

Introduction à la Programmation 1

Objectifs:

- Découverte du type String.
- Comprendre qu'il y a des types différents.
- Maîtriser les expressions booléennes dans les conditions.
- Comprendre l'utilisation des boucles avec dépendances.

1 Les chaînes de caractères : le type String

Les chaînes de caractères _____[COURS]

- Les constantes de chaînes de caractères sont écrites entre guillemets doubles.
- Attention aux guillemets dans la chaîne, on les écrit \".
- Opérateur de concaténation de deux chaînes est +
- Le caractère \n représente le passage à une nouvelle ligne.
- Les variables de type String peuvent être utilisées pour stocker des messages que l'on souhaite afficher à l'écran avec par exemple la procédure `void printString (String s)` que l'on vous donne dans ce sujet.

Exercice 1 (Des exemples simples, ★)

Quelle est la valeur des variables à la suite de l'exécution des instructions suivantes ?

```
1 String s = " rentre chez lui.";
2 String s2 = "Paul";
3 String s3 = "Michel";
4 String s4 = s2 + s;
5 String s5 = s3 + s;
6 String s6 = s2 + " et " + s3 + " rentrent chez eux.\n";
```

Réponse ☐

2 Utilisation des différents types `int` et String

Les chaînes de caractères et les entiers sont des types différents _____[COURS]

- Une variable `x` de type `int` ne peut pas contenir de chaîne de caractères.
- Une variable `s` de type String ne peut pas contenir d'entier.
- On peut utiliser les deux types dans une même suite d'instructions.

```

1 String s = "Ceci est une chaine";
2 int a = 2;
3 int b = a * a;

```

Exercice 2 (Des exemples simples avec des variables de différents type, ☆)

1. Quelle est la valeur de la variable ch après l'exécution des instructions suivantes ?

```

1 int x = 3;
2 String ch = "Salut";
3 if (x <= 2) {
4     ch = "Hallo";
5 } else {
6     ch = "Hi";
7 }

```

2. Que fait la suite d'instructions suivantes ?

```

1 int x = 3;
2 String st = "Ca va ?";
3 String ch = "";
4 x = x / 2;
5 if (x == 1) {
6     ch = "How are you ?";
7 } else {
8     ch = "Comment allez-vous ?";
9 }
10 ch = ch + "\n";
11 printString (ch);

```

Réponse ☐

3 Fonctions avec le type String

Les fonctions peuvent utiliser le type String _____[COURS]

- Une fonction peut retourner une valeur de type String.
- Les valeurs de type String peuvent être utilisées comme paramètres d'une fonction.
- Les deux types ne sont pas forcément identiques – on peut avoir une fonction qui prend en paramètre une chaîne de caractères et renvoie un entier ou une fonction prenant en paramètre un entier et retournant une chaîne de caractères.

On considère les deux fonctions données ci-dessous :

```

1 public static String concatenate (String s, String s2) {
2     return s + s2;
3 }
4
5 public static String neymarOrNot (int x) {
6     String st = "";
7     if (x == 10) {
8         st = "C'est le numéro de Neymar.";
9     } else {

```

```

10     st = "Ce n'est pas le numéro de Neymar.";
11 }
12 return st;
13 }

```

Elles peuvent être utilisées de la façon suivante :

```

1 String s = concatenate ("Hello", "World!\n");
2 String s2 = concatenate (s, "Ca va ?");
3 String s3 = neymarOrNot (1);

```

Les fonctions peuvent être utilisées par exemple pour :

- calculer la longueur d'une chaîne de caractères avec "`int stringLength (String s)`";
 - obtenir le $(i+1)$ -ème caractère d'une chaîne avec "`String characterAtPos (String s, int i)`" (Le premier caractère est à la position 0 et si i est plus grand que la longueur, la fonction renvoie la chaîne vide "").);
 - Transformer un entier en chaîne de caractères avec "`String intToString (int a)`".
- (De telles fonctions vous seront fournies cette année.)

Exercice 3 (Utilisation de fonctions données, ★)

1. On considère la fonction suivante :

```

1 public static String addA (String ch) {
2     return (ch + "A");
3 }

```

Que vaut la variable `st` après exécution du code suivant ?

```

1 String s = "Ma lettre finale est ";
2 String st = addA (s);

```

2. Quelle est la valeur de la variable `a` après les instructions suivantes où "`int stringLength(String s)`" est la fonction décrite ci-dessus ?

```

1 String ch = "Ceci est ";
2 String ch2 = "une chaîne";
3 ch = ch + ch2;
4 int a = stringLength (ch);

```

3. Que vaut la variable `cha` dans l'exemple suivant où "`String characterAtPos (String s, int i)`" est la fonction décrite ci-dessus ?

```

1 String ch = "Avec consonnes";
2 String cha = characterAtPos (ch, 0);
3 cha = cha + characterAtPos (ch, 2);
4 cha = cha + characterAtPos (ch, 6);
5 cha = cha + characterAtPos (ch, 9);
6 cha = cha + characterAtPos (ch, 12);

```

Réponse ☐

Exercice 4 (Modification d'une fonction, ★★)

On considère la fonction suivante où la fonction "`String intToString (int x)`" est celle décrite ci-dessus.

```

1 public static String result (int x) {
2     return intToString (x);
3 }

```

Et on considère la liste d'instructions suivante :

```

1 int a = 4;
2 int b = a * a;
3 a = b - a;
4 String s = result (a);
5 printString (s);

```

1. Quelle valeur a chent ces instructions ?
2. Est-il possible de modifier la fonction `String result (int x)` de telle sorte que le programme a che à la fin "Le resultat du calcul est n" (où *n* est la valeur contenue dans la variable entière a) ?
3. Définir une fonction "`int square(int n)`" qui calcule le carré du nombre donné en paramètre et remplacer à l'endroit adéquat le morceau de code dans les instructions précédentes par l'utilisation de cette fonction.

Réponse ☐

4 Expressions booléennes dans les conditionnelles

Des expressions booléennes plus complexes dans les tests _____[COURS]

– Voici trois opérateurs booléens :

1. le **et** (en Java &&). Une expression avec un **et** est vraie si les expressions à gauche et à droite du **et** sont vraies ;
2. le **ou** (en Java ||). Une expression avec un **ou** est vraie si au moins une des expressions à gauche et à droite du **ou** est vraie ;
3. la négation (en Java !). Une expression avec une négation est vraie si l'expression après la négation est fausse.

– Exemple d'expressions utilisant ces opérateurs :

- `(2 < x && x <= 5)`
- `(x == 1 || x == 2)`
- `!(x == 0)` (qui est équivalent à `x != 0`)

– On peut combiner ces opérateurs, par exemple en écrivant "`(x == 1 || x == 2) && y > 5`"

– Attention, sur les chaînes de caractères, on ne peut pas tester l'égalité de deux chaînes avec `==`. Pour cela, on vous donne une fonction "`boolean stringEquals (String st1, String st2)`" à utiliser dans les tests

– Voici un exemple :

```

1 String nom = "Michel";
2 if (stringEquals (nom, "Paul") || stringEquals (nom, "Pierre")) {
3     printString("On a trouve Paul ou Pierre.");
4 } else {
5     printString("Ce n'est ni Paul, ni Pierre.");
6 }

```

Exercice 5 (Savoir évaluer une condition, ★)

Quelle est la valeur de la variable *x* après la suite d'instructions suivante ?

```

1 int a = 2;
2 a = a * a * a * a + 1;
3 int b = a / 2;
4 int c = a - 1;
5 int x = 0;
6 if ((b == 0) && (c == 16)) {
7     x = 1;
8 } else {
9     x = 2;
10 }

```

Réponse ☐

Exercice 6 (Équivalence d'expressions, **)

On dit que deux expressions booléennes sont équivalentes quand ces deux expressions s'évaluent toujours vers la même valeur booléenne, peu importe les valeurs des variables.

Par exemple, " $x > 10 \ \&\& \ x < 12$ " est équivalente à " $x == 11$ ". Par contre, " $x > y$ " et " $x != y$ " ne sont pas équivalentes.

Dire si les expressions suivantes sont équivalentes :

1. " $x > y \ || \ x < y$ " et " $x != y$ ";
2. " $x != 3 \ \&\& \ x != 4 \ \&\& \ x != 5$ " et " $x <= 2 \ || \ x >= 6$ ";
3. " $x == y \ \&\& \ x == z$ " et " $x == z$ ";
4. " $x == y \ \&\& \ x == z$ " et " $x == y \ \&\& \ y == z$ ".

Réponse ☐

Exercice 7 (Simplification des expressions booléennes, **)

Simplifier une expression booléenne signifie la remplacer par une autre expression booléenne équivalente qui est plus courte. Simplifier les expressions suivantes :

- $x > 5 \ \&\& \ x > 7$
- $x == y \ \&\& \ x == z \ \&\& \ y == z$
- $x == 17 \ || \ (x != 17 \ \&\& \ x == 42)$
- $x > 5 \ || \ (x <= 5 \ \&\& \ y > 5)$

Réponse ☐

5 Instructions conditionnelles imbriquées

Instructions composées _____[COURS]

- Certaines des instructions présentées à la Session 1 sont des instructions *composées* : elles peuvent contenir d'autres instructions.
- Exemples d'instructions composées déjà vus : les conditionnelles, et les boucles.
- Par exemple :

```

1 int x = 1;
2 int y = 2;
3 int z = 0;
4 if (x == 1) {
5     if (y == 2) {
6         z = 3;

```

```

7   } else {
8       z = 5;
9   }
10  } else {
11      z = 7;
12  }

```

L'exécution du programme précédent a pour effet d'affecter la valeur 3 à la variable z :

- On a le droit d'imbriquer arbitrairement des instructions : des conditionnelles dans des conditionnelles dans des conditionnelles ...
- Avec des instructions imbriquées sur plusieurs niveaux, il est absolument indispensable (dans l'intérêt des lecteurs humains) de suivre strictement une discipline d'indentation du code source.
- On parlera des boucles imbriquées à la Session 3.

Exercice 8 (Suppression des opérateurs booléens, **)

Réécrire les suites d'instructions suivantes en n'utilisant que des conditions sans opérateur booléen :

1.

```

1 String text = "Ceci est une phrase.\n";
2 int l = stringLength (text);
3 int x = 0;
4 if (l != 0 && l <= 10) {
5     x = 1;
6 } else {
7     x = 2;
8 }

```

2.

```

1 int a = 26;
2 int b = 2 * a;
3 b = a - a;
4 if (b == a || b > 43) {
5     b = 0;
6 } else {
7     b = 1;
8 }

```

3.

```

1 int c = 45;
2 int d = c / 9;
3 int e = 0;
4 if (!(d == 6) && (c >= 2)) {
5     e = 2;
6 } else {
7     e = 3;
8 }

```

Réponse ☐

6 Boucles

Boucles et conditionnelles

[COURS]

Le corps d'une boucle peut contenir n'importe quelle instruction. En particulier, il peut contenir une conditionnelle.

```
1 for (int i = 0; i < 100; i++) {
2     if (i % 2 == 0) {
3         printString ("Pair.");
4     } else {
5         printString ("Impair.");
6     }
7 }
```

7 Fonctions utilisées

Liste des fonctions

[COURS]

```
1 /* Affiche un entier. */
2 public static void printInt (int x) {
3     System.out.print (x);
4 }
5
6 /* Affiche une chaîne de caractères. */
7 public static void printString (String s) {
8     System.out.print (s);
9 }
10
11 /* Retourne la longueur d'une chaîne de caractères. */
12 public static int stringLength (String s) {
13     return s.length ();
14 }
15
16 /* Transforme un entier x en une chaîne de caractères. */
17 * Retourne la chaîne de caractères représentant la valeur de x.
18 */
19 public static String intToString(int x) {
20     return String.valueOf (x);
21 }
22
23 /*
24 * Retourne une chaîne constituée du caractère se trouvant à la
25 * position $i$ de la chaîne s.
26 * Si i est négatif ou en dehors de la chaîne, retourne la chaîne vide.
27 */
28
29 public static String characterAtPos(String s,int i) {
30     String res = "";
31     char a;
32     if(i >= 0 && i < s.length ()){
33         res = String.valueOf (s.charAt (i));
34     }
35     return res;
```

```
36 }
37
38 /*
39  * Teste si deux chaînes de caractères st1 et st2 ont le même contenu.
40  * Retourne vrai si st1 et st2 sont égales.
41  */
42 public static boolean stringEquals (String st1, String st2) {
43     return st1.equals(st2);
44 }
```


8 Réponses aux exercices

Exercice 1, page 1

s vaut "rentre chez lui."
s2 vaut "Paul"
s3 vaut "Michel"
s4 vaut "Paul rentre chez lui."
s5 vaut "Michel rentre chez lui."
s6 vaut "Michel et Paul rentrent chez eux."

En exécutant le programme suivant la réponse sera affichée.

```
1 public class ExoPaulAndMichel {
2
3     /* Affiche une chaîne de caractères. */
4
5     public static void printString (String s) {
6         System.out.print (s);
7     }
8
9     public static void main (String[] args) {
10        String s = " rentre chez lui.";
11        String s2 = "Paul ";
12        String s3 = "Michel ";
13        String s4 = s2 + s;
14        String s5 = s3 + s;
15        String s6 = s2 + " et " + s3 + " rentrent chez eux.\n";
16        s6 = s2 + " et " + s3 + " rentrent chez eux.\n";
17        printString (s2);
18        printString (s3);
19        printString (s4);
20        printString (s5);
21        printString (s6);
22    }
23 }
```

code/ExoPaulAndMichel.java

Exercice 2, page 2

1. La variable ch vaut "Hi".
2. La suite d'instructions affiche "How are you ?".

En exécutant les programmes suivants, la réponse sera affichée.

```
1.
1 public class ExoSalut {
2
3     /* Affiche une chaîne de caractères. */
4
5     public static void printString (String s) {
6         System.out.print (s);
7     }
8
9     public static void main (String[] args) {
```

```

10     int x = 3;
11     String ch = "Salut";
12     if (x <= 2) {
13         ch = "Hallo";
14     } else {
15         ch = "Hi";
16     }
17     printString (ch);
18 }
19 }

```

code/ExoSalut.java

```

1 public class ExoHow {
2
3     /* Affiche une chaîne de caractères. */
4
5     public static void printString (String s) {
6         System.out.print (s);
7     }
8
9     public static void main (String[] args) {
10         int x = 3;
11         String st = "Ca va ?";
12         String ch = "";
13         x = x / 2;
14         if (x == 1) {
15             ch = "How are you ?";
16         } else {
17             ch = "Comment allez-vous ?";
18         }
19         ch = ch + "\n";
20         printString (ch);
21     }
22 }

```

code/ExoHow.java

Exercise 3, page 3

1. La variable st vaut "Ma lettre finale est "A"
2. La variable a contient l'entier 19 (on compte les espaces comme des caractères)
3. La variable ch contient la chaîne de caractères " Ae ooe"

En exécutant les programmes suivants, la réponse sera affichée.

1.

```

1 public class ExoAddLetter {
2
3     /* Affiche une chaîne de caractères. */
4
5     public static void printString (String s) {
6         System.out.print (s);
7     }
8
9     public static String addA (String ch) {
10         return (ch + "A");

```

```

11     }
12
13     public static void main (String[] args) {
14         String s = "Ma lettre finale est ";
15         String st = addA (s);
16         printString (st);
17     }
18 }

```

code/ExoAddLetter.java

2.

```

1 public class ExoLengthComputing {
2
3
4     /* Retourne la longueur d'une chaine de caracteres. */
5
6     public static int stringLength (String s) {
7         return s.length();
8     }
9
10    /* Affiche une chaine de caracteres. */
11
12    public static void printString (String s) {
13        System.out.print (s);
14    }
15
16    /*
17     * Transforme un entier x en une chaine de caracteres.
18     * Retourne la chaine de caracteres representant x.
19     */
20
21    public static String intToString (int x) {
22        return String.valueOf (x);
23    }
24
25    public static void main (String[] args) {
26        String ch = "Ceci est ";
27        String ch2 = "une chaine";
28        ch = ch + ch2;
29        int a = stringLength (ch);
30        String res = intToString (a);
31        printString (res);
32    }
33 }

```

code/ExoLengthComputing.java

3.

```

1 public class ExoDeleteConsonant {
2
3     /*
4      * Retourne une chaine constituée du caractère se trouvant à la
5      * position $i$ de la chaine s.
6      * Si i est négatif ou en dehors de la chaine, retourne la chaine
7      * vide.
8      */
9
10

```

```

9      public static String characterAtPos (String s, int i) {
10          String res = "";
11          char a;
12          if (i >= 0 && i < s.length() ) {
13              res = String.valueOf (s.charAt (i) );
14          }
15          return res;
16      }
17
18      /* Affiche une chaîne de caractères. */
19
20      public static void printString (String s) {
21          System.out.print (s);
22      }
23
24      public static void main (String[] args) {
25          String ch = "Avec consonnes";
26          String cha = characterAtPos (ch, 0);
27          cha = cha + characterAtPos (ch, 2);
28          cha = cha + characterAtPos (ch, 6);
29          cha = cha + characterAtPos (ch, 9);
30          cha = cha + characterAtPos (ch, 12);
31          printString (cha);
32      }
33  }

```

code/ExoDeleteConsonant.java

Exercice 4, page 4

1. Il a été demandé le chapitre 12. Pour vérifier il suffit d'exécuter le programme suivant :

```

1  public class ExoComputeWithSquare {
2
3      /* Affiche une chaîne de caractères. */
4
5      public static void printString (String s) {
6          System.out.print (s);
7      }
8
9      /* Transforme un entier en une chaîne de caractères. */
10
11     public static String intToString (int x) {
12         return String.valueOf (x);
13     }
14
15     public static String result (int x) {
16         return "Le resultat du calcul est " + intToString (x);
17     }
18
19     public static void main (String[] args) {
20         int a = 4;
21         int b = a * a;
22         a = b - a;
23         String s = result (a);
24         printString (s);
25     }

```

26 }

code/ExoComputeWithSquare.java

2. Il faut modifier la fonction de la façon suivante :

```
1 public static String resultat (int x) {  
2     return "Le résultat du calcul est " + intToString (x);  
3 }
```

3. Le programme devient le suivant :

```
1 public class ExoComputeWithSquare2 {  
2  
3     /* Affiche une chaîne de caractères. */  
4  
5     public static void printString (String s) {  
6         System.out.print (s);  
7     }  
8  
9     /*  
10    * Transforme un entier x en une chaîne de caractères.  
11    * Retourne la chaîne de caractères représentant x.  
12    */  
13  
14    public static String intToString (int x) {  
15        return String.valueOf (x);  
16    }  
17  
18    public static String result (int x) {  
19        return "Le resultat du calcul est " + intToString (x);  
20    }  
21  
22    static int square (int n) {  
23        return n * n;  
24    }  
25  
26    public static void main (String[] args) {  
27        int a = 4;  
28        int b = square (a);  
29        a = b - a;  
30        String s = result (a);  
31        printString (s);  
32    }  
33 }
```

code/ExoComputeWithSquare2.java

Exercice 5, page 5

La variable x contient la valeur 2. Pour le voir, il suffit d'exécuter le programme suivant :

```
1 public class ExoTest1 {  
2  
3     /* Affiche un entier. */  
4  
5     public static void printInt (int x) {  
6         System.out.print (x);
```

```

7      }
8
9
10     public static void main (String[] args) {
11         int a = 2;
12         a = a * a * a * a + 1;
13         int x = 0;
14         if ( (a / 2 == 0) && (a - 1 == 16) ) {
15             x = 1;
16         } else {
17             x = 2;
18         }
19         printInt (x);
20     }
21 }

```

code/ExoTest1.java

Exercice 6, page 5

- Oui
- Oui
- Non : Si x et z valent 1 et y vaut 5, alors la première expression est fausse et la deuxième est vraie.
- Oui : chacune des deux expressions exprime le fait que les trois variables ont la même valeur.

Exercice 7, page 5

- $x > 7$
- $x == y \ \&\& \ y == z$ (par exemple)
- $x == 17 \ || \ x == 42$
- $x > 5 \ || \ y > 5$

Exercice 8, page 6

1.

```

1 String text = "Ceci est une phrase.\n";
2 int l = stringLength(text);
3 int x = 0;
4 if (l != 0) {
5     if (l <= 10) {
6         x = 1;
7     } else {
8         x = 2;
9     }
10 } else {
11     x = 2;
12 }

```

2.

```

1 int a = 26;
2 int b = 2 * a;
3 b = a - a;
4 if (b == a) {

```

```

5   b = 0;
6 } else {
7   if (b > 43) {
8     b = 0;
9   } else {
10    b = 1;
11  }
12 }

```

3.

```

1  int c = 45;
2  int d = c / 9;
3  int e = 0;
4  if (d == 6) {
5    e = 3;
6  } else {
7    if (c >= 2) {
8      e = 2;
9    } else {
10     e = 3;
11   }
12 }

```

ou

```

1  int c = 45;
2  int d = c / 9;
3  int e = 0;
4  if (d != 6) {
5    if (c >= 2) {
6      e = 2;
7    } else {
8      e = 3;
9    }
10 } else {
11   e = 3;
12 }

```