

Exercice :

Concevoir un programme qui :

- Demande à l'utilisateur son age
- Si une phrase est entrée au lieu d'un entier, afficher le message d'erreur ("Nous voulons un entier").
- Sinon, afficher : "Tu as XX ans"
- Faire en sorte que le programme s'arrête s'il affiche un message d'erreur grâce à System.exit(0)

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;
    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");
}finally {
    if (f != null)
        f.close();
}
```

Exercice :

- Tester le code !

Les assertions

Il faut activer les assertions :

- Windows -> preferences
- Taper "Java"
- Sélectionner "Installed JREs"
- Sélectionner la première JRE (la seule normalement) et faire "edit"
- Dans "Default VM arguments", écrire : -ea (avec le tiret)

Les assertions

Une assertion permet de vérifier une condition considérée comme vraie.

Si cette condition est vraie, alors l'assertion sera muette.

Si elle est fausse, alors une erreur sera produite.

Une assertion s'introduit via le mot clé **assert** et elle est suivie de la condition à vérifier et éventuellement d'un message d'erreur.

```
package module1;
```

```
import java.util.Scanner;
```

```
public class Exceptions {
```

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

```
        System.out.println("Quel temps fait-il cet été ?");
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int temperature = sc.nextInt();
```

```
        assert temperature > 0 : "La température ne peut pas être négative";
```

```
        System.out.println("Il fait " + temperature + "degrés");
```

```
        sc.close();
```

```
    }
```

```
}
```

TP

