

**CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS
CENTRE RÉGIONAL DE NOUVELLE-AQUITAINE**

ANNÉE UNIVERSITAIRE 2019 / 2020

SUJET EXAMEN dans le cadre du COVID-19

Code et intitulé de l'UE :

NFA031 - Programmation avec Java : notions de base

N° semestre : 1

N° session : 2

Nom de l'intervenant : Laurent FALLOT

Date : 12 juin 2020

Horaire : 18h00-20h00

Durée : 2h (+15mn pour la prise en main de l'outil et la remise du devoir)

Nombre de pages du sujet (page de garde incluse) : 4

Barème : voir titres

Calculatrice	Oui
Dictionnaire	Oui
Notes de cours	Oui

Pour le rendu du devoir, merci de vous référer aux consignes indiquées sur le tutoriel « participer à un examen en ligne » qui vous a été fourni et indiquées dans la rubrique d'examen sur Moodle.

1. Cours (4 points)

1.1. Comment une méthode peut signaler au programme qu'un traitement anormal s'est produit ? Quel code doit-on générer ?

1.2. Dans le bout de code qui suit sachant que la méthode `Math.sqrt` calcule la racine carrée du double pris en paramètre et retourne un double

```
dst = Math.sqrt(x*x + y*y);
```

- a) De quoi s'agit-il ?
- b) Qu'est-ce que `dst` ?
- c) Quels sont les opérateurs ?
- d) Quels sont les opérandes de `+` ?
- e) Quel peut être le type de `x` et de `y` ?

2. Erreurs (2 points)

2.1. Le programme suivant permet de transformer une chaîne de caractères en nombre entier et d'afficher à l'écran le résultat. Des erreurs empêchent la compilation de ce code. Donnez 7 erreurs et proposez une correction pour chacune d'elle.

```
1. public class StringToInt {  
2.     public static boolean stringToInt(String s) {  
3.         int res = 0.0;  
4.         int i = 0;  
5.         int signe = 1;  
6.  
7.         if (s.charAt(0) == "-") {  
8.             signe = -1;  
9.             i = 1;  
10.        } else if (s.charAt(0) == '+')  
11.            i = 1;  
12.  
13.        for (; i < s.length(), i++) {  
14.            char c = s.charAt(i);  
15.            if (c < '0' || c > '9') {  
16.                throw new Error("caractère invalide !");  
17.                res == res * 10 + c - '0';  
18.            }  
19.            return signe * res;  
20.            Terminal.ecrireStringln("le résultat est : "  
21.                                   + signe * res);  
22.        }  
23.  
24.        public static void main(String [] arguments) {  
25.            Terminal.ecrireIntln(StringToInt('1234'));  
26.        }  
27.    }
```

3. Algorithmique (4 points)

3.1. Ecrire un algorithme (pas de Java svp) qui calcule la moyenne des valeurs stockées dans un tableau et inférieures ou égales à un seuil donné. Définir les entrées, les sorties, un jeu d'exemples et l'algorithme.

3.2.L'algorithme suivant permet de calculer le nombre minimum de caractères qu'il faut pour afficher un nombre entier. Le traduire sous forme de méthode.

Entrées

nombre : le nombre de départ.

Sortie

res : le nombre minimum de caractères nécessaires pour afficher le nombre de départ.

Variables locales

actuel : un entier contenant ce qu'il reste à triater du nombre de départ.

Algorithme

```
si nombre est égal à 0
    retourner 1
si nombre est négatif
    res = 1
    actuel = - nombre
sinon
    res = 0
    actuel = nombre
fin si

tant que actuel est différent de 0
    actuel = actuel / 10
    res = res + 1
fin tant que

retourner res
```

fin de l'algorithme

4. Simulation (2 points)

4.1.Présenter sous forme de tableau l'exécution ligne à ligne du programme suivant.

```
1. public class Simulation {
2.     public static void main(String[] args) {
3.         int[] tab = { 1, 5, 10, 7 };
4.         Terminal.ecrireStringln("résultat = "
5.                                 + calcul(tab, 5, 8));
6.     }
7.
8.     public static int calcul(int[] tab, int min, int max) {
9.         int res = 0;
10.        for (int i = 0; i < tab.length; i++) {
11.            if (tab[i] < min) {
12.                res = res * 10 + 1;
13.            } else if (tab[i] >= max) {
14.                res = res * 10 + 3;
15.            } else {
16.                res = res * 10 + 2;
17.            }
18.        }
19.    }
20. }
```

```

18.         }
19.         return res;
20.     }
21. }

```

5. Exercices (8 points)

Tous les exercices de cette partie devront être écrits en Java.

- 5.1.** Ecrire une méthode calculant la valeur absolue d'un nombre double. **Ne pas faire usage de la bibliothèque Math !**

Rappel : La valeur absolue d'un nombre positif est le nombre lui-même, celle d'un nombre négatif n est son opposé : $-n$.

- 5.2.** On appelle distance urbaine entre deux points, la somme des valeurs absolues de écarts coordonnée par coordonnée. Cette distance s'appelle ainsi car elle calcule la longueur du chemin à parcourir pour aller d'un point A à un point B dans une ville où toutes les rues sont soit parallèles soit perpendiculaire. Ainsi, si sur un plan de la ville, A est en x_A, y_A et B en x_B, y_B , la distance urbaine entre A et B est alors

$$\text{valeur_absolue}(x_A - x_B) + \text{valeur_absolue}(y_A - y_B).$$

Les valeurs coordonnées des points x_A, x_B, y_A et y_B seront prises en paramètre, le résultat de la méthode sera la distance urbaine entre A et B.

- 5.3.** Ecrire une méthode qui prend en paramètre un verbe du premier groupe à l'infinitif et retourne sous forme d'un tableau de String la conjugaison du verbe au présent de l'indicatif. Il faudra vérifier que le verbe se termine bien par er. Ne traiter que le cas général ne pas considérer les cas particuliers comme envoyer, acheter, peler, ...

- 5.4.** Ecrire une méthode Java prenant un paramètre un tableau de double nommé « tab » et un nombre entier positif nommé n , et qui décale vers la droite le contenu du tableau d'autant de cases que n . Les valeurs contenues dans les n cases de droite seront perdues. Les n cases de gauche recevront la valeur 0. Pensez à traiter le cas où n est négatif. Quand n est plus grand ou égal à la taille de tab, au retour, tab ne contiendra que des 0.

Exemples

tab avant appel	n	tab après appel
{1, 2, 3, 4, 5}	3	{0, 0, 0, 1, 2}
{1, 2, 3, 4, 5}	1	{0, 1, 2, 3, 4}
{1, 2, 3}	4	{0, 0, 0}

- 5.5.** Ecrire une fonction permettant de fusionner deux ArrayList (al1 et al2) de nombres entiers rangés par ordre croissant. al1 et al2 seront pris en paramètre et le résultat sera une nouvelle ArrayList où les nombres devront être aussi triés par ordre croissant.

FIN DU SUJET D'EXAMEN