

Cours et photocopiés autorisés. Autres documents interdits

On prendra soin d'écrire le code java le plus propre possible, en particulier en ce qui concerne les spécificateurs et les exceptions.

Exercice 1

On considère la classe :

```
public class Utilitaire {
    public double carre(double x) {
        return x*x;
    }

    public static void main(args[]) {
        double y= 5.0;
        System.out.println("le carré est "+ carre(y));
    }
}
```

Question 1.1

Ce code ne compile pas. Pourquoi ? corrigez l'erreur.

Question 1.2

Peut-on remplacer 5.0 par 5 dans la ligne

```
double y= 5.0;
```

Justifiez votre réponse.

Question 1.3

Soit la classe Test :

```
public class Test {
    public static void main(args[]) {
        // ... Afficher la valeur de 4.5 au carré en utilisant la méthode
        // définie dans Utilitaire.
    }
}
```

complétez la classe.

Exercice 2

Écrire le programme EnMajuscule. Celui-ci prend deux arguments (des noms de fichiers textes). Il copiera le premier dans le second en mettant toutes les lettres en majuscules.

Noter que si le second fichier existait déjà, son ancien contenu sera perdu.

Si un problème se produit (en particulier que le premier fichier n'existe pas), le programme doit avertir l'utilisateur.

Exercice 3

Question 3.1

Écrire le programme `CompteurBinaire`, qui compte combien de fois un octet donné (passé en argument) apparaît dans un fichier.

L'appel du programme se fera de la manière suivante :

```
java CompteurBinaire toto 56
```

Pour chercher combien de fois l'octet 56 apparaît dans le fichier `toto`.

Question 3.2 (*pas facile*)

Écrire le programme `Compteur`, qui compte combien de fois une chaîne de caractères donnée (passée en paramètre) apparaît dans un fichier texte. Exemple d'utilisation :

```
java Compteur et, toto.txt
```

renverra le nombre de fois où la chaîne « et, » apparaît dans le fichier `toto.txt`.

Exercice 4

On suppose qu'un fichier binaire `toto.dat` a été rempli avec des entiers en utilisant un `RandomAccessFile`

Écrire un programme qui prendra en argument un entier, et qui supprimera toutes les occurrences de cet entier dans le fichier `toto.dat`. La taille finale du fichier devra tenir compte des suppressions.

Notez l'existence dans `RandomAccessFile` de la méthode

```
void setLength (long newLength)  
                throws IOException  
    fixe la longueur du fichier à newLength.
```

Documentation

On rappelle quelques méthodes. Certaines vous seront utiles ici, d'autres pas.

Dans la classe `String`

```
char charAt (int index)  
    retourne le caractère numéro i de la chaîne.
```

Dans la classe `Character`

```
static boolean isDigit (char ch)  
    Détermine si ch est un chiffre.
```

```
static boolean isLetter (char ch)  
    Détermine si ch est une lettre.
```

```
static boolean isLowerCase (char ch)  
    Détermine si ch est une lettre en minuscules.
```

static boolean **isUpperCase** (char ch)
Détermine si ch est une lettre en majuscules.

static boolean **isWhitespace** (char ch)
Détermine si ch est un caractère d'espacement.

static char **toLowerCase** (char ch)
Convertit ch en minuscules.

static char **toUpperCase** (char ch)
Convertit ch en majuscules.