Les nombres romains

M.C. Lagasquie

10 octobre 2013

On souhaite étudier la manipulation de nombres représentés soit comme un entier, soit comme une chaine de caractères. Pour cela on se donne une classe Nombre, et des tableaux de correspondance entre valeur entière et représentation sous forme de chaînes de caractères. (note : pour tester on prendra des nombres strictement inférieurs à 4000). Dans un premier temps, on considère les nombres romains. L'algorithme de conversion d'`entier` vers `nombre romain` est le suivant :

```
int reste = valeur initiale;
String representation = "";
Pour tout i appartenant à [0, VALEURS_ROMAINS.length[ Faire
    Tant que reste est supérieur ou égal à VALEURS_ROMAINS[i] Faire
        representation = representation + REP_ROMAINS[i]; // concat
        reste = reste - VALEURS_ROMAINS[i];
    Fin Tant que
Fin Pour
```

L'algorithme de conversion de `nombre romain` vers `entier` est le suivant :

```
int valeur = 0;
String rep = representation initiale;
Pour tout i appartenant à [0, REPRESENTATION_ROMAINS.length[ Faire
   Tant que representation commence par REP_ROMAINS[i] Faire
       valeur = valeur + VALEURS_ROMAINS[i];
       enlever à rep les caractères de REP_ROMAINS[i];
   Fin Tant que
Fin Pour
```

1. Appliquer les 2 algorithmes pour convertir 402 en nombre romain et CMXII en nombre entier. A partir des exemples, déduire les tables de correspondance entre valeur et représentation VALEURS_ROMAINS et REPRESENTATION_ROMAINS.
2. Toutes les représentations à base de chiffres romains ne sont pas valides. Par exemple, 9 se représente "IX" et pas "VIIII" qui est incorrect. Proposer une solution permettant de vérifier la validité d'un nombre romain saisi par l'utilisateur au clavier, sous la forme d'une chaîne de caractères toute en majuscule.
3. Ecrire le code de la classe NombreRomainInit à partir du javadoc suivant :

Field Summary	
static int	NOMBRE_MAX Borne max, juste pour simplifier
private static java.lang.String[]	REPRESENTATION_ROMAINS constantes String représentant les nombres romains
protected int	valeur la représentation interne entière du nombre
private static int[]	VALEURS_ROMAINS constantes int représentant les entiers correspondant aux nombres romains

Constructor Summary
NomBreRomainInit(int value) construit le nombre en stockant sa valeur entière
NomBreRomainInit(java.lang.String s) convertit un nombre romain valide en entier et stocke sa valeur

Method Summary	
java.lang.String	intToNombre() Retourne la représentation romaine de la valeur entière
void	moins(Nombre n) soustraction d'un nombre au nombre courant
int	nombreToInt() Retourne la représentation entière
void	plus(Nombre n) addition d'un nombre au nombre courant

4. On veut maintenant écrire la classe `NombreEntier` sur le même schéma que la classe `NombreRomain` qui à partir d'un `int` saisi au clavier par l'utilisateur va le convertir en chaîne de caractères et à partir d'une chaîne de caractères va calculer sa valeur entière (SANS utiliser les fonctions de conversion prédéfinies de Java telles que `parseInt` ou `toString`). Pour cela, on va généraliser dans une classe parente abstraite `Nombre`, tout ce qui est commun à `NombreRomain` et `NombreEntier`. Ce qui est spécifique étant soient les constructeurs, soient des méthodes abstraites, soient les constantes déclarées dans chaque classe fille. Définir la classe `Nombre` à partir des éléments communs aux représentations de nombre.
5. Donner les tableaux de correspondance nécessaires aux conversions de entier vers entiers.
6. Ecrire le code de la classe `NombreEntier`.
7. Ecrire le code de la classe `NombreRomain` en tenant compte de la classe `Nombre`.