BPO
TP 9 - les flots

Avertissement : Les questions de la première partie premier sont destinées à vous familiariser avec l'utilisation des flots ; les classes ne sont pas structurées pour gagner du temps. Il est inutile d'en faire plus que demandé ; l'essentiel est de savoir retrouver les bonnes classes/fonctions à utiliser. La seconde partie concerne la classe ***PaquetDeCartes****.*

Partie 1 : quelques tests sur les flots

1. Dans la fonction **main** de la classe **tests.EcrireIdentite**, écrire une séquence d'instructions qui crée un fichier et écrit dans ce fichier vos nom, prénom et adresse, un élément par ligne. Le nom du fichier est fixé dans le code.

   Un peu d'aide : utiliser la page 16 du cours

   exemple d'appel : **java tests.EcrireIdentite**

2. Dans la fonction **main** de la classe **tests.LireIdentite** écrire une séquence d'instructions qui relit complètement le fichier et affiche sur la sortie standard le nombre de chaînes lues. Le nom du fichier est fixé dans le code.

   Un peu d'aide : utiliser la page 22 du cours

   exemple d'appel : **java tests.LireIdentité**

3. Dans la fonction **main** de la classe **tests.Ecrire**, écrire une séquence d'instructions qui crée le fichier **nb.txt** et écrit dans ce fichier les nombres entiers de 1 à 1000, un par ligne.

   Un peu d'aide : utiliser la page 16 du cours

   exemple d'appel : **java tests.Ecrire**

4. Pour éviter que le nom de fichier soit codé en dur dans le programme, on le passe en argument de la ligne de commande ; ces arguments se retrouvent dans le paramètre de la fonction **main**. Modifier les instructions de la classe **tests.Ecrire** pour utiliser ce nom de fichier.

   Un peu d'aide : utiliser la classe ***File*** *pour tester l'existence du fichier*

   exemple d'appel : **java tests.Ecrire monFichier.txt**

5. Dans la fonction **main** de la classe **tests.Lire**, écrire une séquence d'instructions qui relit un fichier et affiche sur la sortie standard le nombre d'entiers lus et la moyenne des nombres. Le nom du fichier est en argument de la ligne de commande.

   Un peu d'aide : utiliser la classe ***Scanner*** *pour analyser chaque ligne*

exemple d'appel : **java tests.Lire monFichier.txt**

6. Dans **tests.Ecrire**, si le fichier existe déjà, déclencher une exception **IOException** ; dans **tests.Lire**, si le fichier n'existe pas, déclencher une exception **IOException.** Lors de la capture de l'exception, on affiche un message d'erreur.

   *Un peu d'aide : **throw** permet déclencher une exception ; **try .. catch** capture une exception*

7. Dans la fonction **main** de la classe **tests.EcrireData**, générer un fichier dont le nom est fourni en argument de la ligne de commande ; ce fichier contient un nombre de lignes pris au hasard entre 10 et 30 Chaque ligne du fichier doit contenir une lettre minuscule suivie d'au moins un espace et d'un nombre entier ; la première lettre est prise au hasard dans l'alphabet, les autres lettres se suivent (après **z**, on repart à **a**). Le nombre entier est pris au hasard entre -5 et 7.

8. Dans la fonction **main** de la classe **tests.LireData**, écrire une séquence d'instructions qui relit un fichier créé par **tests.EcrireData** et affiche sur la sortie standard la fréquence de chaque lettre.

Partie 2 : écrire/lire des cartes

Ajouter la classe **PaquetDeCartes** dans le projet.

On souhaite compléter la classe **PaquetDeCartes**, pour permettre l'écriture / la lecture d'un paquet de cartes dans un fichier. Sur une ligne du fichier, se trouve la description d'une seule carte, c'est-à-dire son nom, son numéro, suivi d'un seul espace, suivi de sa couleur.

Dans la classe **PaquetDeCartes**, écrire la fonction **void ecrire(String nomDeFichier) throws ErreurFichier** qui permet d'écrire toutes les cartes du paquet dans le fichier dont le nom est passé en paramètre. Si la sauvegarde pose un problème, cette fonction déclenche l'exception **ErreurFichier**.

Ajouter la fonction **void lire(String nomDeFichier) throws ErreurFichier** qui permet d'initialiser un paquet de cartes avec toutes les cartes décrites dans le fichier dont le nom est passé en paramètre. Si la lecture pose un problème, cette fonction déclenche l'exception **ErreurFichier**.

Il n'est pas simple de tester ces deux fonctions comme on le fait habituellement. Pour tester **ecrire**, il faut ouvrir le fichier avec un éditeur de texte et vérifier son contenu. Pour tester **lire**, il faut utiliser un fichier préalablement construit avec **ecrire**.