

TD de programmation orientée objet en Java

Les objets

Exercice 1 : Calcul d'un coefficient binomial arbitraire long

Écrire un programme intitulé `BinomialCoefficient` qui demande à l'utilisateur de saisir deux entiers arbitraires longs n et k ($0 \leq k \leq n$) puis affiche le coefficient binomial C_n^k , calculé en utilisant la formule suivante :

$$C_n^k = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2) \cdots (n-k+1)}{k(k-1)(k-2) \cdots 1}$$

Afin de pouvoir traiter des entiers de grandes valeurs, vous devez utiliser des objets de type `BigInteger`¹ ainsi que l'ensemble des méthodes (d'instance et de classe) qu'ils mettent à votre disposition. L'exemple suivant montre un résultat d'exécution :

```
n = 50
k = 30
C(50, 30) = 47129212243960
```

Le programme demande à l'utilisateur de saisir n (ici l'utilisateur choisit 50) puis k (ici 30) et affiche le coefficient binomial correspondant.

Exercice 2 : Conversion d'une chaîne de caractères en un entier

Écrire un programme qui s'appelle `StringToInt` qui demande à l'utilisateur de saisir une chaîne de caractères correspondant à un entier. La chaîne de caractères doit impérativement être lue avec la méthode `nextLine()` des objets de type `Scanner` et non pas avec la méthode `nextInt()`. Le programme doit alors convertir cette chaîne de caractères en un entier et l'afficher à l'écran. Attention ! Votre programme doit vérifier que la chaîne de caractères saisie correspond bien à un entier. Dans le cas contraire, le programme affichera un message d'erreur et interrompra son exécution. Un déroulement correct du programme doit produire un affichage comme suit :

```
Saisissez un entier: 5367
n = 5367
```

L'utilisateur a saisi le nombre 5367 qui a été lu par le programme par une instruction du type `String s = scan.nextLine();`. Le programme s'est chargé ensuite

de convertir `s` en un entier qu'il a affiché à l'utilisateur. Par contre, lorsque l'utilisateur saisit une chaîne de caractères qui n'est pas valide (ex. contenant une lettre), le programme lui indique que cette chaîne ne correspond pas à un entier :

```
Saisissez un entier: 347k24
Votre chaîne de caractères ne correspond pas à un entier!
```

Quelques éléments d'aide :

- Vous pouvez accéder aux différents caractères composant la chaîne grâce à la méthode d'instance `charAt(int i)` des objets de type `String`;
- Pour vérifier si un caractère correspond bien à un chiffre, on peut utiliser la méthode de classe `Character.isDigit(char c)`;
- Pour convertir un caractère en un entier correspondant au chiffre qu'il représente, on peut utiliser la méthode de classe `Character.getNumericValue(char c)`.

Exercice 3 : Inverser une chaîne de caractères

Écrire un programme intitulé `InvertString` qui inverse une chaîne de caractères saisie par l'utilisateur et l'affiche à l'écran. Pour ce faire, vous utiliserez des objets de type `StringBuilder` qui sont des objets qui permettent de stocker des chaînes de caractères (à l'instar des objets de type `String`) et qui mettent à votre disposition plein de méthodes pour manipuler ces chaînes. Notamment, la méthode d'instance `append(char c)` qui peut vous être très utile pour atteindre votre objectif. Un affichage possible est donné par l'exemple suivant :

```
Entrez une chaîne de caractères: Bonjour
L'inversion de cette chaîne donne: ruojnoB
```

Exercice 4 : Recherche d'un mot dans un texte

Écrire un programme intitulé `WordSearch` qui permet à l'utilisateur de saisir un texte et un mot à chercher à l'intérieur de ce texte. L'utilisateur commence par saisir le texte à inspecter en premier (on supposera que celui-ci ne contient que des mots séparés par des espaces et ne contient pas de signes de ponctuation). Ensuite, il saisit le mot recherché. Le programme doit signaler à l'utilisateur la position de chaque occurrence du mot dans le texte s'il arrive à le trouver et indiquer, au cas

1. cf. <http://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html>

contraire, que le mot est introuvable.

L'exemple suivant montre le déroulement attendu du programme. Ce dernier sollicite l'utilisateur pour donner le texte à fouiller puis demande le mot à chercher. Il affiche pour chaque occurrence la position du mot dans le texte (ici une seule occurrence à la 8e position).

```
Texte à fouiller:
Lorem ipsum dolor sit amet consectetur adipiscing elit Sed non risus
    Suspendisse lectus tortor dignissim sit amet adipiscing nec ultricies
    sed dolor
Mot à trouver:
elit
Le mot elit apparait à la place no.8.
```

Exercice 5 : Produit matriciel

Écrire un programme nommé `MatrixMultiplication` qui permet de calculer le produit matriciel ordinaire de deux matrices² d'entiers. Les deux matrices sont stockées dans deux fichiers séparés. Le programme demandera donc à l'utilisateur de saisir le chemin vers chacun des deux fichiers et se chargera de les lire (sous forme d'un tableau à deux dimensions `int[][]`). La première ligne du fichier contenant la matrice spécifie le nombre de lignes et le nombre de colonnes de celle-ci tandis que les lignes suivantes spécifient son contenu. L'exemple suivant illustre le contenu d'un fichier décrivant une matrice de taille 4 x 3 :

```
4 3
1 2 0
4 3 -1
5 -3 1
8 6 7
```

Le programme vérifiera si les deux matrices sont compatibles puis calculera le produit et l'affichera à l'écran. Un affichage possible est donné dans l'exemple suivant³ :

```
Entrez le chemin complet vers le fichier contenant la première matrice :
/home/rossi/Documents/A.txt
Entrez le chemin complet vers le fichier contenant la deuxième matrice :
/home/rossi/Documents/B.txt
| 1 2 0 | x | 5 1 | = | 9 7 |
| 3 -1 9 | | 2 3 | | 23 9 |
          | 3 4 |
```

2. cf. http://fr.wikipedia.org/wiki/Produit_matriciel#Produit_matriciel_ordinaire

3. Vous pouvez récupérer les fichiers des deux matrices à l'adresse <http://ihazocd.org/teaching/oop/labs/matrices.zip>

Exercice 6 : Manipulation d'un fichier CSV

Écrire un programme appelé `ManipCSV` qui prend en entrée le fichier CSV (Comma-Separated Values) suivant⁴ :

```
Id,Prénom,Couleur,Age
1,Titi,Jaune,5
2,Médor,Noir,10
3,Pitié,Noir,5
4,Juju,Gris,5
5,Vanille,Blanc,7
6,Chocolat,Marron,12
7,Milou,Blanc,3
8,Idefix,Blanc,14
9,Pluto,Jaune,17
10,Dingo,Roux,1
```

Ce fichier contient des données décrivant des chiens (leurs prénoms, âge et couleur). Le programme se charge de lire le fichier en utilisant des objets de type `BufferedReader` et `FileReader` puis affichera à l'utilisateur la moyenne d'âge de ces chiens, la couleur la plus observée ainsi que les prénoms des chiens ayant cette couleur. Le programme doit produire un affichage similaire à l'exemple suivant :

```
Saisir le chemin vers votre fichier CSV :
/home/rossi/Documents/chiens.txt

Moyenne d'âge : 7.9
Couleur la plus observée : Blanc
Les chiens ayant la couleur Blanc sont: Vanille, Milou, Idefix
```

4. Vous pouvez télécharger le fichier à l'adresse <http://ihazocd.org/teaching/oop/labs/chiens.txt>