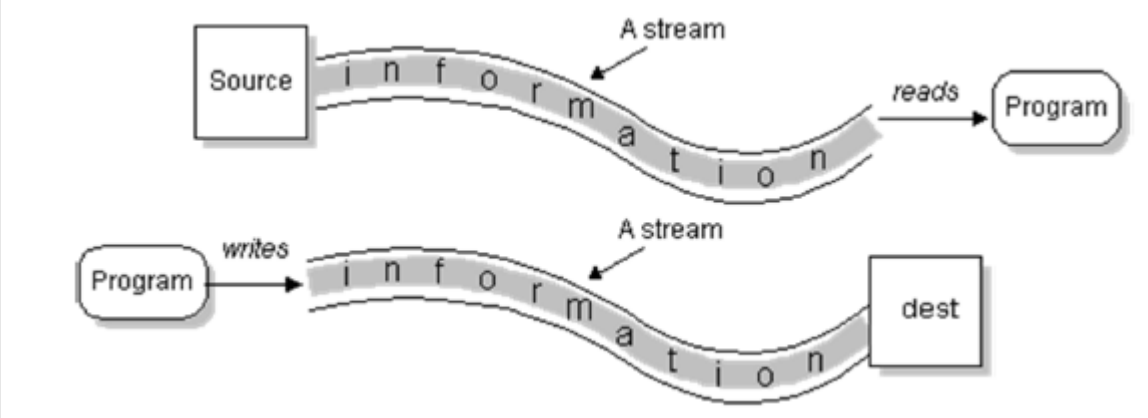
Lecture Ecriture Fichier

Dans Java les échanges d'information entre le programme et un périphérique extérieur (écran, clavier, fichier, connexion réseau) font appel à la notion de flux (stream).



Lecture Ecriture Fichier

Lorsque le flux est associé à un fichier , celui-ci est représenté par un objet instance de la classe Path

Cet objet représente la source du Flux si le flux est sortant (outputStream) ou sa destination si le flux est entrant (inputStream)

```
Path fichierSource = Paths.get("monFichier.txt");
```

Lecture Ecriture Fichier
Fichier Binaire

Java propose des Classes différentes selon si les données du fichier lues (ou écrites) sont du texte (caractères) ou des données (Ex: relevés de valeurs de type int , double,..).

Les Flux de données de type binaire sont associés à la Classe InputStream (pour la lecture) ou OutputStream (pour l'écriture)

```
Path chemin = Paths.get("data.bin") ;

InputStream in = Files.newInputStream(chemin);
```

ou

```
OutputStream out = Files.newOutputStream(chemin);
```

Lecture Ecriture Fichier
Fichier Binaire

Java permet d'ajouter des fonctionnalités aux flux avec des Classes faisant parties des filtres, notamment:

BufferedInputStream, (BufferedOutputStream) :

Ces deux classes bufférisent les données limitant ainsi les accès au fichier.

```
Path chemin = Paths.get("data.bin") ;

InputStream in = Files.newInputStream(chemin);

BufferedInputStream bis = new BufferedInputStream (in);
```

Lecture Ecriture Fichier
Fichier Binaire

Il existe également des fonctionnalités permettant de formater les donnée lues ou écrites dans les flux.

DataInputStream, (DataOutputStream)

```
Path chemin = Paths.get("data.bin") ;
InputStream in = Files.newInputStream(chemin);
BufferedInputStream bis = new BufferedInputStream (in);
DataIntPutStream  dis =  new DataInputStream (bis);
```

Lecture Ecriture Fichier
Fichier Binaire

Les objets de type DataInputSteam (DataOutputStream) possèdent des méthodes qui permettent de lire (écrire) des données

Lecture :

int readInt() lit 4 octets dans le flux et retourne un int constitué à partir des 4 octets

byte readByte(), float readFloat(), double readDouble(), ….

Ecriture

void writeInt (int v), void writeFloat (float v), void writeDouble(double v)

void writeByte(int v)

Lecture Ecriture Fichier
Fichier Binaire

Exemple:

```
Path chemin = Paths.get("data.bin") ;
InputStream in = Files.newInputStream(chemin);
BufferedInputStream bis = new BufferedInputStream (in);
DataIntPutStream  dis =  new DataInputStream (bis);
int entierLu = dis.readInt();
dis.close();
System.out.println("  valeur lue : " +entierLu);
```

Lecture Ecriture Fichier
Fichier Binaire

Exercice :
Le fichier source.bin contient des valeurs de températures stockées en double.

Créer une application qui:

- Ouvre le fichier en mode binaire,
- Lit et affiche les valeurs,
- Calcule et affiche la moyenne,
- Stocke dans un fichier binaire cible.bin le nombre de valeurs lues (int) et la moyenne(double),
- Verifie les données en ouvrant le fichier cible.bin pour afficher les valeurs contenues,