

Évaluation n°1 du 26 mai 2017

Java 8 AL ESGI

Mathieu Pauly @ Arolla

Durée : 1h

Documents autorisés

Le code est à envoyer à mathieu.pauly@arolla.fr au format ZIP.

Exercice 1 (10 points)

Les permissions de fichier Unix sont encodées en octal (par ex. 0755). En revanche, elles sont affichées à l'utilisateur sous la forme de 3 "rwx". Par exemple, lister les fichiers donnera :

```
$ ll /bin/echo
-rwxr-xr-x 1 thieux 197121 28352 août 28 2015 /bin/echo*
```

Pour extraire :

- le 4 dans 0754 on calcule le reste de la division entière de 0754 et de 8.
- le 5 dans 0754 on calcule le reste de la division entière de 0754/8 et de 8.
- le 7 dans 0754 on calcule le reste de la division entière de 0754/8/8 et de 8.

Chaque chiffre ainsi extrait peut varier entre 0 et 7. Voici comment on les interprète :

- $7 = 4 + 2 + 1 \Rightarrow 111 \Rightarrow \text{rwx}$
- $5 = 4 + 0 + 1 \Rightarrow 101 \Rightarrow \text{r-x}$
- $4 = 4 + 0 + 0 \Rightarrow 100 \Rightarrow \text{r--}$

Noter que le reste de la division entière :

- de 5 et 2 donne 1 ;
- de 5/2 et 2 donne 0 ;
- de 5/2/2 et 2 donne 1.

Consignes

Créer une classe `UnixPermissions` et une méthode `public String permissionsFor(int mode)` qui suit le contrat suivant :

- `permissionsFor(0777).equals("rwx rwx rwx");`
- `permissionsFor(0000).equals("--- --- ---");`
- `permissionsFor(0700).equals("rwx --- ---");`
- `permissionsFor(0770).equals("rwx rwx ---");`
- `permissionsFor(0666).equals("rw- rw- rw-");`
- `permissionsFor(0555).equals("r-x r-x r-x");`

On pourra utiliser :

- la classe `ArrayList`
- la méthode `Collections.reverse(List<?> list)`
- la méthode `String.join(CharSequence delimiter, Iterable<? extends CharSequence> elements)`

Exercice 2 (10 points)

Etant donné la suite d'instructions suivante :

```
ArrayList<Shape> shapes = new ArrayList<>();
shapes.add(new Circle(0, 10));
shapes.add(new Triangle(0, 0, 0, 1, 1, 0));
shapes.add(new Rectangle(0, 0, 10, 15));

for (Shape shape : shapes) {
    System.out.println(shape.dumpContent());
}
```

Et les constructeurs suivants :

```
Circle(double center, double radius)
Triangle(double x1, double y1, double x2, double y2, double x3, double y3)
Rectangle(double topLeftX, double topLeftY, double width, double height)
```

Consignes

1. Définir l'interface `Shape`.
2. Implémenter les classes `Circle`, `Triangle` et `Rectangle`.

Annexes

Documentation Java : <https://docs.oracle.com/javase/8/docs/api/>