

Travaux Dirigés

PROGRAMMATION ORIENTEE OBJET
Langage Java

TD 1 : Types simples, chaînes de caractères et tableaux, quelques classes utiles

Le but de ce TD est de vous familiariser avec la syntaxe de Java et d'étudier les différences avec le langage C. Vous étudierez également quelques classes utiles.

Exercice 1 : Corriger les programmes suivants :

```
public class Scope {
    public static void main(String args[]){
        int i=0;
        for (int i=0 ;i<5 ;i++) {System.out.print(i+",") ;}
        System.out.print("\n") ;
    }
}
```

```
public class Variables {
    public static void main(String args[]){
        float a=3.0 ;
        double b=4 ;
        float c ;
        c=Math.sqrt(a*a+b*b) ;
        System.out.print("c="+c) ;
    }
}
```

```
public class Promote {
    public static void main(String args[]){
        byte b=42 ;
        char c='a' ;
        short s=1024 ;
        int i=50000 ;
        float f=5.67f,
        double d=.1234 ;
        double resultat=(f*b)+(i/c)-(d*s) ;
        System.out.print((f*b)+ "+"+(i/c)+ "-"+(d*s)) ;
        System.out.println("="+resultat) ;

        byte b2=10 ;
        byte b3=b2*b;
        System.out.println("b3="+b3) ;
    }
}
```

Exercice 2: Soit le fichier source java ci-après :

```
public class Array {
    public static void main(String args[]){
        String jour_semaine[]=new String [7];
        int jour=Integer.parseInt(args[0]) ;
        jour_semaine[0]="dimanche" ;
        jour_semaine[1]= "lundi" ;
        jour_semaine[2]= "mardi" ;
        jour_semaine[3]= "mercredi" ;
        jour_semaine[4]= "jeudi" ;
        jour_semaine[5]= "vendredi" ;
        jour_semaine[6]= "ça me dit" ;
        System.out.println("Moi, le préfère le "+jour_semaine[jour]) ;
    }
}
```

```
}  
}
```

Que se passera-t-il si on lance les commandes :

\$java Array 2

\$java Array 7

Expliquer.

Exercice 3 : Le but de cet exercice est d'étudier les possibilités offertes par certaines classes du langage java. A partir de l'analyse de chaque programme, indiquez ce qui s'affiche à l'écran.

Classes String et StringBuffer

```
public class DemoString2 {  
    static String arr[] = {  
        "Voici", "venu", "le", "temps", "pour", "tous", "les", "bons",  
        "citoyens", "de", "venir", "en", "aide", "à", "leur", "pays"  
    };  
  
    public static void main(String args[]) {  
        String s;  
        int a = 42;  
        StringBuffer sb = new StringBuffer(40);  
        s = sb.append("a = ").append(a).append("!").toString();  
        System.out.println(s);  
  
        String s1 = "Hello";  
        String s2 = "Hello";  
        String s3 = "Goodbye";  
        String s4 = "HELLO";  
        System.out.println(s1 + " identique à " + s2 + " -> " + s1.equals(s2));  
        System.out.println(s1 + " identique à " + s3 + " -> " + s1.equals(s3));  
        System.out.println(s1 + " identique à " + s4 + " -> " + s1.equals(s4));  
        System.out.println(s1 + " identique à (sans distinction entre min/maj) " + s4 + " -> "  
+s1.equalsIgnoreCase(s4));  
  
        String sc = "Ceci est une démo de la méthode getChars.";  
        int debut = 13;  
        int fin = 17;  
        char buf[] = new char[fin - debut];  
        sc.getChars(debut, fin, buf, 0);  
        System.out.println(buf);  
  
        String si = "Le moment est venu pour tous les bons citoyens " +  
            "de venir en aide à leur pays "+  
            "et de payer leurs impots";  
        System.out.println(si);  
        int indexd1=si.indexOf('d');  
        System.out.println("index du premier 'd' = " +si.indexOf('d'));  
        System.out.println("index du premier 'de' = " + si.indexOf("de"));  
        System.out.println("index du deuxième 'd' = " + si.indexOf('d', indexd1+1));  
        System.out.println("index du dernier 'de' = " + si.lastIndexOf("de"));  
  
        StringBuffer sb2 = new StringBuffer("hello world!");  
        sb2.insert(6, "there ");  
        System.out.println(sb2);
```

```

StringBuffer sb3 = new StringBuffer("Hello");
System.out.println("sb3 avant = " + sb3);
System.out.println("charAt(1) avant = " + sb3.charAt(1));
sb3.setCharAt(1, 'i');
sb3.setLength(2);
System.out.println("sb3 avant = " + sb3);
System.out.println("charAt(1) après = " + sb3.charAt(1));

StringBuffer sb4 = new StringBuffer("Hello");
System.out.println("contenu = " + sb4);
System.out.println("taille = " + sb4.length());
System.out.println("capacité = " + sb4.capacity());

String s5=new String("Salut à tous");
String s5bis;
s5bis=s5.substring(2,6);
System.out.println("s5 = " + s5bis);

}
}

```

Classe HashTable

```

import java.util.Dictionary;
import java.util.Hashtable;
class HTDemo {
    public static void main(String args[]) {
        Hashtable ht = new Hashtable();
        ht.put("title", "The Java Handbook");
        ht.put("author", "Patrick Naughton");
        ht.put("email", "naughton@starwave.com");
        ht.put("age", new Integer(30));
        show(ht);
    }

    static void show(Dictionary d) {
        System.out.println("Title: " + d.get("title"));
        System.out.println("Author: " + d.get("author"));
        System.out.println("Email: " + d.get("email"));
        System.out.println("Age: " + d.get("age"));
    }
}

```

Exercice 4: Analyser la classe ci-après :

```

class Mois {
    String nom;
    int jours;

    Mois(String nom, int jours) {
        this.nom = nom;
        this.jours = jours;
    }
}

```

En utilisant un tableau, complétez la classe TestMois ci-après pour qu'elle affiche le nombre de jours dans un mois donné, par exemple :

\$java TestMois 10
Le mois de novembre a 30 jours

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

        int mon_mois = Integer.parseInt(args[0]);
        // code à fournir
        System.out.print("Le mois de " + moisJours[mon_mois].nom + " a ");
        System.out.println(moisJours[mon_mois].jours + " jours");
    }
}
class Mois {
    String nom;
    int jours;

    Mois(String nom, int jours) {
        this.nom = nom;
        this.jours = jours;
    }
}
```

Exercice 5 : On considère la méthode suivante :

```
1 public static String buildString(String s) {
2     String r="";
3     for(int i=0;i<s.length();i++) {
4         if(keepChar(s.charAt(i),i)) {
5             r=r+s.charAt(i);
6         }
7     }
8     return r;
9 }
```

On remarque que la méthode utilise la méthode keepChar.

1. On suppose que la méthode keepChar est la suivante :

```
1 public static boolean keepChar(char c,int position) {
2     return true;
3 }
```

Que fait alors la méthode buildString (justifiez votre réponse) ?

2. On suppose que la méthode keepChar est la suivante :

```
1 public static boolean keepChar(char c,int position) {
2     if(position%2==1) {
3         return true;
4     } else {
5         return false;
6     }
7 }
```

Quel est le résultat de la méthode buildString si on lui transmet la chaîne "NBVCX" ?
Que fait la méthode buildString ?

3. Pour chaque résultat souhaité pour la méthode buildString proposez une méthode keep-

Char qui permette son obtention :

(a) La chaîne renvoyée par `buildString` doit contenir un caractère sur trois de la chaîne paramètre, en conservant le caractère de position 1, puis celui de position 4, etc.
L'image de "ABCDEF" est donc "BE".

(b) La chaîne renvoyée par `buildString` doit contenir les caractères qui ne sont pas des chiffres contenus dans la chaîne paramètre. L'image de "AB2C3D6EF" est donc "ABCDEF".

Vous pourrez utiliser la classe `Character` qui propose une méthode `isDigit`¹ qui à un char associe `true` si et seulement si ce caractère correspond à un chiffre.

¹ Exemple: `boolean test=Character.isDigit('A');`