

Saisir des données envoyées par le clavier



Si vous voulez lire des données envoyées tout simplement par le clavier, nous vous indiquons deux méthodes illustrées ci-dessous.

Les deux programmes d'illustration ont pour objectif de lire des entiers envoyés par l'intermédiaire du clavier, et d'en faire la somme. Leurs cahiers des charges diffèrent uniquement sur la façon d'indiquer la fin de la saisie.

Nous notons en rouge les instructions significatives sur le thème de la saisie des données au clavier.

Première méthode : utiliser un Bufferedreader

On utilise deux classes de java.io, la classe InputStreamReader et la classe BufferedReader.

La classe InputStreamReader admet un constructeur InputStreamReader(InputStream), c'est-à-dire un constructeur qui admet en paramètre un flot d'entrée. System.in est une instance de la classe InputStream. Avec une instance de InputStreamReader, on ne peut grosso modo que lire des caractères un à un.

La classe BufferedReader a un constructeur qui prend en argument une instance de Reader dont hérite la classe InputStreamReader. Cette classe permet en particulier de lire une ligne d'un texte, mais en revanche, on ne peut pas lui demander de lire un entier, un double etc...

On lit ici ce qui est envoyé par le clavier ligne par ligne et on découpe le contenu de chaque ligne avec un StringTokenizer pour récupérer les entiers attendus.

L'utilisateur devra taper return deux fois de suite pour interrompre la saisie.

Voici le programme que nous proposons.

```
import import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.util.StringTokenizer;

class SaisieClavier
{
    public static void main (String[] argv) throws IOException,
        NumberFormatException
    {
        int somme = 0;
        String ligne;
        StringTokenizer st;
        BufferedReader entree = new BufferedReader
            (new InputStreamReader(System.in));

        ligne = entree.readLine();
        while(ligne.length() > 0)
        {
            st = new StringTokenizer(ligne);
            while(st.hasMoreTokens())
                somme += Integer.parseInt(st.nextToken());
            ligne = entree.readLine();
        }
        System.out.println("La somme vaut : "+somme);
    }
}
```

Voici une execution :

```
$ java SaisieClavier
3 1
2

La somme vaut : 6
```

Vous pouvez [récupérer ce programme](#).

Deuxième méthode : utiliser un Scanner

Depuis le JDK 5.0 , la saisie à partir de la fenêtre d'exécution est plus simple. La classe java.util.Scanner, qui possède différents constructeurs dont un de type InputStream que nous utilisons dans notre exemple, permet de saisir différents types de données.

```
class SaisieClavierBis {
    public static void main (String[] arg) {
        java.util.Scanner entree = new java.util.Scanner\(System.in\);

        System.out.println("Donnez votre prénom et votre nom");
        String prenom = entree.next\(\);;
        String nom = entree.next();
        System.out.println("Donnez votre âge");
        int age = entree.nextInt\(\);;
        entree.nextLine();
        System.out.println("Ecrire votre phrase");
        String phrase = entree.nextLine\(\);;
        System.out.println(prenom + " " + nom + ", " +
                           age + " ans, dit : " + phrase);
    }
}
```

Voici une execution :

```
Donnez votre prénom et votre nom
Lou
Charon
Donnez votre âge
12
Ecrire votre phrase
Coucou de Lou
Lou Charon, 12 ans, dit : Coucou de Lou
```

Vous pouvez [récupérer ce programme](#).

new java.util.Scanner(System.in); : on instancie la clase Scanner en indiquant par le constructeur le flux de données issu du clavier.

entree.next() : renvoie la prochaine chaîne de caractères ; par défaut, les séparateurs des chaînes de caractères sont les caractères d'espacement (espace, saut de ligne, tabulation...).

entree.nextInt()entree.nextInt() : pour que cet appel fonctionne, il est nécessaire que la prochaine donnée corresponde à un entier, que la méthode renvoie ; sinon la méthode lancera une exception.

entree.nextLine(); : renvoie la prochaine ligne ou la fin de la ligne courante ; ici, on utilise cette instruction pour ignorer la fin de la ligne ayant servi à donner l'âge, cette fin ligne ne contiendra sans doute à l'exécution que le caractère de fin de ligne.