

Exemple de sérialisation

Java offre un mécanisme puissant et facile d'utilisation pour sauvegarder et restituer des objets complexes sur disque.

Un objet est qualifié de complexe s'il contient des références à d'autres objets, qui eux-mêmes peuvent contenir des références à d'autres objets. Les objets références peuvent l'être plusieurs fois.

Il s'agit donc de parcourir récursivement un graphe avec des occurrences multiples.

Les classes désirant utiliser cette fonctionnalité doivent implémenter **l'interface Serializable du package java.io.**

Tous les composants graphiques de AWT sont sérialisables.

Les constructeurs pour déclarer des flux d'entrée et de sortie et les méthodes associées sont :

ObjectInputStream (FileInputStream)

méthodes : **Serializable readObject ()**
 close()

ObjectOutputStream (FileOutputStream)

méthodes : **writeObject (Serializable)**
 flush()
 close()

exemple :

// fichier Customer.java

import java.io.*;

public class Customer implements Serializable

{

 private String name;

 private int age;

 public Customer()

 {

 this("Tahiti", 22);

 }

 public Customer(String theName, int theAge)

 {

 name = theName;

 age = theAge;

 }

 public String toString()

 {

 String temp;

 temp = "Name: " + name + "\n";

 temp += "Age: " + age + "\n";

 return temp;

 }

 public void save(String fileName) throws IOException

 {

 FileOutputStream fos;

ObjectOutputStream oos;

 fos = new FileOutputStream(fileName);

oos = new ObjectOutputStream(fos);

oos.writeObject(this);

```

        oos.flush();
        oos.close();
    }

    public static Customer load(String fileName) throws
IOException, ClassNotFoundException
    {
        FileInputStream fis;
        ObjectInputStream ois;

        Customer tempCustomer = null;

        fis = new FileInputStream(fileName);
        ois = new ObjectInputStream(fis);

        tempCustomer = (Customer) ois.readObject();
        ois.close();
        return tempCustomer;
    }
}

```

// fichier CustomerDriver.java

```
import java.io.*;
```

```

public class CustomerDriver
{
    public static void main(String arg[])
    {
        Customer customerA;
        Customer customerB;
        customerA = new Customer("Janine", 24);

        try {
            // sauvegarde customerA
            customerA.save("foo.bar");
            System.out.println("Customer A");
            System.out.println(customerA);

            // charge l'objet customer du fichier
            // foo.bar
            customerB = null;

```

```
        customerB = Customer.load("foo.bar");
        System.out.println("Customer B");
        System.out.println(customerB);
    }
    catch (IOException exc)
    {
        System.out.println(exc);
    }
    catch (ClassNotFoundException exc)
    {
        System.out.println(exc);
    }
}
}
```