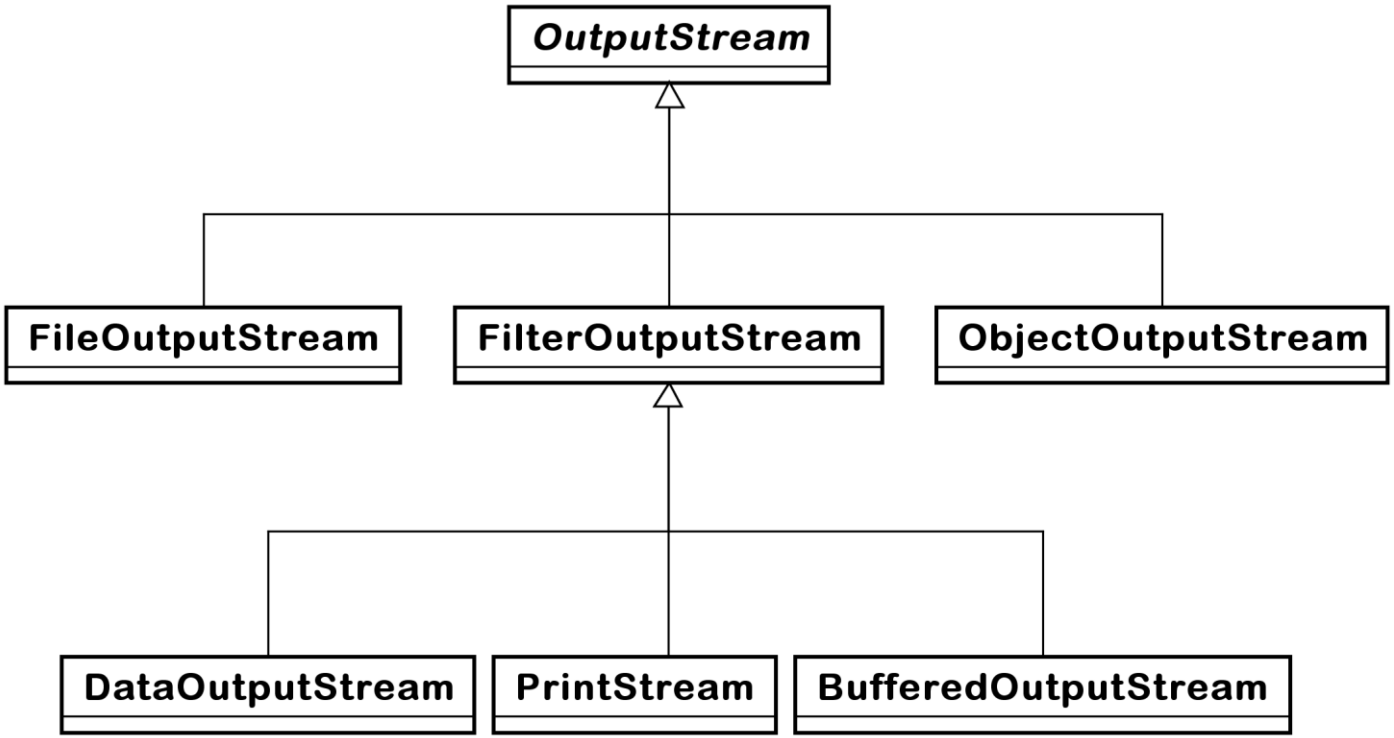
Serialização

Classes de persistência

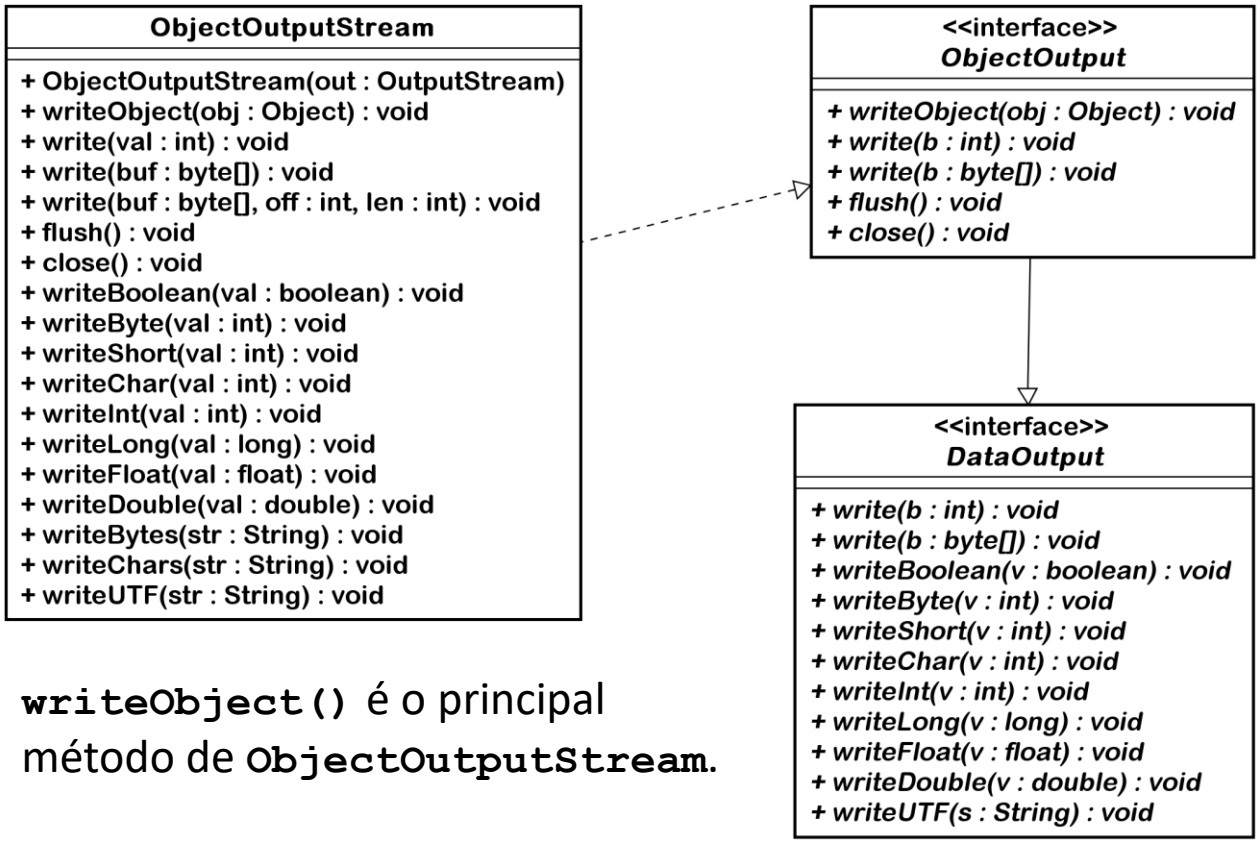


ObjectOutputStream e ObjectInputStream

A **serialização** de objetos consistem em converter os dados de estado de um objeto numa representação binária ou textual, a fim de armazená-lo num meio de armazenamento secundário

A operação inversa (a partir de uma cadeia de bits reconstruir um objeto), é chamada de **desserialização**

ObjectOutputStream



`writeObject()` é o principal método de `ObjectOutputStream`.

Exemplo - Serialização

```
File f = new File("Objetos.bin");
try (FileOutputStream fos = new FileOutputStream(f);
     ObjectOutputStream o = new ObjectOutputStream(fos)) {
    o.writeObject("Boa noite");
    o.writeObject(new ContaBancaria("1","Lucas",300));
    o.writeObject(new Veiculo("MIE-3932","Anderson"));
}
```

Observações

- Podem ser gravados vários objetos no *stream*.
- Para um objeto ser serializado, este deve implementar a interface **`Serializable`**
  - **`Serializable`** é uma *interface de marcação*
- Valores de variáveis estáticas não são serializados.
- Para não serializar determinada variável de instância, introduzir o modificador **`transient`**, como abaixo:

```
private transient String atributo;
```

Exemplo - Desserialização

```
String objeto1;
ContaBancaria objeto2;
Veiculo objeto3;

try (FileInputStream fis = new FileInputStream(f);
     ObjectInputStream ois = new ObjectInputStream(fis)) {
    objeto1 = (String) ois.readObject();
    objeto2 = (ContaBancaria) ois.readObject();
    objeto3 = (Veiculo) ois.readObject();
}
```

Para ler os objetos do *stream*, deve-se executar as operações de leitura na mesma ordem em que foram executadas as operações de gravação

Exceções lançadas

`ClassNotFoundException`

Ocorre quando se tenta desserializar objeto de uma classe desconhecida

`NotSerializableException`.

Ocorre quando se quer serializar um objeto de uma classe que não implementa a interface **`Serializable`**

Serialização - Versionamento

- É possível “versionar” uma classe ao persistir um objeto.
- O versionamento garante que objetos somente possam ser reconstituídos quando recuperados a partir da mesma versão de classe que persistiu o objeto.
- Para estabelecer um número de versão, acrescentar:

```
private static final long serialVersionUID = 1L;
```

Serialização - Versionamento

- *serialVersionUID* deveria ser um identificador único para uma versão.
- Quando não há declaração de serialVersionUID, Java gera um número de versão
- É recomendável que todas as classes serializáveis declarem explicitamente um valor para serialVersionUID.
- A exceção **`java.io.InvalidClassException`** é lançada quando ocorre tentativa de desserializar um objeto que foi serializado com versão diferente.

StreamCorruptedException

- Uma exceção da classe **`StreamCorruptedException`** pode ser lançada ao ler um arquivo com objetos persistidos
- Ocorre ao ler um arquivo que foi gravado com **`FileOutputStream`** com objetivo de acrescentar dados.
- Falha ocorre porque sempre que **`ObjectOutputStream`** é criado, grava um novo cabeçalho no arquivo, constituído de 4 bytes

00	01	02	03	04	05	06	07	08	09
ac	ed	00	05	74	00	09	42	6f	61
74	00	0e	53	65	67	75	6e	64	61