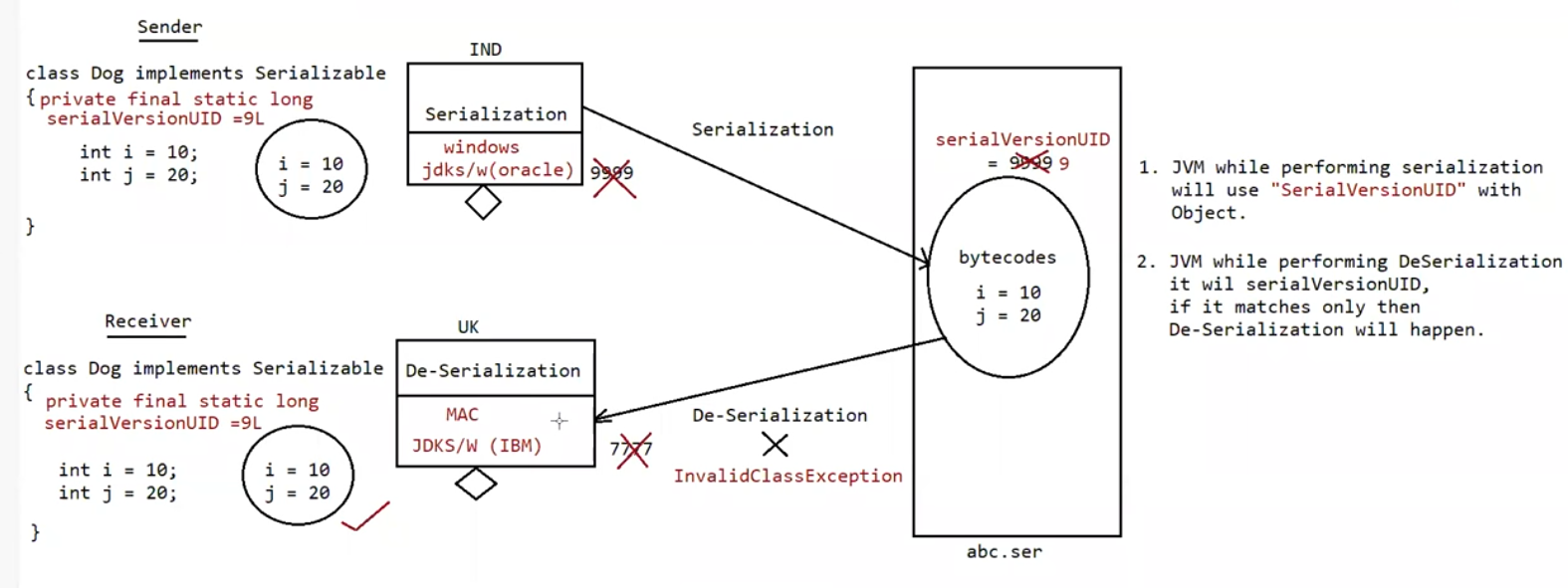
SerialVersionUID



Eg: Serial\_Version\_Uid\_Sender

Eg:Serial\_Version\_Uid\_Reciever

serialVersionUID

=> To perform Serialization & Deserialization internally JVM will use a unique identifier,which is nothing but serialVersionUID

=> At the time of serialization JVM will save serialVersionUID with object.

=> At the time of Deserialization JVM will compare serialVersionUID and if it is matched then only object will be Deserialized otherwise we will get RuntimeException saying "InvalidClassException".

The process in depending on default serialVersionUID are :

1. After Serializing object if we change the .class file then we can't perform deserialization because of mismatch in serialVersionUID of local class and serialized object in this case at the time of Deserialization we will get RuntimeException saying in "InvalidClassException".

2. Both sender and receiver should use the same version of JVM if there any incompatability in JVM versions then receive unable to deserializable because of different serialVersionUID, in this case receiver will get RuntimeException saying "InvalidClassException".

3. To generate serialVersionUID internally JVM will use complex algorithm which may create performance problems.

Usage of StringTokenizer :

It is a part of java.util package

It is used to split the entire string into multiple tokens based delimiter we supply

Eg: String data = “sachin Ramesh Tendulkar”

StringTokenizer stk = new StringTokenizer(data)

StringTokenizer stk = new StringTokenizer(data,” “)

Eg: StringTokenizer\_Eg1