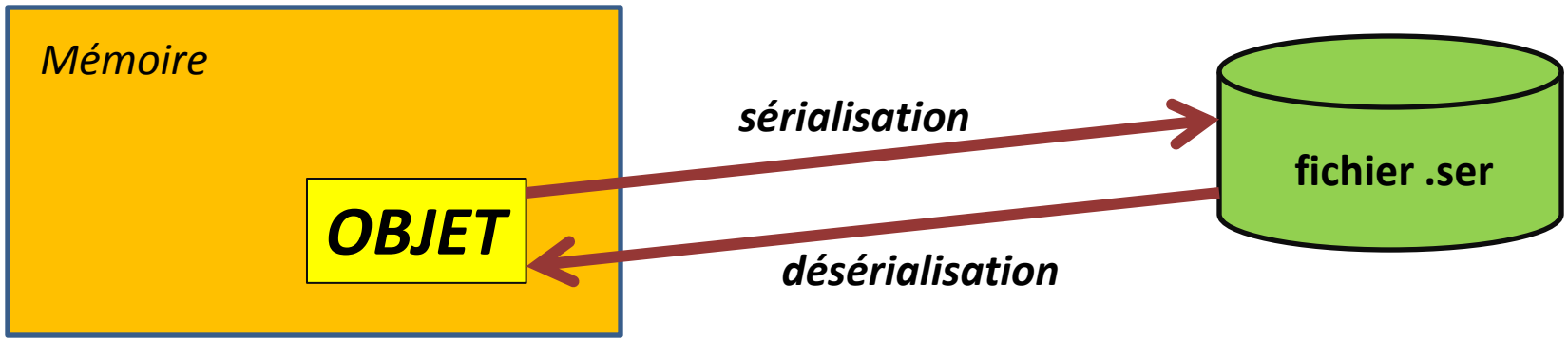
Sérialisation : introduction

Ahcène Bounceur

Intérêt

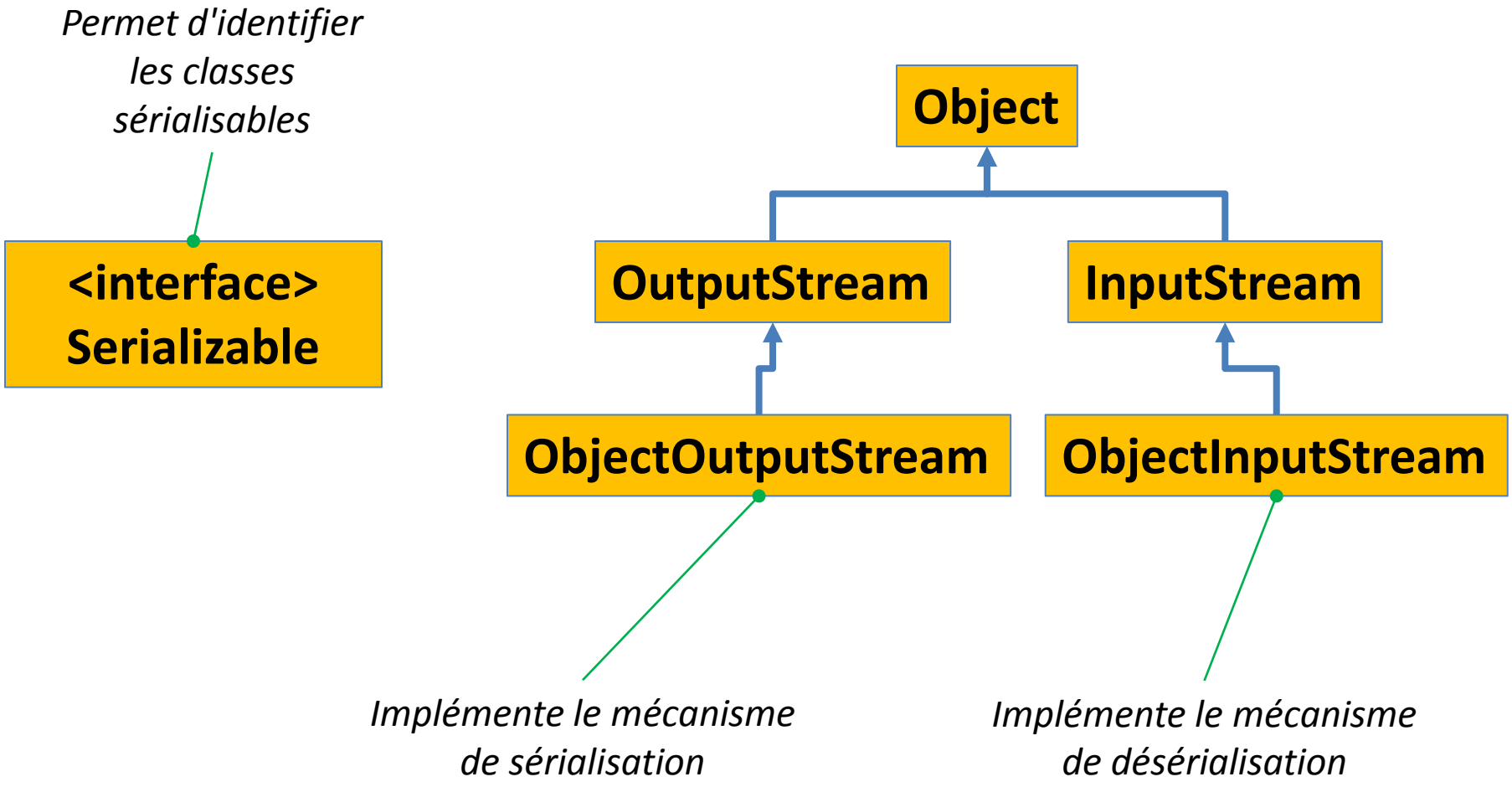
- Traduit un objet en un flot d'octets
  - permet de transmettre des objets sur le réseau
- Permet de rendre des objets persistants
  - quand l'application se termine elle sauve dans un fichier certains objets
  - quand elle est relancée, elle relit ces objets précédemment sauvegardés



Introduction

- **Sérialisation binaire**
  - *Permet de rendre des objets persistants en écrivant les données présentes en mémoire vers un flux de données binaires*
  - *introduction dans le JDK 1.1 d'un mécanisme de sérialisation*
    - *permet de sérialiser les objets de manière transparente et indépendante du système d'exploitation.*
    - *S'appuie sur les flux d'entrée sortie (**java.io**)*

Introduction



Interface sérialisable

- Pour qu'un objet puisse être sérialisé il faut que sa classe implémente l'interface **`Serializable`** ou hériter d'une classe elle-même sérialisable.

```
public interface Serializable {
}
```

Interface qui ne contient ni attributs ni méthode. Interface de marquage (tag interface)

- La sérialisation d'un objet consiste à écrire ses attributs sur un flux de sortie binaire.
- Tout attribut est sérialisé si :
  - *Il est de type primitif (int, char, boolean, ….) ou est une référence dont le type est un type sérialisable (dans ce cas l'objet référencé est sérialisé)*
  - *Il n'est pas déclaré static*
  - *Il n'est pas déclaré transient*
  - *Il n'est pas hérité d'une classe mère sauf si celle-ci est elle-même sérialisable*

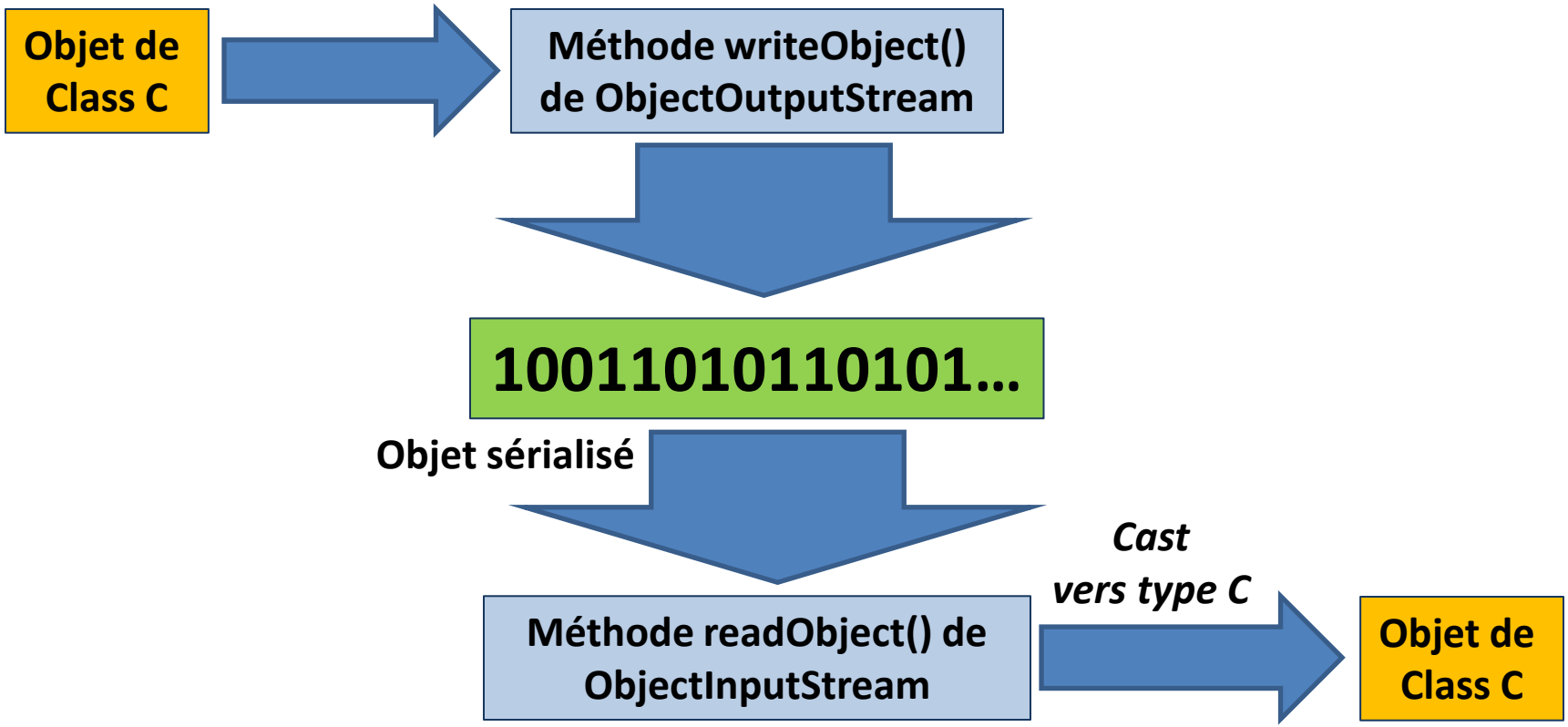
Classes ObjectOutputStream et ObjectInputStream (1/2)

- **`ObjectOutputStream`** représente "un flux objet" qui permet de sérialiser un objet grâce à la méthode **`writeObject()`**
- **`ObjectInputStream`** représente "un flux objet" qui permet de désérialiser un objet grâce à la méthode **`readObject()`**

Ne pas oublier : Les exceptions

- `private void readObject(ObjectInputStream stream)` **`throws IOException, ClassNotFoundException`**
- `private void writeObject(ObjectOutputStream stream)` **`throws IOException`**

Classes ObjectOutputStream et ObjectInputStream (2/2)



Exemple 1 (1/3)

```
import java.io.Serializable;

public class A implements Serializable {
   private int v = 10 ;
   private String s ;

   public A(int v, String s) {
        this.v = v ;
        this.s = s ;
   }
   public float getV() {
        return  v ;
   }
   public String getS() {
        return s ;
   }
}
```

Exemple 1 (2/3)

```
import java.io.ByteArrayOutputStream;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;

public class TestSerial {
   public static void main(String[] args) throws Exception {
        A objA = new A(35,"salut");
        FileOutputStream fos = new FileOutputStream("objeta.ser");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(objA);
   }
}
```

Le fichier objeta.ser est créé sur le disque dur

Exemple 1 (3/3)

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.ObjectInputStream;

public class TestSerialLecture {
   public static void main(String[] args) throws FileNotFoundException,
   IOException, ClassNotFoundException {
        ObjectInputStream ois = new ObjectInputStream(new
   FileInputStream("objeta.ser"));
        A a = (A)(ois.readObject()) ;
        System.out.println(a.getV());
        System.out.println(a.getS());
   }
}
```

Console :

```
35
salut
```

Exemple 2 (1/3)

```
import java.io.Serializable;

public class A implements Serializable {
   private transient int v = 10 ;
   private String s ;

   public A(int v, String s) {
        this.v = v ;
        this.s = s ;
   }
   public float getV() {
        return  v ;
   }
   public String getS() {
        return s ;
   }
}
```

Exemple 2 (2/3)

```
import java.io.ByteArrayOutputStream;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;

public class TestSerial {
   public static void main(String[] args) throws Exception {
        A objA = new A(35,"salut");
        FileOutputStream fos = new FileOutputStream("objeta.ser");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(objA);
   }
}
```

Le fichier objeta.ser est créé sur le disque dur

Exemple 2 (3/3)

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.ObjectInputStream;

public class TestSerialLecture {
   public static void main(String[] args) throws FileNotFoundException,
   IOException, ClassNotFoundException {
        ObjectInputStream ois = new ObjectInputStream(new
   FileInputStream("objeta.ser"));
        A a = (A)(ois.readObject()) ;
        System.out.println(a.getV());
        System.out.println(a.getS());
   }
}
```

Console :

```
0
salut
```

Questions