Les Entrées-Sorties

Dr Pélagie HOUNGUE

Introduction

- Pour obtenir des données, un programme ouvre un flux de données sur une source de données (fichier, clavier, etc)
- De la même façon pour écrire des données dans un fichier, un programme ouvre un flux de données
- Java fournit le paquetage *java.io* qui permet de gérer les flux de données en entrée et en sortie, sous forme de caractères (exemple fichiers textes) ou sous forme binaire (octets, byte)
- La classe Scanner du paquetage java.util est aussi fournie pour les mêmes causes.

Introduction (2)

- Java fournit plusieurs hiérarchies de classes pour gérer les flux de données
- Pour les flux binaires :
  - La classe *InputStream* et ses sous-classes pour lire des octets (*FileInputStream*)
  - La classe *OuputStream* et ses sous-classes pour écrire des octets (*FileOuputStream*)
- Pour les flux de caractères :
  - La classe *Reader* et ses sous-classes pour lire des caractères

  (*BufferedReader, FileReader*)
  - La classe *Writer* et ses sous-classes pour écrire des caractères (*BufferedWriter, FileWriter*)
- Pour les autres types primitifs et les chaines :
  - La classe Scanner

La classe java.util.Scanner

- La classe Scanner permet la lecture de données sur l'entrée standard (clavier) ou dans un fichier.
- Un objet de type Scanner, scanne les données de types primitifs ou de la classe String.
- Pour utiliser la classe Scanner, il faut d'abord l'importer avec
  - import java.util.Scanner;
- Pour lire des données sur l'entrée standard, il faut créer un objet de la classe
  - Scanner ob = new Scanner(System.in);
- Pour ensuite récupérer les données saisies, il faut faire appel sur l'objet ob avec des méthodes.

La classe java.util.Scanner (2)

- **Scanner ob = Scanner(System.in);**
  - boolean b = ob.nextBoolean();
  - byte by = ob.nextByte();
  - short sh = ob.nextShort();
  - int i = ob.nextInt();
  - long l = ob.nextLong();
  - float f = ob.nextFloat();
  - double d = ob.nextDouble();
  - String s = ob.nextLine(); //délimité par le caractère de validation
  - String s = ob.next(); //délimité par l'espace
- Par défaut, les espaces aguissent comme des délimiteurs

Exemple 1

```
//Lire le  nom et l'âge d'une personne au clavier
//et les afficher
import java.util.Scanner;
class Bonjour{
public static void main (String[] args){
Scanner ob = new Scanner(System.in);
System.out.println("Entrez votre nom : ");
String nom = ob.next();
System.out.println("Bonjour :" + nom);
System.out.println("Entrez votre âge : ") ;
int age = ob.nextInt() ;
System.out.println("Votre age est :" + age);
}
}
```

Exemple 2

```
// Lire et afficher le carré d'un nombre
import java.util.Scanner; // disponible à partir de java 5
public class Carre {
public static void main(String[] args) {
  double x;
  double result = 0.0;
  Scanner t = new Scanner(System.in);
  System.out.print("Entrer un nombre : ");
  x = t.nextDouble();
  result = x * x;
  System.out.println(x + " au carre = " + result);
  }
  }
```

Lire avec la classe BufferedReader

```
//Lecture d'une chaine et affichage
import java.io.*;
public class Test{
public static void main(String[] argv) {
String chaine =" ";
System.out.println("Veuillez saisir votre nom :");
try {
BufferedReader buff = new BufferedReader(new InputStreamReader(System.in));
//« Convertit » un objet de type InputStream en Reader
chaine = buff.readLine(); //Lit la ligne jusqu'au prochain retour chariot
buff.close();
} catch (Exception e) {
System.out.println("Intercepté : " + e);
}
if (chaine != null)
System.out.println("Votre nom est :" + chaine);
}
}
```

Lire avec la classe BufferedReader (2)

```
//Lecture d'un caractère et affichage
import java.io.*;
public class Test{
public static void main(String[] argv) {
char caract='\0';
System.out.println("Veuillez entrer un caractère:");
try {
BufferedReader buff = new BufferedReader(new InputStreamReader(System.in));
//« Convertit » un objet de type InputStream en Reader
caract = buff.readLine().charAt(0);
buff.close();
} catch (Exception e) {
System.out.println("Intercepté : " + e);
}
if (caract!= '\0')
System.out.println("Vous avez entré le caractère :" + caract);

}
}
```

Ecrire du texte dans un fichier

```
import java.io.*;
public class TestIO {
public static void main(String[] argv) {
try {
FileWriter Fichier = new FileWriter(“text.txt”); //FileWriter
hérite de
//Writer et permet de manipuler un flux texte associé à un
fichier
Fichier.write(“Ceci est ma première ligne dans un fichier");
//Ecriture d’une //ligne de texte dans le fichier «text.txt »
Fichier.close();//Fermeture du flux Fichier
} catch (Exception e) {
System.out.println("Intercepté : " + e);
}
}
}
```

Exemple de lecture et d'écriture dans un fichier

```
import java.io.*;
public class TestIO {
public static void main(String[] argv) {
try {
FileReader in = new FileReader("text.txt");
FileWriter out = new FileWriter("ecrire.txt");
int a;
while ((a = in.read()) != -1) {//read() retourne le code ASCII du caractère lu si tout va
out.write(a); //Transfert de données jusqu'à ce que in ne fournisse plus rien
}
//Fermeture des flux et par conséquent des fichiers respectifs
in.close();
out.close();
} catch (Exception e) {
System.out.println("Intercepté : " + e);
}
}
}
```

Exercice 1

- Ecrire un programme qui demande à l'utilisateur de saisir un nombre entier n positif et non nul. Le programme affiche ensuite, le produit des n premiers nombres entiers positifs non nuls.

Exercice 2

- Ecrire un programme qui demande à l'utilisateur de saisir une série de nombres entiers positifs. Pour arrêter la saisie, l'utilisateur entre la valeur zéro. Puis le programme affiche la plus grande des valeurs et la plus petite.