

La cabecera se crea cada vez que se pone *new ObjectOutputStream(fichero)*. Para que no se añadan estas cabeceras lo que se hace es *redefinir la clase ObjectOutputStream creando una nueva clase que la herede (extends)*. Y dentro de esa clase se redefine el método *writeStreamHeader()* que es el que escribe las cabeceras, y hacemos que ese método no haga nada. De manera que si el fichero ya se ha creado se llamará a ese método de la clase redefinida.

La clase redefinida quedará así:

```
public class MiObjectOutputStream extends ObjectOutputStream
{
    public MiObjectOutputStream(OutputStream out) throws IOException
    {
        super(out);
    }
    protected MiObjectOutputStream()
        throws IOException, SecurityException
    {
        super();
    }
    // Redefinición del método de escribir la cabecera
    // para que no haga nada.
    protected void writeStreamHeader() throws IOException
    {
    }
}
```

Y dentro de nuestro programa a la hora de abrir el fichero para añadir nuevos objetos se pregunta si ya existe, si existe, se crea el objeto con la clase redefinida, y si no existe, el fichero se crea con la clase *ObjectOutputStream*:

```
File fichero = new File(nombrefichero);
ObjectOutputStream dataOS;
if (!fichero.exists())
{
    //Si el fichero no existe crea un ObjectOutputStream, la primera vez
    FileOutputStream fileout;
    fileout = new FileOutputStream(fichero);
    dataOS = new ObjectOutputStream(fileout);
}
else
{
    // Si ya existe el fichero creará un ObjectOutputStream

    // con el método writeStreamHeader redefinido (sin hacer nada)
    dataOS = new MiObjectOutputStream
        (new FileOutputStream(fichero,true));
} //fin if
```