

```

1  Charset Charset.defaultCharset() //renvoie l'encodage du systeme
   byte[] maString.getBytes(Charset e) //code maString avec e
3  new String(byte[] tab, Charset e) //cree une nouvelle String represente
   // par tab dans l'encodage e
5  new String(byte[] tab) //nouvelle chaine codee avec l'encodage systeme

7  Locale.getDefault(); //renvoie la langue du systeme
   new Locale(String langue, String pays); //deux caracteres chacun
9  new Locale(String langue) //sans pays
  
```

Antis che

Exercice 1.  crire une classe `Convertisseur` qui fournit la m thode statique `reencode`. Cette m thode prend en argument une cha ne mal-cod  et deux encodages, et renvoie la cha ne correctement cod . Elle a  t  interpr t e en utilisant le premier encodage, alors qu'elle  tait encod e dans le second.

Exercice 2.  crire une classe `StringMultiLangues` qui repr sente une cha ne de caract res traduite dans plusieurs langues; il faut donc une m thode pour ajouter des langues. On ne peut pas savoir   l'avance quelle sera la langue du syst me de l'utilisateur, r impl menter la m thode `toString()` (h rit e de `Object`) de fa on   ce que ce soit toujours la cha ne dans la bonne langue qui soit renvoy e.

Dans un second temps, rendre cette classe plus s re : traiter le cas o  la cha ne n'est pas traduite dans la langue du syst me et r pondre au mieux (s'il n'y a pas de traduction qu becoise ou belge, la traduction en fran ais de France ou de Suisse est meilleure que la traduction anglaise).

Exercice 3. Nous voulons effectuer les deux op rations suivantes.

- Parcourir un r pertoire et tous ses sous-r pertoires et afficher le nom de tous les fichiers qu'ils contiennent.
- Parcourir un r pertoire et tous ses sous-r pertoires et rendre ex cutable chaque fichier qui termine par `".exe"`.

Proposer une mani re de faire cela. (Attention, factoriser le code!).

Exercice 4. D river la classe `BufferedReader` en `BufferedReaderEncodage` qui a un constructeur prenant une `File` et deux `Charset` (`in` et `out`). Il s'agit d' crire une m thode `readLineByte()` pour qui renvoie un tableau de `byte` utilisant l'encodage `out`, sachant que le fichier est encod  dans l'encodage `in`. (En r alit , il faudrait utiliser `FileInputStream` et `InputStreamReader` mais pour le besoin de l'exercice, passons-nous en.)

Exercice 5.  crire une m thode qui prend en entr e un fichier `un/chemin/nom` dont l'encodage est celui par d faut du syst me, et qui copie son contenu dans le fichier `un/chemin/utf8/nom_utf8` en changeant son encodage en UTF8 et les fins de lignes en `\n`.

Exercice 6 (Pr -calcul de `StringMultiLangues`).  crire une classe `PreCalculateur` qui contient

- un attribut `localeList` qui est une liste de `Locale`;
- une m thode prenant en argument deux `File` (`in` et `out`)
- qui construit une `StringMultiLangues`, en suivant la convention que la i - me ligne de `in` correspond au message dans la i - me langue de `localeListe`;

- puis écrit cet objet dans `out` ;
- le fichier `in` est donc fourni par vous alors que le fichier `out` est destiné à n'être utilisé que sur cette machine.

La méthode proposé dans cet exercice est incohérente par rapport au service rendu. Pourquoi ? et que devrait-on plutôt faire ?