

BPO

TP 9 - les flots

Avertissement : ces petits exercices sont destinés à vous apprendre l'utilisation des flots ; les classes ne sont pas structurées pour gagner du temps ...

Exercice 1 :

1. dans la fonction **main** de la classe **tests.EcrireSemaine**, écrire une séquence d'instructions qui crée un fichier et écrit dans ce fichier les 7 jours de la semaine, un nom par ligne. Le nom du fichier est codé en dur.

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

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

2. dans la fonction **main** de la classe **tests.LireSemaine**, écrire une séquence d'instructions qui relit le fichier et affiche sur la sortie standard le nombre de chaînes lues. Le nom du fichier est codé en dur.

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

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

Exercice 2 :

1. dans la fonction **main** de la classe **tests.Ecrire**, écrire une séquence d'instructions qui crée un fichier et écrit dans ce fichier les nombres entiers de 1 à 1000, un par ligne. Le nom du fichier est codé en dur.

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

2. le nom du fichier est le premier argument de la ligne de commande ; ces arguments sont 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**

3. dans la fonction **main** de la classe **tests.Lire**, écrire une séquence d'instructions qui relit le 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**

4. 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**. Dans la capture, on affiche un message d'erreur.

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

5. le nombre d'entiers à écrire dans le fichier est pris au hasard entre 500 et 1000. Modifier **tests.Ecrire** en conséquence. La classe **tests.Lire** fonctionne-t-elle encore ? Si oui, elle était donc bien écrite. Si non, il faut la corriger ...

Exercice 3 :

1. 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 50 et 100. Chaque ligne du fichier doit contenir une lettre majuscule 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 -100 et 100.
2. 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.