
Dans le ponctuel :

- le nom du projet doit être *ep01grxx-nomDeFamille-revSCA*
 - le nom de l'auteur de la classe doit apparaître dans un commentaire au début de chaque classe
 - et le nom du fichier .jar remis doit être *ep01grxx-nomDeFamille-revSCA*
-

- 1) Codez la classe *Logiciel* en respectant les spécifications suivantes :
 - a) Un objet de type *Logiciel* permet de mémoriser le nom, le nombre d'utilisateurs et son coût.
 - b) La classe doit avoir un constructeur avec trois paramètres : le nom, le nombre d'utilisateurs et le coût.
 - c) La classe doit avoir un constructeur avec deux paramètres : le nom et le coût. Dans ce cas, le nombre d'utilisateurs est fixé à 1.
 - d) La classe doit avoir une méthode *afficher* qui affiche la valeurs de chacune des propriétés.
 - e) La classe doit avoir une méthode *info* qui retourne une chaîne de caractères qui inclut la valeur courante de chacune des propriétés.
 - f) La classe doit avoir une méthode d'accès et une méthode d'altération pour le nombre d'utilisateurs.
 - g) La classe doit avoir une méthode *valeurRevente* qui calcule et retourne la valeur estimée de revente du logiciel usagé, en suivant la règle suivante : les logiciels de moins de 100\$ peuvent être revendus à leur coût original, tandis que les autres peuvent être revendus à 90% de leur coût original.
 - h) La classe doit avoir une méthode *ajouterVersion* qui prend en paramètre une valeur réelle représentant un numéro de version. Ce numéro doit être concaténé au nom courant du logiciel (le nom du logiciel est donc dorénavant modifié).
 - i) L'instruction *System.out.println(logicielPrefere)* , où *logicielPrefere* est de type *Logiciel*, doit afficher la valeur courant de chacune des propriétés.
 - j) La classe doit permettre de comparer deux objets de type *Logiciel*. Deux logiciels sont considérés égaux s'ils ont le même nom.
- 2) Codez la méthode *main* de la classe *Main* qui fait appel à au moins une reprise à chacune des méthodes de la classe *Logiciel*, dans un contexte pertinent.
- 3) Codez la classe *Mat* dont la méthode *main* permet de créer un tableau à 2 dimensions de NB lignes et de NB colonnes dont tous les éléments sont égaux à 0, au départ de chacun des numéros a), b) et c) et **d'affecter** les valeurs indiquées aux éléments de la

matrice. Votre programme doit produire les résultats attendus **peu importe les dimensions** de la matrice carrée.

Afin de faciliter la correction, utilisez la variable *lig* pour l'indice de ligne et la variable *col* pour l'indice de colonne.

a) 1 0 0 0 0 Si la valeur de NB est égale à 5
 0 2 0 0 0
 0 0 3 0 0
 0 0 0 4 0
 0 0 0 0 5

b) 1 2 3 4 Si la valeur de NB est égale à 4
 1 2 3 4
 1 2 3 4
 1 2 3 4

c) 1 2 3 4 5 6 Si la valeur de NB est égale à 6
 0 1 2 3 4 5
 0 0 1 2 3 4
 0 0 0 1 2 3
 0 0 0 0 1 2
 0 0 0 0 0 1

4) Complétez le code de la méthode *main* de la classe *MotAleatoire* qui génère un mot aléatoirement selon les spécifications ci-dessous. La méthode doit :

- Lire le nombre de lettres du mot à générer (nom de variable *longMot*)
- Alternier les consonnes (b, c, d,...x, z) et les voyelles (a, e, i, o, u, y)
- Commencer par une consonne
- Faire en sorte qu'une lettre n'apparaisse pas plus d'une fois dans le mot généré (nom de variable *motGenere*)
- La méthode *Math.random()* génère un nombre aléatoire entre 0.0 et 0.9999

```
public static void main(String[] args) {
    String motGenere, consonnes, voyelles;
    int longMot, noCar;
    char car;

    consonnes= "bcdfghjklmnpqrstvwxyz";
    voyelles= "aeiouy";
    longMot= Integer.parseInt(
                JOptionPane.showInputDialog("Entrez la
                longueur"));
    motGenere = "";

    // à compléter

    System.out.println ("Mot généré = " + motGenere);
}
```