	Conception Orientée Objet *M1 – Info* S. *Caffiau - année 2015/2016* d'après le TD "Introduction à JAVA" de Devismes et Falcone	

Introductions

Objectifs de la séance : Consolider les connaissances sur (ou découvrir) JAVA qui sera utilisé pour programmer en orienté objet. Illustrer les limitations de la conception fonctionnelle.

Ce sujet est à terminer en 2 séances.
Tous les exercices sont à faire par binôme. Les binômes seront constitués par l'enseignant.
Ces exercices sont à réaliser sans IDE.
Complétez les questions au fur et à mesure, nous y reviendrons dans 2 semaines.

Exercice 1 : Hello World!

Pour écrire un programme Java, il vous faut :
– Un éditeur de texte (commande kwrite, emacs, nedit ou vi. . .)
– La plateforme Java contenant en particulier :
- Le compilateur Java (commande javac)
- La machine virtuelle Java (commande java)

Question 1 : Taper le texte ci-dessous dans un éditeur de texte et enregistrer le dans un fichier nommé Hello.java.

```
/* Hello: affiche "Hello World !*/

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

Question 2 : Compiler Hello.java en tapant la commande javac Hello.java dans un terminal. Ce processus crée un ou plusieurs fichiers de byte-code portant l'extension .class, ici Hello.class. Le compilateur n'affiche normalement aucun message ; s'il y en a, cela signifie qu'il y a des erreurs dans le texte du programme.

Question 3 : Exécuter le programme en tapant la commande java Hello. Ce processus exécute le byte-code dans la machine virtuelle.

Question 4 : Que signifie `/* Hello: affiche...*/` dans le code ? _____
_____

Question 5 : Que signifie le mot-clé `public` dans le code ? _____
_____
Que se passe-t-il si vous le supprimez ?_____

Question 6 : Renommez Hello.java en Hello2.java (commande mv Hello.java Hello2.java sous linux). Modifiez le code pour qu'il affiche "Bonjour à vous!". Exécutez Hello.class. Que se passe-t-il ?_____

Question 7 : Compiler Hello2.java. Que se passe-t-il ?_____
Décrivez, avec vos mots, le problème._____
_____
Corrigez le.

Question 8 : Renommez `main` par `main2`. Compilez de nouveau. Que se passe-t-il ?
Décrivez, avec vos mots, le problème._____
_____
Corrigez le. Qu'est ce que cette fonction (ie méthode en java) "spéciale" ?_____
_____

Exercice 2 : Comparaison d'arguments en ligne de commande

Les arguments de la ligne de commande sont accessibles à partir du tableau en paramètre dans `main(String[] arg)`. Le nombre d'éléments du tableau est donné par `arg.length`. Les arguments de la ligne de commande seront donc dans `arg[0] ... arg[arg.length-1]`.

Question 1 : Écrire un programme qui compare deux arguments entiers passés en paramètre. Par exemple, la commande `java Arg 2 3` affichera « 2 est inferieur à 3 ».

Indication : Pour convertir une chaîne de caractères en valeur entière, on utilise la fonction `Integer.parseInt(s)` qui retourne la valeur entière correspondant à la chaîne s.

Notez les erreurs que vous rencontrez à la compilation :_____
_____
_____
_____

Question 2 : Exécutez `java Arg a 3`. Que se passe-t-il ?_____

_____

Question 3 : Exécutez `java Arg 2 3 7`. Que se passe-t-il ? et pourquoi ?_____

_____

Exercice 3 : Première boucle

Écrire un programme `Occ.java` lisant tous les arguments passés sur la ligne de commande et recherchant si le premier argument existe dans la suite de la ligne.

Par exemple `java Occ aaaa bbb AA cc` affichera : Non trouve

et `java Occ aaaa bbb AA aaaa cc` affichera : Trouve.

Indication : Pour comparer deux chaines de caractères s1 et s2 (ou deux objets en général, cf. plus tard dans le TD), il y a deux possibilités utiliser la méthode `.equals()` ou utiliser l'opérateur **==** :

– `s1.equals(s2)` renvoie true (vrai) si et seulement si s1 et s2 sont deux chaînes de caractères identiques au sens usuels. Les chaînes de caractères sont comparées structuellement.

– `s1 == s2` renvoie true (vrai) si et seulement si s1 et s2 sont physiquement (dans la mémoire) deux chaînes de caractères identiques. Les addresses des chaînes de caractères sont comparées.

Notez les erreurs que vous rencontrez à la compilation :_____

_____

_____

_____

Exercice 4 : Entrée/sortie en ligne de commande

Question 1 : Ecrivez, compilez et exécutez le code suivant :

```
1   import java.io.*;
2   //Saisie un entier au clavier
3   public class Saisie {
4       public static void main (String [] arg) throws IOException {
5            BufferedReader inr = new BufferedReader (new InputStreamReader(System.in));
6            System.out.println("Taper une valeur entiere au clavier");
7            int val = Integer.parseInt(inr.readLine());
8            int somme= 0;
9            while (val != -1){
10                  somme += val;
11                  System.out.println("Taper une valeur entiere au clavier");
12                  val = Integer.parseInt(inr.readLine());
13           }
14           System.out.println("La somme des nombre que vous avez saisis est " + somme);
15      }
16  }
```

Question 2 : Qu'est ce que le code de la ligne 1 ?_____

_____

Question 3 : Qu'est ce que le code de la ligne 2 ?_____

Question 4 : Qu'est ce que le code "`throws IOException`" de la ligne 4 ? A quoi cela sert-il ?

Question 5 : Qu'est ce que veut dire le code de la ligne 5 ?

Question 6 : En vous basant sur le programme précédent, écrire un programme qui effectue la somme des entiers naturels saisis au clavier, la saisie s'arrête lorsque l'utilisateur saisit -1. Notez les erreurs que vous rencontrez à la compilation :

Exercice 5 : Crible d'Eratosthène

Le crible d'Ératosthène (IIIème siècle avant J-C.) est une méthode qui détermine tous les nombres premiers inférieurs ou égaux à un entier donné. Si on souhaite par exemple déterminer tous les nombres premiers inférieurs ou égaux à 100, nous procédons comme suit. Nous écrivons tous les nombres entiers qui vont de 2 à 100. Le premier entier écrit est 2. Il est premier : il est entouré, et tous ses multiples sont barrés. Le premier entier non barré après 2 est 3 : il est premier, tous ses multiples sont barrés. Le premier entier non barré après 3 est 5 : il est premier,tous ses multiples sont barrés. Et on procède comme ceci jusqu'à atteindre le dernier entier. . .Ceux qui ne sont pas barrés sont exactement tous les nombres premiers !

Question 1 : Écrivez un programme qui calcule les nombres premiers inférieurs ou égaux à 100. Pour cela, vous utiliserez un tableau de booléens. On crée ce tableau avec l'instruction `boolean [] tab = new boolean[101];` puis on accède à la ième case du tableau (0 i 100) par `tab[i]`.

Question 2 : Que modifieriez vous dans votre code pour que les nombres premiers soient compris entre 1 et N, N étant donné en ligne de commande par l'utilisateur (pour obtenir le même résultat qu'à la question 1, N serait égal à 100) ?

Exercice 6 : Première fonction

Compléter le code suivant en implémentant un tri à bulle :

```
/* tri a bulle*/
public class Tri {
   public static void bulle (int [] tab){
   }

   public static void main (String [] arg){
         int t[] = {2,3,7,5,6,11,0};
         System.out.println("Avant");
         for (int i=0; i<t.length; i++){
               System.out.println(t[i] + " ");
         }
         System.out.println("Apres");
         Tri.bulle(t);
         for (int i=0; i<t.length; i++){
               System.out.println(t[i] + " ");
         }
   }
}
```

Exercice 7 : Code anonyme

Que fait le code suivant :

```
public class Anonyme {
   public static void A (int un, int un1, int cpt){
         int un2=un + un1;
         if (cpt<31){
               cpt++;
               A(un1, un2, cpt);
         }
         System.out.println(un2);
   }

   public static void main (String [] arg){
         A(1, 1, 2);
         System.out.println("1");
         System.out.println("1");
```

```
    }
}
```

Comment avez vous fait pour en comprendre le sens ?_____
_____
_____

Exercice 8 : Palindrome

On appelle palindrome un mot pouvant se lire indifféremment dans les deux sens, par exemple ici, rêver sont des palindromes.
En utilisant la classe String, écrivez un programme qui teste si un mot passé en argument est un palindrome.

Indication (Classe String) Soit s une chaîne de caractères. Les caractères sont indexés dans s de 0 à s.length()−1. L'appel s.charAt(i) renvoit le ième caractère de s.
Notez les erreurs que vous rencontrez à la compilation :_____
_____
_____
_____

Exercice 9 : Triangle de Pascal

En utilisant un objet ArrayList, écrivez un programme qui affiche un triangle de Pascal de N lignes.
Exemple pour N = 6 :

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

Notez les erreurs que vous rencontrez à la compilation :_____
_____
_____
_____

Exercice 10 : Itérateurs

Les itérateurs servent à parcourir les collections. Par exemple, en Java, les classes `Vector` et `ArrayList` sont des collections. Un itérateur est une sorte de curseur dont le travail est de se déplacer dans une séquence d'objets. Il offre un accès simple à chaque objet de cette séquence sans se préoccuper de la structure de stockage sous-jacente. Java met à disposition un itérateur standard — la classe Iterator — pour parcourir les éléments d'une collection. Par exemple, supposons

l’existence d’une liste d’entiers l (ArrayList<Integer>). Alors, on peut créer un itérateur comme suit :

```
Iterator i = l. iterator ();
```

Ensuite, deux solutions existent pour parcourir la liste :

```
while (i. hasNext ()) {
// extrait l’element suivant
Integer j= ( Integer )i. next ();
// On traite l’element : Ici , on l’affiche
System . out. print (j);
}
```

Ou, plus simplement :

```
for( Integer j : L) {
System .out . print (j);
}
```

Reprenez la solution de l’Exercice 9 et utilisez un itérateur lorsque cela est possible.