

## Contrôle de Algorithmique et Programmation JAVA

Durée : 1h30

Aucun document autorisé

*Note : la qualité des commentaires, avec notamment la présence d'affirmations significatives, ainsi que les noms donnés aux variables, l'emploi à bon escient des majuscules et la bonne indentation rentreront pour une part importante dans l'appréciation du travail.*

## 1 Tableau

Écrivez la méthode statique `décompte(int []t)` qui donne le nombre d'occurrences de chaque nombre contenu dans le tableau `t`. Cette méthode renvoie un tableau indiquant, à l'indice `i`, le nombre d'occurrences du nombre `i` dans `t`.

**Par exemple** : pour un tableau `t` qui contient les valeurs `[3,1,0,3,5,6,5]`, l'exécution de la méthode `décompte(t)` renvoie le tableau `[1,1,0,2,0,2,1]` car le nombre 0 et le nombre 1 sont présents 1 fois chacun, le nombre 2 n'est pas présent, le nombre 3 est présent 2 fois, le nombre 4 n'est pas présent, le nombre 5 est présent 2 fois, et le nombre 6 est présent une seule fois.

## 2 Fichiers

Un professeur souhaite corriger les rendus de TD de ses élèves. Les logins des élèves sont dans un fichier de nom `listeEleve.txt`. On supposera que chaque élève a rendu un fichier de nom `TP<login>.java` (exemple : `TPtoto.java` pour un élève de login `toto`).

Le professeur inscrit les notes de ses élèves dans un fichier `notes.txt`. Il y écrit chaque nom de login suivi de la note pour cet élève. La note attribuée à un élève est un entier tiré au hasard entre 0 et 20, sauf s'il n'y a aucun rendu (dans ce cas, on met 0).

On rappelle qu'on ouvre un fichier en lecture ou en écriture grâce aux classes `FileInputStream` ou `FileOutputStream`, et que l'exception `FileNotFoundException` est émise si le fichier ne peut être ouvert pour une raison ou une autre.

D'autre part, les méthodes `readUTF()` et `readInt()` (resp. `writeUTF()` et `writeInt()`) de la classe `DataInputStream` (resp. `DataOutputStream`) permettent de lire (resp. écrire) une chaîne de caractères et un entier dans le fichier.

Dans une classe `Correction`, écrivez la méthode `main` qui remplit le fichier de notes `notes.txt` selon les indications précédentes.

## 3 Héritage

Soient les déclarations de classes et d'interfaces Java suivantes :

```
abstract class ClassA {
    public void methode(){System.out.println(1);}
}
class ClassB extends ClassA {
}
abstract class ClassC extends ClassA implements Interface1 {
    public void methode(){System.out.println(2);}
    public void methode(int x){System.out.println(3);}
}
```

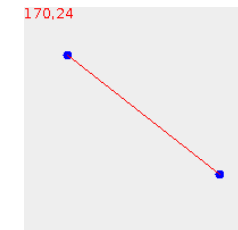
```
class ClassD extends ClassC {
    private void methode(){System.out.println(4);}
}
class ClassE extends ClassC {
    public void methode(String s){System.out.println(5);}
    public void methode(int x){System.out.println(6);}
}
interface Interface1 {
    public void methode(int x);
}
```

- 1. Dessinez le graphe d'héritage des classes et de l'interface précédentes.
- 2. Pour chacune des lignes du code suivant, indiquez s'il y a une erreur, d'où vient l'erreur et, si la ligne est censée afficher quelque chose, indiquez ce qu'elle affiche.

```
1 : ClassA c1 = new ClassB();
2 : ClassA c2 = new ClassC();
3 : ClassD c3 = new ClassD();
4 : ClassB c4 = new ClassD();
5 : ClassC c5 = new ClassE();
6 : c1.methode();
7 : c2.methode();
8 : c3.methode();
9 : c3.methode(3);
10 : c4.methode();
11 : c5.methode();
12 : c5.methode(3);
```

## 4 Swing

On souhaite écrire une petite application graphique qui permet à l'utilisateur de dessiner deux points dans un carré, de les relier par une droite et d'afficher la distance entre les deux points. L'image ci-dessous donne une idée du rendu de l'application.



Le carré est représenté par un `JPanel`, et la méthode `public void paintComponent(Graphics g)` (que vous devrez écrire), vous permettra de dessiner dans le graphique `g` associé. Le point de coordonnées `(0,0)` est le point nord-ouest du graphique.

`Graphics` possède en particulier les méthodes `fillOval`, `drawLine` et `drawString` dont vous aurez besoin. La documentation Java de ces méthodes est donnée en annexe.

On rappelle que l'interface `MouseListener` est l'auditeur des événements souris, et la méthode `public void mousePressed(MouseEvent e)` est déclenchée lorsqu'on appuie sur un des boutons de la souris. Les méthodes `getX()` et `getY()` d'un événement de type `MouseEvent` donnent les coordonnées du point  $(x, y)$  sur lequel l'utilisateur a cliqué.

Écrivez la classe `Carré` qui hérite de `JPanel` et implémente `MouseListener` et qui permet à l'utilisateur de tracer une droite entre 2 points sélectionnés à la souris, et d'afficher la distance entre ces points. **Notez bien que la droite et la distance ne s'affichent qu'après le second clic de souris.** Un 3ème clic de souris efface le carré.

On rappelle que la distance dans le plan cartésien entre deux points  $a$  et  $b$  de coordonnées  $(x_a, y_a)$  et  $(x_b, y_b)$  est égale à  $\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$ .

---

## Annexe

### fillOval

```
public void fillOval(int x, int y, int width, int height)
```

Fills an oval bounded by the specified rectangle with the current color.

Parameters:

- x - the x coordinate of the upper left corner of the oval to be filled.
- y - the y coordinate of the upper left corner of the oval to be filled.
- width - the width of the oval to be filled.
- height - the height of the oval to be filled.

### drawLine

```
public void drawLine(int x1, int y1, int x2, int y2)
```

Draws a line, using the current color, between the points  $(x_1, y_1)$  and  $(x_2, y_2)$  in this graphics context's coordinate system.

Parameters:

- x1 - the first point's x coordinate.
- y1 - the first point's y coordinate.
- x2 - the second point's x coordinate.
- y2 - the second point's y coordinate.

### drawString

```
public void drawString(String str, int x, int y)
```

Draws the text given by the specified string, using this graphics context's current font and color. The baseline of the leftmost character is at position  $(x, y)$  in this graphics context's coordinate system.

Parameters:

- str - the string to be drawn.
- x - the x coordinate.
- y - the y coordinate.