

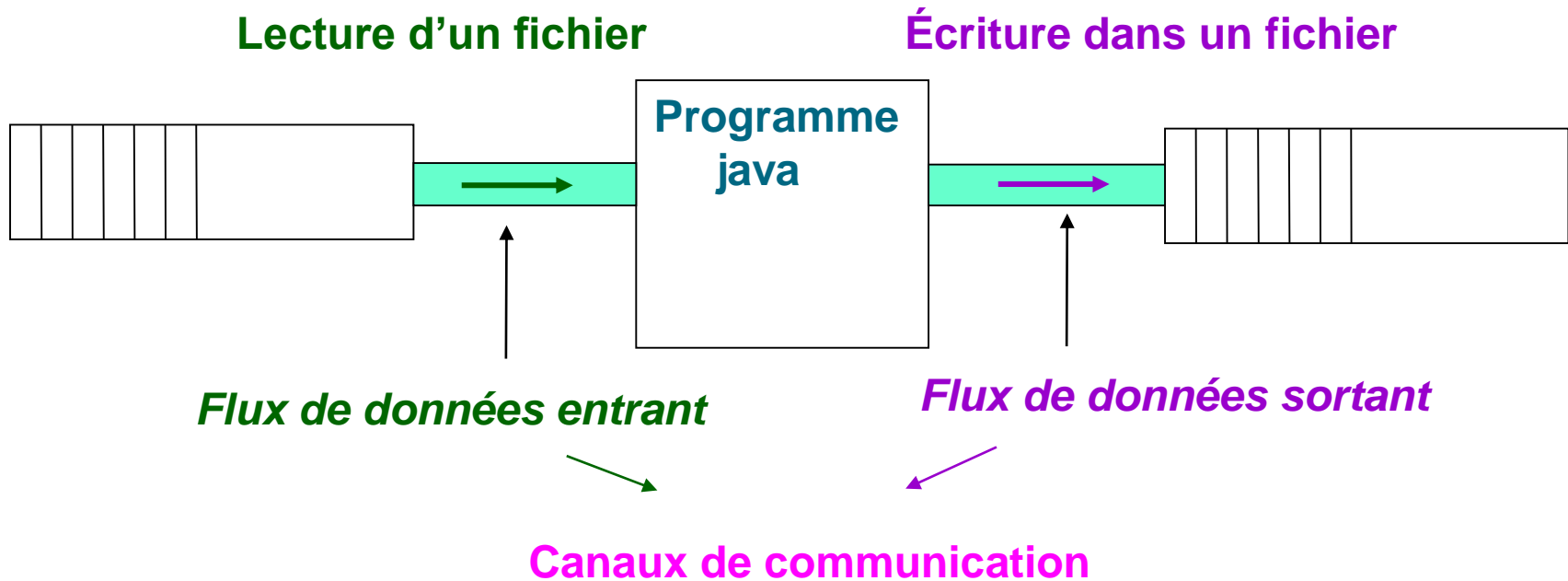


Chapitre 5

Fichiers

Enregistrement d'objets sérialisés dans des fichiers

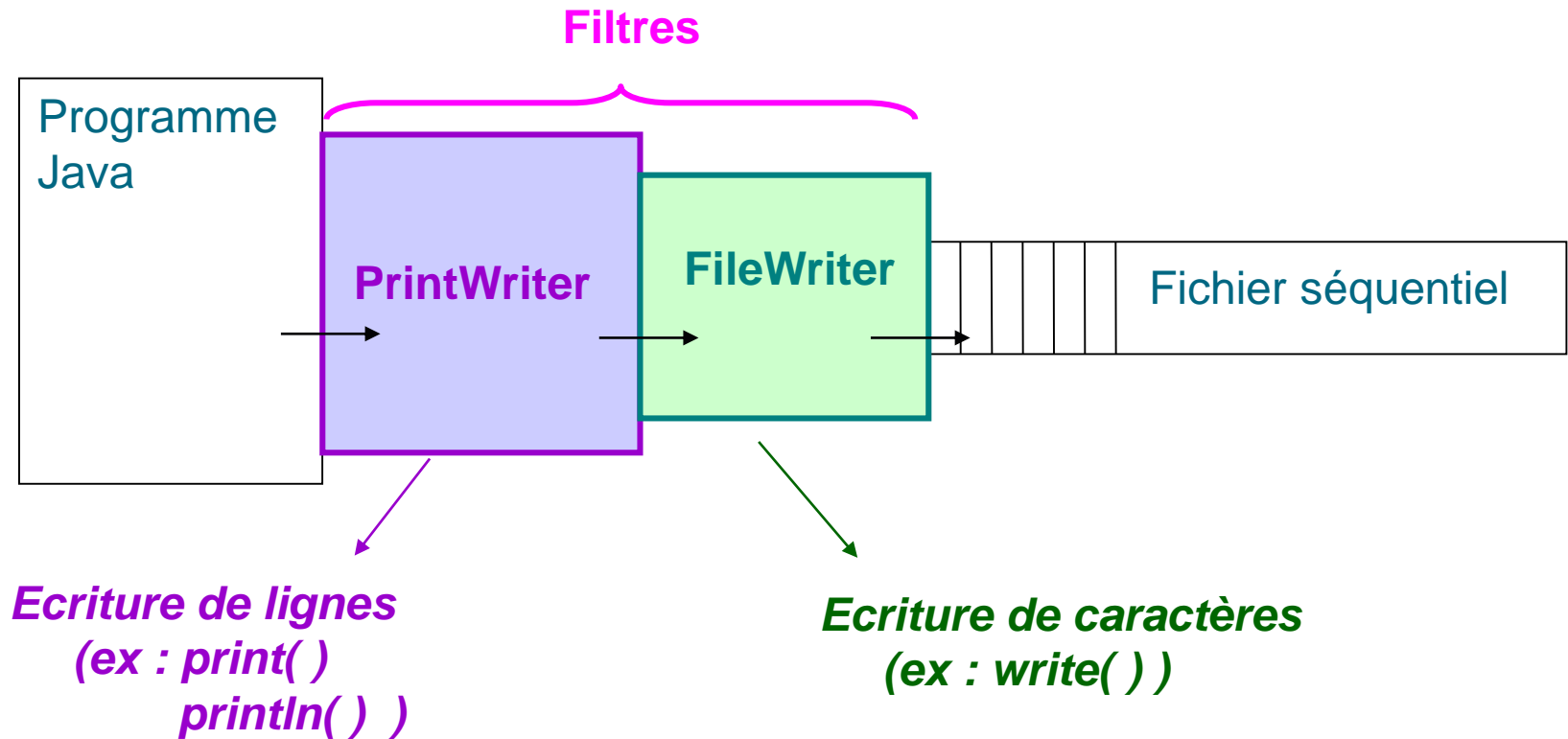
Fichiers



1. Ecriture / lecture de chaînes de caractères

- Ecriture

Ecriture de chaînes de caractères



Ecriture de chaînes de caractères

```
public class Vehicule {  
    private String plaque;  
    private String proprio;  
  
    public Vehicule (String plaque, String proprio) {  
        ...  
    }  
  
    ... // Getters et setters  
}
```

Ecriture de chaînes de caractères

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

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

Ecriture de chaînes de caractères

```
try {  
    sortie = new PrintWriter (new FileWriter ("vehicule.txt", false ));  
    // Ouverture du fichier : écrase son contenu éventuel  
  
    vehicule = new Vehicule("1-CVA-123","Julot");  
    sortie.print(vehicule.getPlaque());  
    sortie.println(vehicule.getProprio());  
    vehicule = new Vehicule("1-DFG-456","Mario");  
    sortie.print(vehicule.getPlaque());  
    sortie.println(vehicule.getProprio());  
    sortie.close( );  
}
```

Filtre : écriture plus aisée

Ecriture de chaînes de caractères

```
sortie = new PrintWriter(new FileWriter("vehicule.txt",true ));
```

```
    // Réouverture du fichier en mode append
```

```
    vehicule = new Vehicule("1-TVR-789","Anne");
```

```
    sortie.print(vehicule.getPlaque());
```

```
    sortie.println(vehicule.getProprio());
```

```
}
```

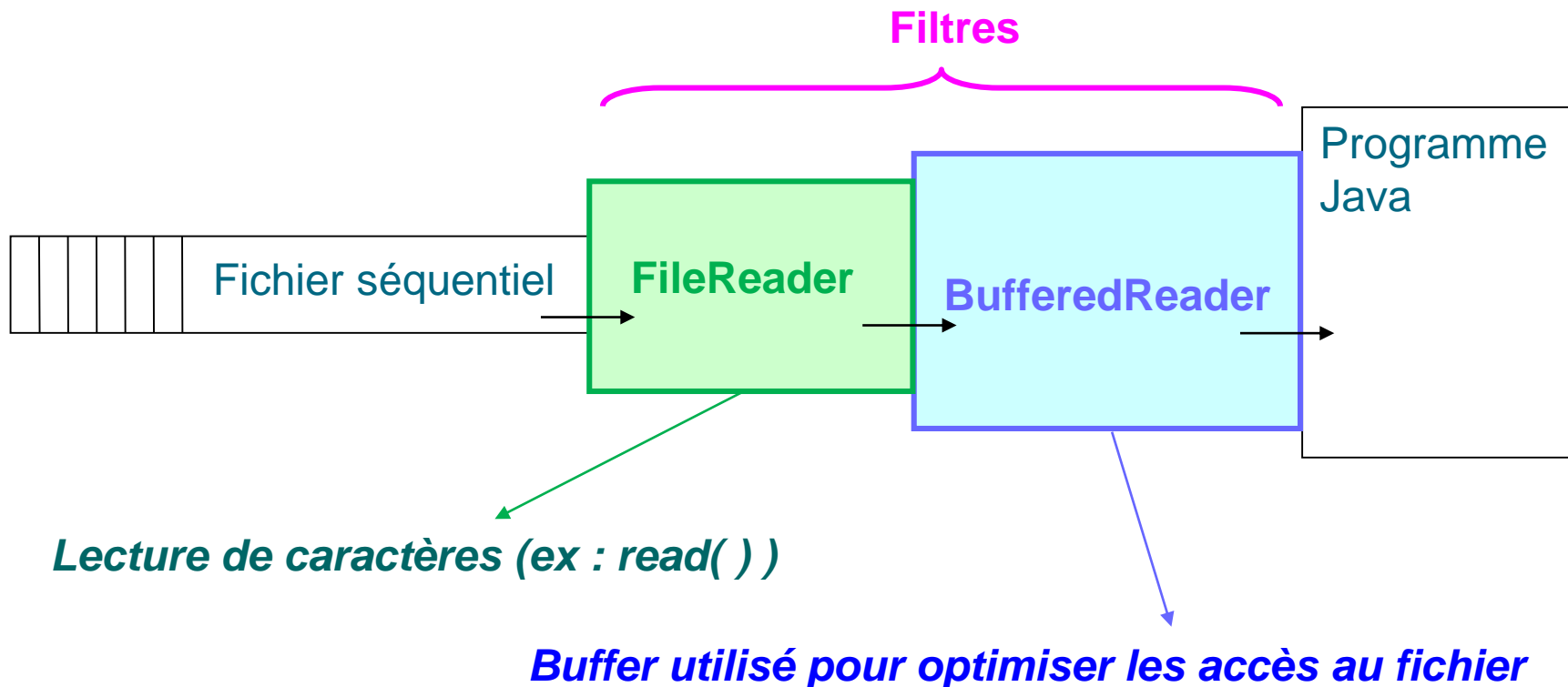

Ecriture de chaînes de caractères

```
catch ( IOException ioException) {  
    JOptionPane.showMessageDialog(null, ioException.getMessage( ));  
}  
  
finally {  
    if (sortie != null)  
        sortie.close( );  
}  
}  
}
```

1. Ecriture / lecture de chaînes de caractères

- Ecriture
- Lecture

Lecture de chaînes de caractères



Lecture de chaînes de caractères

Filtre : optimise accès au disque

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

Lecture de chaînes de caractères

```
catch (IOException ioException) {  
    JOptionPane.showMessageDialog(null, ioException.getMessage( ));  
}  
  
finally {  
    if (entree != null) {  
        try {  
            entree.close();  
        }  
        catch ( IOException exception) {  
            ...  
        }  
    }  
}
```

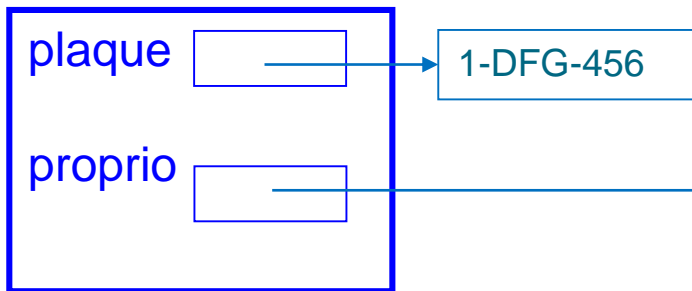
1. Ecriture / lecture de chaînes de caractères

- Ecriture
- Lecture

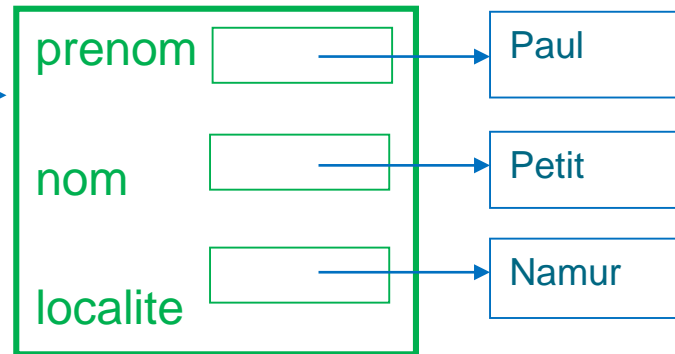
2. Ecriture / lecture d'objets

Lecture / écriture d'objets

Vehicule vehicule



Personne personne



Lecture / écriture d'objets

```
import java.io.*;
public class Personne implements Serializable {
    private String prenom;
    private String nom;
    private String localite;

    public Personne(String prenom, String nom, String localite) {
        ...
    }

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


Lecture / écriture d'objets

```
import java.io.*;
public class Vehicule implements Serializable {
    private String plaque;
    private Personne proprio;
    public Vehicule (String plaque, Personne proprio) {
        ...
    }
    public String toString( ) {
        return "Le véhicule " + plaque + " appartient à " + proprio;
    }
}
```

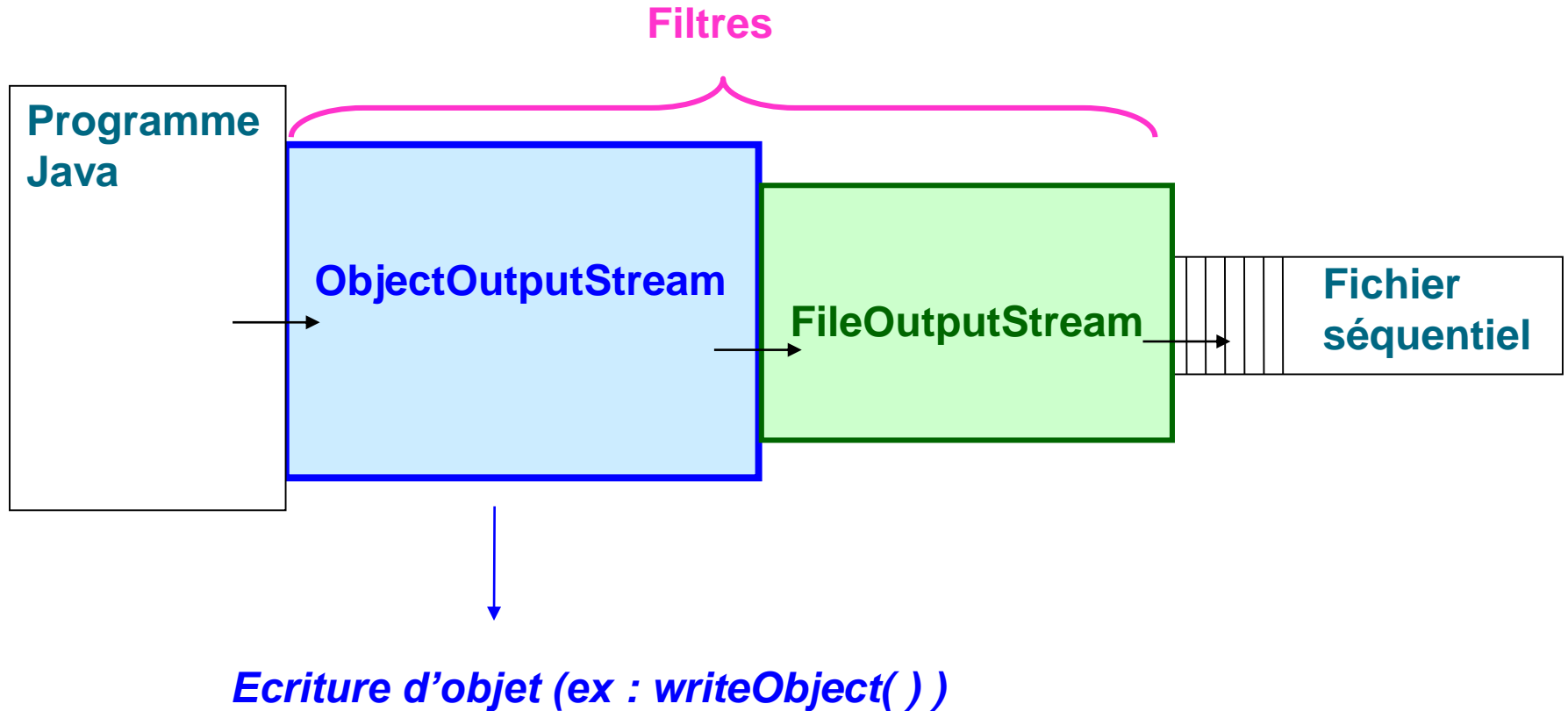
1. Ecriture / lecture de chaînes de caractères

- Ecriture
- Lecture

2. Ecriture / lecture d'objets

- Ecriture

Ecriture d'objets



Ecriture d'objets

```
public class Programme {  
    public static void main(String[ ] args) {  
        ObjectOutputStream sortie = null;  
        try {  
            sortie = new ObjectOutputStream(new FileOutputStream("vehic.txt"));  
            sortie.writeObject(new Vehicule ("1-TDE-123",  
                                                new Personne("Marie","Leroy","Liege")));  
            sortie.writeObject (new Vehicule ("1-BCA-456",  
                                                new Personne("Paul","Dond","Namur")));  
            sortie.writeObject (new Vehicule ("1-LOL-789",  
                                                new Personne("Jules","Claude","Mons")));  
        }  
    }  
}
```

Flux d'objets

Flux de données

Ecriture d'objets

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

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

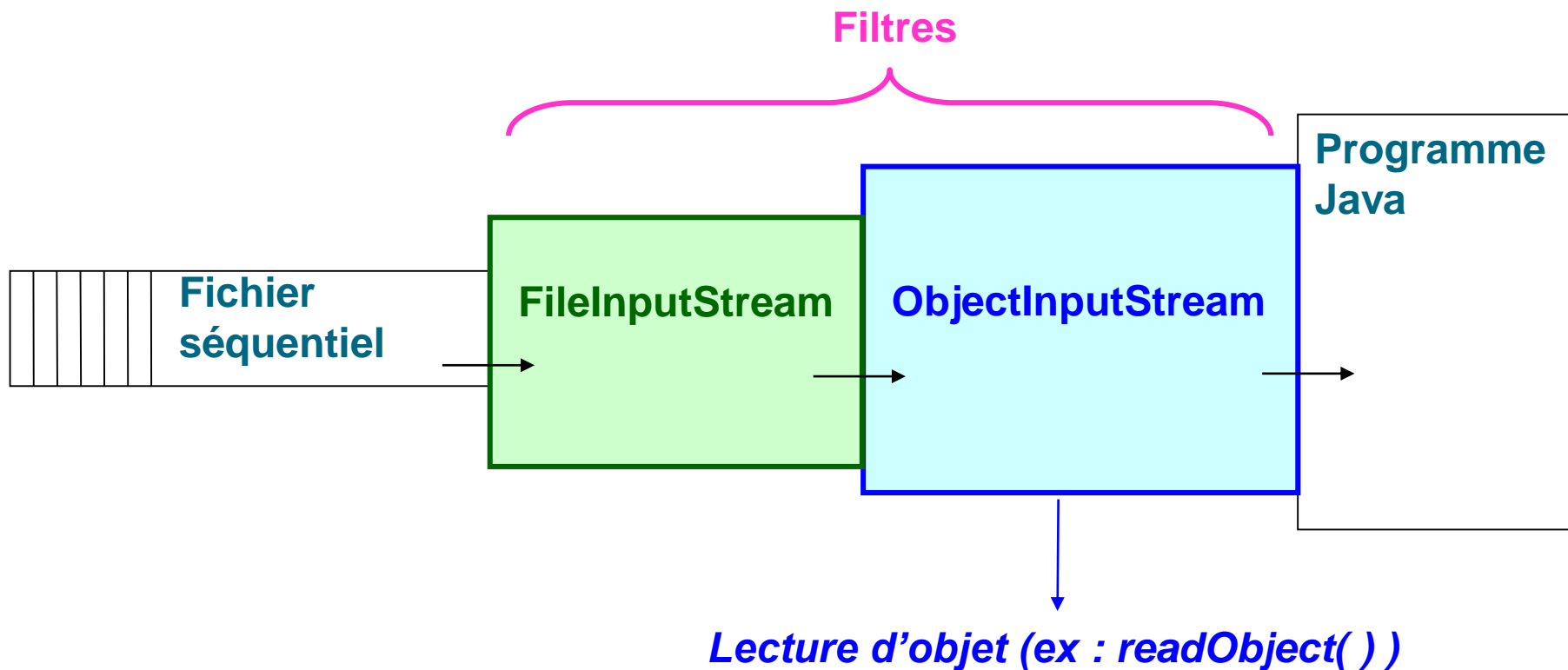
1. Ecriture / lecture de chaînes de caractères

- Ecriture
- Lecture

2. Ecriture / lecture d'objets

- Ecriture
- Lecture

Lecture d'objets



Lecture d'objets

```
ObjectInputStream entree = null;
```

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


Lecture d'objets

```
catch (EOFException eofException) {
    System.out.println("Plus d'enregistrement");
}
catch (IOException ioException) {
    JOptionPane.showMessageDialog(null,"Erreur lecture fichier");
}
catch (Exception exception) {
    JOptionPane.showMessageDialog(null,"Erreur autre que lecture");
}
Finally {
    if (entree != null) {
        try { entree.close();
        }
        catch ( IOException iException) { ...
        }
    }
}
} }
```