IUT Lannion **S2 (P5)**
Département informatique

TP3 (M213)

Exercice 1/

Cet exercice a pour but de définir une méthode de classe qui fait la moyenne de notes (des entiers) passées en argument de la ligne de commande.

Question 1/ Proposez une classe munie d'une méthode de classe `moyenne` qui délivre la moyenne des notes passées par la ligne de commande. Le paramètre de cette méthode est un tableau de `String` qui contient les paramètres de la ligne de commandes. :

```
static int moyenne(String [] valeurs)…
```

La conversion en type primitif `int` de chaque argument se fera en utilisant la méthode de classe `parseInt` de la classe `Integer`. On utilisera l'exception `NumberFormatException` pour vérifier que l'argument est bien du type attendu. Ainsi, si un argument ne représente pas un entier, `l'`exception est levée mais le calcul continue avec les arguments qui suivent.

Question 2/ Définissez la classe `MonException`. Le constructeur affichera un message significatif à l'écran. Prévoyez également de redéfinir `toString()`. Vous utiliserez cette exception (`MonException`) pour vérifier que le nombre de notes passées en argument est bien différent de zéro.

Exercice 2/

On souhaite coder du texte en remplaçant certains mots par d'autres mots. Ainsi, si le codage du mot "aujourd'hui" est "AZ" et celui du mot "beau" est "QS", le codage de la phrase :

"aujourd'hui il fait beau"

devient :

"AZ il fait QS".

Un codeur peut se voir comme un objet qui gère des associations (mot, codage_du_mot) dans une instance de `HashMap` et qui possède une méthode d'instance `public String Coder(string CH1)` qui délivre le résultat du codage de la chaîne CH1.

Question 1/ Proposez la classe `Codeur` (prévoir un constructeur, une méthode pour ajouter une association et la méthode `Coder`).

Question 2/ Modifier la méthode précédente qui devient :

```
public int Coder(string CH1, String CH2)
```

où la chaîne `CH1` est en entrée et est susceptible de contenir des nombres. La chaîne `CH2` est le résultat du codage, comme précédemment, et le résultat de la méthode est la somme des entiers apparaissant dans la chaîne `CH1`. Pour découper la chaîne en mots, on utilisera une instance de `StringTokenizer`.

Par exemple, si `CH1` = "aujourd'hui à 18 heures il fait beau et la température est de 30 degrés" alors `CH2` = "AZ à 18 heures il fait QS et la température est de 30 degrés" et le résultat délivré par la méthode est de 48 (18+30).

Par exemple, si `CH1` = "aujourd'hui il fait beau" alors `CH2` = "AZ il fait QS" et le résultat délivré par la méthode est de 0.

Pour savoir si un mot de la chaîne est un entier ou non, on utilisera la méthode public `static int` **`parseInt`**`(String s)` de la classe Integer :

public static int **parseInt**(String s)
throws NumberFormatException

Parses the string argument as a signed decimal integer.
Parameters: `s` - a `String` containing the `int` representation to be parsed
Returns: the integer value represented by the argument in decimal.
Throws: `NumberFormatException` - if the string does not contain a parsable integer.

L'exception `NumberFormatException` sera traitée pour savoir si un mot est une chaîne ou si il correspond à un entier.

Exercice 3/

On souhaite définir des listes d'objets où il est possible d'appliquer une méthode à tous les éléments de la liste. Par exemple, si L est une instance de liste, la méthode `appliquer` :

```
L.appliquer("plusun ") ;
```

permet d'appliquer la méthode `plusun` à chacun de ses éléments. Si la méthode n'est pas définie pour certains objets de la liste, un message d'erreur apparaît à l'écran mais le traitement continu.

Question 1/ Proposer la classe Liste (sous classe de `LinkedList`).

Question 2/ Proposer la méthode de la classe `Liste` :

```
public void appliquer(String nomMethode) ;
```

qui permet d'appliquer la méthode de nom `nomMethode` à chaque élément de la liste. Si la méthode n'est pas définie pour certains objets de la liste, un message d'erreur apparaît à l'écran mais le traitement se poursuit. Etudiez les méthodes des classes `Class` et `Method`.