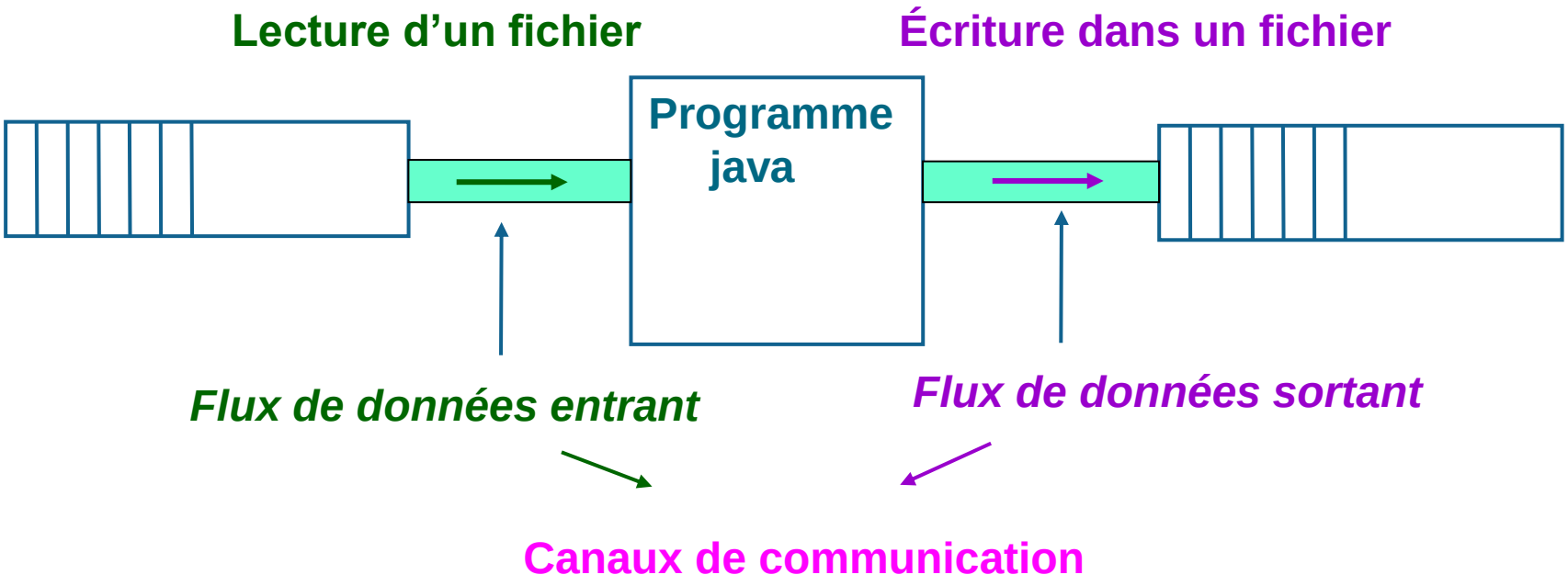
Chapitre 5
Fichiers

Enregistrement d'objets sérialisés dans des fichiers

Fichiers

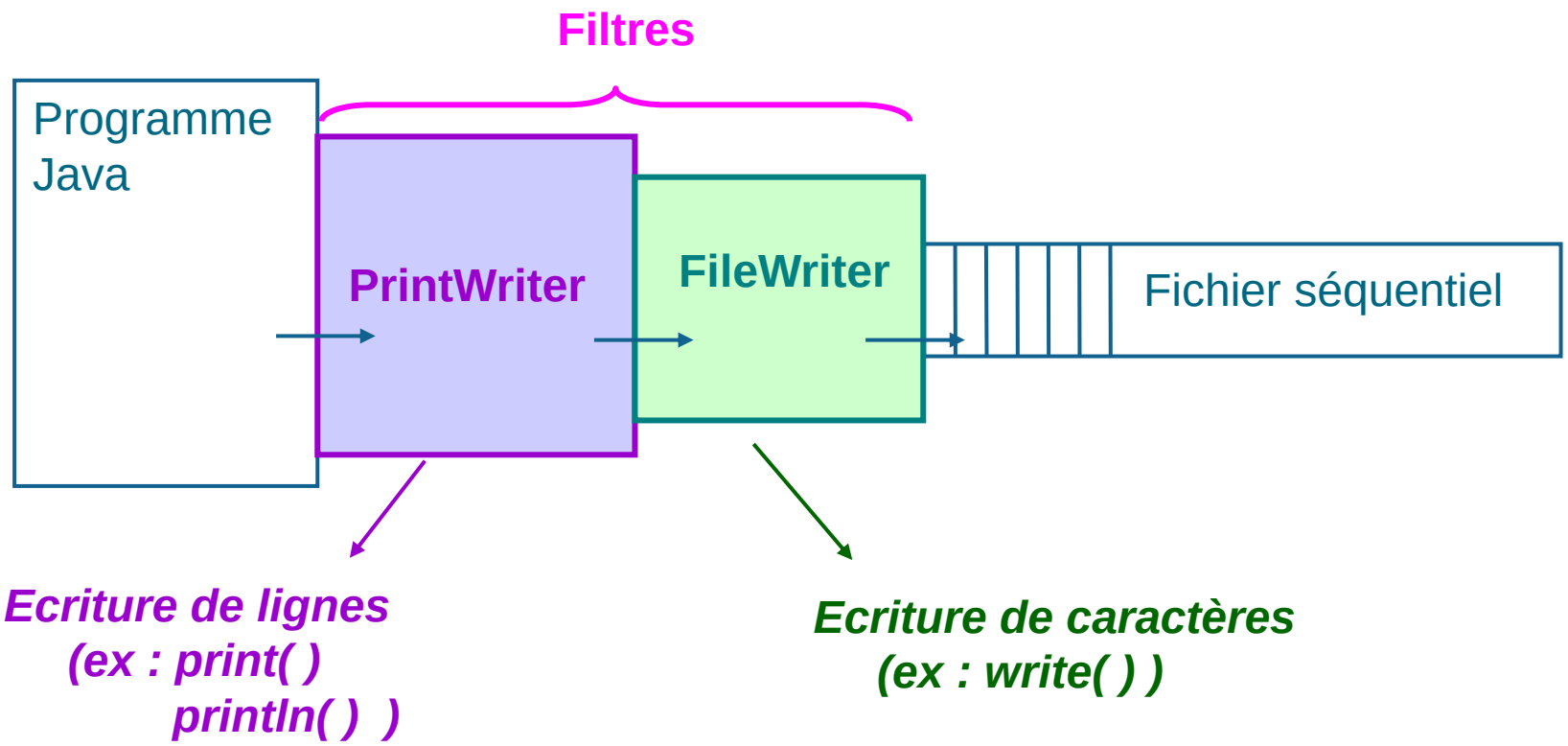

Lecture d'un fichier
Écriture dans un fichier
Programme java
Flux de données entrant
Flux de données sortant
Canaux de communication

Fichiers

1. Ecriture / lecture de chaînes de caractères

- Ecriture

Ecriture de chaînes de caractères



Try-with-resource

Statement **try-with-resource**

try dans lequel on déclare les ressources qui doivent être fermées quand le programme a terminé avec elles

Ces ressources doivent être des objets de classes implémentant l'interface **Closeable**

Ecriture de chaînes de caractères

```
public class Vehicle {

        private String plateNumber ;
        private String owner ;

    public Vehicle(String plateNumber, String owner) {
            …
     }

     …   // Getters et setters

}
```

Ecriture de chaînes de caractères

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


public class Main {
  public static void main(String[ ] args) {
          Vehicle vehicle;
```

Ecriture de chaînes de caractères

try-with-resource

***Ouverture** du fichier*

***Écrase** son contenu éventuel*

```
try ( FileWriter fileOutput = new FileWriter("vehicle.txt", false);
      PrintWriter output = new PrintWriter(fileOutput) ) {
```

Filtre :
écriture plus aisée

```
    vehicle = new Vehicle("1-CVA-123", "Julot");
    output.print(vehicle.getPlateNumber());
    output.println(vehicle.getOwner());
    vehicle = new Vehicle("1-DFG-456", "Mario");
    output.print(vehicle.getPlateNumber());
    output.println(vehicle.getOwner());
}
```

Ecriture de chaînes de caractères

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

Ecriture de chaînes de caractères

*Ouverture du fichier en mode **append***

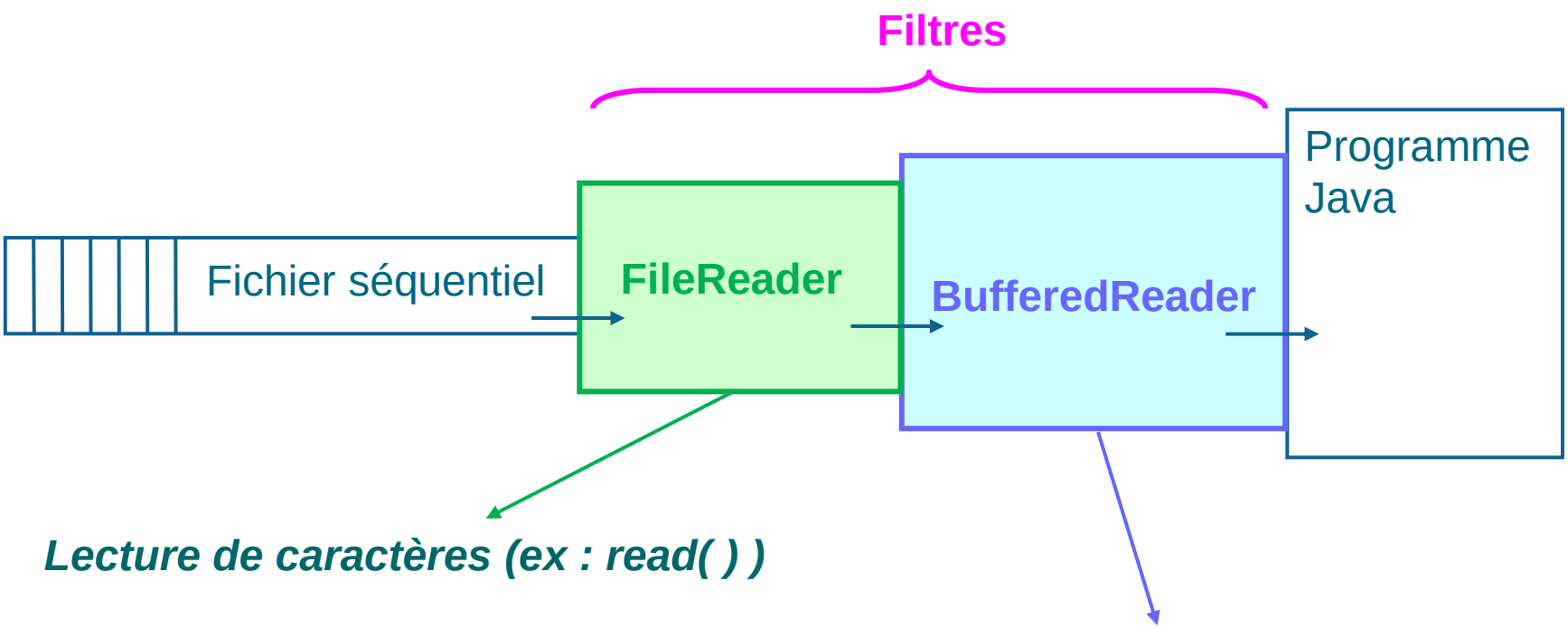
```
try (FileWriter fileOutput = new FileWriter("vehicle.txt", true);

    PrintWriter output = new PrintWriter(fileOutput)) {
```

Fichiers

1. Ecriture / lecture de chaînes de caractères

- Ecriture
- Lecture

Lecture de chaînes de caractères



Lecture de chaînes de caractères

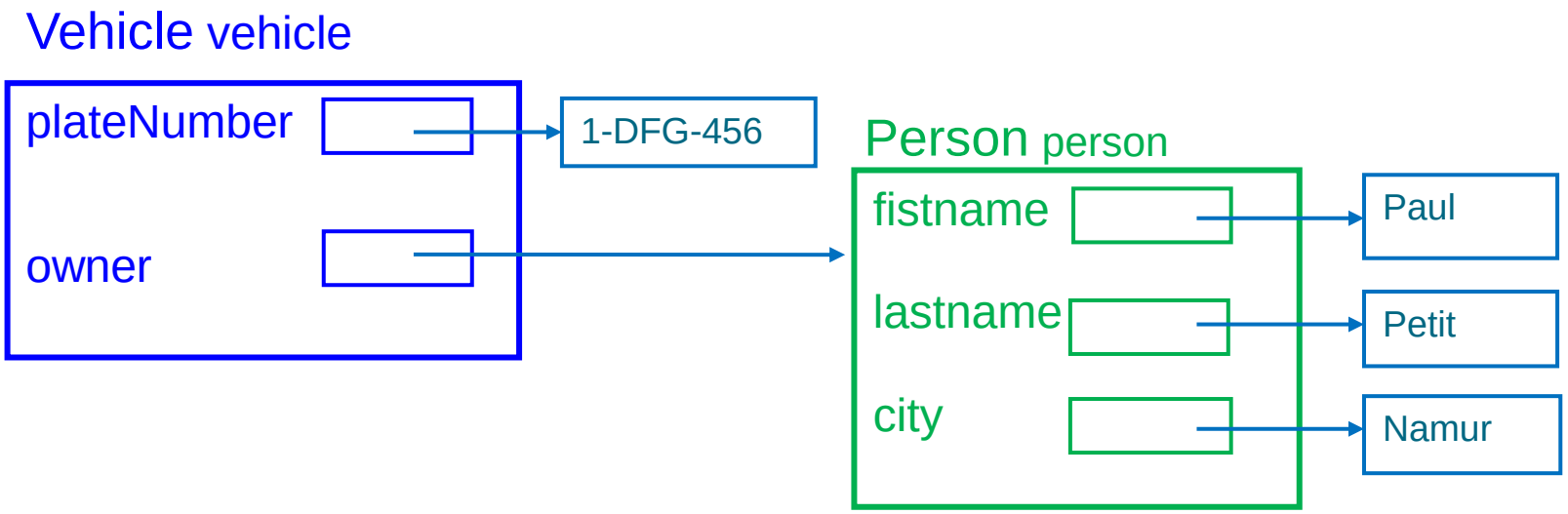
```
try ( FileReader fileInput = new FileReader("vehicle.txt");
      BufferedReader input = new BufferedReader(fileInput) ) {
          String readText;
          readText = input.readLine();
          while (readText != null) {
              …
              readText = input.readLine();
          }
}
catch (IOException ioException) {
        …
}
```

Filtre : optimise accès au disque

Fichiers

1. **Ecriture / lecture de chaînes de caractères**
   - Ecriture
   - Lecture
2. **Ecriture / lecture d'objets**

Lecture / écriture d'objets



Lecture / écriture d'objets

```
import java.io.* ;
public class Person implements Serializable  {
    private String firstname ;
    private String lastname ;
    private String city ;

    public Person(String firstname, String lastname, String city) {
        ...
    }
    …
    public String toString( ) {
        return firstname + " " + lastname +"  habitant à " + city ;
    }
}
```

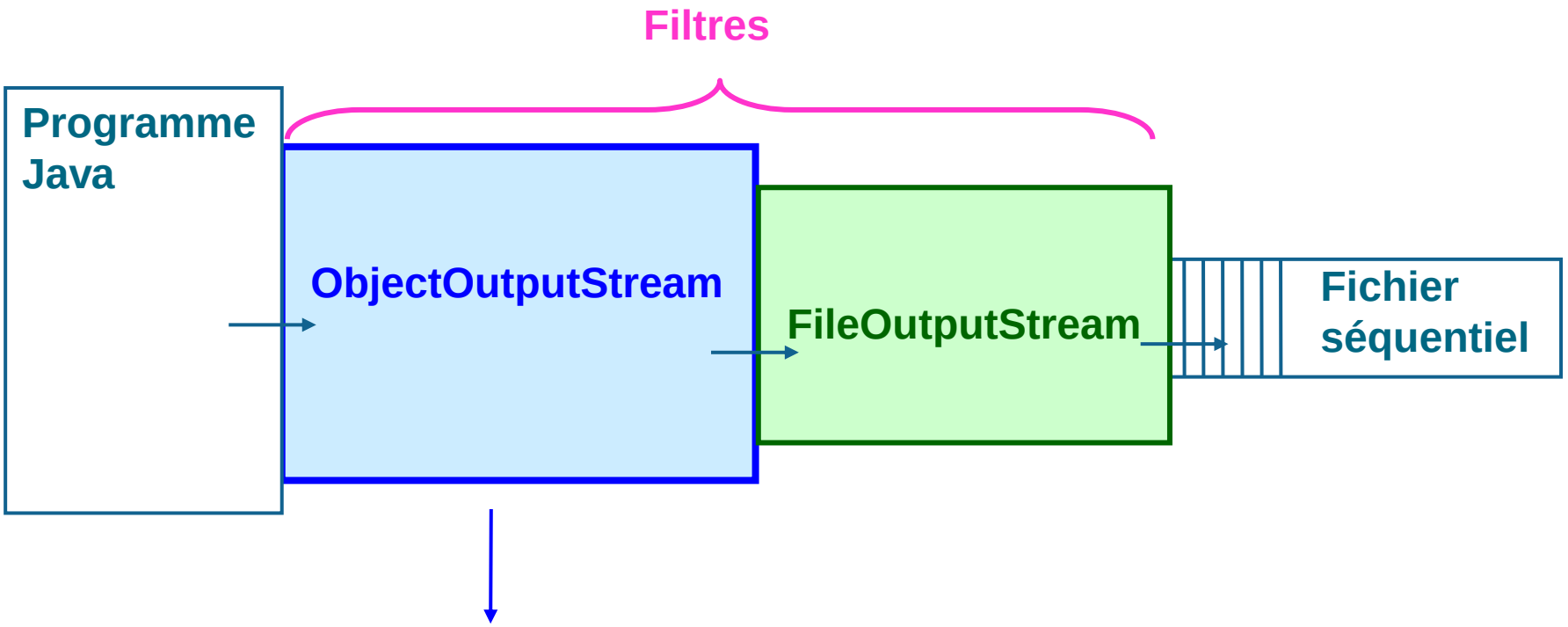
Lecture / écriture d'objets

```
import java.io.* ;
public class Vehicle implements Serializable {
    private String plateNumber ;
    private Person owner ;

    public Vehicle(String plateNumber, Person owner) {
        this.plateNumber = plateNumber;
        this.owner = owner;
    }

    …

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

Fichiers

1. **Ecriture / lecture de chaînes de caractères**
   - Ecriture
   - Lecture
2. **Ecriture / lecture d'objets**
   - Ecriture

Ecriture d'objets



Ecriture d'objet (ex : writeObject( ))

Ecriture d'objets

Flux de données

```
try ( FileOutputStream fileOutput = new FileOutputStream("vehicle.txt");
      ObjectOutputStream output = new ObjectOutputStream(fileOutput) ) {
```

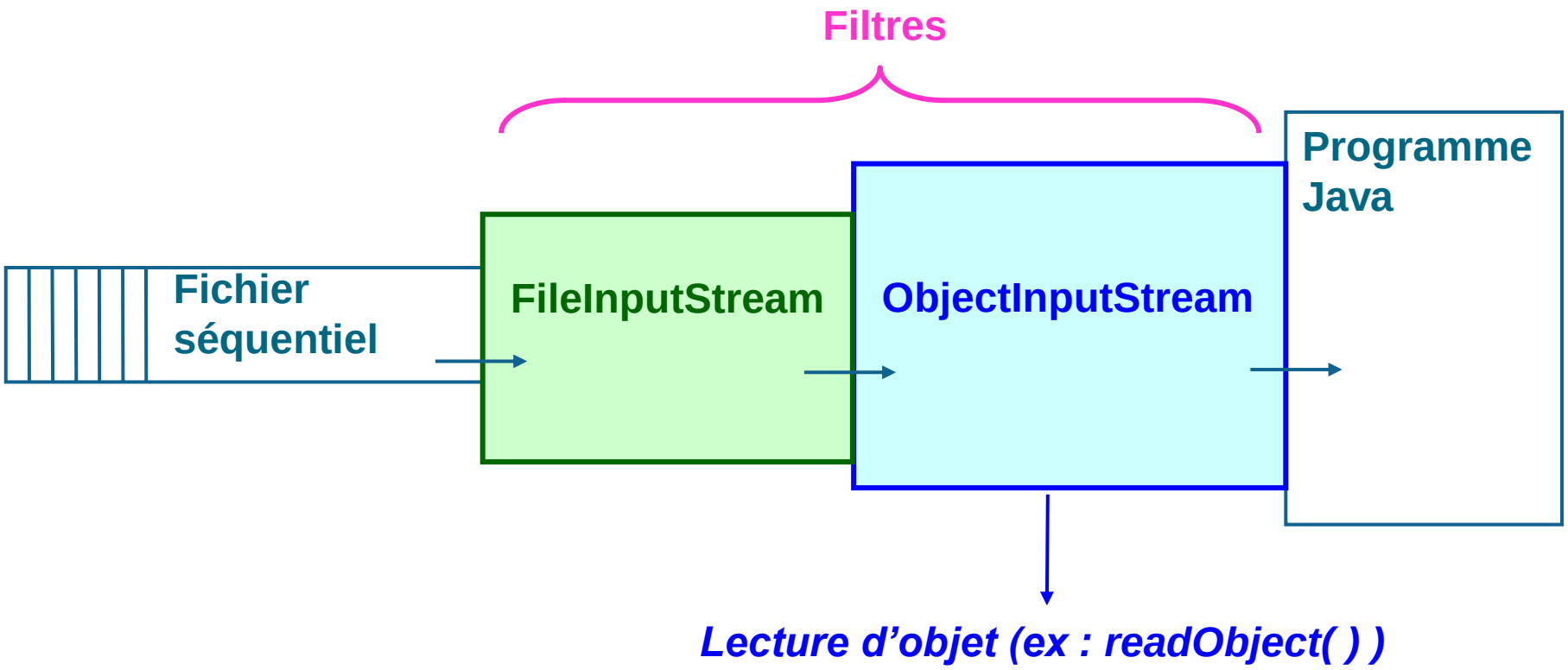
Flux d'objets

```
      output.writeObject (new Vehicle ("1-LOL-658",new Person(…))) ;
      output.writeObject (new Vehicle ("1-GAG-789",new Person(…))) ;
      output.writeObject (new Vehicle ("1-TKC-789",new Person(…))) ;
}
catch (IOException ioException) {
      ...
}
```

Fichiers

1. **Ecriture / lecture de chaînes de caractères**
   - Ecriture
   - Lecture
2. **Ecriture / lecture d'objets**
   - Ecriture
   - Lecture

Lecture d'objets



Lecture d'objets

```
Vehicle vehicle;
try ( FileInputStream fileInput = new FileInputStream("vehicle.txt");
      ObjectInputStream input = new ObjectInputStream(fileInput) ) {

       while (fileInput.available() > 0) {
            vehicle = (Vehicle) input.readObject() ;
            …
       }
}
```

Flux de données

Flux d'objets

Attention au casting obligatoire si on veut appeler une méthode de la classe Vehicle

Lecture d'objets

```
catch (ClassNotFoundException classNotFoundException) {
        ...
}

catch (IOException ioException) {
        ...
}
```