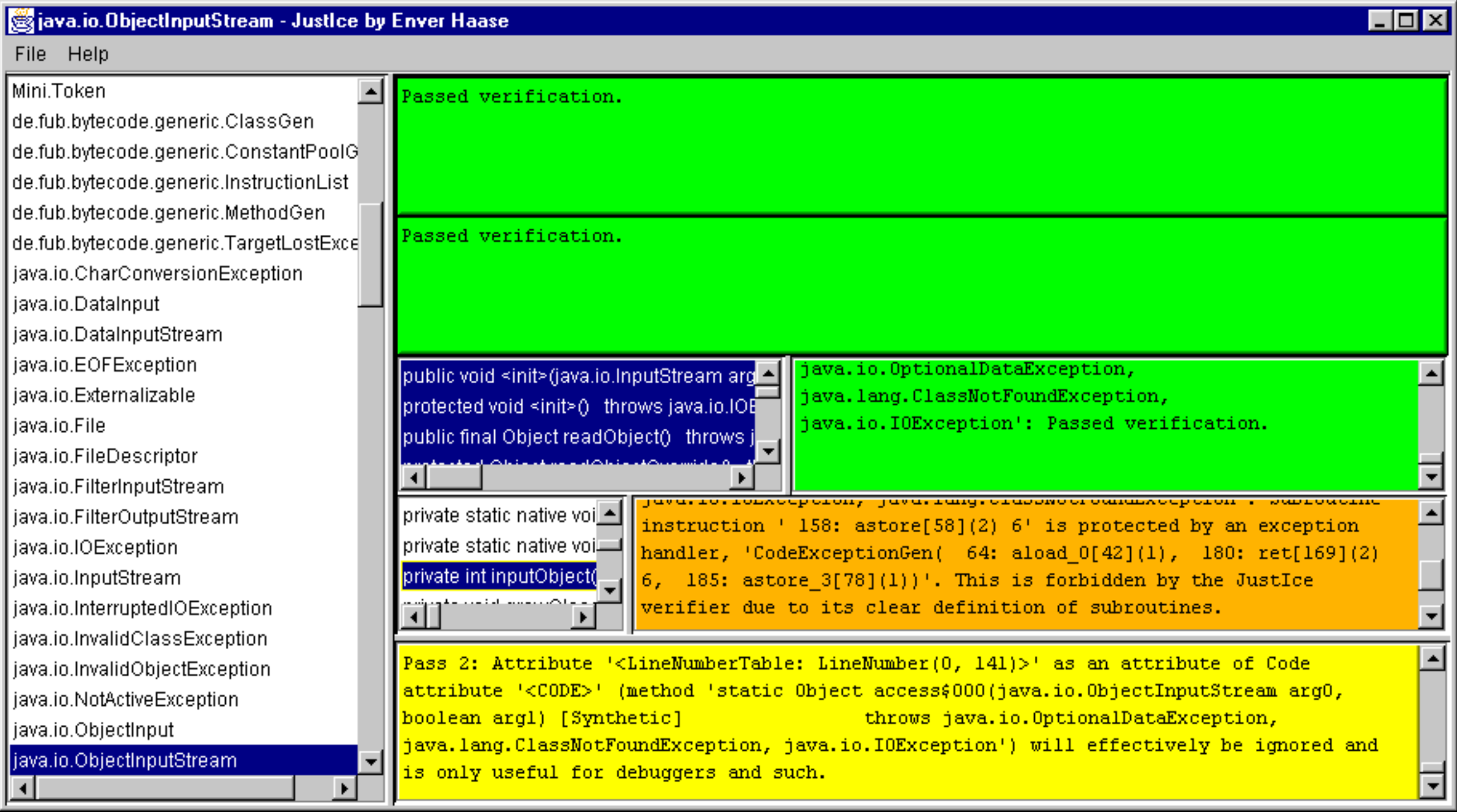
java.io.ObjectInputStream - JustIce by Enver Haase
File
Help
Mini.Token
de.fub.bytecode.generic.ClassGen
de.fub.bytecode.generic.ConstantPoolG
de.fub.bytecode.generic.InstructionList
de.fub.bytecode.generic.MethodGen
de.fub.bytecode.generic.TargetLostExce
java.io.CharConversionException
java.io.DataInput
java.io.DataInputStream
java.io.EOFException
java.io.Externalizable
java.io.File
java.io.FileDescriptor
java.io.FilterInputStream
java.io.FilterOutputStream
java.io.IOException
java.io.InputStream
java.io.InterruptedIOException
java.io.InvalidClassException
java.io.InvalidObjectException
java.io.NotActiveException
java.io.ObjectInput
java.io.ObjectInputStream
Passed verification.
Passed verification.
public void <init>(java.io.InputStream arg
protected void <init>() throws java.io.IOE
public final Object readObject() throws j
java.io.OptionalDataException,
java.lang.ClassNotFoundException,
java.io.IOException': Passed verification.
private static native voi
private static native voi
private int inputObject(
instruction ' 158: astore[58](2) 6' is protected by an exception
handler, 'CodeExceptionGen(64: aload_0[42](1), 180: ret[169](2)
6, 185: astore_3[78](1))'. This is forbidden by the JustIce
verifier due to its clear definition of subroutines.
Pass 2: Attribute '<LineNumberTable: LineNumber(0, 141)>' as an attribute of Code
attribute '<CODE>' (method 'static Object access$000(java.io.ObjectInputStream arg0,
boolean arg1) [Synthetic] throws java.io.OptionalDataException,
java.lang.ClassNotFoundException, java.io.IOException') will effectively be ignored and
is only useful for debuggers and such.