

# Introduction à la Programmation (IP1)

## Partiel (Sujet A) – Durée : 2 heures

Université Paris-Diderot – Samedi 25 octobre 2014

- Aucun document ni aucune machine ne sont autorisés. Les téléphones doivent être rangés.
- Les exercices sont tous indépendants.
- Une réponse peut utiliser les réponses attendues à une question précédente (même si elle est non traitée).
- Les fragments de code Java doivent être correctement indentés.

On rappelle que le type `boolean` est constitué des deux valeurs `true` et `false`.

### Exercice 1

1. Remplir les pointillés avec un type cohérent.

```
1 public static ... f (... s, ... t) {
2     t = t + 5;
3     s = s + t;
4     return s;
5 }
```

2. Donner 5 erreurs dans la fonction suivante en les expliquant à l'aide d'une courte phrase.

```
1 public static f (string[] a) {
2     for (int i = 0; i < 100; i++) {
3         printInt (i)
4         s = s + i
5     return s;
6 }
```

□

### Exercice 2

1. Que valent les variables `a`, `b` et `s` à la fin de l'exécution du fragment de code suivant ?

```
1 int a = 0;
2 int b = 0;
3 String s = "";
4 for (int i = 0; i < 11; i++) {
5     if (i % 3 == 0) {
6         s = s + "a";
7         a = a + 1;
8     } else {
9         s = s + "b";
10        b = b + 1;
11    }
12 }
13 if (a > b) {
14     printString ("a!");
15 } else {
16     printString ("b!");
17 }
```

2. Qu'affiche ce programme ?

□

### Exercice 3

1. Pourquoi la fonction suivante ne calcule-t-elle pas la factorielle de l'entier  $n$  pris en paramètre ? (On rappelle que la factorielle de l'entier  $n$  est le produit des entiers non nuls plus petits que  $n$ ,  $n$  compris. Ainsi, la factorielle de 5 est  $1 * 2 * 3 * 4 * 5$ ).

```

1 public static int fact (int n) {
2     int result = 1;
3     for (int i = 0; i < n; i++) {
4         result = result * i;
5     }
6     return result;
7 }

```

2. Que renvoie précisément cette fonction (par rapport à  $n$ ) ?
3. Comment modifier cette fonction pour qu'elle calcule effectivement la factorielle de  $n$  ?

□

### Exercice 4

1. Écrire une fonction « `public static String mirror (String s)` » qui renvoie le miroir de  $s$ , c'est-à-dire la chaîne qui contient les mêmes caractères que  $s$  mais apparaissant dans l'ordre inverse de celui dans lequel ils apparaissent dans  $s$ .  
(Par exemple, "cachet aspirine" donne "eniripsa tehcac".)

Fonctions et procédures pouvant être utilisées :

- « `public static String charAt (String s, int i)`; » qui renvoie la chaîne formée du caractère de  $s$  situé en position  $i$  sous la forme d'une chaîne. (La première position est la position 0.)
- « `public static int stringLength (String s)`; » qui retourne la longueur de la chaîne  $s$  pris en entrée.

□

### Exercice 5

1. Écrire une fonction `equal` qui prend en argument deux tableaux d'entiers et qui renvoie `true` si et seulement si les deux tableaux sont égaux (c'est-à-dire s'ils ont la même longueur et si pour tout  $i$ , la case d'indice  $i$  du premier tableau contient la même valeur que la case d'indice  $i$  du second tableau).

Fonctions et procédures pouvant être utilisées :

- « `public static int intArrayLength (int[] t)`; » qui retourne la longueur du tableau  $t$ .

□

### Exercice 6

Le but de cet exercice est d'afficher un entier sous une représentation "francisée". Par exemple, l'entier 3483201 est représenté par « 3 millions 483 mille 201 » en français. On supposera que les entiers fournis n'atteignent pas le milliard et sont positifs.

1. Écrire une fonction « `public static int numberOfDigits (int n)`; » qui renvoie le nombre de chiffres de l'entier  $n$  fourni en paramètre. (Indication : on pourra compter combien de fois il est possible de diviser  $n$  par 10 sans obtenir zéro.)
2. Écrire une fonction « `public static void printNumberInFrench (int n)`; » qui affiche la version francisée de l'entier fourni en paramètre. (On rappelle que mille est invariable, mais pas million.)  
Pour cela, on pourra d'abord écrire trois procédures traitant de l'affichage des millions, des milliers et des unités et ensuite, utiliser ces procédures pour construire la procédure `printNumberInFrench`.

Fonctions et procédures pouvant être utilisées :

- « `public static void printString (String s)`; » qui affiche une chaîne  $s$  à l'écran.
- « `public static void printInt (int x)`; » qui affiche un entier  $x$  à l'écran.

□