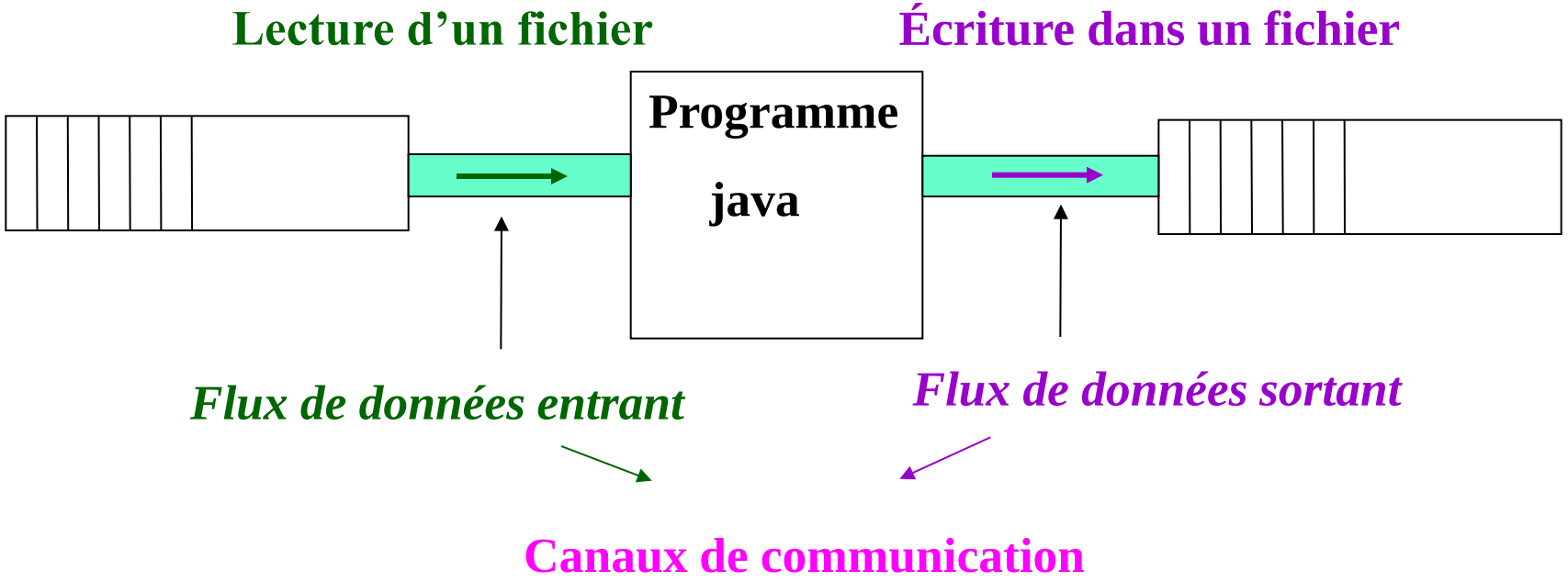
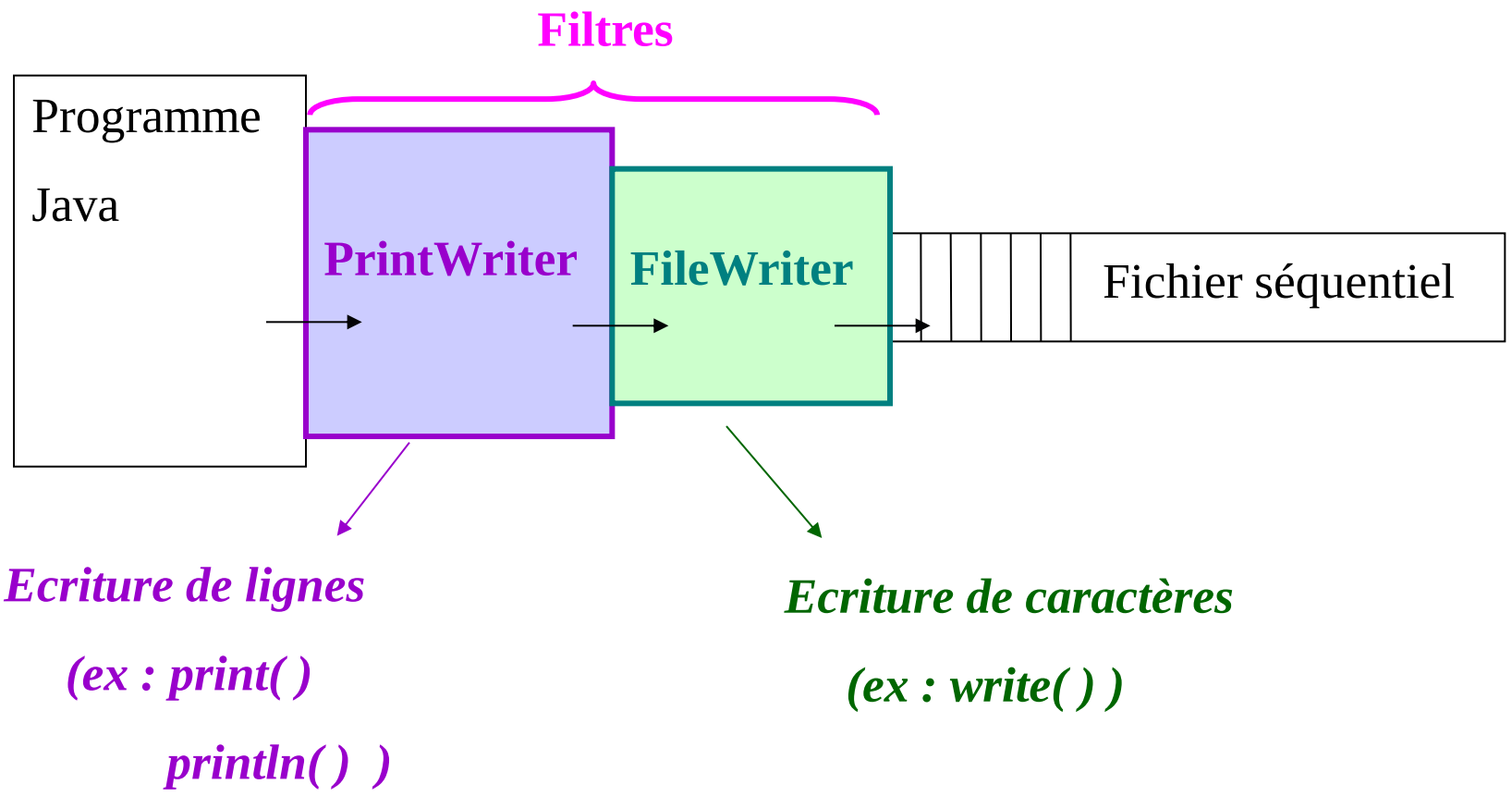
5. Fichiers


Lecture d'un fichier
Écriture dans un fichier
Programme
java
Flux de données entrant
Flux de données sortant
Canaux de communication

5. Fichiers

- 5.1. Ecriture/lecture de chaînes de caractères
  - Ecriture


Filtres
Programme
Java
PrintWriter
FileWriter
Fichier séquentiel
Ecriture de lignes
(ex : print( )
println( ))
Ecriture de caractères
(ex : write( ))

```
public class Vehicule
        {private String plaque, proprio;

        public Vehicule(String pla, String pro)
                {  plaque = pla; proprio = pro;}

        // gettors et settors

        }
```

```
import java.io.*;
import javax.swing.*;


public class Programme

{  public static void main(String[ ] args)
        {       Vehicule veh;
                PrintWriter sortie = null;
```

Filtre : écriture plus aisée

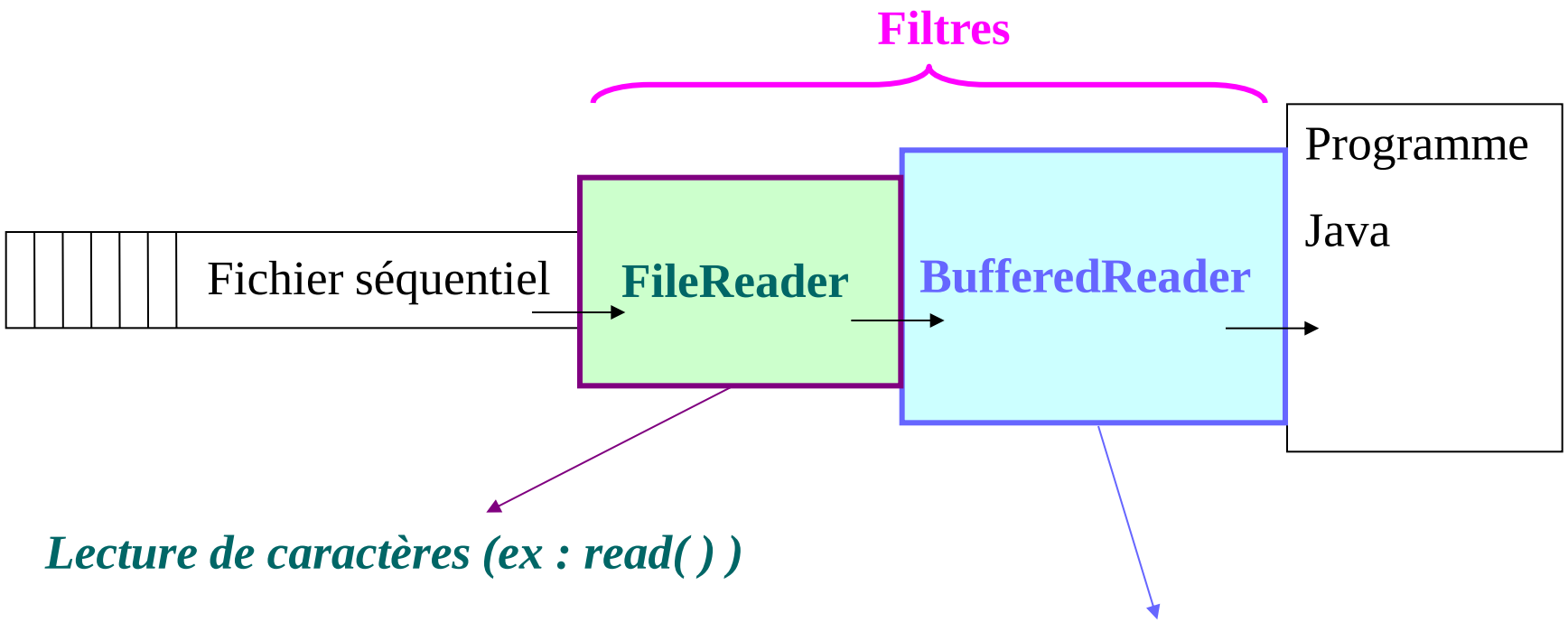
```
try {
sortie = new PrintWriter (new FileWriter ("vehicule.txt" , false ));
        /*ouverture du fichier: écrase son contenu éventuel */

veh = new Vehicule("cv123","julot");
sortie.print(veh.getPlaque());
sortie.println(veh.getProprio())
veh = new Vehicule("di456","mario");
sortie.print(veh.getPlaque());
sortie.println(veh.getProprio());
sortie.close( );
```

```
sortie = new PrintWriter(new FileWriter("vehicule.txt",true ));
                    /*réouverture du fichier en mode append */
        veh = new Vehicule("re789","polo");
        sortie.print(veh.getPlaque());
        sortie.println(veh.getProprio());

}
```

```
catch ( IOException e)
 {JOptionPane.showMessageDialog(null,e.getMessage( )); }

finally
  {if (sortie != null)
            {try {sortie.close( );}
              catch ( IOException e)  {…}
            }
   }
```

5. Fichiers

- 5.1. Ecriture/lecture de chaînes de caractères
  - Ecriture
  - Lecture


Filtres
Programme
Java
Fichier séquentiel
FileReader
BufferedReader
Lecture de caractères (ex : read( ))
Buffer utilisé pour optimiser les accès au fichier

Filtre : optimise

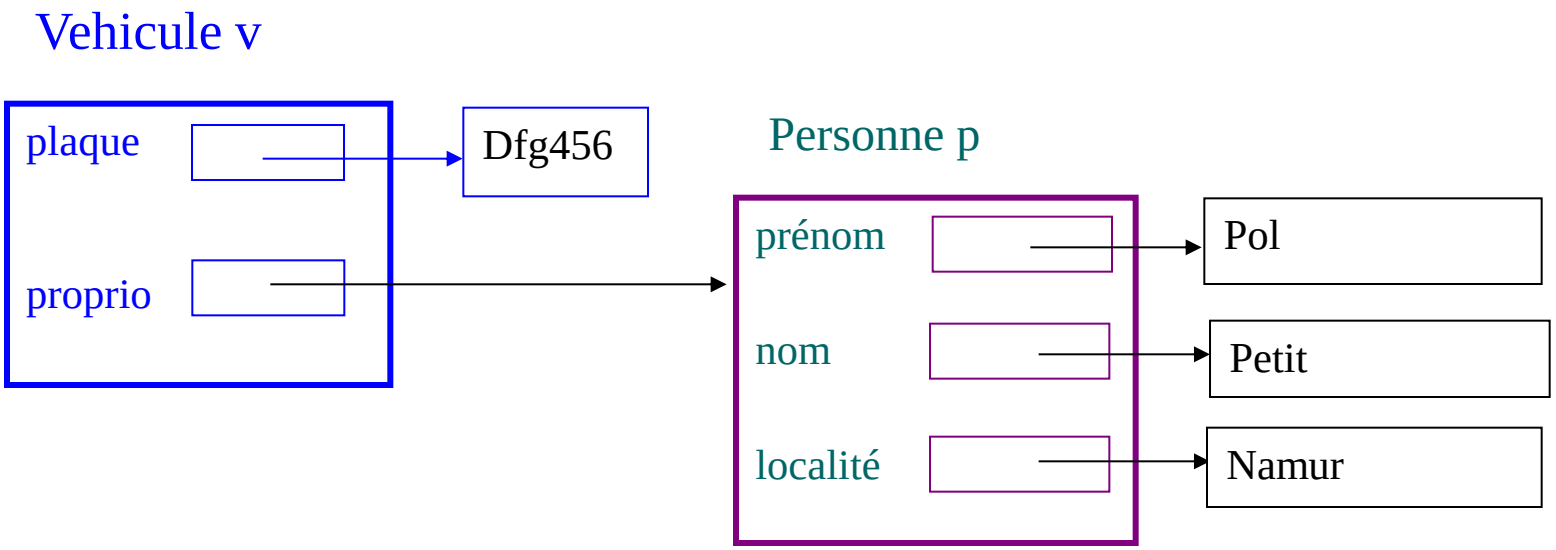
accès au disque

```
BufferedReader entree = null;
try {entree = new BufferedReader(new FileReader ("vehicule.txt"));
      String s;
      s = entree.readLine( );
      while (s != null)
        { System.out.println (s);
         s = entree.readLine( );    }
   }
```

```
catch (IOException e)
      {JOptionPane.showMessageDialog(null,e.getMessage( )); }

finally
 {if (entree != null)
            {try {entree.close( );}
              catch ( IOException e)  {…}
            }
  }
}}
```

5. Fichiers

- 5.1. Ecriture/lecture de chaînes de caractères
- 5.2. Ecriture/lecture d'objets

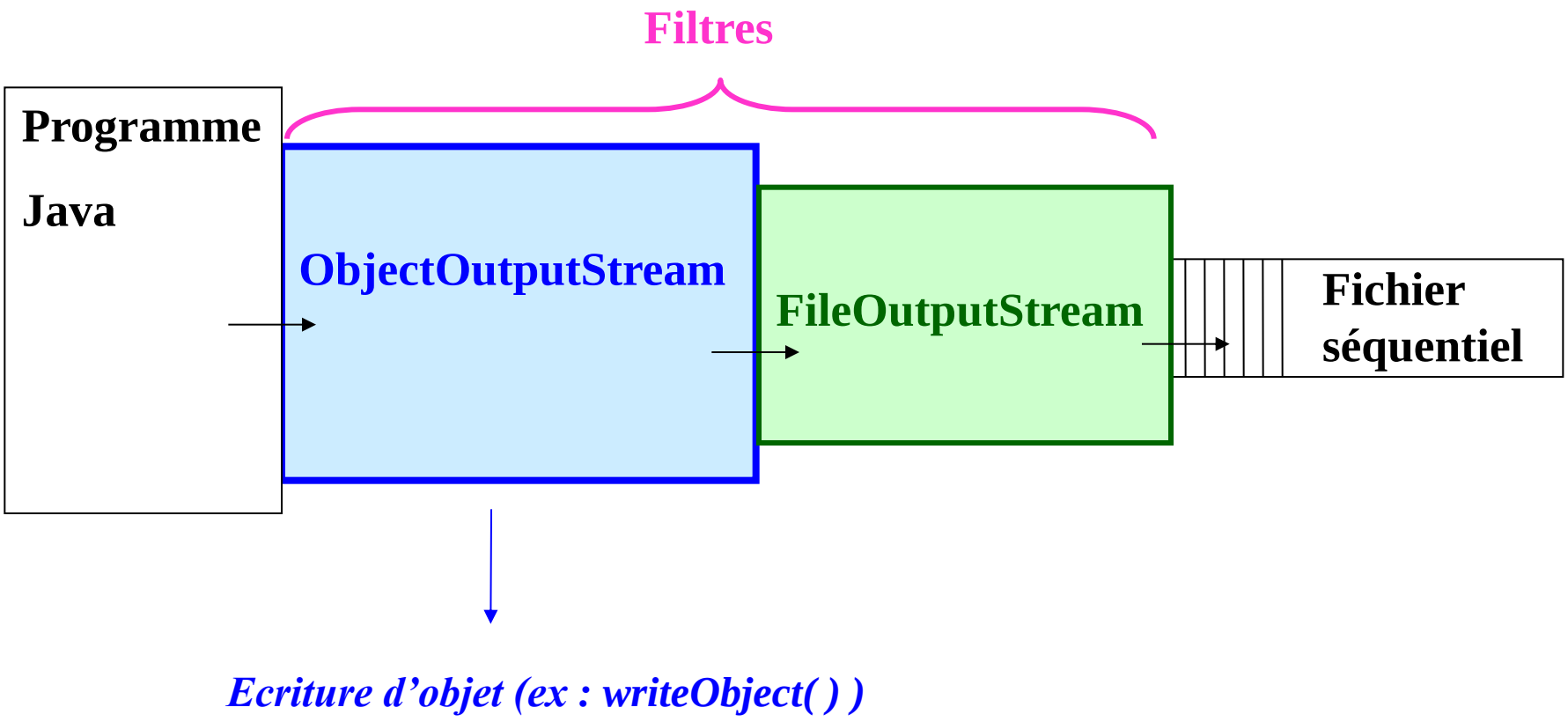

Vehicule v
plaque
proprio
Dfg456
Personne p
prénom
nom
localité
Pol
Petit
Namur

```
import java.io.*;

public class Personne implements Serializable

{ private String prenom, nom, localite;

  public Personne(String p, String n, String l)
        {prenom = p; nom = n; localite = l;}

  public String toString( )
              {return "la personne " + prenom + " " + nom + "
               habitant à " + localite;}}
```

```
import java.io.*;

public class Vehicule implements Serializable

{ private String plaque;
   private Personne proprio;

  public Vehicule(String pla, Personne pro)
            { plaque = pla;  proprio = pro;}

  public String toString( )
  {return "Le véhicule " + plaque + " appartient à " + proprio; }
}
```

5. Fichiers

- 5.1. Ecriture/lecture de chaînes de caractères
- 5.2. Ecriture/lecture d'objets
  - Ecriture


Filtres
Programme
Java
ObjectOutputStream
FileOutputStream
Fichier
séquentiel
Ecriture d'objet (ex : writeObject( ))

Flux d'objets

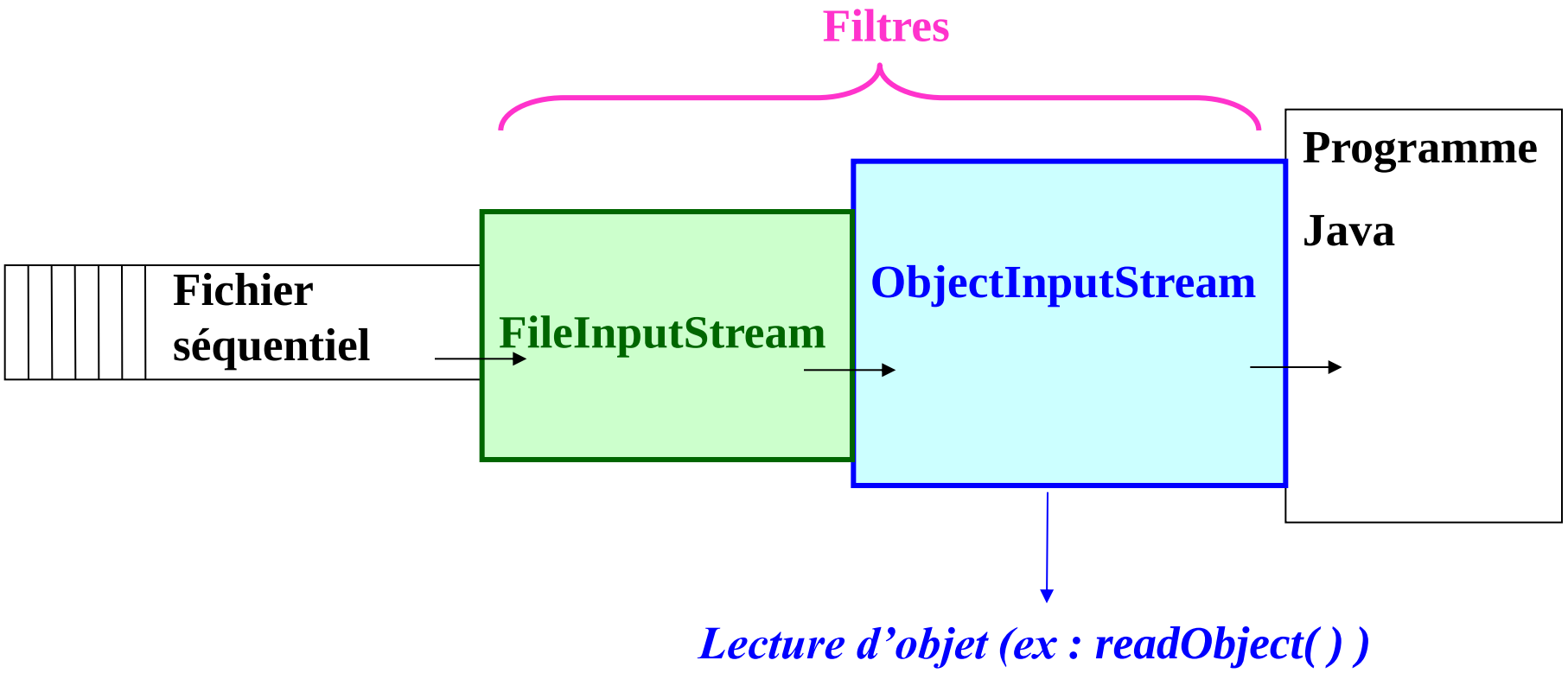
Flux de données

```
public class Programme
{ public static void main(String[ ] args)
{ ObjectOutputStream sortie = null;
  try {sortie =new ObjectOutputStream(new FileOutputStream("vehic.txt"));
        sortie.writeObject(new Vehicule ("xw123",new
                                  Personne("Marie","Leroy","Liege")));
        sortie.writeObject(new Vehicule ("df456",new
                                  Personne("Pol","Dupond","Namur")));
        sortie.writeObject(new Vehicule ("rt789",new
                                  Personne("Jules","Claude","Mons")));
       }
```

```
catch ( IOException e)
 {JOptionPane.showMessageDialog(null,e.getMessage( )); }

finally
  {if (sortie != null)
            {try {sortie.close( );}
              catch ( IOException e)  {…}
            }
   }
```

5. Fichiers

- 5.1. Ecriture/lecture de chaînes de caractères
- 5.2. Ecriture/lecture d'objets
  - Ecriture
  - Lecture


Filtres
Programme
Java
Fichier
séquentiel
FileInputStream
ObjectInputStream
Lecture d'objet (ex : readObject( ))

```
 ObjectInputStream entree =  null;
try {
  entree = new ObjectInputStream(new FileInputStream("vehic.txt"));
  Vehicule veh;
   veh = (Vehicule) entree.readObject( );
                                   /* attention au casting obligatoire */
   while (veh !=null)
        { System.out.println(veh);
          veh = (Vehicule) entree.readObject( ); }
}
```

```
catch (EOFException e)
        {System.out.println("Plus d'enregistrement"); }
catch (IOException e)
        {    JOptionPane.showMessageDialog(null,"erreur lecture
             fichier");}
catch (Exception e)
        { JOptionPane.showMessageDialog(null,"erreur autre que
             lecture fichier");}
finally
  {if (entree != null)
             {try {entree.close( );}
               catch ( IOException e)  {…}
             }
   } }}
```