BUT Informatique 1ère année

R2.02: Développement d'applications avec IHM

Cours 3 : lecture et écriture dans des fichiers (texte, binaire)

Sébastien Lefèvre
sebastien.lefevre@univ-ubs.fr

Vannes iut : Université Bretagne Sud

iutvannes.fr

Vannes
iut :
Université Bretagne Sud

Lire et écrire en mode texte

iutvannes.fr

Ecrire dans une console

```
...
System.out.println("mon premier test");
...
```

On appelle la méthode println de l'attribut out de la classe System ?

➔ Javadoc !

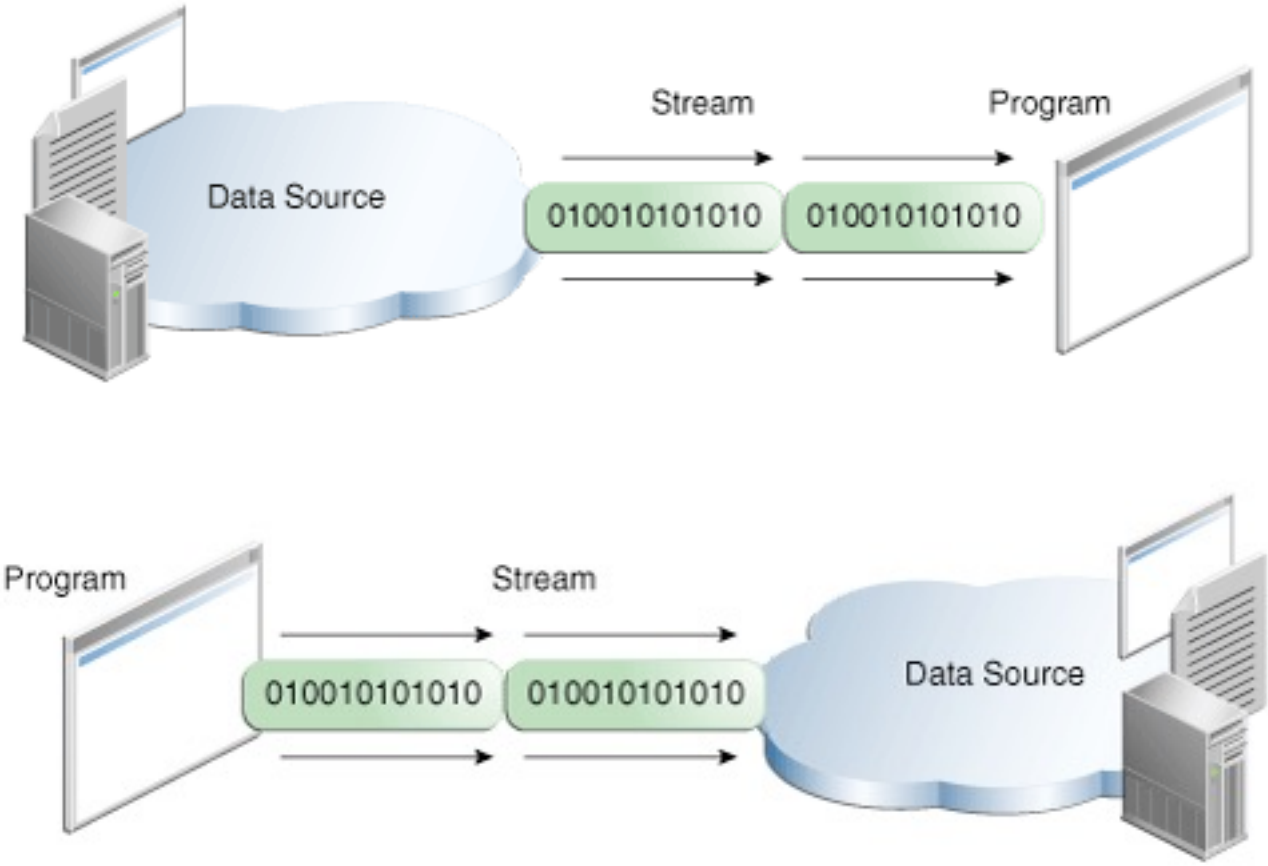
out est un PrintStream

Stream = flot/flux de données
  avec un début et une fin
  en lecture/in ou en écriture/out

De nombreuses classes dans java.io
➔ Javadoc !

4 classes mères abstraites :
  InputStream et OutputStream (bits)
  Reader et Writer (caractères)

Les flux de données



IN = lecture

OUT = écriture

Vannes
iut
Université Bretagne Sud

Ecrire dans un fichier

Idem mais on remplace la console par un fichier

```
...
out.println("mon premier test");
...
```

On appelle la méthode println de l'objet out.

Comment spécifier que out pointe vers le bon fichier ?

➔ Javadoc

Vannes iut : Université Bretagne Sud

Flux en écriture vers un fichier

```
PrintStream out = new PrintStream ("test.txt");
out.println("mon premier test");
```

ou

```
PrintWriter out = new PrintWriter ("test.txt");
out.println("mon premier test");
```

Une solution simple… mais incomplète !

Vannes iut : Université Bretagne Sud

Flux en écriture vers un fichier

```
try
{
 PrintWriter out = new PrintWriter ("test.txt");
 out.println("mon premier test");
}
catch (FileNotFoundException ex)
{
 System.out.println(ex.getMessage());
 // ou ex.printStackTrace();
}
```

Une solution simple… mais incomplète !

Vannes iut : Université Bretagne Sud

Flux en écriture vers un fichier

```
try
{
 PrintWriter out = new PrintWriter ("test.txt");
 out.println("mon premier test");
 out.close();
}
catch (FileNotFoundException ex)
{
 System.out.println(ex.getMessage());
}
```

Une solution simple… mais limitée !

Vannes iut : Université Bretagne Sud

Un meilleur contrôle sur le flux de sortie vers un fichier

Fichier = File…

➔ FileOutputStream, FileWriter, etc

Combiner FileWriter et PrintWriter ?

```
FileWriter f = new FileWriter ("test.txt");
PrintWriter p = new PrintWriter (f);

PrintWriter out =
new PrintWriter (new FileWriter ("test.txt"));

PrintWriter out =
new PrintWriter (new FileWriter ("test.txt", true));
```

Vannes iut : Université Bretagne Sud

Encore plus de contrôle

```
FileWriter file = new FileWriter ("test.txt");
BufferedWriter buf = new BufferedWriter(file);
PrintWriter out = new PrintWriter (buf);
out.println("une ligne à écrire");
out.close();
```

La mise en tampon permet de gagner en efficacité !

Toutes ces classes héritent de Writer

C'est un patron de conception bien connu (décorateur)…
cf. en 2ème année !

Vannes
iut
Université Bretagne Sud

Lire depuis la console

```
Scanner sc = new Scanner(System.in);
String s = sc.nextLine();
```

System.in est un objet de type InputStream.

Peut-on lire directement dans un fichier ?

➔ Javadoc !

Lire depuis un fichier texte

Comme pour l'écriture, plusieurs classes aux fonctionnalités complémentaires

- FileReader
- BufferedReader

```
FileReader file = new FileReader ("test.txt");
BufferedReader in = new BufferedReader(file);
String s=in.readLine();
while(s!=null)
{
 System.out.println(s);
 s=in.readLine();
}
in.close();
```

Vannes
iut
Université Bretagne Sud

Remplacer la classe Scanner

Peut-on écrire une classe Scanner ?

```
BufferedReader br = new BufferedReader(System.in);
String s = br.readLine();
```

➔ Javadoc !

Vannes
iut :
Université Bretagne Sud

Décomposer les lignes du fichier

- La classe java.util.StringTokenizer

(cf. Scanner) et les méthodes hasMoreTokens et nextToken

- La classe String et la méthode split

- La conversion en types primitifs avec les classes Wrappers: Integer, Double, Boolean, etc

Vannes
iut :
Université Bretagne Sud

Vannes
iut :
Université Bretagne Sud

Lire et écrire en binaire

iutvannes.fr

Les fichiers textes, des cas particuliers

L'ordinateur ne distingue pas un fichier texte d'un fichier binaire :
c'est l'humain qui le fait !

Un fichier texte est un fichier binaire où :

- chaque octet correspond au code ascii du caractère,
- chaque ligne se termine par un caractère de fin de ligne (CR),
- et le fichier se termine par le caractère de fin de fichier (EOF).

Les éditeurs de texte savent lire les fichiers texte.

Exemple !

Vannes
iut :
Université Bretagne Sud

Les fichiers binaires

- Composé de 0 et de 1
- Comment le lire ? (exemples)

- Avantages
  - rapidité de lecture/écriture
  - compacité
  - protection (nécessite le bon décodeur)
- Inconvénients
  - pas lisible facilement
  - nécessite de connaître le format (dans l'en-tête ou ailleurs)

Vannes
iut
Université Bretagne Sud

Lecture et écriture de données binaires

La classe DataInputStream

➔ Javadoc

La classe DataOutputStream

➔ Javadoc

Exemple

Vannes
iut :
Université Bretagne Sud

Aller plus loin

Java est un langage objet

➔ peut-on lire / écrire des objets ?

➔ pourquoi le faire ?

Lire / écrire des objets

Personne = nom, prénom, naissance

Date = jour, mois, année

Ecrire le programme qui lit ou écrit une personne dans un fichier

- en texte
- en binaire

Exemple

Vannes iut : Université Bretagne Sud

Le mécanisme de sérialisation

Java offre un mécanisme qui permet d'écrire / lire facilement des objets (et tous leurs constituants de façon récursive)

- Interface Serializable
- Lecture : ObjectInputStream
- Ecriture : ObjectOutputStream

Exemple

Vannes iut : Université Bretagne Sud

Aller plus loin

Spécifier le mécanisme de lecture / écriture en surchargeant les méthodes readObject / writeObject

Exemple