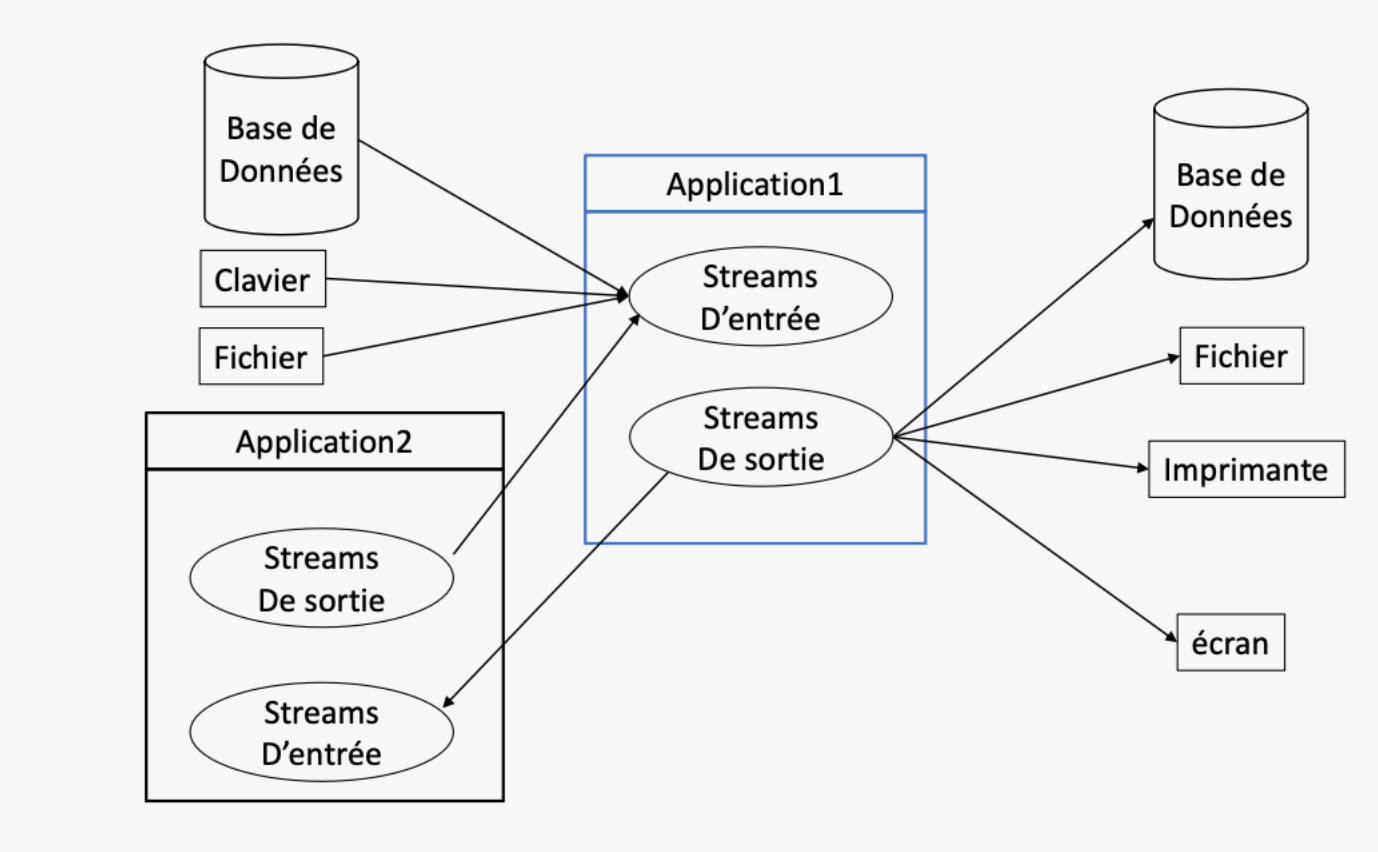
ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT
TECHNIQUE DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE CASABLANCA

Programmation Orientée Objet

Les entrées sorties

Mme Hassna BENSAG
Email: **h.bensag@gmail.com**

Entrées/Sorties



Principe des entrées/sorties

- Afin d'effectuer une entrée ou une sortie de données dans un programme Java, le principe est simple et se résume aux opérations suivantes :
  - Ouverture d'un moyen de communication.
  - Écriture ou lecture des données.
  - Fermeture du moyen de communication.
- En Java, les moyens de communication sont représentés par des objets particuliers appelés streams (flux).

Les streams

- **Streams de Communication:** établissent l'interaction entre systèmes(source/programme ou programme/destination).

	Stream de caractères	Stream binaires
Stream d'entrée	Reader	InputStream
Stream de sortie	Writer	OutputStream

  - Exemple de stream binaires: **FileInputStream,FileOutputStream**
  - Exemple de stream de caractères: **FileReader, FileWriter**

- **Streams de traitement:** **p**ermettent de traiter les données, et doivent être associés à des streams d'entrées ou de sorties.
  - Exemple de stream binaires: **BufferedInputStream**
  - Exemple de stream de caractères: **BufferedReader, BufferedWriter**

La classe File

- La classe **File** peut représenter un fichier ou un répertoire.
- La classe File contient plusieurs méthodes pour manipuler des fichiers, les supprimer, les renommer, etc.
- Exemple:

App1.java

```
1  import java.io.File;
2
3  public class App1 {
4      public static void main(String[] args) {
5          File file1= new File( pathname: "myfile.txt");
6          File file2= new File( pathname: "//Users//hassnabensag//Documents");
7          File file3= new File( pathname: "//Users//hassnabensag//Documents//myfile.dat");
8
       }
10 }
```

La classe File

- Méthodes

Méthode	Description
boolean createNewFile()	Crée un nouveau fichier vide.
boolean delete()	Supprime le fichier ou le répertoire.
boolean exists()	Teste si le fichier ou le répertoire existe.
String getName()	Renvoie le nom du fichier ou du répertoire.
String getAbsolutePath()	Renvoie la chaîne de chemin absolu du fichier ou du répertoire.
String getPath()	Renvoie la chaîne de chemin relatif du fichier ou du répertoire.
boolean isDirectory()	Teste si le fichier désigné par le chemin d'accès est un répertoire.
long length()	Renvoie la longueur du fichier indiqué par le chemin.
String[] list()	Renvoie un tableau de chaînes contenant les noms des fichiers et des répertoires du répertoire .
boolean mkdir()	Permet de créer un répertoire.
boolean renameTo(File dest)	Permet de renommer un fichier.

Lire et écrire un fichier texte

- Exemple pour lire et écrire dans un fichier texte en utilisant les classes **FileReader** et **FileWriter, permettant de traiter les fichiers caractère par cractère**

App1.java

```java
import java.io.*;

public class App1 {
    public static void main(String[] args) {
        File file1 = new File( pathname: "myfile1.txt");
        File file2 = new File( pathname: "myfile2.txt");
        FileReader fileR=null;
        FileWriter fileW=null;
        try {
             fileR= new FileReader(file1);
             fileW= new FileWriter(file2);
            int caractere;
            while ((caractere = fileR.read()) != -1)
                fileW.write(caractere);

        }catch (FileNotFoundException e){
            System.out.println(""+e.getMessage());
        }catch (IOException e){
            System.out.println(e.getMessage());
        }finally {
            try {
                if(fileR!=null)
                    fileR.close();
                if(fileR!=null)
                    fileW.close();
            }catch (IOException e){
                System.out.println(e.getMessage());
            }
        }

    }
}
```

Lire et écrire un fichier binaire

- Exemple pour lire et écrire dans un fichier binaire en utilisant les classes **FileInputStream** et **FileOutputStream permettant de traiter les fichiers octer par octer**

App2.java

```java
import java.io.*;

public class App2 {
    public static void main(String[] args) {
        File f1= new File( pathname: "java.png");
        File f2= new File( pathname: "javacopie.png");
        FileInputStream InStream=null;
        FileOutputStream OutStream=null;
        try{
            InStream=new FileInputStream(f1);
            OutStream=new FileOutputStream(f2);
            int octet;
            while((octet=InStream.read())!=-1){
                OutStream.write(octet);
            }
        }catch (FileNotFoundException e){
            System.out.println("Erreur: Fichier introuvable"+e.getMessage());
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }finally {
            try {
                if (InStream != null) {
                    InStream.close();
                }
                if (OutStream != null) {
                    OutStream.close();
                }
            }catch (IOException e) {
                System.out.println(e.getMessage());
            }
```

Lire et écrire un fichier texte ligne par ligne

- Exemple pour lire et écrire dans un fichier texte en utilisant les classes **BufferedReader** et **BufferedWriter**

App3.java

```java
import java.io.*;

public class App3 {
    public static void main(String[] args) {
        File file1 = new File( pathname: "myfilev1.txt");
        File file2 = new File( pathname: "myfilev2.txt");
        FileReader fileR=null;
        FileWriter fileW=null;
        BufferedReader bfReader;
        BufferedWriter bfWriter;
        try{
            fileR=new FileReader(file1);
            fileW=new FileWriter(file2);
            bfReader=new BufferedReader(fileR);
            bfWriter=new BufferedWriter(fileW);
            String ligne=null;
            while ((ligne=bfReader.readLine())!=null){
                bfWriter.write(ligne);
                bfWriter.newLine();
            }
            bfReader.close();
            bfWriter.close();
        }
        catch (IOException e){
            System.out.println(e.getMessage());
        }
    }
}
```