

Illustration des entrées-sorties

Exemple: Affiche les infos sur les fichiers dans d'un repertoire

```
package fichiers;

//Affiche les infos sur les fichiers dans d'un repertoire.
import java.io.*;

public class DirList {
    public static void main(String[] args) {
        File path = new File(".");
        File[] list = path.listFiles();

        for(int i = 0; i < list.length; i++) {
            System.out.println("Name: " + list[i].getName());
            System.out.println("Can read: " + list[i].canRead());
            System.out.println("Can write: " + list[i].canWrite());
            System.out.println("Length: " + list[i].length());
            System.out.println("Last modif: " + list[i].lastModified());

            if(list[i].isFile())
                System.out.println("it's a file");
            else if(list[i].isDirectory())
                System.out.println("it's a directory");
        }
    }
}
```

Exemple: Lecture d'entrée par lignes

Accès orienté caractère Unicode (16-bit) (Java 1.1)

FileReader est un décorateur qui modifie ou filtre le flux de caractères d'un **BufferedReader**.

```
package fichiers;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.File;

import java.io.FileNotFoundException;
import java.io.IOException;

public class ReadUnicode1 {

    public static void main(String args[])
    {
        BufferedReader in = null;
        File myFile = new File ("./src/fichiers/ReadUnicode1.java");

        try {
            in = new BufferedReader(new FileReader(myFile));
            String s= "";
            StringBuilder data = new StringBuilder();

            while((s = in.readLine()) != null)
                data.append(s + '\n');

            System.out.println(data);
        }
        catch (FileNotFoundException e)
        {
            System.out.println(e.getMessage());
        }
        catch (IOException e)
        {
            System.out.println(e.getMessage());
        }
        finally {
            try {
                in.close();
            }
            catch (IOException e)
            {
                System.out.println(e.getMessage());
            }
        }
    }
}
```

Exemple: Lecture depuis le clavier

Accès orienté caractère Unicode (16-bit) (Java 1.1)

```
package fichiers;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

// Lecture depuis le clavier
public class ReadUnicode2 {
    public static void main(String args[])
    {
        BufferedReader stdin = null;

        // in de type InputStream
        stdin = new BufferedReader(new InputStreamReader(System.in));
        do {
            String input = null;
            System.out.print("Enter a line (q to quit):");
            try {
                input = stdin.readLine();
                System.out.println(input);
            }
            catch (IOException e)
            {
                System.out.println(e.getMessage());
            }
        } while (!"q".equals(input)); // input peut être null
    }
}
```

Exemple: Entrée depuis un String en mémoire
Accès orienté caractère Unicode (16-bit) (Java 1.1)

```
package fichiers;

import java.io.StringReader;

import java.io.IOException;

public class ReadUnicode3 {

    public static void main(String args[])
    {
        String data = "...";
        StringReader in = new StringReader(data);

        try {
            int ch;
            // ch = in.read()  renvoie le code entier du char
            while ( (ch = in.read()) != -1) {
                // faut transtyper pour avoir le caractère
                System.out.print((char)ch);
                ;
            }
        }
        catch (IOException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

Exemple:Écriture vers un fichier à partir d'un fichier Accès orienté caractère Unicode (16-bit) (Java 1.1)

```
package fichiers;

import java.io.PrintWriter;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.File;

import java.io.IOException;

public class WriteUnicode1 {

public static void main(String args[])
{
    BufferedReader in = null;
    PrintWriter out = null;

    File myFileInIn = new File ("./src/fichiers/WriteUnicode1.java");
    File myFileInOut = new File ("./src/fichiers/WriteUnicode1Bis.java");

    try {
        in = new BufferedReader(new FileReader(myFileIn));

        // écriture tamponné, par caractères, vers un fichier
        out = new PrintWriter(
            new BufferedWriter(new FileWriter(myFileOut)));

        int lineCount = 1;
        String s;
        while ((s = in.readLine()) != null)
            out.println(lineCount++ + ": " + s);
    }
    catch (IOException e)
    {
        System.out.println(e.getMessage());
    }
    finally
    {
        try {
            in.close();
            out.close();
        }
        catch (IOException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

Exemple:Écriture vers un fichier à partir d'une saisie clavier

Accès orienté caractère Unicode (16-bit) (Java 1.1)

```
package fichiers;

import java.io.PrintWriter;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.FileWriter;
import java.io.File;
import java.io.IOException;

public class WriteUnicode2 {

    public static void main(String args[])
    {
        BufferedReader stdin = null;

        // in de type InputStream
        stdin = new BufferedReader(new InputStreamReader(System.in));

        PrintWriter out = null;
        File myFileOut = new File ("./src/fichiers/WriteUnicode2Bis.java");

        // ecriture tamponné, par caractères, vers un fichier
        try {
            out = new PrintWriter(
                new BufferedWriter(new FileWriter(myFileOut)));
        }
        catch (IOException e)
        {
            System.out.println(e.getMessage());
        }

        String input = null;
        int lineCount = 1;
        do {

            System.out.print("Enter a line (q to quit):");
            try {
                input = stdin.readLine();
                out.println(lineCount++ + ": " + input);
            }
            catch (IOException e)
            {
                System.out.println(e.getMessage());
            }

        } while (!"q".equals(input)); // input peut être null

        try {
            stdin.close();
            out.close();
        } catch (IOException e) { System.out.println(e.getMessage()); }

    }
}
```

Exemple:Écriture vers un fichier depuis un String en mémoire

Accès orienté caractère Unicode (16-bit) (Java 1.1)

```
package fichiers;

import java.io.PrintWriter;
import java.io.BufferedWriter;
import java.io.StringReader;
import java.io.FileWriter;
import java.io.File;
import java.io.IOException;

public class WriteUnicode3 {

    public static void main(String args[])
    {
        String data = "...";
        StringReader in = new StringReader(data);

        PrintWriter out = null;
        File myFileOut = new File ("./src/fichiers/WriteUnicode3Bis.java");

        // ecriture tamponné, par caractères, vers un fichier
        try {
            out = new PrintWriter(
                new BufferedWriter(new FileWriter(myFileOut)));
        }
        catch (IOException e)
        {
            System.out.println(e.getMessage());
        }

        try {
            int ch;
            // ch = in.read() renvoie le code entier du char
            while ( (ch = in.read()) != -1) {
                // faut transtyper pour avoir le caractère
                out.print((char)ch)
                ;
            }
        }
        catch (IOException e)
        {
            System.out.println(e.getMessage());
        }
        finally
        {
            out.close();
        }
    }
}
```

Exemple:Retour sur la classe Clavier du TP1
Accès orienté caractère Unicode (16-bit) (Java 1.1)

```
package tpl;

import java.io.*;

public class Clavier {

    public static void main(String[] args) {
        System.out.print("saisir une chaine: ");
        String s = Clavier.lireString();
        System.out.println(s);
        System.out.print("saisir un entier: ");
        int n = Clavier.lireInt();
        System.out.println(n);
    }

    public static String lireString(){
        String ligneLue = null;
        try{
            BufferedReader br = new BufferedReader(
                new InputStreamReader(System.in));
            ligneLue = br.readLine();
        }
        catch(IOException e){
            System.out.println(e.getMessage());
            System.exit(0);
        }
        return ligneLue;
    }

    public static int lireInt(){
        int n = 0;
        try{
            n = Integer.parseInt(lireString());
        }
        catch(NumberFormatException e){
            System.out.println(e.getMessage());
            System.exit(0);
        }
        return n;
    }
}
```


Exemple: Écriture et lecture de données formatées

Accès orienté byte (Java 1.0)

```
package fichiers;
```

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
```

```
public class ReadWriteDonneesFormattees {
    public static void main(String args[])
    {
        try {
            DataOutputStream out = new DataOutputStream (
                new BufferedOutputStream (new FileOutputStream("Data.txt")));

            out.writeDouble(3.14159);
            out.writeChar('a');
            out.writeInt (10);
            out.writeBytes ("this is a string\n");
            out.close();
        }
    }
}
```

```
// DataInputStream pour lire les données formatées (sauf String)
```

```
DataInputStream in = new DataInputStream(
    new BufferedInputStream (new FileInputStream("Data.txt")));

System.out.println(in.readDouble());
System.out.println(in.readChar());
System.out.println(in.readInt());
```

```
// BufferedReader pour lire le String
```

```
BufferedReader br = new BufferedReader (new InputStreamReader(in));

System.out.println(br.readLine());
}
catch (IOException e)
{
    System.out.println(e.getMessage());
}
}
```

Exemple: S rialiser des objets ou donn es primitives dans un fichier
puis pour des rialiser les m mes objets et les m mes donn es

```
package serialisation;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Date;

public class Serialisation1 {
    public static void main(String args[])
    {
        FileOutputStream fos = null;
        try {
            fos = new FileOutputStream("t.tmp");
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
        }

        ObjectOutputStream oos = null;
        try {
            oos = new ObjectOutputStream(fos);
            oos.writeInt(10325);
            oos.writeObject("Today"); // conversion en String
            oos.writeObject(new Date());
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
        finally
        {
            try {
                oos.close();
            } catch (IOException e) {
                System.out.println(e.getMessage());
            }
        }

        FileInputStream fis = null;
        try {
            fis = new FileInputStream("t.tmp");
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
        }

        ObjectInputStream ois = null;

        try {
            ois = new ObjectInputStream(fis);
            int i = ois.readInt();
            String today = (String) ois.readObject();
            Date date = (Date) ois.readObject();

            System.out.println(i + " " + today + " " + date);
        }
    }
}
```

```
    } catch (IOException e) {
        System.out.println(e.getMessage());
    } catch (ClassNotFoundException e) {
        System.out.println(e.getMessage());
    }
    finally
    {
        try {
            ois.close();
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}
}
```

Exemple de gestion de jetons (Tokenizer)

```
package tokenizer;

import java.io.*;

/** Classe qui lit une séquence d'entiers (max 1000), stocké dans un
 * fichier text et les place dans un tableau d'entiers Java.
 */
public class Tokenizer1 {
    public static void main (String[] args) throws Exception{
        BufferedReader in = new BufferedReader(new FileReader(args[0]));
        StreamTokenizer st = new StreamTokenizer(in);

        int[] lesInts = new int[1000];
        int tokenType = 0;
        int i = 0;

        do {
            tokenType = st.nextToken ();
            if ( tokenType == StreamTokenizer.TT_NUMBER ) {
                lesInts[i] = (int)st.nval;
                System.out.println(i + ": " + lesInts[i]);
                i++;
            }
        } while ( tokenType != StreamTokenizer.TT_EOF && i < 1000 ) ;
    }
}
```