Template version: proc-RFC_template-1_01_011004.dot

RFC 39 - Small Execution Environment

Confidential, Draft
rfc-0039-Small_EE_for_Bundles

60 Pages

Abstract

This RFC contains a proposed specification for a small execution environment for bundles. The proposed specification is small but useful "subset" of Java.

0 Document Information

0.1 Table of Contents

0.2 Status

This document specifies the minimal set of classes that can be expected by a bundle, in the terms laid out by RFC 6. Distribution of this document is unlimited within OSGi.

0.3 Acknowledgement

The developers and customers of IBM VisualAge Micro Edition were significant contributors to the development of the information in this document.

0.4 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [1].

```
Source code is shown in this typeface.
```

0.5 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	6 September 2001	Rick DeNatale, IBM/OTI, Rick_DeNatale@oti.com
First review draft	28 September 2001	Rick DeNatale, IBM/OTI, Rick_DeNatale@oti.com BJ Hargrave, IBM, Hargrave@us.ibm.com
Update copyright/license	04 October 2001	Updated copyright/license to new OSGi template. BJ Hargrave, IBM, Hargrave@us.ibm.com
Second draft	08 October 2001	Updated to remove methods not in Foundation profile based upon analysis done by David Bowen of Sun. BJ Hargrave, IBM, Hargrave@us.ibm.com
Third draft	09 October 2001	Forgot to remove deprecated methods from ThreadGroup. BJ Hargrave, IBM, Hargrave@us.ibm.com
Fourth draft	02 November 2001	Corrected character converter names and aliases. Added reference to sources. BJ Hargrave, IBM, Hargrave@us.ibm.com
Fifth draft	14 December 2001	Added missing public inner classes/interfaces. Re-generated class and member information from the source code using a more simple format to significantly reduce the length of the document. BJ Hargrave, IBM, Hargrave@us.ibm.com

1 Introduction

This document proposes a minimal Execution Environment[2] for bundles. This proposal meets the following requirements.

1. The proposed execution environment is a proper subset of J2SE (Java 2 Standard Edition)

2. The proposed execution environment is a proper subset of J2ME (Java 2 Micro Edition) Foundation Profile

3. The proposed execution environment supports the existing, published OSGi API (Service Platform Release 2) signatures.

2 Motivation and Rationale

RFC 6[2] defines an Execution Environment to be "the collection of public and protected methods and fields in public classes and interfaces provided by the runtime environment (RE) that an OSGi Bundle can assume exists and can be accessed by the bundle at runtime." The goal is to provide a specification of what a bundle can reasonable expect to be available to it, without overly constraining implementations of the OSGi framework or the underlying Java platform.

The present RFC is proposed as an alternative to RFC 26[3]. This RFC is based on customer requirements to provide a java class library for the OSGi environment with a small (<1MB) footprint.

Since OSGi pre-dates J2ME, we have the situation that existing gateway implementations and bundles have been built to a subset of the function in J2SE. In particular, CDC and J2SE differ in the contents of the package java.net, and the substitution of APIs in javax.microedition.io. This RFC provides support for this legacy.

In the interest of footprint, it does not include the entire set of collection classes, nor does it contain the BigInteter and BigDecimal classes, there is sufficient API provided to allow these to be added as a separate bundle.

This document uses generated formal parameter names that are not considered to be part of the specification; implementers are free to name formal parameters as they wish.

3 Technical Discussion

3.1 Classes and members

This section details the classes and members that are part of this execution environment.

Note: The values of the final static fields are not part of this specification. They are present to permit the source code to compile.

```
package java.io;
public class BufferedInputStream extends java.io.FilterInputStream {
    public BufferedInputStream(java.io.InputStream var0) { super(null); }
    public BufferedInputStream(java.io.InputStream var0, int var1) { super(null); }
    public int available() throws java.io.IOException { return 0; }
    public void close() throws java.io.IOException { }
    public void mark(int var0) { }
    public boolean markSupported() { return false; }
    public int read() throws java.io.IOException { return 0; }
    public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
    public void reset() throws java.io.IOException { }
    public long skip(long var0) throws java.io.IOException { return 0l; }
    protected byte[] buf;
    protected int count;
    protected int marklimit;
    protected int markpos;
    protected int pos;
}

package java.io;
public class BufferedOutputStream extends java.io.FilterOutputStream {
    public BufferedOutputStream(java.io.OutputStream var0) { super(null); }
    public BufferedOutputStream(java.io.OutputStream var0, int var1) { super(null); }
    public void flush() throws java.io.IOException { }
    public void write(byte[] var0, int var1, int var2) throws java.io.IOException { }
    public void write(int var0) throws java.io.IOException { }
    protected byte[] buf;
    protected int count;
}

package java.io;
public class BufferedReader extends java.io.Reader {
    public BufferedReader(java.io.Reader var0) { }
    public BufferedReader(java.io.Reader var0, int var1) { }
    public void close() throws java.io.IOException { }
    public void mark(int var0) throws java.io.IOException { }
    public boolean markSupported() { return false; }
    public int read() throws java.io.IOException { return 0; }
    public int read(char[] var0, int var1, int var2) throws java.io.IOException { return 0; }
    public java.lang.String readLine() throws java.io.IOException { return null; }
    public boolean ready() throws java.io.IOException { return false; }
    public void reset() throws java.io.IOException { }
    public long skip(long var0) throws java.io.IOException { return 0l; }
}

package java.io;
public class BufferedWriter extends java.io.Writer {
    public BufferedWriter(java.io.Writer var0) { }
    public BufferedWriter(java.io.Writer var0, int var1) { }
    public void close() throws java.io.IOException { }
    public void flush() throws java.io.IOException { }
    public void newLine() throws java.io.IOException { }
    public void write(char[] var0, int var1, int var2) throws java.io.IOException { }
    public void write(int var0) throws java.io.IOException { }
    public void write(java.lang.String var0, int var1, int var2) throws java.io.IOException { }
}

package java.io;
public class ByteArrayInputStream extends java.io.InputStream {
    public ByteArrayInputStream(byte[] var0) { }
    public ByteArrayInputStream(byte[] var0, int var1, int var2) { }
    public int available() { return 0; }
    public void close() throws java.io.IOException { }
    public void mark(int var0) { }
    public boolean markSupported() { return false; }
    public int read() { return 0; }
    public int read(byte[] var0, int var1, int var2) { return 0; }
    public void reset() { }
    public long skip(long var0) { return 0l; }
    protected byte[] buf;
    protected int pos;
    protected int mark;
```

```
        protected int count;
    }

    package java.io;
    public class ByteArrayOutputStream extends java.io.OutputStream {
        public ByteArrayOutputStream() { }
        public ByteArrayOutputStream(int var0) { }
        public void close() throws java.io.IOException { }
        public void reset() { }
        public int size() { return 0; }
        public byte[] toByteArray() { return null; }
        public java.lang.String toString() { return null; }
        public void write(byte[] var0, int var1, int var2) { }
        public void write(int var0) { }
        public void writeTo(java.io.OutputStream var0) throws java.io.IOException { }
        protected byte[] buf;
        protected int count;
    }

    package java.io;
    public class CharConversionException extends java.io.IOException {
        public CharConversionException() { }
        public CharConversionException(java.lang.String var0) { }
    }

    package java.io;
    public abstract interface DataInput {
        public abstract boolean readBoolean() throws java.io.IOException;
        public abstract byte readByte() throws java.io.IOException;
        public abstract char readChar() throws java.io.IOException;
        public abstract double readDouble() throws java.io.IOException;
        public abstract float readFloat() throws java.io.IOException;
        public abstract void readFully(byte[] var0) throws java.io.IOException;
        public abstract void readFully(byte[] var0, int var1, int var2) throws java.io.IOException;
        public abstract int readInt() throws java.io.IOException;
        public abstract java.lang.String readLine() throws java.io.IOException;
        public abstract long readLong() throws java.io.IOException;
        public abstract short readShort() throws java.io.IOException;
        public abstract int readUnsignedByte() throws java.io.IOException;
        public abstract int readUnsignedShort() throws java.io.IOException;
        public abstract java.lang.String readUTF() throws java.io.IOException;
        public abstract int skipBytes(int var0) throws java.io.IOException;
    }

    package java.io;
    public class DataInputStream extends java.io.FilterInputStream implements java.io.DataInput {
        public DataInputStream(java.io.InputStream var0) { super(null); }
        public final int read(byte[] var0) throws java.io.IOException { return 0; }
        public final int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
        public final boolean readBoolean() throws java.io.IOException { return false; }
        public final byte readByte() throws java.io.IOException { return 0; }
        public final char readChar() throws java.io.IOException { return 0; }
        public final double readDouble() throws java.io.IOException { return 0.0d; }
        public final float readFloat() throws java.io.IOException { return 0.0f; }
        public final void readFully(byte[] var0) throws java.io.IOException { }
        public final void readFully(byte[] var0, int var1, int var2) throws java.io.IOException { }
        public final int readInt() throws java.io.IOException { return 0; }
        public final java.lang.String readLine() throws java.io.IOException { return null; }
        public final long readLong() throws java.io.IOException { return 0l; }
        public final short readShort() throws java.io.IOException { return 0; }
        public final int readUnsignedByte() throws java.io.IOException { return 0; }
        public final int readUnsignedShort() throws java.io.IOException { return 0; }
        public final java.lang.String readUTF() throws java.io.IOException { return null; }
        public final int skipBytes(int var0) throws java.io.IOException { return 0; }
    }

    package java.io;
    public abstract interface DataOutput {
        public abstract void write(byte[] var0) throws java.io.IOException;
        public abstract void write(byte[] var0, int var1, int var2) throws java.io.IOException;
        public abstract void write(int var0) throws java.io.IOException;
        public abstract void writeBoolean(boolean var0) throws java.io.IOException;
        public abstract void writeByte(int var0) throws java.io.IOException;
        public abstract void writeBytes(java.lang.String var0) throws java.io.IOException;
        public abstract void writeChar(int var0) throws java.io.IOException;
        public abstract void writeChars(java.lang.String var0) throws java.io.IOException;
        public abstract void writeDouble(double var0) throws java.io.IOException;
        public abstract void writeFloat(float var0) throws java.io.IOException;
        public abstract void writeInt(int var0) throws java.io.IOException;
        public abstract void writeLong(long var0) throws java.io.IOException;
        public abstract void writeShort(int var0) throws java.io.IOException;
        public abstract void writeUTF(java.lang.String var0) throws java.io.IOException;
    }

    package java.io;
    public class DataOutputStream extends java.io.FilterOutputStream implements java.io.DataOutput {
        public DataOutputStream(java.io.OutputStream var0) { super(null); }
        public void flush() throws java.io.IOException { }
        public final int size() { return 0; }
        public void write(byte[] var0, int var1, int var2) throws java.io.IOException { }
        public void write(int var0) throws java.io.IOException { }
        public final void writeBoolean(boolean var0) throws java.io.IOException { }
        public final void writeByte(int var0) throws java.io.IOException { }
        public final void writeBytes(java.lang.String var0) throws java.io.IOException { }
        public final void writeChar(int var0) throws java.io.IOException { }
        public final void writeChars(java.lang.String var0) throws java.io.IOException { }
        public final void writeDouble(double var0) throws java.io.IOException { }
        public final void writeFloat(float var0) throws java.io.IOException { }
        public final void writeInt(int var0) throws java.io.IOException { }
        public final void writeLong(long var0) throws java.io.IOException { }
        public final void writeShort(int var0) throws java.io.IOException { }
        public final void writeUTF(java.lang.String var0) throws java.io.IOException { }
        protected int written;
    }

    package java.io;
    public class EOFException extends java.io.IOException {
        public EOFException() { }
        public EOFException(java.lang.String var0) { }
    }

    package java.io;
    public abstract interface Externalizable extends java.io.Serializable {
        public abstract void readExternal(java.io.ObjectInput var0) throws java.io.IOException,
    java.lang.ClassNotFoundException;
        public abstract void writeExternal(java.io.ObjectOutput var0) throws java.io.IOException;
    }

    package java.io;
    public class File implements java.io.Serializable, java.lang.Comparable {
        public File(java.io.File var0, java.lang.String var1) { }
        public File(java.lang.String var0) { }
        public File(java.lang.String var0, java.lang.String var1) { }
        public boolean canRead() { return false; }
        public boolean canWrite() { return false; }
        public int compareTo(java.lang.Object var0) { return 0; }
        public int compareTo(java.io.File var0) { return 0; }
        public boolean delete() { return false; }
        public void deleteOnExit() { }
        public boolean equals(java.lang.Object var0) { return false; }
        public boolean exists() { return false; }
        public java.lang.String getAbsolutePath() { return null; }
        public java.lang.String getCanonicalPath() throws java.io.IOException { return null; }
        public java.lang.String getName() { return null; }
        public java.lang.String getParent() { return null; }
        public java.lang.String getPath() { return null; }
        public int hashCode() { return 0; }
        public boolean isAbsolute() { return false; }
        public boolean isDirectory() { return false; }
        public boolean isFile() { return false; }
        public long lastModified() { return 0l; }
        public long length() { return 0l; }
        public java.lang.String[] list() { return null; }
        public boolean mkdir() { return false; }
        public boolean mkdirs() { return false; }
        public boolean createNewFile() throws java.io.IOException { return false; }
        public static java.io.File createTempFile(java.lang.String var0, java.lang.String var1, java.io.File
    var2) throws java.io.IOException { return null; }
        public boolean renameTo(java.io.File var0) { return false; }
        public java.lang.String toString() { return null; }
        public final static char separatorChar; static { separatorChar = 0; }
        public final static java.lang.String separator; static { separator = null; }
        public final static char pathSeparatorChar; static { pathSeparatorChar = 0; }
        public final static java.lang.String pathSeparator; static { pathSeparator = null; }
    }

    package java.io;
    public final class FileDescriptor {
        public FileDescriptor() { }
        public void sync() throws java.io.SyncFailedException { }
        public boolean valid() { return false; }
        public final static java.io.FileDescriptor in; static { in = null; }
        public final static java.io.FileDescriptor out; static { out = null; }
        public final static java.io.FileDescriptor err; static { err = null; }
    }

    package java.io;
    public class FileInputStream extends java.io.InputStream {
        public FileInputStream(java.io.File var0) throws java.io.FileNotFoundException { }
        public FileInputStream(java.io.FileDescriptor var0) { }
        public FileInputStream(java.lang.String var0) throws java.io.FileNotFoundException { }
        public int available() throws java.io.IOException { return 0; }
        public void close() throws java.io.IOException { }
        protected void finalize() throws java.io.IOException { }
        public final java.io.FileDescriptor getFD() throws java.io.IOException { return null; }
        public int read() throws java.io.IOException { return 0; }
        public int read(byte[] var0) throws java.io.IOException { return 0; }
        public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
        public long skip(long var0) throws java.io.IOException { return 0l; }
    }

    package java.io;
    public class FileNotFoundException extends java.io.IOException {
        public FileNotFoundException() { }
        public FileNotFoundException(java.lang.String var0) { }
    }

    package java.io;
    public class FileOutputStream extends java.io.OutputStream {
        public FileOutputStream(java.io.File var0) throws java.io.FileNotFoundException { }
        public FileOutputStream(java.io.FileDescriptor var0) { }
        public FileOutputStream(java.lang.String var0) throws java.io.FileNotFoundException { }
        public FileOutputStream(java.lang.String var0, boolean var1) throws java.io.FileNotFoundException {
    }
        public void close() throws java.io.IOException { }
        protected void finalize() throws java.io.IOException { }
        public final java.io.FileDescriptor getFD() throws java.io.IOException { return null; }
        public void write(byte[] var0) throws java.io.IOException { }
        public void write(byte[] var0, int var1, int var2) throws java.io.IOException { }
        public void write(int var0) throws java.io.IOException { }
    }

    package java.io;
    public final class FilePermission extends java.security.Permission implements java.io.Serializable {
        public FilePermission(java.lang.String var0, java.lang.String var1) { super(null); }
        public java.lang.String getActions() { return null; }
        public boolean equals(java.lang.Object var0) { return false; }
        public boolean implies(java.security.Permission var0) { return false; }
        public java.security.PermissionCollection newPermissionCollection() { return null; }
        public int hashCode() { return 0; }
    }

    package java.io;
    public class FileReader extends java.io.InputStreamReader {
        public FileReader(java.io.File var0) throws java.io.FileNotFoundException { super(null); }
        public FileReader(java.io.FileDescriptor var0) { super(null); }
        public FileReader(java.lang.String var0) throws java.io.FileNotFoundException { super(null); }
    }

    package java.io;
    public class FileWriter extends java.io.OutputStreamWriter {
        public FileWriter(java.io.File var0) throws java.io.IOException { super(null); }
        public FileWriter(java.io.FileDescriptor var0) { super(null); }
        public FileWriter(java.lang.String var0) throws java.io.IOException { super(null); }
        public FileWriter(java.lang.String var0, boolean var1) throws java.io.IOException { super(null); }
    }

    package java.io;
    public class FilterInputStream extends java.io.InputStream {
        protected FilterInputStream(java.io.InputStream var0) { }
        public int available() throws java.io.IOException { return 0; }
        public void close() throws java.io.IOException { }
        public void mark(int var0) { }
        public boolean markSupported() { return false; }
        public int read() throws java.io.IOException { return 0; }
        public int read(byte[] var0) throws java.io.IOException { return 0; }
        public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
        public void reset() throws java.io.IOException { }
        public long skip(long var0) throws java.io.IOException { return 0l; }
        protected java.io.InputStream in;
    }

    package java.io;
    public class FilterOutputStream extends java.io.OutputStream {
        public FilterOutputStream(java.io.OutputStream var0) { }
        public void close() throws java.io.IOException { }
        public void flush() throws java.io.IOException { }
        public void write(byte[] var0) throws java.io.IOException { }
        public void write(byte[] var0, int var1, int var2) throws java.io.IOException { }
        public void write(int var0) throws java.io.IOException { }
        protected java.io.OutputStream out;
    }

    package java.io;
    public abstract class InputStream {
        public InputStream() { }
        public int available() throws java.io.IOException { return 0; }
        public void close() throws java.io.IOException { }
        public void mark(int var0) { }
        public boolean markSupported() { return false; }
        public abstract int read() throws java.io.IOException;
        public int read(byte[] var0) throws java.io.IOException { return 0; }
        public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
        public void reset() throws java.io.IOException { }
        public long skip(long var0) throws java.io.IOException { return 0l; }
    }

    package java.io;
    public class InputStreamReader extends java.io.Reader {
        public InputStreamReader(java.io.InputStream var0) { }
        public InputStreamReader(java.io.InputStream var0, java.lang.String var1) throws
    java.io.UnsupportedEncodingException { }
        public void close() throws java.io.IOException { }
        public java.lang.String getEncoding() { return null; }
        public int read() throws java.io.IOException { return 0; }
        public int read(char[] var0, int var1, int var2) throws java.io.IOException { return 0; }
        public boolean ready() throws java.io.IOException { return false; }
    }

    package java.io;
    public class InterruptedIOException extends java.io.IOException {
        public InterruptedIOException() { }
        public InterruptedIOException(java.lang.String var0) { }
        public int bytesTransferred;
    }

    package java.io;
    public class InvalidClassException extends java.io.ObjectStreamException {
        public InvalidClassException(java.lang.String var0) { }
        public InvalidClassException(java.lang.String var0, java.lang.String var1) { }
        public java.lang.String getMessage() { return null; }
        public java.lang.String classname;
    }

    package java.io;
    public class InvalidObjectException extends java.io.ObjectStreamException {
        public InvalidObjectException(java.lang.String var0) { }
    }

    package java.io;
    public class IOException extends java.lang.Exception {
        public IOException() { }
        public IOException(java.lang.String var0) { }
    }

    package java.io;
    public class NotActiveException extends java.io.ObjectStreamException {
        public NotActiveException() { }
        public NotActiveException(java.lang.String var0) { }
    }

    package java.io;
    public class NotSerializableException extends java.io.ObjectStreamException {
        public NotSerializableException() { }
        public NotSerializableException(java.lang.String var0) { }
    }

    package java.io;
    public abstract interface ObjectInput extends java.io.DataInput {
        public abstract int available() throws java.io.IOException;
        public abstract void close() throws java.io.IOException;
        public abstract int read() throws java.io.IOException;
        public abstract int read(byte[] var0) throws java.io.IOException;
        public abstract int read(byte[] var0, int var1, int var2) throws java.io.IOException;
        public abstract java.lang.Object readObject() throws java.lang.ClassNotFoundException,
    java.io.IOException;
        public abstract long skip(long var0) throws java.io.IOException;
    }

    package java.io;
    public class ObjectInputStream extends java.io.InputStream implements java.io.ObjectInput,
    java.io.ObjectStreamConstants {
        protected ObjectInputStream() throws java.io.IOException, java.lang.SecurityException { }
        public ObjectInputStream(java.io.InputStream var0) throws java.io.StreamCorruptedException,
    java.io.IOException { }
        public int available() throws java.io.IOException { return 0; }
        public void close() throws java.io.IOException { }
        public void defaultReadObject() throws java.io.IOException, java.lang.ClassNotFoundException,
    java.io.NotActiveException { }
        protected boolean enableResolveObject(boolean var0) throws java.lang.SecurityException { return
    false; }
        public int read() throws java.io.IOException { return 0; }
        public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
        public boolean readBoolean() throws java.io.IOException { return false; }
        public byte readByte() throws java.io.IOException { return 0; }
        public char readChar() throws java.io.IOException { return 0; }
        public double readDouble() throws java.io.IOException { return 0.0d; }
        public GetField readFields() throws java.io.IOException, java.lang.ClassNotFoundException,
    java.io.NotActiveException { return null; }
        public float readFloat() throws java.io.IOException { return 0.0f; }
        public void readFully(byte[] var0) throws java.io.IOException { }
        public void readFully(byte[] var0, int var1, int var2) throws java.io.IOException { }
        public int readInt() throws java.io.IOException { return 0; }
        public java.lang.String readLine() throws java.io.IOException { return null; }
        public long readLong() throws java.io.IOException { return 0l; }
        protected java.io.ObjectStreamClass readClassDescriptor() throws java.io.IOException,
    java.lang.ClassNotFoundException { return null; }
        protected java.lang.Class resolveProxyClass(java.lang.String[] var0) throws java.io.IOException,
    java.lang.ClassNotFoundException { return null; }
        public final java.lang.Object readObject() throws java.io.OptionalDataException,
    java.lang.ClassNotFoundException, java.io.IOException { return null; }
        protected java.lang.Object readObjectOverride() throws java.io.OptionalDataException,
    java.lang.ClassNotFoundException, java.io.IOException { return null; }
        public short readShort() throws java.io.IOException { return 0; }
        protected void readStreamHeader() throws java.io.IOException, java.io.StreamCorruptedException { }
        public int readUnsignedByte() throws java.io.IOException { return 0; }
        public int readUnsignedShort() throws java.io.IOException { return 0; }
        public java.lang.String readUTF() throws java.io.IOException { return null; }
        public void registerValidation(java.io.ObjectInputValidation var0, int var1) throws
    java.io.NotActiveException, java.io.InvalidObjectException { }
        protected java.lang.Class resolveClass(java.io.ObjectStreamClass var0) throws java.io.IOException,
    java.lang.ClassNotFoundException { return null; }
        protected java.lang.Object resolveObject(java.lang.Object var0) throws java.io.IOException { return
    null; }
        public int skipBytes(int var0) throws java.io.IOException { return 0; }
        public static abstract class GetField {
            public GetField() { }
            public abstract java.io.ObjectStreamClass getObjectStreamClass();
            public abstract boolean defaulted(java.lang.String var0) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract boolean get(java.lang.String var0, boolean var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract char get(java.lang.String var0, char var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract byte get(java.lang.String var0, byte var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract short get(java.lang.String var0, short var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract int get(java.lang.String var0, int var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract long get(java.lang.String var0, long var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract float get(java.lang.String var0, float var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract double get(java.lang.String var0, double var1) throws java.io.IOException,
    java.lang.IllegalArgumentException;
            public abstract java.lang.Object get(java.lang.String var0, java.lang.Object var1) throws
    java.io.IOException, java.lang.IllegalArgumentException;
        }
    }

    package java.io;
    public abstract interface ObjectInputValidation {
        public abstract void validateObject() throws java.io.InvalidObjectException;
    }

    package java.io;
    public abstract interface ObjectOutput extends java.io.DataOutput {
        public abstract void close() throws java.io.IOException;
        public abstract void flush() throws java.io.IOException;
        public abstract void write(byte[] var0) throws java.io.IOException;
        public abstract void write(byte[] var0, int var1, int var2) throws java.io.IOException;
        public abstract void write(int var0) throws java.io.IOException;
        public abstract void writeObject(java.lang.Object var0) throws java.io.IOException;
```

```
}

package java.io;
public class ObjectOutputStream extends java.io.OutputStream implements java.io.ObjectOutput,
java.io.ObjectStreamConstants {
    protected ObjectOutputStream() throws java.io.IOException, java.lang.SecurityException { }
    public ObjectOutputStream(java.io.OutputStream var0) throws java.io.IOException { }
    protected void annotateClass(java.lang.Class var0) throws java.io.IOException { }
    protected void annotateProxyClass(java.lang.Class var0) throws java.io.IOException { }
    public void close() throws java.io.IOException { }
    public void defaultWriteObject() throws java.io.IOException { }
    protected void drain() throws java.io.IOException { }
    protected boolean enableReplaceObject(boolean var0) throws java.lang.SecurityException { return
false; }
    public void flush() throws java.io.IOException { }
    public PutField putFields() throws java.io.IOException { return null; }
    protected java.lang.Object replaceObject(java.lang.Object var0) throws java.io.IOException { return
null; }
    public void reset() throws java.io.IOException { }
    public void useProtocolVersion(int var0) throws java.io.IOException { }
    public void write(byte[] var0) throws java.io.IOException { }
    public void write(byte[] var0, int var1, int var2) throws java.io.IOException { }
    public void write(int var0) throws java.io.IOException { }
    public void writeBoolean(boolean var0) throws java.io.IOException { }
    public void writeByte(int var0) throws java.io.IOException { }
    public void writeBytes(java.lang.String var0) throws java.io.IOException { }
    public void writeChar(int var0) throws java.io.IOException { }
    public void writeChars(java.lang.String var0) throws java.io.IOException { }
    public void writeDouble(double var0) throws java.io.IOException { }
    public void writeFields() throws java.io.IOException { }
    public void writeFloat(float var0) throws java.io.IOException { }
    public void writeInt(int var0) throws java.io.IOException { }
    public void writeLong(long var0) throws java.io.IOException { }
    protected void writeClassDescriptor(java.io.ObjectStreamClass var0) throws java.io.IOException { }
    public final void writeObject(java.lang.Object var0) throws java.io.IOException { }
    protected void writeObjectOverride(java.lang.Object var0) throws java.io.IOException { }
    public void writeShort(int var0) throws java.io.IOException { }
    protected void writeStreamHeader() throws java.io.IOException { }
    public void writeUTF(java.lang.String var0) throws java.io.IOException { }
    public static abstract class PutField {
        public PutField() { }
        public abstract void put(java.lang.String var0, boolean var1);
        public abstract void put(java.lang.String var0, char var1);
        public abstract void put(java.lang.String var0, byte var1);
        public abstract void put(java.lang.String var0, short var1);
        public abstract void put(java.lang.String var0, int var1);
        public abstract void put(java.lang.String var0, long var1);
        public abstract void put(java.lang.String var0, float var1);
        public abstract void put(java.lang.String var0, double var1);
        public abstract void put(java.lang.String var0, java.lang.Object var1);
        public abstract void write(java.io.ObjectOutput var0) throws java.io.IOException;
    }
}

package java.io;
public class ObjectStreamClass implements java.io.Serializable {
    ObjectStreamClass() { }
    public java.lang.Class forClass() { return null; }
    public java.io.ObjectStreamField getField(java.lang.String var0) { return null; }
    public java.io.ObjectStreamField[] getFields() { return null; }
    public java.lang.String getName() { return null; }
    public long getSerialVersionUID() { return 0l; }
    public static java.io.ObjectStreamClass lookup(java.lang.Class var0) { return null; }
    public java.lang.String toString() { return null; }
    public final static java.io.ObjectStreamField[] NO_FIELDS; static { NO_FIELDS = null; }
}

package java.io;
public abstract interface ObjectStreamConstants {
    public final static short STREAM_MAGIC = -21267;
    public final static short STREAM_VERSION = 5;
    public final static byte TC_BASE = 112;
    public final static byte TC_NULL = 112;
    public final static byte TC_REFERENCE = 113;
    public final static byte TC_CLASSDESC = 114;
    public final static byte TC_OBJECT = 115;
    public final static byte TC_STRING = 116;
    public final static byte TC_ARRAY = 117;
    public final static byte TC_CLASS = 118;
    public final static byte TC_BLOCKDATA = 119;
    public final static byte TC_ENDBLOCKDATA = 120;
    public final static byte TC_RESET = 121;
    public final static byte TC_BLOCKDATALONG = 122;
    public final static byte TC_EXCEPTION = 123;
    public final static byte TC_LONGSTRING = 124;
    public final static byte TC_PROXYCLASSDESC = 125;
    public final static byte TC_MAX = 125;
    public final static int baseWireHandle = 8257536;
    public final static int PROTOCOL_VERSION_1 = 1;
    public final static int PROTOCOL_VERSION_2 = 2;
    public final static java.io.SerializablePermission SUBCLASS_IMPLEMENTATION_PERMISSION = null;
    public final static java.io.SerializablePermission SUBSTITUTION_PERMISSION = null;
    public final static byte SC_WRITE_METHOD = 1;
    public final static byte SC_SERIALIZABLE = 2;
    public final static byte SC_EXTERNALIZABLE = 4;
    public final static byte SC_BLOCK_DATA = 8;
}

package java.io;
public abstract class ObjectStreamException extends java.io.IOException {
    protected ObjectStreamException() { }
    protected ObjectStreamException(java.lang.String var0) { }
}

package java.io;
public class ObjectStreamField implements java.lang.Comparable {
    public ObjectStreamField(java.lang.String var0, java.lang.Class var1) { }
    ObjectStreamField(java.lang.String var0, java.lang.String var1) { }
    public int compareTo(java.lang.Object var0) { return 0; }
    public java.lang.String getName() { return null; }
    public int getOffset() { return 0; }
    public java.lang.Class getType() { return null; }
    public char getTypeCode() { return 0; }
    public java.lang.String getTypeString() { return null; }
    public boolean isPrimitive() { return false; }
    protected void setOffset(int var0) { }
    public java.lang.String toString() { return null; }
}

package java.io;
public class OptionalDataException extends java.io.ObjectStreamException {
    OptionalDataException() { }
    OptionalDataException(java.lang.String var0) { }
    public boolean eof;
    public int length;
}

package java.io;
public abstract class OutputStream {
    public OutputStream() { }
    public void close() throws java.io.IOException { }
    public void flush() throws java.io.IOException { }
    public void write(byte[] var0) throws java.io.IOException { }
    public void write(byte[] var0, int var1, int var2) throws java.io.IOException { }
    public abstract void write(int var0) throws java.io.IOException;
}

package java.io;
```

```
public class OutputStreamWriter extends java.io.Writer {
    public OutputStreamWriter(java.io.OutputStream var0) { }
    public OutputStreamWriter(java.io.OutputStream var0, java.lang.String var1) throws
java.io.UnsupportedEncodingException { }
    public void close() throws java.io.IOException { }
    public void flush() throws java.io.IOException { }
    public java.lang.String getEncoding() { return null; }
    public void write(char[] var0, int var1, int var2) throws java.io.IOException { }
    public void write(int var0) throws java.io.IOException { }
    public void write(java.lang.String var0, int var1, int var2) throws java.io.IOException { }
}

package java.io;
public class PrintStream extends java.io.FilterOutputStream {
    public PrintStream(java.io.OutputStream var0) { super(null); }
    public PrintStream(java.io.OutputStream var0, boolean var1) { super(null); }
    public boolean checkError() { return false; }
    public void close() { }
    public void flush() { }
    public void print(char[] var0) { }
    public void print(char var0) { }
    public void print(double var0) { }
    public void print(float var0) { }
    public void print(int var0) { }
    public void print(long var0) { }
    public void print(java.lang.Object var0) { }
    public void print(java.lang.String var0) { }
    public void print(boolean var0) { }
    public void println() { }
    public void println(char[] var0) { }
    public void println(char var0) { }
    public void println(double var0) { }
    public void println(float var0) { }
    public void println(int var0) { }
    public void println(long var0) { }
    public void println(java.lang.Object var0) { }
    public void println(java.lang.String var0) { }
    public void println(boolean var0) { }
    protected void setError() { }
    public void write(byte[] var0, int var1, int var2) { }
    public void write(int var0) { }
}

package java.io;
public class PrintWriter extends java.io.Writer {
    public PrintWriter(java.io.OutputStream var0) { }
    public PrintWriter(java.io.OutputStream var0, boolean var1) { }
    public PrintWriter(java.io.Writer var0) { }
    public PrintWriter(java.io.Writer var0, boolean var1) { }
    public boolean checkError() { return false; }
    public void close() { }
    public void flush() { }
    public void print(char[] var0) { }
    public void print(char var0) { }
    public void print(double var0) { }
    public void print(float var0) { }
    public void print(int var0) { }
    public void print(long var0) { }
    public void print(java.lang.Object var0) { }
    public void print(java.lang.String var0) { }
    public void print(boolean var0) { }
    public void println() { }
    public void println(char[] var0) { }
    public void println(char var0) { }
    public void println(double var0) { }
    public void println(float var0) { }
    public void println(int var0) { }
    public void println(long var0) { }
    public void println(java.lang.Object var0) { }
    public void println(java.lang.String var0) { }
    public void println(boolean var0) { }
    protected void setError() { }
    public void write(char[] var0) { }
    public void write(char[] var0, int var1, int var2) { }
    public void write(int var0) { }
    public void write(java.lang.String var0) { }
    public void write(java.lang.String var0, int var1, int var2) { }
    protected java.io.Writer out;
}

package java.io;
public class PushbackInputStream extends java.io.FilterInputStream {
    public PushbackInputStream(java.io.InputStream var0) { super(null); }
    public PushbackInputStream(java.io.InputStream var0, int var1) { super(null); }
    public int available() throws java.io.IOException { return 0; }
    public void close() throws java.io.IOException { }
    public boolean markSupported() { return false; }
    public int read() throws java.io.IOException { return 0; }
    public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
    public long skip(long var0) throws java.io.IOException { return 0l; }
    public void unread(byte[] var0) throws java.io.IOException { }
    public void unread(byte[] var0, int var1, int var2) throws java.io.IOException { }
    public void unread(int var0) throws java.io.IOException { }
    protected byte[] buf;
    protected int pos;
}

package java.io;
public class RandomAccessFile implements java.io.DataInput, java.io.DataOutput {
    public RandomAccessFile(java.io.File var0, java.lang.String var1) throws
java.io.FileNotFoundException { }
    public RandomAccessFile(java.lang.String var0, java.lang.String var1) throws
java.io.FileNotFoundException { }
    public void close() throws java.io.IOException { }
    public final java.io.FileDescriptor getFD() throws java.io.IOException { return null; }
    public long getFilePointer() throws java.io.IOException { return 0l; }
    public long length() throws java.io.IOException { return 0l; }
    public int read() throws java.io.IOException { return 0; }
    public int read(byte[] var0) throws java.io.IOException { return 0; }
    public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
    public final boolean readBoolean() throws java.io.IOException { return false; }
    public final byte readByte() throws java.io.IOException { return 0; }
    public final char readChar() throws java.io.IOException { return 0; }
    public final double readDouble() throws java.io.IOException { return 0.0d; }
    public final float readFloat() throws java.io.IOException { return 0.0f; }
    public final void readFully(byte[] var0) throws java.io.IOException { }
    public final void readFully(byte[] var0, int var1, int var2) throws java.io.IOException { }
    public final int readInt() throws java.io.IOException { return 0; }
    public final java.lang.String readLine() throws java.io.IOException { return null; }
    public final long readLong() throws java.io.IOException { return 0l; }
    public final short readShort() throws java.io.IOException { return 0; }
    public final int readUnsignedByte() throws java.io.IOException { return 0; }
    public final int readUnsignedShort() throws java.io.IOException { return 0; }
    public final java.lang.String readUTF() throws java.io.IOException { return null; }
    public void seek(long var0) throws java.io.IOException { }
    public void setLength(long var0) throws java.io.IOException { }
    public int skipBytes(int var0) throws java.io.IOException { return 0; }
    public void write(byte[] var0) throws java.io.IOException { }
    public void write(byte[] var0, int var1, int var2) throws java.io.IOException { }
    public void write(int var0) throws java.io.IOException { }
    public final void writeBoolean(boolean var0) throws java.io.IOException { }
    public final void writeByte(int var0) throws java.io.IOException { }
    public final void writeBytes(java.lang.String var0) throws java.io.IOException { }
    public final void writeChar(int var0) throws java.io.IOException { }
    public final void writeChars(java.lang.String var0) throws java.io.IOException { }
    public final void writeDouble(double var0) throws java.io.IOException { }
    public final void writeFloat(float var0) throws java.io.IOException { }
    public final void writeInt(int var0) throws java.io.IOException { }
    public final void writeLong(long var0) throws java.io.IOException { }
    public final void writeShort(int var0) throws java.io.IOException { }
    public final void writeUTF(java.lang.String var0) throws java.io.IOException { }
}

package java.io;
public abstract class Reader {
    protected Reader() { }
    protected Reader(java.lang.Object var0) { }
    public abstract void close() throws java.io.IOException;
    public void mark(int var0) throws java.io.IOException { }
    public boolean markSupported() { return false; }
    public int read() throws java.io.IOException { return 0; }
    public int read(char[] var0) throws java.io.IOException { return 0; }
    public abstract int read(char[] var0, int var1, int var2) throws java.io.IOException;
    public boolean ready() throws java.io.IOException { return false; }
    public void reset() throws java.io.IOException { }
    public long skip(long var0) throws java.io.IOException { return 0l; }
    protected java.lang.Object lock;
}

package java.io;
public abstract interface Serializable {
}

package java.io;
public final class SerializablePermission extends java.security.BasicPermission {
    public SerializablePermission(java.lang.String var0) { super(null, null); }
    public SerializablePermission(java.lang.String var0, java.lang.String var1) { super(null, null); }
}

package java.io;
public class StreamCorruptedException extends java.io.ObjectStreamException {
    public StreamCorruptedException() { }
    public StreamCorruptedException(java.lang.String var0) { }
}

package java.io;
public class StringReader extends java.io.Reader {
    public StringReader(java.lang.String var0) { }
    public void close() { }
    public void mark(int var0) throws java.io.IOException { }
    public boolean markSupported() { return false; }
    public int read() throws java.io.IOException { return 0; }
    public int read(char[] var0, int var1, int var2) throws java.io.IOException { return 0; }
    public boolean ready() throws java.io.IOException { return false; }
    public void reset() throws java.io.IOException { }
    public long skip(long var0) throws java.io.IOException { return 0l; }
}

package java.io;
public class StringWriter extends java.io.Writer {
    public StringWriter() { }
    public StringWriter(int var0) { }
    public void close() throws java.io.IOException { }
    public void flush() { }
    public java.lang.StringBuffer getBuffer() { return null; }
    public java.lang.String toString() { return null; }
    public void write(char[] var0, int var1, int var2) { }
    public void write(int var0) { }
    public void write(java.lang.String var0) { }
    public void write(java.lang.String var0, int var1, int var2) { }
}

package java.io;
public class SyncFailedException extends java.io.IOException {
    public SyncFailedException(java.lang.String var0) { }
}
```

```
package java.io;
public class UnsupportedEncodingException extends java.io.IOException {
    public UnsupportedEncodingException() { }
    public UnsupportedEncodingException(java.lang.String var0) { }
}

package java.io;
public class UTFDataFormatException extends java.io.IOException {
    public UTFDataFormatException() { }
    public UTFDataFormatException(java.lang.String var0) { }
}

package java.io;
public class WriteAbortedException extends java.io.ObjectStreamException {
    public WriteAbortedException(java.lang.String var0, java.lang.Exception var1) { }
    public java.lang.String getMessage() { return null; }
    public java.lang.Exception detail;
}

package java.io;
public abstract class Writer {
    protected Writer() { }
    protected Writer(java.lang.Object var0) { }
    public abstract void close() throws java.io.IOException;
    public abstract void flush() throws java.io.IOException;
    public void write(char[] var0) throws java.io.IOException { }
    public abstract void write(char[] var0, int var1, int var2) throws java.io.IOException;
    public void write(int var0) throws java.io.IOException { }
    public void write(java.lang.String var0) throws java.io.IOException { }
    public void write(java.lang.String var0, int var1, int var2) throws java.io.IOException { }
    protected java.lang.Object lock;
}

package java.lang;
public class AbstractMethodError extends java.lang.IncompatibleClassChangeError {
    public AbstractMethodError() { }
    public AbstractMethodError(java.lang.String var0) { }
}

package java.lang;
public class ArithmeticException extends java.lang.RuntimeException {
    public ArithmeticException() { }
    public ArithmeticException(java.lang.String var0) { }
}

package java.lang;
public class ArrayIndexOutOfBoundsException extends java.lang.IndexOutOfBoundsException {
    public ArrayIndexOutOfBoundsException() { }
    public ArrayIndexOutOfBoundsException(int var0) { }
    public ArrayIndexOutOfBoundsException(java.lang.String var0) { }
}

package java.lang;
public class ArrayStoreException extends java.lang.RuntimeException {
    public ArrayStoreException() { }
    public ArrayStoreException(java.lang.String var0) { }
}

package java.lang;
public final class Boolean implements java.io.Serializable {
    public Boolean(java.lang.String var0) { }
    public Boolean(boolean var0) { }
    public boolean booleanValue() { return false; }
    public boolean equals(java.lang.Object var0) { return false; }
    public static boolean getBoolean(java.lang.String var0) { return false; }
    public int hashCode() { return 0; }
    public java.lang.String toString() { return null; }
    public static java.lang.Boolean valueOf(java.lang.String var0) { return null; }
    public final static java.lang.Class TYPE; static { TYPE = null; }
    public final static java.lang.Boolean TRUE; static { TRUE = null; }
    public final static java.lang.Boolean FALSE; static { FALSE = null; }
}

package java.lang;
public final class Byte extends java.lang.Number implements java.lang.Comparable {
    public Byte(byte var0) { }
    public Byte(java.lang.String var0) throws java.lang.NumberFormatException { }
    public byte byteValue() { return 0; }
    public int compareTo(java.lang.Byte var0) { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public double doubleValue() { return 0.0d; }
    public boolean equals(java.lang.Object var0) { return false; }
    public float floatValue() { return 0.0f; }
    public int hashCode() { return 0; }
    public int intValue() { return 0; }
    public long longValue() { return 0l; }
    public static byte parseByte(java.lang.String var0) throws java.lang.NumberFormatException { return
0; }
    public static byte parseByte(java.lang.String var0, int var1) throws java.lang.NumberFormatException
{ return 0; }
    public short shortValue() { return 0; }
    public java.lang.String toString() { return null; }
    public static java.lang.Byte valueOf(java.lang.String var0, int var1) throws
java.lang.NumberFormatException { return null; }
    public final static byte MAX_VALUE = 127;
    public final static byte MIN_VALUE = -128;
    public final static java.lang.Class TYPE; static { TYPE = null; }
}

package java.lang;
public final class Character implements java.io.Serializable, java.lang.Comparable {
    public Character(char var0) { }
    public char charValue() { return 0; }
    public int compareTo(java.lang.Character var0) { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public static int digit(char var0, int var1) { return 0; }
    public boolean equals(java.lang.Object var0) { return false; }
    public static char forDigit(int var0, int var1) { return 0; }
    public static int getType(char var0) { return 0; }
    public int hashCode() { return 0; }
    public static boolean isDigit(char var0) { return false; }
    public static boolean isLetter(char var0) { return false; }
    public static boolean isLowerCase(char var0) { return false; }
    public static boolean isSpaceChar(char var0) { return false; }
    public static boolean isUpperCase(char var0) { return false; }
    public static boolean isWhitespace(char var0) { return false; }
    public static char toLowerCase(char var0) { return 0; }
    public java.lang.String toString() { return null; }
    public static char toUpperCase(char var0) { return 0; }
    public final static char MIN_VALUE = 0;
    public final static char MAX_VALUE = 65535;
    public final static int MIN_RADIX = 2;
    public final static int MAX_RADIX = 36;
    public final static java.lang.Class TYPE; static { TYPE = null; }
    public final static byte UNASSIGNED = 0;
    public final static byte UPPERCASE_LETTER = 1;
    public final static byte LOWERCASE_LETTER = 2;
    public final static byte TITLECASE_LETTER = 3;
    public final static byte MODIFIER_LETTER = 4;
    public final static byte OTHER_LETTER = 5;
    public final static byte NON_SPACING_MARK = 6;
    public final static byte ENCLOSING_MARK = 7;
    public final static byte COMBINING_SPACING_MARK = 8;
    public final static byte DECIMAL_DIGIT_NUMBER = 9;
    public final static byte LETTER_NUMBER = 10;
    public final static byte OTHER_NUMBER = 11;
    public final static byte SPACE_SEPARATOR = 12;
    public final static byte LINE_SEPARATOR = 13;
    public final static byte PARAGRAPH_SEPARATOR = 14;
    public final static byte CONTROL = 15;
    public final static byte FORMAT = 16;
    public final static byte PRIVATE_USE = 18;
    public final static byte SURROGATE = 19;
    public final static byte DASH_PUNCTUATION = 20;
    public final static byte START_PUNCTUATION = 21;
    public final static byte END_PUNCTUATION = 22;
    public final static byte CONNECTOR_PUNCTUATION = 23;
    public final static byte OTHER_PUNCTUATION = 24;
    public final static byte MATH_SYMBOL = 25;
    public final static byte CURRENCY_SYMBOL = 26;
    public final static byte MODIFIER_SYMBOL = 27;
    public final static byte OTHER_SYMBOL = 28;
}

package java.lang;
public final class Class implements java.io.Serializable {
    private Class() { }
    public static java.lang.Class forName(java.lang.String var0) throws java.lang.ClassNotFoundException
{ return null; }
    public static java.lang.Class forName(java.lang.String var0, boolean var1, java.lang.ClassLoader
var2) throws java.lang.ClassNotFoundException { return null; }
    public java.lang.Class[] getClasses() { return null; }
    public java.lang.ClassLoader getClassLoader() { return null; }
    public java.lang.Class getComponentType() { return null; }
    public java.lang.reflect.Constructor getConstructor(java.lang.Class[] var0) throws
java.lang.NoSuchMethodException, java.lang.SecurityException { return null; }
    public java.lang.reflect.Constructor[] getConstructors() throws java.lang.SecurityException { return
null; }
    public java.lang.Class[] getDeclaredClasses() throws java.lang.SecurityException { return null; }
    public java.lang.reflect.Constructor getDeclaredConstructor(java.lang.Class[] var0) throws
java.lang.NoSuchMethodException, java.lang.SecurityException { return null; }
    public java.lang.reflect.Constructor[] getDeclaredConstructors() throws java.lang.SecurityException
{ return null; }
    public java.lang.reflect.Field getDeclaredField(java.lang.String var0) throws
java.lang.NoSuchFieldException, java.lang.SecurityException { return null; }
    public java.lang.reflect.Field[] getDeclaredFields() throws java.lang.SecurityException { return
null; }
    public java.lang.reflect.Method getDeclaredMethod(java.lang.String var0, java.lang.Class[] var1)
throws java.lang.NoSuchMethodException, java.lang.SecurityException { return null; }
    public java.lang.reflect.Method[] getDeclaredMethods() throws java.lang.SecurityException { return
null; }
    public java.lang.Class getDeclaringClass() { return null; }
    public java.lang.reflect.Field getField(java.lang.String var0) throws
java.lang.NoSuchFieldException, java.lang.SecurityException { return null; }
    public java.lang.reflect.Field[] getFields() throws java.lang.SecurityException { return null; }
    public java.lang.Class[] getInterfaces() { return null; }
    public java.lang.reflect.Method getMethod(java.lang.String var0, java.lang.Class[] var1) throws
java.lang.NoSuchMethodException, java.lang.SecurityException { return null; }
    public java.lang.reflect.Method[] getMethods() throws java.lang.SecurityException { return null; }
    public int getModifiers() { return 0; }
    public java.lang.String getName() { return null; }
    public java.security.ProtectionDomain getProtectionDomain() { return null; }
    public java.net.URL getResource(java.lang.String var0) { return null; }
    public java.io.InputStream getResourceAsStream(java.lang.String var0) { return null; }
    public java.lang.Class getSuperclass() { return null; }
    public boolean isArray() { return false; }
    public boolean isAssignableFrom(java.lang.Class var0) { return false; }
    public boolean isInstance(java.lang.Object var0) { return false; }
    public boolean isInterface() { return false; }
    public boolean isPrimitive() { return false; }
    public java.lang.Object newInstance() throws java.lang.IllegalAccessException,
java.lang.InstantiationException { return null; }
    public java.lang.String toString() { return null; }
}

package java.lang;
public class ClassCastException extends java.lang.RuntimeException {
```

All Page Within This Box

```
    public ClassCastException() { }
    public ClassCastException(java.lang.String var0) { }
}

package java.lang;
public class ClassCircularityError extends java.lang.LinkageError {
    public ClassCircularityError() { }
    public ClassCircularityError(java.lang.String var0) { }
}

package java.lang;
public class ClassFormatError extends java.lang.LinkageError {
    public ClassFormatError() { }
    public ClassFormatError(java.lang.String var0) { }
}

package java.lang;
public abstract class ClassLoader {
    protected ClassLoader() { }
    protected ClassLoader(java.lang.ClassLoader var0) { }
    protected final java.lang.Class defineClass(java.lang.String var0, byte[] var1, int var2, int var3)
throws java.lang.ClassFormatError { return null; }
    protected final java.lang.Class defineClass(java.lang.String var0, byte[] var1, int var2, int var3,
java.security.ProtectionDomain var4) throws java.lang.ClassFormatError { return null; }
    protected java.lang.Class findClass(java.lang.String var0) throws java.lang.ClassNotFoundException {
return null; }
    protected final java.lang.Class findLoadedClass(java.lang.String var0) { return null; }
    protected final java.lang.Class findSystemClass(java.lang.String var0) throws
java.lang.ClassNotFoundException { return null; }
    public final java.lang.ClassLoader getParent() { return null; }
    public java.net.URL getResource(java.lang.String var0) { return null; }
    public final java.util.Enumeration getResources(java.lang.String var0) throws java.io.IOException {
return null; }
    public java.io.InputStream getResourceAsStream(java.lang.String var0) { return null; }
    public static java.lang.ClassLoader getSystemClassLoader() { return null; }
    public static java.net.URL getSystemResource(java.lang.String var0) { return null; }
    public static java.util.Enumeration getSystemResources(java.lang.String var0) throws
java.io.IOException { return null; }
    public static java.io.InputStream getSystemResourceAsStream(java.lang.String var0) { return null; }
    public java.lang.Class loadClass(java.lang.String var0) throws java.lang.ClassNotFoundException {
return null; }
    protected java.lang.Class loadClass(java.lang.String var0, boolean var1) throws
java.lang.ClassNotFoundException { return null; }
    protected final void resolveClass(java.lang.Class var0) { }
    protected java.net.URL findResource(java.lang.String var0) { return null; }
    protected java.util.Enumeration findResources(java.lang.String var0) throws java.io.IOException {
return null; }
    protected java.lang.String findLibrary(java.lang.String var0) { return null; }
}

package java.lang;
public class ClassNotFoundException extends java.lang.Exception {
    public ClassNotFoundException() { }
    public ClassNotFoundException(java.lang.String var0) { }
    public ClassNotFoundException(java.lang.String var0, java.lang.Throwable var1) { }
    public void printStackTrace() { }
    public void printStackTrace(java.io.PrintStream var0) { }
}

package java.lang;
public abstract interface Cloneable {
}

package java.lang;
public class CloneNotSupportedException extends java.lang.Exception {
    public CloneNotSupportedException() { }
    public CloneNotSupportedException(java.lang.String var0) { }
}
```

```
package java.lang;
public abstract interface Comparable {
    public abstract int compareTo(java.lang.Object var0);
}

package java.lang;
public final class Double extends java.lang.Number implements java.lang.Comparable {
    public Double(double var0) { }
    public Double(java.lang.String var0) throws java.lang.NumberFormatException { }
    public int compareTo(java.lang.Double var0) { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public byte byteValue() { return 0; }
    public static long doubleToLongBits(double var0) { return 0l; }
    public double doubleValue() { return 0.0d; }
    public boolean equals(java.lang.Object var0) { return false; }
    public float floatValue() { return 0.0f; }
    public int hashCode() { return 0; }
    public int intValue() { return 0; }
    public static boolean isInfinite(double var0) { return false; }
    public static boolean isNaN(double var0) { return false; }
    public static double longBitsToDouble(long var0) { return 0.0d; }
    public long longValue() { return 0l; }
    public static double parseDouble(java.lang.String var0) throws java.lang.NumberFormatException {
return 0.0d; }
    public short shortValue() { return 0; }
    public java.lang.String toString() { return null; }
    public static java.lang.String toString(double var0) { return null; }
    public static java.lang.Double valueOf(java.lang.String var0) throws java.lang.NumberFormatException
{ return null; }
    public final static double MAX_VALUE = 1.7976931348623157E308d;
    public final static double MIN_VALUE = 4.9E-324d;
    public final static double NaN = 0.0d / 0.0d;
    public final static double POSITIVE_INFINITY = 1.0d / 0.0d;
    public final static double NEGATIVE_INFINITY = -1.0d / 0.0d;
    public final static java.lang.Class TYPE; static { TYPE = null; }
}

package java.lang;
public class Error extends java.lang.Throwable {
    public Error() { }
    public Error(java.lang.String var0) { }
}

package java.lang;
public class Exception extends java.lang.Throwable {
    public Exception() { }
    public Exception(java.lang.String var0) { }
}

package java.lang;
public class ExceptionInInitializerError extends java.lang.LinkageError {
    public ExceptionInInitializerError() { }
    public ExceptionInInitializerError(java.lang.String var0) { }
    public ExceptionInInitializerError(java.lang.Throwable var0) { }
    public java.lang.Throwable getException() { return null; }
    public void printStackTrace() { }
    public void printStackTrace(java.io.PrintStream var0) { }
}

package java.lang;
public final class Float extends java.lang.Number implements java.lang.Comparable {
    public Float(float var0) { }
    public Float(double var0) { }
    public Float(java.lang.String var0) throws java.lang.NumberFormatException { }
    public int compareTo(java.lang.Float var0) { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public byte byteValue() { return 0; }
    public double doubleValue() { return 0.0d; }
    public boolean equals(java.lang.Object var0) { return false; }
    public static int floatToIntBits(float var0) { return 0; }
    public float floatValue() { return 0.0f; }
    public int hashCode() { return 0; }
    public static float intBitsToFloat(int var0) { return 0.0f; }
    public int intValue() { return 0; }
    public static boolean isInfinite(float var0) { return false; }
    public static boolean isNaN(float var0) { return false; }
    public long longValue() { return 0l; }
    public static float parseFloat(java.lang.String var0) throws java.lang.NumberFormatException {
return 0.0f; }
    public short shortValue() { return 0; }
    public java.lang.String toString() { return null; }
    public static java.lang.String toString(float var0) { return null; }
    public static java.lang.Float valueOf(java.lang.String var0) throws java.lang.NumberFormatException
{ return null; }
    public final static float MAX_VALUE = 3.4028235E38f;
    public final static float MIN_VALUE = 1.4E-45f;
    public final static float NaN = 0.0f / 0.0f;
    public final static float POSITIVE_INFINITY = 1.0f / 0.0f;
    public final static float NEGATIVE_INFINITY = -1.0f / 0.0f;
    public final static java.lang.Class TYPE; static { TYPE = null; }
}

package java.lang;
public class IllegalAccessError extends java.lang.IncompatibleClassChangeError {
    public IllegalAccessError() { }
    public IllegalAccessError(java.lang.String var0) { }
}

package java.lang;
public class IllegalAccessException extends java.lang.Exception {
    public IllegalAccessException() { }
    public IllegalAccessException(java.lang.String var0) { }
}

package java.lang;
public class IllegalArgumentException extends java.lang.RuntimeException {
    public IllegalArgumentException() { }
    public IllegalArgumentException(java.lang.String var0) { }
}

package java.lang;
public class IllegalMonitorStateException extends java.lang.RuntimeException {
    public IllegalMonitorStateException() { }
    public IllegalMonitorStateException(java.lang.String var0) { }
}

package java.lang;
public class IllegalStateException extends java.lang.RuntimeException {
    public IllegalStateException() { }
    public IllegalStateException(java.lang.String var0) { }
}

package java.lang;
public class IllegalThreadStateException extends java.lang.IllegalArgumentException {
    public IllegalThreadStateException() { }
    public IllegalThreadStateException(java.lang.String var0) { }
}

package java.lang;
public class IncompatibleClassChangeError extends java.lang.LinkageError {
    public IncompatibleClassChangeError() { }
    public IncompatibleClassChangeError(java.lang.String var0) { }
}

package java.lang;
public class IndexOutOfBoundsException extends java.lang.RuntimeException {
    public IndexOutOfBoundsException() { }
    public IndexOutOfBoundsException(java.lang.String var0) { }
```

```
}

package java.lang;
public class InstantiationError extends java.lang.IncompatibleClassChangeError {
    public InstantiationError() { }
    public InstantiationError(java.lang.String var0) { }
}

package java.lang;
public class InstantiationException extends java.lang.Exception {
    public InstantiationException() { }
    public InstantiationException(java.lang.String var0) { }
}

package java.lang;
public final class Integer extends java.lang.Number implements java.lang.Comparable {
    public Integer(int var0) { }
    public Integer(java.lang.String var0) throws java.lang.NumberFormatException { }
    public byte byteValue() { return 0; }
    public int compareTo(java.lang.Integer var0) { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public static java.lang.Integer decode(java.lang.String var0) throws java.lang.NumberFormatException
{ return null; }
    public double doubleValue() { return 0.0d; }
    public boolean equals(java.lang.Object var0) { return false; }
    public float floatValue() { return 0.0f; }
    public static java.lang.Integer getInteger(java.lang.String var0) { return null; }
    public static java.lang.Integer getInteger(java.lang.String var0, int var1) { return null; }
    public int hashCode() { return 0; }
    public int intValue() { return 0; }
    public long longValue() { return 0l; }
    public static int parseInt(java.lang.String var0) throws java.lang.NumberFormatException { return 0;
}
    public static int parseInt(java.lang.String var0, int var1) throws java.lang.NumberFormatException {
return 0; }
    public short shortValue() { return 0; }
    public static java.lang.String toBinaryString(int var0) { return null; }
    public static java.lang.String toHexString(int var0) { return null; }
    public java.lang.String toString() { return null; }
    public static java.lang.String toString(int var0) { return null; }
    public static java.lang.Integer valueOf(java.lang.String var0, int var1) throws
java.lang.NumberFormatException { return null; }
    public final static int MAX_VALUE = 2147483647;
    public final static int MIN_VALUE = -2147483648;
    public final static java.lang.Class TYPE; static { TYPE = null; }
}

package java.lang;
public class InternalError extends java.lang.VirtualMachineError {
    public InternalError() { }
    public InternalError(java.lang.String var0) { }
}

package java.lang;
public class InterruptedException extends java.lang.Exception {
    public InterruptedException() { }
    public InterruptedException(java.lang.String var0) { }
}

package java.lang;
public class LinkageError extends java.lang.Error {
    public LinkageError() { }
    public LinkageError(java.lang.String var0) { }
}

package java.lang;
public final class Long extends java.lang.Number implements java.lang.Comparable {
    public Long(long var0) { }
    public Long(java.lang.String var0) throws java.lang.NumberFormatException { }
    public byte byteValue() { return 0; }
    public int compareTo(java.lang.Long var0) { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public double doubleValue() { return 0.0d; }
    public boolean equals(java.lang.Object var0) { return false; }
    public float floatValue() { return 0.0f; }
    public int hashCode() { return 0; }
    public int intValue() { return 0; }
    public long longValue() { return 0l; }
    public static long parseLong(java.lang.String var0) throws java.lang.NumberFormatException { return
0l; }
    public static long parseLong(java.lang.String var0, int var1) throws java.lang.NumberFormatException
{ return 0l; }
    public short shortValue() { return 0; }
    public static java.lang.String toBinaryString(long var0) { return null; }
    public static java.lang.String toHexString(long var0) { return null; }
    public java.lang.String toString() { return null; }
    public static java.lang.String toString(long var0) { return null; }
    public static java.lang.Long valueOf(java.lang.String var0, int var1) throws
java.lang.NumberFormatException { return null; }
    public final static long MAX_VALUE = 9223372036854775807l;
    public final static long MIN_VALUE = -9223372036854775808l;
    public final static java.lang.Class TYPE; static { TYPE = null; }
}

package java.lang;
public final class Math {
    private Math() { }
    public static double abs(double var0) { return 0.0d; }
    public static float abs(float var0) { return 0.0f; }
    public static int abs(int var0) { return 0; }
    public static long abs(long var0) { return 0l; }
    public static double acos(double var0) { return 0.0d; }
    public static double asin(double var0) { return 0.0d; }
    public static double atan(double var0) { return 0.0d; }
    public static double atan2(double var0, double var1) { return 0.0d; }
    public static double ceil(double var0) { return 0.0d; }
    public static double cos(double var0) { return 0.0d; }
    public static double exp(double var0) { return 0.0d; }
    public static double floor(double var0) { return 0.0d; }
    public static double IEEEremainder(double var0, double var1) { return 0.0d; }
    public static double log(double var0) { return 0.0d; }
    public static double max(double var0, double var1) { return 0.0d; }
    public static float max(float var0, float var1) { return 0.0f; }
    public static int max(int var0, int var1) { return 0; }
    public static long max(long var0, long var1) { return 0l; }
    public static double min(double var0, double var1) { return 0.0d; }
    public static float min(float var0, float var1) { return 0.0f; }
    public static int min(int var0, int var1) { return 0; }
    public static long min(long var0, long var1) { return 0l; }
    public static double pow(double var0, double var1) { return 0.0d; }
    public static double rint(double var0) { return 0.0d; }
    public static long round(double var0) { return 0l; }
    public static int round(float var0) { return 0; }
    public static double sin(double var0) { return 0.0d; }
    public static double sqrt(double var0) { return 0.0d; }
    public static double tan(double var0) { return 0.0d; }
    public static double random() { return 0.0d; }
    public static double toRadians(double var0) { return 0.0d; }
    public static double toDegrees(double var0) { return 0.0d; }
    public final static double E; static { E = 0.0d; }
    public final static double PI; static { PI = 0.0d; }
}

package java.lang;
public class NegativeArraySizeException extends java.lang.RuntimeException {
    public NegativeArraySizeException() { }
    public NegativeArraySizeException(java.lang.String var0) { }
}
```

```
package java.lang;
public class NoClassDefFoundError extends java.lang.LinkageError {
    public NoClassDefFoundError() { }
    public NoClassDefFoundError(java.lang.String var0) { }
}

package java.lang;
public class NoSuchFieldError extends java.lang.IncompatibleClassChangeError {
    public NoSuchFieldError() { }
    public NoSuchFieldError(java.lang.String var0) { }
}

package java.lang;
public class NoSuchFieldException extends java.lang.Exception {
    public NoSuchFieldException() { }
    public NoSuchFieldException(java.lang.String var0) { }
}

package java.lang;
public class NoSuchMethodError extends java.lang.IncompatibleClassChangeError {
    public NoSuchMethodError() { }
    public NoSuchMethodError(java.lang.String var0) { }
}

package java.lang;
public class NoSuchMethodException extends java.lang.Exception {
    public NoSuchMethodException() { }
    public NoSuchMethodException(java.lang.String var0) { }
}

package java.lang;
public class NullPointerException extends java.lang.RuntimeException {
    public NullPointerException() { }
    public NullPointerException(java.lang.String var0) { }
}

package java.lang;
public abstract class Number implements java.io.Serializable {
    public Number() { }
    public byte byteValue() { return 0; }
    public abstract double doubleValue();
    public abstract float floatValue();
    public abstract int intValue();
    public abstract long longValue();
    public short shortValue() { return 0; }
}

package java.lang;
public class NumberFormatException extends java.lang.IllegalArgumentException {
    public NumberFormatException() { }
    public NumberFormatException(java.lang.String var0) { }
}

package java.lang;
public class Object {
    public Object() { }
    protected java.lang.Object clone() throws java.lang.CloneNotSupportedException { return null; }
    public boolean equals(java.lang.Object var0) { return false; }
    protected void finalize() throws java.lang.Throwable { }
    public final java.lang.Class getClass() { return null; }
    public int hashCode() { return 0; }
    public final void notify() { }
    public final void notifyAll() { }
    public java.lang.String toString() { return null; }
    public final void wait() throws java.lang.InterruptedException { }
    public final void wait(long var0) throws java.lang.InterruptedException { }
    public final void wait(long var0, int var1) throws java.lang.InterruptedException { }
}
```

```
package java.lang;
public class OutOfMemoryError extends java.lang.VirtualMachineError {
    public OutOfMemoryError() { }
    public OutOfMemoryError(java.lang.String var0) { }
}

package java.lang;
public abstract class Process {
    public Process() { }
    public abstract void destroy();
    public abstract int exitValue();
    public abstract java.io.InputStream getErrorStream();
    public abstract java.io.InputStream getInputStream();
    public abstract java.io.OutputStream getOutputStream();
    public abstract int waitFor() throws java.lang.InterruptedException;
}

package java.lang.ref;
public class PhantomReference extends java.lang.ref.Reference {
    public java.lang.Object get() { return null; }
    public PhantomReference(java.lang.Object var0, java.lang.ref.ReferenceQueue var1) { }
}

package java.lang.ref;
public abstract class Reference {
    public void clear() { }
    public boolean enqueue() { return false; }
    public java.lang.Object get() { return null; }
    public boolean isEnqueued() { return false; }
    Reference() { }
}

package java.lang.ref;
public class ReferenceQueue {
    public java.lang.ref.Reference poll() { return null; }
    public java.lang.ref.Reference remove() throws java.lang.InterruptedException { return null; }
    public java.lang.ref.Reference remove(long var0) throws java.lang.IllegalArgumentException,
java.lang.InterruptedException { return null; }
    public ReferenceQueue() { }
}

package java.lang.ref;
public class SoftReference extends java.lang.ref.Reference {
    public SoftReference(java.lang.Object var0, java.lang.ref.ReferenceQueue var1) { }
    public SoftReference(java.lang.Object var0) { }
    public java.lang.Object get() { return null; }
}

package java.lang.ref;
public class WeakReference extends java.lang.ref.Reference {
    public WeakReference(java.lang.Object var0, java.lang.ref.ReferenceQueue var1) { }
    public WeakReference(java.lang.Object var0) { }
}

package java.lang.reflect;
public class AccessibleObject {
    protected AccessibleObject() { }
    public boolean isAccessible() { return false; }
    public static void setAccessible(java.lang.reflect.AccessibleObject[] var0, boolean var1) throws
java.lang.SecurityException { }
    public void setAccessible(boolean var0) throws java.lang.SecurityException { }
}

package java.lang.reflect;
public final class Array {
    private Array() { }
    public static java.lang.Object get(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return null; }
```

```
    public static boolean getBoolean(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return false; }
    public static byte getByte(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return 0; }
    public static char getChar(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return 0; }
    public static double getDouble(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return 0.0d; }
    public static float getFloat(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return 0.0f; }
    public static int getInt(java.lang.Object var0, int var1) throws java.lang.IllegalArgumentException,
java.lang.ArrayIndexOutOfBoundsException { return 0; }
    public static int getLength(java.lang.Object var0) throws java.lang.IllegalArgumentException {
return 0; }
    public static long getLong(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return 0l; }
    public static short getShort(java.lang.Object var0, int var1) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { return 0; }
    public static java.lang.Object newInstance(java.lang.Class var0, int[] var1) throws
java.lang.NegativeArraySizeException, java.lang.IllegalArgumentException { return null; }
    public static java.lang.Object newInstance(java.lang.Class var0, int var1) throws
java.lang.NegativeArraySizeException { return null; }
    public static void set(java.lang.Object var0, int var1, java.lang.Object var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setBoolean(java.lang.Object var0, int var1, boolean var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setByte(java.lang.Object var0, int var1, byte var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setChar(java.lang.Object var0, int var1, char var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setDouble(java.lang.Object var0, int var1, double var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setFloat(java.lang.Object var0, int var1, float var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setInt(java.lang.Object var0, int var1, int var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setLong(java.lang.Object var0, int var1, long var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
    public static void setShort(java.lang.Object var0, int var1, short var2) throws
java.lang.IllegalArgumentException, java.lang.ArrayIndexOutOfBoundsException { }
}

package java.lang.reflect;
public final class Constructor extends java.lang.reflect.AccessibleObject implements
java.lang.reflect.Member {
    private Constructor() { }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.Class getDeclaringClass() { return null; }
    public java.lang.Class[] getExceptionTypes() { return null; }
    public int getModifiers() { return 0; }
    public java.lang.String getName() { return null; }
    public java.lang.Class[] getParameterTypes() { return null; }
    public int hashCode() { return 0; }
    public java.lang.Object newInstance(java.lang.Object[] var0) throws
java.lang.InstantiationException, java.lang.IllegalAccessException, java.lang.IllegalArgumentException,
java.lang.reflect.InvocationTargetException { return null; }
    public java.lang.String toString() { return null; }
}

package java.lang.reflect;
public final class Field extends java.lang.reflect.AccessibleObject implements java.lang.reflect.Member
{
    private Field() { }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.Object get(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return null; }
    public boolean getBoolean(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return false; }
```

```
    public byte getByte(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return 0; }
    public char getChar(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return 0; }
    public java.lang.Class getDeclaringClass() { return null; }
    public double getDouble(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return 0.0d; }
    public float getFloat(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return 0.0f; }
    public int getInt(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return 0; }
    public long getLong(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return 0l; }
    public int getModifiers() { return 0; }
    public java.lang.String getName() { return null; }
    public short getShort(java.lang.Object var0) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { return 0; }
    public java.lang.Class getType() { return null; }
    public int hashCode() { return 0; }
    public void set(java.lang.Object var0, java.lang.Object var1) throws
java.lang.IllegalAccessException, java.lang.IllegalArgumentException { }
    public void setBoolean(java.lang.Object var0, boolean var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public void setByte(java.lang.Object var0, byte var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public void setChar(java.lang.Object var0, char var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public void setDouble(java.lang.Object var0, double var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public void setFloat(java.lang.Object var0, float var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public void setInt(java.lang.Object var0, int var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public void setLong(java.lang.Object var0, long var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public void setShort(java.lang.Object var0, short var1) throws java.lang.IllegalAccessException,
java.lang.IllegalArgumentException { }
    public java.lang.String toString() { return null; }
}

package java.lang.reflect;
public abstract interface InvocationHandler {
    public abstract java.lang.Object invoke(java.lang.Object var0, java.lang.reflect.Method var1,
java.lang.Object[] var2) throws java.lang.Throwable;
}

package java.lang.reflect;
public class InvocationTargetException extends java.lang.Exception {
    protected InvocationTargetException() { }
    public InvocationTargetException(java.lang.Throwable var0) { }
    public InvocationTargetException(java.lang.Throwable var0, java.lang.String var1) { }
    public java.lang.Throwable getTargetException() { return null; }
    public void printStackTrace() { }
    public void printStackTrace(java.io.PrintStream var0) { }
}

package java.lang.reflect;
public abstract interface Member {
    public abstract java.lang.Class getDeclaringClass();
    public abstract int getModifiers();
    public abstract java.lang.String getName();
    public final static int PUBLIC = 0;
    public final static int DECLARED = 1;
}

package java.lang.reflect;
public final class Method extends java.lang.reflect.AccessibleObject implements java.lang.reflect.Member
{
    private Method() { }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.Class getDeclaringClass() { return null; }
    public java.lang.Class[] getExceptionTypes() { return null; }
    public int getModifiers() { return 0; }
    public java.lang.String getName() { return null; }
    public java.lang.Class[] getParameterTypes() { return null; }
    public java.lang.Class getReturnType() { return null; }
    public int hashCode() { return 0; }
    public java.lang.Object invoke(java.lang.Object var0, java.lang.Object[] var1) throws
java.lang.IllegalAccessException, java.lang.IllegalArgumentException,
java.lang.reflect.InvocationTargetException { return null; }
    public java.lang.String toString() { return null; }
}

package java.lang.reflect;
public class Modifier {
    public Modifier() { }
    public static boolean isAbstract(int var0) { return false; }
    public static boolean isFinal(int var0) { return false; }
    public static boolean isInterface(int var0) { return false; }
    public static boolean isNative(int var0) { return false; }
    public static boolean isPrivate(int var0) { return false; }
    public static boolean isProtected(int var0) { return false; }
    public static boolean isPublic(int var0) { return false; }
    public static boolean isStatic(int var0) { return false; }
    public static boolean isStrict(int var0) { return false; }
    public static boolean isSynchronized(int var0) { return false; }
    public static boolean isTransient(int var0) { return false; }
    public static boolean isVolatile(int var0) { return false; }
    public static java.lang.String toString(int var0) { return null; }
    public final static int PUBLIC = 1;
    public final static int PRIVATE = 2;
    public final static int PROTECTED = 4;
    public final static int STATIC = 8;
    public final static int FINAL = 16;
    public final static int SYNCHRONIZED = 32;
    public final static int VOLATILE = 64;
    public final static int TRANSIENT = 128;
    public final static int NATIVE = 256;
    public final static int INTERFACE = 512;
    public final static int ABSTRACT = 1024;
    public final static int STRICT = 2048;
}

package java.lang.reflect;
public class Proxy implements java.io.Serializable {
    private Proxy() { }
    protected Proxy(java.lang.reflect.InvocationHandler var0) { }
    public static java.lang.Class getProxyClass(java.lang.ClassLoader var0, java.lang.Class[] var1)
throws java.lang.IllegalArgumentException { return null; }
    public static java.lang.Object newProxyInstance(java.lang.ClassLoader var0, java.lang.Class[] var1,
java.lang.reflect.InvocationHandler var2) throws java.lang.IllegalArgumentException { return null; }
    public static boolean isProxyClass(java.lang.Class var0) { return false; }
    public static java.lang.reflect.InvocationHandler getInvocationHandler(java.lang.Object var0) throws
java.lang.IllegalArgumentException { return null; }
    protected java.lang.reflect.InvocationHandler h;
}

package java.lang.reflect;
public final class ReflectPermission extends java.security.BasicPermission {
    public ReflectPermission(java.lang.String var0) { super(null, null); }
    public ReflectPermission(java.lang.String var0, java.lang.String var1) { super(null, null); }
}

package java.lang.reflect;
public class UndeclaredThrowableException extends java.lang.RuntimeException {
    public UndeclaredThrowableException(java.lang.Throwable var0) { }
    public UndeclaredThrowableException(java.lang.Throwable var0, java.lang.String var1) { }
    public java.lang.Throwable getUndeclaredThrowable() { return null; }
    public void printStackTrace() { }
    public void printStackTrace(java.io.PrintStream var0) { }
}

package java.lang;
public abstract interface Runnable {
    public abstract void run();
}

package java.lang;
public class Runtime {
    private Runtime() { }
    public java.lang.Process exec(java.lang.String[] var0) throws java.io.IOException { return null; }
    public java.lang.Process exec(java.lang.String[] var0, java.lang.String[] var1) throws
java.io.IOException { return null; }
    public java.lang.Process exec(java.lang.String var0) throws java.io.IOException { return null; }
    public java.lang.Process exec(java.lang.String var0, java.lang.String[] var1) throws
java.io.IOException { return null; }
    public void exit(int var0) { }
    public long freeMemory() { return 0l; }
    public void gc() { }
    public static java.lang.Runtime getRuntime() { return null; }
    public void load(java.lang.String var0) { }
    public void loadLibrary(java.lang.String var0) { }
    public void runFinalization() { }
    public long totalMemory() { return 0l; }
}

package java.lang;
public class RuntimeException extends java.lang.Exception {
    public RuntimeException() { }
    public RuntimeException(java.lang.String var0) { }
}

package java.lang;
public final class RuntimePermission extends java.security.BasicPermission {
    public RuntimePermission(java.lang.String var0) { super(null, null); }
    public RuntimePermission(java.lang.String var0, java.lang.String var1) { super(null, null); }
}

package java.lang;
public class SecurityException extends java.lang.RuntimeException {
    public SecurityException() { }
    public SecurityException(java.lang.String var0) { }
}

package java.lang;
public class SecurityManager {
    public SecurityManager() { }
    public void checkAccept(java.lang.String var0, int var1) { }
    public void checkAccess(java.lang.Thread var0) { }
    public void checkAccess(java.lang.ThreadGroup var0) { }
    public void checkConnect(java.lang.String var0, int var1) { }
    public void checkConnect(java.lang.String var0, int var1, java.lang.Object var2) { }
    public void checkCreateClassLoader() { }
    public void checkDelete(java.lang.String var0) { }
    public void checkExec(java.lang.String var0) { }
    public void checkExit(int var0) { }
    public void checkLink(java.lang.String var0) { }
    public void checkListen(int var0) { }
    public void checkMemberAccess(java.lang.Class var0, int var1) { }
    public void checkMulticast(java.net.InetAddress var0) { }
    public void checkMulticast(java.net.InetAddress var0, byte var1) { }
    public void checkPackageAccess(java.lang.String var0) { }
    public void checkPackageDefinition(java.lang.String var0) { }
    public void checkPropertiesAccess() { }
    public void checkPropertyAccess(java.lang.String var0) { }
    public void checkRead(java.io.FileDescriptor var0) { }
    public void checkRead(java.lang.String var0) { }
    public void checkRead(java.lang.String var0, java.lang.Object var1) { }
    public void checkSecurityAccess(java.lang.String var0) { }
    public void checkSetFactory() { }
    public boolean checkTopLevelWindow(java.lang.Object var0) { return false; }
    public void checkSystemClipboardAccess() { }
    public void checkAwtEventQueueAccess() { }
    public void checkPrintJobAccess() { }
    public void checkWrite(java.io.FileDescriptor var0) { }
    public void checkWrite(java.lang.String var0) { }
    protected java.lang.Class[] getClassContext() { return null; }
    public java.lang.ThreadGroup getThreadGroup() { return null; }
    public java.lang.Object getSecurityContext() { return null; }
    public void checkPermission(java.security.Permission var0) { }
    public void checkPermission(java.security.Permission var0, java.lang.Object var1) { }
}

package java.lang;
public final class Short extends java.lang.Number implements java.lang.Comparable {
    public Short(java.lang.String var0) throws java.lang.NumberFormatException { }
    public Short(short var0) { }
    public byte byteValue() { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public int compareTo(java.lang.Short var0) { return 0; }
    public double doubleValue() { return 0.0d; }
    public boolean equals(java.lang.Object var0) { return false; }
    public float floatValue() { return 0.0f; }
    public int hashCode() { return 0; }
    public int intValue() { return 0; }
    public long longValue() { return 0l; }
    public static short parseShort(java.lang.String var0) throws java.lang.NumberFormatException {
return 0; }
    public static short parseShort(java.lang.String var0, int var1) throws
java.lang.NumberFormatException { return 0; }
    public short shortValue() { return 0; }
    public java.lang.String toString() { return null; }
    public static java.lang.Short valueOf(java.lang.String var0, int var1) throws
java.lang.NumberFormatException { return null; }
    public final static short MAX_VALUE = 32767;
    public final static short MIN_VALUE = -32768;
    public final static java.lang.Class TYPE; static { TYPE = null; }
}

package java.lang;
public class StackOverflowError extends java.lang.VirtualMachineError {
    public StackOverflowError() { }
    public StackOverflowError(java.lang.String var0) { }
}

package java.lang;
public final class String implements java.io.Serializable, java.lang.Comparable {
    public String() { }
    public String(byte[] var0) { }
    public String(byte[] var0, int var1, int var2) { }
    public String(byte[] var0, int var1, int var2, java.lang.String var3) throws
java.io.UnsupportedEncodingException { }
    public String(byte[] var0, java.lang.String var1) throws java.io.UnsupportedEncodingException { }
    public String(char[] var0) { }
    public String(char[] var0, int var1, int var2) { }
    public String(java.lang.String var0) { }
    public char charAt(int var0) { return 0; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public int compareTo(java.lang.String var0) { return 0; }
    public int compareToIgnoreCase(java.lang.String var0) { return 0; }
    public java.lang.String concat(java.lang.String var0) { return null; }
    public boolean endsWith(java.lang.String var0) { return false; }
    public boolean equals(java.lang.Object var0) { return false; }
    public boolean equalsIgnoreCase(java.lang.String var0) { return false; }
    public byte[] getBytes() { return null; }
    public byte[] getBytes(java.lang.String var0) throws java.io.UnsupportedEncodingException { return
null; }
    public void getChars(int var0, int var1, char[] var2, int var3) { }
    public int hashCode() { return 0; }
    public int indexOf(int var0) { return 0; }
    public int indexOf(int var0, int var1) { return 0; }
    public int indexOf(java.lang.String var0) { return 0; }
    public int indexOf(java.lang.String var0, int var1) { return 0; }
    public java.lang.String intern() { return null; }
    public int lastIndexOf(int var0) { return 0; }
    public int lastIndexOf(int var0, int var1) { return 0; }
    public int lastIndexOf(java.lang.String var0) { return 0; }
    public int lastIndexOf(java.lang.String var0, int var1) { return 0; }
    public int length() { return 0; }
    public boolean regionMatches(int var0, java.lang.String var1, int var2, int var3) { return false; }
    public boolean regionMatches(boolean var0, int var1, java.lang.String var2, int var3, int var4) {
return false; }
    public java.lang.String replace(char var0, char var1) { return null; }
    public boolean startsWith(java.lang.String var0) { return false; }
    public boolean startsWith(java.lang.String var0, int var1) { return false; }
    public java.lang.String substring(int var0) { return null; }
    public java.lang.String substring(int var0, int var1) { return null; }
    public char[] toCharArray() { return null; }
    public java.lang.String toLowerCase() { return null; }
    public java.lang.String toString() { return null; }
    public java.lang.String toUpperCase() { return null; }
    public java.lang.String trim() { return null; }
    public static java.lang.String valueOf(char var0) { return null; }
    public static java.lang.String valueOf(double var0) { return null; }
    public static java.lang.String valueOf(float var0) { return null; }
    public static java.lang.String valueOf(int var0) { return null; }
    public static java.lang.String valueOf(long var0) { return null; }
    public static java.lang.String valueOf(java.lang.Object var0) { return null; }
    public static java.lang.String valueOf(boolean var0) { return null; }
    public final static java.util.Comparator CASE_INSENSITIVE_ORDER; static { CASE_INSENSITIVE_ORDER =
null; }
}

package java.lang;
public final class StringBuffer implements java.io.Serializable {
    public StringBuffer() { }
    public StringBuffer(int var0) { }
    public StringBuffer(java.lang.String var0) { }
    public java.lang.StringBuffer append(char[] var0) { return null; }
    public java.lang.StringBuffer append(char[] var0, int var1, int var2) { return null; }
    public java.lang.StringBuffer append(char var0) { return null; }
    public java.lang.StringBuffer append(double var0) { return null; }
    public java.lang.StringBuffer append(float var0) { return null; }
    public java.lang.StringBuffer append(int var0) { return null; }
    public java.lang.StringBuffer append(long var0) { return null; }
    public java.lang.StringBuffer append(java.lang.Object var0) { return null; }
    public java.lang.StringBuffer append(java.lang.String var0) { return null; }
    public java.lang.StringBuffer append(boolean var0) { return null; }
    public int capacity() { return 0; }
    public char charAt(int var0) { return 0; }
    public java.lang.StringBuffer delete(int var0, int var1) { return null; }
    public java.lang.StringBuffer deleteCharAt(int var0) { return null; }
    public void ensureCapacity(int var0) { }
    public void getChars(int var0, int var1, char[] var2, int var3) { }
    public java.lang.StringBuffer insert(int var0, char[] var1) { return null; }
    public java.lang.StringBuffer insert(int var0, char var1) { return null; }
    public java.lang.StringBuffer insert(int var0, double var1) { return null; }
    public java.lang.StringBuffer insert(int var0, float var1) { return null; }
    public java.lang.StringBuffer insert(int var0, int var1) { return null; }
    public java.lang.StringBuffer insert(int var0, long var1) { return null; }
    public java.lang.StringBuffer insert(int var0, java.lang.Object var1) { return null; }
    public java.lang.StringBuffer insert(int var0, java.lang.String var1) { return null; }
    public java.lang.StringBuffer insert(int var0, boolean var1) { return null; }
    public int length() { return 0; }
    public java.lang.StringBuffer reverse() { return null; }
    public void setCharAt(int var0, char var1) { }
    public void setLength(int var0) { }
    public java.lang.String substring(int var0, int var1) { return null; }
    public java.lang.String toString() { return null; }
}

package java.lang;
public class StringIndexOutOfBoundsException extends java.lang.IndexOutOfBoundsException {
    public StringIndexOutOfBoundsException() { }
    public StringIndexOutOfBoundsException(int var0) { }
    public StringIndexOutOfBoundsException(java.lang.String var0) { }
}

package java.lang;
public final class System {
    public static void setIn(java.io.InputStream var0) { }
    public static void setOut(java.io.PrintStream var0) { }
    public static void setErr(java.io.PrintStream var0) { }
    private System() { }
    public static void arraycopy(java.lang.Object var0, int var1, java.lang.Object var2, int var3, int
var4) { }
    public static long currentTimeMillis() { return 0l; }
    public static void exit(int var0) { }
    public static void gc() { }
    public static java.util.Properties getProperties() { return null; }
    public static java.lang.String getProperty(java.lang.String var0) { return null; }
    public static java.lang.String getProperty(java.lang.String var0, java.lang.String var1) { return
null; }
    public static java.lang.SecurityManager getSecurityManager() { return null; }
    public static int identityHashCode(java.lang.Object var0) { return 0; }
    public static void loadLibrary(java.lang.String var0) { }
    public static void runFinalization() { }
    public static void setProperties(java.util.Properties var0) { }
    public static void setSecurityManager(java.lang.SecurityManager var0) { }
    public static java.lang.String mapLibraryName(java.lang.String var0) { return null; }
    public final static java.io.InputStream in; static { in = null; }
    public final static java.io.PrintStream out; static { out = null; }
    public final static java.io.PrintStream err; static { err = null; }
}

package java.lang;
public class Thread implements java.lang.Runnable {
    public Thread() { }
    private Thread(int var0, java.lang.String var1, java.lang.Object var2, int var3, boolean var4) { }
    public Thread(java.lang.Runnable var0) { }
    public Thread(java.lang.Runnable var0, java.lang.String var1) { }
    public Thread(java.lang.String var0) { }
    public Thread(java.lang.ThreadGroup var0, java.lang.Runnable var1) { }
    public Thread(java.lang.ThreadGroup var0, java.lang.Runnable var1, java.lang.String var2) { }
    public Thread(java.lang.ThreadGroup var0, java.lang.String var1) { }
    public final void checkAccess() { }
    public static java.lang.Thread currentThread() { return null; }
    public final java.lang.String getName() { return null; }
    public final int getPriority() { return 0; }
    public final java.lang.ThreadGroup getThreadGroup() { return null; }
    public void interrupt() { }
    public static boolean interrupted() { return false; }
    public final boolean isAlive() { return false; }
    public final boolean isDaemon() { return false; }
    public boolean isInterrupted() { return false; }
    public final void join() throws java.lang.InterruptedException { }
    public final void join(long var0) throws java.lang.InterruptedException { }
    public final void join(long var0, int var1) throws java.lang.InterruptedException { }
    public void run() { }
    public final void setDaemon(boolean var0) { }
    public final void setName(java.lang.String var0) { }
    public final void setPriority(int var0) { }
    public static void sleep(long var0) throws java.lang.InterruptedException { }
    public static void sleep(long var0, int var1) throws java.lang.InterruptedException { }
    public void start() { }
    public java.lang.String toString() { return null; }
    public static void yield() { }
    public final static int MAX_PRIORITY = 10;
    public final static int MIN_PRIORITY = 1;
    public final static int NORM_PRIORITY = 5;
}

package java.lang;
public class ThreadDeath extends java.lang.Error {
    public ThreadDeath() { }
}

package java.lang;
public class ThreadGroup {
    public ThreadGroup(java.lang.String var0) { }
    private ThreadGroup(java.lang.String var0, java.lang.ThreadGroup var1) { }
    public ThreadGroup(java.lang.ThreadGroup var0, java.lang.String var1) { }
    public int activeCount() { return 0; }
    public int activeGroupCount() { return 0; }
    public final void checkAccess() { }
    public final void destroy() { }
    public int enumerate(java.lang.Thread[] var0) { return 0; }
    public int enumerate(java.lang.Thread[] var0, boolean var1) { return 0; }
    public int enumerate(java.lang.ThreadGroup[] var0) { return 0; }
    public int enumerate(java.lang.ThreadGroup[] var0, boolean var1) { return 0; }
    public final int getMaxPriority() { return 0; }
    public final java.lang.String getName() { return null; }
    public final java.lang.ThreadGroup getParent() { return null; }
    public final boolean isDaemon() { return false; }
    public boolean isDestroyed() { return false; }
    public void list() { }
    public final boolean parentOf(java.lang.ThreadGroup var0) { return false; }
    public final void setDaemon(boolean var0) { }
    public final void setMaxPriority(int var0) { }
    public java.lang.String toString() { return null; }
    public void uncaughtException(java.lang.Thread var0, java.lang.Throwable var1) { }
}

package java.lang;
public class Throwable implements java.io.Serializable {
    public Throwable() { }
    public Throwable(java.lang.String var0) { }
    public java.lang.Throwable fillInStackTrace() { return null; }
    public java.lang.String getMessage() { return null; }
    public void printStackTrace() { }
    public void printStackTrace(java.io.PrintStream var0) { }
    public void printStackTrace(java.io.PrintWriter var0) { }
    public java.lang.String toString() { return null; }
}

package java.lang;
public class UnknownError extends java.lang.VirtualMachineError {
    public UnknownError() { }
    public UnknownError(java.lang.String var0) { }
}

package java.lang;
public class UnsatisfiedLinkError extends java.lang.LinkageError {
    public UnsatisfiedLinkError() { }
    public UnsatisfiedLinkError(java.lang.String var0) { }
}

package java.lang;
public class UnsupportedOperationException extends java.lang.RuntimeException {
    public UnsupportedOperationException() { }
    public UnsupportedOperationException(java.lang.String var0) { }
}
```

```
package java.lang;
public class VerifyError extends java.lang.LinkageError {
    public VerifyError() { }
    public VerifyError(java.lang.String var0) { }
}

package java.lang;
public abstract class VirtualMachineError extends java.lang.Error {
    public VirtualMachineError() { }
    public VirtualMachineError(java.lang.String var0) { }
}

package java.lang;
public final class Void {
    private Void() { }
    public final static java.lang.Class TYPE; static { TYPE = null; }
}

package java.net;
public abstract class Authenticator {
    public Authenticator() { }
    protected java.net.PasswordAuthentication getPasswordAuthentication() { return null; }
    protected final int getRequestingPort() { return 0; }
    protected final java.net.InetAddress getRequestingSite() { return null; }
    protected final java.lang.String getRequestingPrompt() { return null; }
    protected final java.lang.String getRequestingProtocol() { return null; }
    protected final java.lang.String getRequestingScheme() { return null; }
    public static java.net.PasswordAuthentication requestPasswordAuthentication(java.net.InetAddress
var0, int var1, java.lang.String var2, java.lang.String var3, java.lang.String var4) { return null; }
    public static void setDefault(java.net.Authenticator var0) { }
}

package java.net;
public class BindException extends java.net.SocketException {
    public BindException() { }
    public BindException(java.lang.String var0) { }
}

package java.net;
public class ConnectException extends java.net.SocketException {
    public ConnectException() { }
    public ConnectException(java.lang.String var0) { }
}

package java.net;
public abstract class ContentHandler {
    public ContentHandler() { }
    public abstract java.lang.Object getContent(java.net.URLConnection var0) throws java.io.IOException;
}

package java.net;
public abstract interface ContentHandlerFactory {
    public abstract java.net.ContentHandler createContentHandler(java.lang.String var0);
}

package java.net;
public final class DatagramPacket {
    public DatagramPacket(byte[] var0, int var1) { }
    public DatagramPacket(byte[] var0, int var1, int var2) { }
    public DatagramPacket(byte[] var0, int var1, int var2, java.net.InetAddress var3, int var4) { }
    public DatagramPacket(byte[] var0, int var1, java.net.InetAddress var2, int var3) { }
    public java.net.InetAddress getAddress() { return null; }
    public byte[] getData() { return null; }
    public int getLength() { return 0; }
    public int getOffset() { return 0; }
    public int getPort() { return 0; }
    public void setAddress(java.net.InetAddress var0) { }
    public void setData(byte[] var0, int var1, int var2) { }
    public void setData(byte[] var0) { }
    public void setLength(int var0) { }
    public void setPort(int var0) { }
}

package java.net;
public class DatagramSocket {
    public DatagramSocket() throws java.net.SocketException { }
    public DatagramSocket(int var0) throws java.net.SocketException { }
    public DatagramSocket(int var0, java.net.InetAddress var1) throws java.net.SocketException { }
    public void close() { }
    public void connect(java.net.InetAddress var0, int var1) { }
    public void disconnect() { }
    public java.net.InetAddress getInetAddress() { return null; }
    public java.net.InetAddress getLocalAddress() { return null; }
    public int getLocalPort() { return 0; }
    public int getPort() { return 0; }
    public int getReceiveBufferSize() throws java.net.SocketException { return 0; }
    public int getSendBufferSize() throws java.net.SocketException { return 0; }
    public int getSoTimeout() throws java.net.SocketException { return 0; }
    public void receive(java.net.DatagramPacket var0) throws java.io.IOException { }
    public void send(java.net.DatagramPacket var0) throws java.io.IOException { }
    public void setSendBufferSize(int var0) throws java.net.SocketException { }
    public void setReceiveBufferSize(int var0) throws java.net.SocketException { }
    public void setSoTimeout(int var0) throws java.net.SocketException { }
}

package java.net;
public abstract class DatagramSocketImpl implements java.net.SocketOptions {
    public DatagramSocketImpl() { }
    protected abstract void bind(int var0, java.net.InetAddress var1) throws java.net.SocketException;
    protected abstract void close();
    protected abstract void create() throws java.net.SocketException;
    protected java.io.FileDescriptor getFileDescriptor() { return null; }
    protected int getLocalPort() { return 0; }
    public abstract java.lang.Object getOption(int var0) throws java.net.SocketException;
    protected abstract int getTimeToLive() throws java.io.IOException;
    protected abstract void join(java.net.InetAddress var0) throws java.io.IOException;
    protected abstract void leave(java.net.InetAddress var0) throws java.io.IOException;
    protected abstract int peek(java.net.InetAddress var0) throws java.io.IOException;
    protected abstract void receive(java.net.DatagramPacket var0) throws java.io.IOException;
    protected abstract void send(java.net.DatagramPacket var0) throws java.io.IOException;
    public abstract void setOption(int var0, java.lang.Object var1) throws java.net.SocketException;
    protected abstract void setTimeToLive(int var0) throws java.io.IOException;
    protected java.io.FileDescriptor fd;
    protected int localPort;
}

package java.net;
public abstract interface FileNameMap {
    public abstract java.lang.String getContentTypeFor(java.lang.String var0);
}

package java.net;
public abstract class HttpURLConnection extends java.net.URLConnection {
    protected HttpURLConnection(java.net.URL var0) { super(null); }
    public abstract void disconnect();
    public java.io.InputStream getErrorStream() { return null; }
    public static boolean getFollowRedirects() { return false; }
    public java.security.Permission getPermission() throws java.io.IOException { return null; }
    public java.lang.String getRequestMethod() { return null; }
    public int getResponseCode() throws java.io.IOException { return 0; }
    public java.lang.String getResponseMessage() throws java.io.IOException { return null; }
    public static void setFollowRedirects(boolean var0) { }
    public void setRequestMethod(java.lang.String var0) throws java.net.ProtocolException { }
    public abstract boolean usingProxy();
    protected java.lang.String method;
    protected int responseCode;
    protected java.lang.String responseMessage;
    public final static int HTTP_ACCEPTED = 202;
    public final static int HTTP_BAD_GATEWAY = 502;
    public final static int HTTP_BAD_METHOD = 405;
    public final static int HTTP_BAD_REQUEST = 400;
    public final static int HTTP_CLIENT_TIMEOUT = 408;
    public final static int HTTP_CONFLICT = 409;
    public final static int HTTP_CREATED = 201;
    public final static int HTTP_ENTITY_TOO_LARGE = 413;
    public final static int HTTP_FORBIDDEN = 403;
    public final static int HTTP_GATEWAY_TIMEOUT = 504;
    public final static int HTTP_GONE = 410;
    public final static int HTTP_INTERNAL_ERROR = 500;
    public final static int HTTP_LENGTH_REQUIRED = 411;
    public final static int HTTP_MOVED_PERM = 301;
    public final static int HTTP_MOVED_TEMP = 302;
    public final static int HTTP_MULT_CHOICE = 300;
    public final static int HTTP_NO_CONTENT = 204;
    public final static int HTTP_NOT_ACCEPTABLE = 406;
    public final static int HTTP_NOT_AUTHORITATIVE = 203;
    public final static int HTTP_NOT_FOUND = 404;
    public final static int HTTP_NOT_IMPLEMENTED = 501;
    public final static int HTTP_NOT_MODIFIED = 304;
    public final static int HTTP_OK = 200;
    public final static int HTTP_PARTIAL = 206;
    public final static int HTTP_PAYMENT_REQUIRED = 402;
    public final static int HTTP_PRECON_FAILED = 412;
    public final static int HTTP_PROXY_AUTH = 407;
    public final static int HTTP_REQ_TOO_LONG = 414;
    public final static int HTTP_RESET = 205;
    public final static int HTTP_SEE_OTHER = 303;
    public final static int HTTP_USE_PROXY = 305;
    public final static int HTTP_UNAUTHORIZED = 401;
    public final static int HTTP_UNSUPPORTED_TYPE = 415;
    public final static int HTTP_UNAVAILABLE = 503;
    public final static int HTTP_VERSION = 505;
}

package java.net;
public final class InetAddress implements java.io.Serializable {
    InetAddress(int var0) { }
    private InetAddress(int var0, java.lang.String var1) { }
    public boolean equals(java.lang.Object var0) { return false; }
    public byte[] getAddress() { return null; }
    public static java.net.InetAddress[] getAllByName(java.lang.String var0) throws
java.net.UnknownHostException { return null; }
    public static java.net.InetAddress getByName(java.lang.String var0) throws
java.net.UnknownHostException { return null; }
    public java.lang.String getHostAddress() { return null; }
    public java.lang.String getHostName() { return null; }
    public static java.net.InetAddress getLocalHost() throws java.net.UnknownHostException { return
null; }
    public int hashCode() { return 0; }
    public boolean isMulticastAddress() { return false; }
    public java.lang.String toString() { return null; }
}

package java.net;
public abstract class JarURLConnection extends java.net.URLConnection {
    protected JarURLConnection(java.net.URL var0) throws java.net.MalformedURLException { super(null); }
    public java.lang.String getEntryName() { return null; }
    public java.net.URL getJarFileURL() { return null; }
    protected java.net.URLConnection jarFileURLConnection;
}

package java.net;
public class MalformedURLException extends java.io.IOException {
    public MalformedURLException() { }
    public MalformedURLException(java.lang.String var0) { }
}
```

```
package java.net;
public class MulticastSocket extends java.net.DatagramSocket {
    public MulticastSocket() throws java.io.IOException { }
    public MulticastSocket(int var0) throws java.io.IOException { }
    public java.net.InetAddress getInterface() throws java.net.SocketException { return null; }
    public int getTimeToLive() throws java.io.IOException { return 0; }
    public void joinGroup(java.net.InetAddress var0) throws java.io.IOException { }
    public void leaveGroup(java.net.InetAddress var0) throws java.io.IOException { }
    public void send(java.net.DatagramPacket var0, byte var1) throws java.io.IOException { }
    public void setInterface(java.net.InetAddress var0) throws java.net.SocketException { }
    public void setTimeToLive(int var0) throws java.io.IOException { }
}

package java.net;
public final class NetPermission extends java.security.BasicPermission {
    public NetPermission(java.lang.String var0) { super(null, null); }
    public NetPermission(java.lang.String var0, java.lang.String var1) { super(null, null); }
}

package java.net;
public class NoRouteToHostException extends java.net.SocketException {
    public NoRouteToHostException() { }
    public NoRouteToHostException(java.lang.String var0) { }
}

package java.net;
public final class PasswordAuthentication {
    public PasswordAuthentication(java.lang.String var0, char[] var1) { }
    public char[] getPassword() { return null; }
    public java.lang.String getUserName() { return null; }
}

package java.net;
public class ProtocolException extends java.io.IOException {
    public ProtocolException() { }
    public ProtocolException(java.lang.String var0) { }
}

package java.net;
public class ServerSocket {
    public ServerSocket(int var0) throws java.io.IOException { }
    public ServerSocket(int var0, int var1) throws java.io.IOException { }
    public ServerSocket(int var0, int var1, java.net.InetAddress var2) throws java.io.IOException { }
    public java.net.Socket accept() throws java.io.IOException { return null; }
    public void close() throws java.io.IOException { }
    public java.net.InetAddress getInetAddress() { return null; }
    public int getLocalPort() { return 0; }
    public int getSoTimeout() throws java.io.IOException { return 0; }
    protected final void implAccept(java.net.Socket var0) throws java.io.IOException { }
    public static void setSocketFactory(java.net.SocketImplFactory var0) throws java.io.IOException { }
    public void setSoTimeout(int var0) throws java.net.SocketException { }
    public java.lang.String toString() { return null; }
}

package java.net;
public class Socket {
    protected Socket() { }
    public Socket(java.lang.String var0, int var1) throws java.net.UnknownHostException,
java.io.IOException { }
    public Socket(java.lang.String var0, int var1, java.net.InetAddress var2, int var3) throws
java.io.IOException { }
    public Socket(java.net.InetAddress var0, int var1) throws java.io.IOException { }
    public Socket(java.net.InetAddress var0, int var1, java.net.InetAddress var2, int var3) throws
java.io.IOException { }
    protected Socket(java.net.SocketImpl var0) throws java.net.SocketException { }
    public void close() throws java.io.IOException { }
    public java.net.InetAddress getInetAddress() { return null; }
    public java.io.InputStream getInputStream() throws java.io.IOException { return null; }
    public java.net.InetAddress getLocalAddress() { return null; }
    public int getLocalPort() { return 0; }
    public java.io.OutputStream getOutputStream() throws java.io.IOException { return null; }
    public int getPort() { return 0; }
    public int getSoLinger() throws java.net.SocketException { return 0; }
    public int getReceiveBufferSize() throws java.net.SocketException { return 0; }
    public int getSendBufferSize() throws java.net.SocketException { return 0; }
    public int getSoTimeout() throws java.net.SocketException { return 0; }
    public boolean getTcpNoDelay() throws java.net.SocketException { return false; }
    public static void setSocketImplFactory(java.net.SocketImplFactory var0) throws java.io.IOException
{ }
    public void setSendBufferSize(int var0) throws java.net.SocketException { }
    public void setReceiveBufferSize(int var0) throws java.net.SocketException { }
    public void setSoLinger(boolean var0, int var1) throws java.net.SocketException { }
    public void setSoTimeout(int var0) throws java.net.SocketException { }
    public void setTcpNoDelay(boolean var0) throws java.net.SocketException { }
    public java.lang.String toString() { return null; }
}

package java.net;
public class SocketException extends java.io.IOException {
    public SocketException() { }
    public SocketException(java.lang.String var0) { }
}

package java.net;
public abstract class SocketImpl implements java.net.SocketOptions {
    public SocketImpl() { }
    protected abstract void accept(java.net.SocketImpl var0) throws java.io.IOException;
    protected abstract int available() throws java.io.IOException;
    protected abstract void bind(java.net.InetAddress var0, int var1) throws java.io.IOException;
    protected abstract void close() throws java.io.IOException;
    protected abstract void connect(java.lang.String var0, int var1) throws java.io.IOException;
    protected abstract void connect(java.net.InetAddress var0, int var1) throws java.io.IOException;
    protected abstract void create(boolean var0) throws java.io.IOException;
    protected java.io.FileDescriptor getFileDescriptor() { return null; }
    protected java.net.InetAddress getInetAddress() { return null; }
    protected abstract java.io.InputStream getInputStream() throws java.io.IOException;
    protected int getLocalPort() { return 0; }
    public abstract java.lang.Object getOption(int var0) throws java.net.SocketException;
    protected abstract java.io.OutputStream getOutputStream() throws java.io.IOException;
    protected int getPort() { return 0; }
    protected abstract void listen(int var0) throws java.io.IOException;
    public abstract void setOption(int var0, java.lang.Object var1) throws java.net.SocketException;
    public java.lang.String toString() { return null; }
    protected java.net.InetAddress address;
    protected int port;
    protected java.io.FileDescriptor fd;
    protected int localport;
}

package java.net;
public abstract interface SocketImplFactory {
    public abstract java.net.SocketImpl createSocketImpl();
}

package java.net;
public abstract interface SocketOptions {
    public abstract java.lang.Object getOption(int var0) throws java.net.SocketException;
    public abstract void setOption(int var0, java.lang.Object var1) throws java.net.SocketException;
    public final static int SO_LINGER = 128;
    public final static int SO_TIMEOUT = 4102;
    public final static int TCP_NODELAY = 1;
    public final static int IP_MULTICAST_IF = 16;
    public final static int SO_BINDADDR = 15;
    public final static int SO_REUSEADDR = 4;
    public final static int SO_SNDBUF = 4097;
    public final static int SO_RCVBUF = 4098;
}
```

```
package java.net;
public final class SocketPermission extends java.security.Permission implements java.io.Serializable {
    public SocketPermission(java.lang.String var0, java.lang.String var1) { super(null); }
    public boolean equals(java.lang.Object var0) { return false; }
    public int hashCode() { return 0; }
    public java.lang.String getActions() { return null; }
    public boolean implies(java.security.Permission var0) { return false; }
    public java.security.PermissionCollection newPermissionCollection() { return null; }
}

package java.net;
public class UnknownHostException extends java.io.IOException {
    public UnknownHostException() { }
    public UnknownHostException(java.lang.String var0) { }
}

package java.net;
public class UnknownServiceException extends java.io.IOException {
    public UnknownServiceException() { }
    public UnknownServiceException(java.lang.String var0) { }
}

package java.net;
public final class URL implements java.io.Serializable {
    public static void setURLStreamHandlerFactory(java.net.URLStreamHandlerFactory var0) { }
    public URL(java.lang.String var0) throws java.net.MalformedURLException { }
    public URL(java.net.URL var0, java.lang.String var1) throws java.net.MalformedURLException { }
    public URL(java.net.URL var0, java.lang.String var1, java.net.URLStreamHandler var2) throws
java.net.MalformedURLException { }
    public URL(java.lang.String var0, java.lang.String var1, java.lang.String var2) throws
java.net.MalformedURLException { }
    public URL(java.lang.String var0, java.lang.String var1, int var2, java.lang.String var3) throws
java.net.MalformedURLException { }
    public URL(java.lang.String var0, java.lang.String var1, int var2, java.lang.String var3,
java.net.URLStreamHandler var4) throws java.net.MalformedURLException { }
    protected void set(java.lang.String var0, java.lang.String var1, int var2, java.lang.String var3,
java.lang.String var4) { }
    public boolean equals(java.lang.Object var0) { return false; }
    public boolean sameFile(java.net.URL var0) { return false; }
    public int hashCode() { return 0; }
    public final java.lang.Object getContent() throws java.io.IOException { return null; }
    public final java.io.InputStream openStream() throws java.io.IOException { return null; }
    public java.net.URLConnection openConnection() throws java.io.IOException { return null; }
    public java.lang.String toString() { return null; }
    public java.lang.String toExternalForm() { return null; }
    public java.lang.String getFile() { return null; }
    public java.lang.String getHost() { return null; }
    public int getPort() { return 0; }
    public java.lang.String getProtocol() { return null; }
    public java.lang.String getRef() { return null; }
    public java.lang.String getQuery() { return null; }
    public java.lang.String getPath() { return null; }
    public java.lang.String getUserInfo() { return null; }
    public java.lang.String getAuthority() { return null; }
    protected void set(java.lang.String var0, java.lang.String var1, int var2, java.lang.String var3,
java.lang.String var4, java.lang.String var5, java.lang.String var6, java.lang.String var7) { }
}

package java.net;
public abstract class URLConnection {
    protected URLConnection(java.net.URL var0) { }
    public abstract void connect() throws java.io.IOException;
    public boolean getAllowUserInteraction() { return false; }
    public java.lang.Object getContent() throws java.io.IOException { return null; }
    public java.lang.String getContentEncoding() { return null; }
    public int getContentLength() { return 0; }
    public java.lang.String getContentType() { return null; }
    public long getDate() { return 0l; }
    public static boolean getDefaultAllowUserInteraction() { return false; }
    public boolean getDefaultUseCaches() { return false; }
    public boolean getDoInput() { return false; }
    public boolean getDoOutput() { return false; }
    public long getExpiration() { return 0l; }
    public static java.net.FileNameMap getFileNameMap() { return null; }
    public java.lang.String getHeaderField(int var0) { return null; }
    public java.lang.String getHeaderField(java.lang.String var0) { return null; }
    public long getHeaderFieldDate(java.lang.String var0, long var1) { return 0l; }
    public int getHeaderFieldInt(java.lang.String var0, int var1) { return 0; }
    public java.lang.String getHeaderFieldKey(int var0) { return null; }
    public long getIfModifiedSince() { return 0l; }
    public java.io.InputStream getInputStream() throws java.io.IOException { return null; }
    public long getLastModified() { return 0l; }
    public java.io.OutputStream getOutputStream() throws java.io.IOException { return null; }
    public java.security.Permission getPermission() throws java.io.IOException { return null; }
    public java.lang.String getRequestProperty(java.lang.String var0) { return null; }
    public java.net.URL getURL() { return null; }
    public boolean getUseCaches() { return false; }
    protected static java.lang.String guessContentTypeFromName(java.lang.String var0) { return null; }
    public static java.lang.String guessContentTypeFromStream(java.io.InputStream var0) throws
java.io.IOException { return null; }
    public void setAllowUserInteraction(boolean var0) { }
    public static void setContentHandlerFactory(java.net.ContentHandlerFactory var0) { }
    public static void setDefaultAllowUserInteraction(boolean var0) { }
    public void setDefaultUseCaches(boolean var0) { }
    public void setDoInput(boolean var0) { }
    public void setDoOutput(boolean var0) { }
    public static void setFileNameMap(java.net.FileNameMap var0) { }
    public void setIfModifiedSince(long var0) { }
    public void setRequestProperty(java.lang.String var0, java.lang.String var1) { }
    public void setUseCaches(boolean var0) { }
    public java.lang.String toString() { return null; }
    protected java.net.URL url;
    protected long ifModifiedSince;
    protected boolean useCaches;
    protected boolean connected;
    protected boolean doOutput;
    protected boolean doInput;
    protected boolean allowUserInteraction;
}

package java.net;
public class URLDecoder {
    public URLDecoder() { }
    public static java.lang.String decode(java.lang.String var0) { return null; }
}

package java.net;
public class URLEncoder {
    private URLEncoder() { }
    public static java.lang.String encode(java.lang.String var0) { return null; }
}

package java.net;
public abstract class URLStreamHandler {
    public URLStreamHandler() { }
    protected abstract java.net.URLConnection openConnection(java.net.URL var0) throws
java.io.IOException;
    protected void parseURL(java.net.URL var0, java.lang.String var1, int var2, int var3) { }
    protected void setURL(java.net.URL var0, java.lang.String var1, java.lang.String var2, int var3,
java.lang.String var4, java.lang.String var5, java.lang.String var6, java.lang.String var7,
java.lang.String var8) { }
    protected java.lang.String toExternalForm(java.net.URL var0) { return null; }
    protected boolean equals(java.net.URL var0, java.net.URL var1) { return false; }
    protected int getDefaultPort() { return 0; }
    protected java.net.InetAddress getHostAddress(java.net.URL var0) { return null; }
    protected int hashCode(java.net.URL var0) { return 0; }
    protected boolean hostsEqual(java.net.URL var0, java.net.URL var1) { return false; }
    protected boolean sameFile(java.net.URL var0, java.net.URL var1) { return false; }
}

package java.net;
public abstract interface URLStreamHandlerFactory {
    public abstract java.net.URLStreamHandler createURLStreamHandler(java.lang.String var0);
}

package java.security;
public final class AccessControlContext {
    public AccessControlContext(java.security.ProtectionDomain[] var0) { }
    public void checkPermission(java.security.Permission var0) throws
java.security.AccessControlException { }
    public boolean equals(java.lang.Object var0) { return false; }
    public int hashCode() { return 0; }
}

package java.security;
public class AccessControlException extends java.lang.SecurityException {
    public AccessControlException(java.lang.String var0) { }
    public AccessControlException(java.lang.String var0, java.security.Permission var1) { }
    public java.security.Permission getPermission() { return null; }
}

package java.security;
public final class AccessController {
    private AccessController() { }
    public static void checkPermission(java.security.Permission var0) throws
java.security.AccessControlException { }
    public static java.security.AccessControlContext getContext() { return null; }
    public static java.lang.Object doPrivileged(java.security.PrivilegedAction var0) { return null; }
    public static java.lang.Object doPrivileged(java.security.PrivilegedAction var0,
java.security.AccessControlContext var1) { return null; }
    public static java.lang.Object doPrivileged(java.security.PrivilegedExceptionAction var0) throws
java.security.PrivilegedActionException { return null; }
    public static java.lang.Object doPrivileged(java.security.PrivilegedExceptionAction var0,
java.security.AccessControlContext var1) throws java.security.PrivilegedActionException { return null; }
}

package java.security;
public final class AllPermission extends java.security.Permission {
    public AllPermission() { super(null); }
    public AllPermission(java.lang.String var0, java.lang.String var1) { super(null); }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.String getActions() { return null; }
    public int hashCode() { return 0; }
    public boolean implies(java.security.Permission var0) { return false; }
    public java.security.PermissionCollection newPermissionCollection() { return null; }
}

package java.security;
public abstract class BasicPermission extends java.security.Permission implements java.io.Serializable {
    public BasicPermission(java.lang.String var0) { super(null); }
    public BasicPermission(java.lang.String var0, java.lang.String var1) { super(null); }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.String getActions() { return null; }
    public int hashCode() { return 0; }
    public boolean implies(java.security.Permission var0) { return false; }
    public java.security.PermissionCollection newPermissionCollection() { return null; }
}

package java.security.cert;
public abstract class Certificate implements java.io.Serializable {
    protected Certificate(java.lang.String var0) { }
}

package java.security;
public class CodeSource implements java.io.Serializable {
    public CodeSource(java.net.URL var0, java.security.cert.Certificate[] var1) { }
    public boolean equals(java.lang.Object var0) { return false; }
    public int hashCode() { return 0; }
    public final java.net.URL getLocation() { return null; }
    public boolean implies(java.security.CodeSource var0) { return false; }
    public java.lang.String toString() { return null; }
}

package java.security;
public abstract class Permission implements java.io.Serializable {
    public Permission(java.lang.String var0) { }
    public abstract boolean equals(java.lang.Object var0);
    public abstract int hashCode();
    public abstract java.lang.String getActions();
    public final java.lang.String getName() { return null; }
    public abstract boolean implies(java.security.Permission var0);
    public java.security.PermissionCollection newPermissionCollection() { return null; }
    public java.lang.String toString() { return null; }
}

package java.security;
public abstract class PermissionCollection implements java.io.Serializable {
    public PermissionCollection() { }
    public abstract void add(java.security.Permission var0);
    public abstract java.util.Enumeration elements();
    public abstract boolean implies(java.security.Permission var0);
    public boolean isReadOnly() { return false; }
    public void setReadOnly() { }
    public java.lang.String toString() { return null; }
}

package java.security;
public final class Permissions extends java.security.PermissionCollection implements
java.io.Serializable {
    public Permissions() { }
    public void add(java.security.Permission var0) { }
    public java.util.Enumeration elements() { return null; }
    public boolean implies(java.security.Permission var0) { return false; }
}

package java.security;
public abstract class Policy {
    public Policy() { }
    public static java.security.Policy getPolicy() { return null; }
    public static void setPolicy(java.security.Policy var0) { }
    public abstract java.security.PermissionCollection getPermissions(java.security.CodeSource var0);
    public abstract void refresh();
}

package java.security;
public abstract interface PrivilegedAction {
    public abstract java.lang.Object run();
}

package java.security;
public class PrivilegedActionException extends java.lang.Exception {
    public PrivilegedActionException(java.lang.Exception var0) { }
    public java.lang.Exception getException() { return null; }
    public void printStackTrace() { }
    public void printStackTrace(java.io.PrintStream var0) { }
    public void printStackTrace(java.io.PrintWriter var0) { }
    public java.lang.String toString() { return null; }
}

package java.security;
public abstract interface PrivilegedExceptionAction {
    public abstract java.lang.Object run() throws java.lang.Exception;
}

package java.security;
```

All Page Within This Box

```
public class ProtectionDomain {
    public ProtectionDomain(java.security.CodeSource var0, java.security.PermissionCollection var1) { }
    public final java.security.CodeSource getCodeSource() { return null; }
    public final java.security.PermissionCollection getPermissions() { return null; }
    public boolean implies(java.security.Permission var0) { return false; }
    public java.lang.String toString() { return null; }
}

package java.security;
public final class Security {
    private Security() { }
    public static java.lang.String getProperty(java.lang.String var0) { return null; }
    public static void setProperty(java.lang.String var0, java.lang.String var1) { }
}

package java.security;
public final class SecurityPermission extends java.security.BasicPermission {
    public SecurityPermission(java.lang.String var0) { super(null, null); }
    public SecurityPermission(java.lang.String var0, java.lang.String var1) { super(null, null); }
}

package java.security;
public final class UnresolvedPermission extends java.security.Permission implements java.io.Serializable
{
    public UnresolvedPermission(java.lang.String var0, java.lang.String var1, java.lang.String var2,
java.security.cert.Certificate[] var3) { super(null); }
    public boolean equals(java.lang.Object var0) { return false; }
    public boolean implies(java.security.Permission var0) { return false; }
    public java.security.PermissionCollection newPermissionCollection() { return null; }
    public java.lang.String getActions() { return null; }
    public int hashCode() { return 0; }
    public java.lang.String toString() { return null; }
}

package java.util;
public abstract class AbstractCollection implements java.util.Collection {
    protected AbstractCollection() { }
    public boolean add(java.lang.Object var0) { return false; }
    public boolean addAll(java.util.Collection var0) { return false; }
    public void clear() { }
    public boolean contains(java.lang.Object var0) { return false; }
    public boolean containsAll(java.util.Collection var0) { return false; }
    public boolean isEmpty() { return false; }
    public abstract java.util.Iterator iterator();
    public boolean remove(java.lang.Object var0) { return false; }
    public boolean removeAll(java.util.Collection var0) { return false; }
    public boolean retainAll(java.util.Collection var0) { return false; }
    public abstract int size();
    public java.lang.Object[] toArray() { return null; }
    public java.lang.Object[] toArray(java.lang.Object[] var0) { return null; }
    public java.lang.String toString() { return null; }
}

package java.util;
public abstract class AbstractList extends java.util.AbstractCollection implements java.util.List {
    protected AbstractList() { }
    public void add(int var0, java.lang.Object var1) { }
    public boolean add(java.lang.Object var0) { return false; }
    public boolean addAll(int var0, java.util.Collection var1) { return false; }
    public void clear() { }
    public boolean equals(java.lang.Object var0) { return false; }
    public abstract java.lang.Object get(int var0);
    public int hashCode() { return 0; }
    public int indexOf(java.lang.Object var0) { return 0; }
    public java.util.Iterator iterator() { return null; }
    public int lastIndexOf(java.lang.Object var0) { return 0; }
    public java.util.ListIterator listIterator() { return null; }
    public java.util.ListIterator listIterator(int var0) { return null; }
    public java.lang.Object remove(int var0) { return null; }
    protected void removeRange(int var0, int var1) { }
    public java.lang.Object set(int var0, java.lang.Object var1) { return null; }
    public java.util.List subList(int var0, int var1) { return null; }
    protected int modCount;
}

package java.util;
public abstract class AbstractMap implements java.util.Map {
    protected AbstractMap() { }
    public void clear() { }
    public boolean containsKey(java.lang.Object var0) { return false; }
    public boolean containsValue(java.lang.Object var0) { return false; }
    public abstract java.util.Set entrySet();
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.Object get(java.lang.Object var0) { return null; }
    public int hashCode() { return 0; }
    public boolean isEmpty() { return false; }
    public java.util.Set keySet() { return null; }
    public java.lang.Object put(java.lang.Object var0, java.lang.Object var1) { return null; }
    public void putAll(java.util.Map var0) { }
    public java.lang.Object remove(java.lang.Object var0) { return null; }
    public int size() { return 0; }
    public java.lang.String toString() { return null; }
    public java.util.Collection values() { return null; }
}

package java.util;
public abstract class AbstractSequentialList extends java.util.AbstractList {
    protected AbstractSequentialList() { }
    public void add(int var0, java.lang.Object var1) { }
    public boolean addAll(int var0, java.util.Collection var1) { return false; }
    public java.lang.Object get(int var0) { return null; }
    public java.util.Iterator iterator() { return null; }
    public abstract java.util.ListIterator listIterator(int var0);
    public java.lang.Object remove(int var0) { return null; }
    public java.lang.Object set(int var0, java.lang.Object var1) { return null; }
}

package java.util;
public abstract class AbstractSet extends java.util.AbstractCollection implements java.util.Set {
    protected AbstractSet() { }
    public boolean equals(java.lang.Object var0) { return false; }
    public int hashCode() { return 0; }
    public boolean removeAll(java.util.Collection var0) { return false; }
}

package java.util;
public class ArrayList extends java.util.AbstractList implements java.util.List, java.lang.Cloneable,
java.io.Serializable {
    public ArrayList() { }
    public ArrayList(int var0) { }
    public ArrayList(java.util.Collection var0) { }
    public void add(int var0, java.lang.Object var1) { }
    public boolean add(java.lang.Object var0) { return false; }
    public boolean addAll(int var0, java.util.Collection var1) { return false; }
    public boolean addAll(java.util.Collection var0) { return false; }
    public void clear() { }
    public java.lang.Object clone() { return null; }
    public boolean contains(java.lang.Object var0) { return false; }
    public void ensureCapacity(int var0) { }
    public java.lang.Object get(int var0) { return null; }
    public int indexOf(java.lang.Object var0) { return 0; }
    public boolean isEmpty() { return false; }
    public int lastIndexOf(java.lang.Object var0) { return 0; }
    public java.lang.Object remove(int var0) { return null; }
    protected void removeRange(int var0, int var1) { }
    public java.lang.Object set(int var0, java.lang.Object var1) { return null; }
    public int size() { return 0; }
    public java.lang.Object[] toArray() { return null; }
    public java.lang.Object[] toArray(java.lang.Object[] var0) { return null; }
    public void trimToSize() { }
}

package java.util;
public class Arrays {
    private Arrays() { }
    public static java.util.List asList(java.lang.Object[] var0) { return null; }
    public static int binarySearch(byte[] var0, byte var1) { return 0; }
    public static int binarySearch(char[] var0, char var1) { return 0; }
    public static int binarySearch(double[] var0, double var1) { return 0; }
    public static int binarySearch(float[] var0, float var1) { return 0; }
    public static int binarySearch(int[] var0, int var1) { return 0; }
    public static int binarySearch(long[] var0, long var1) { return 0; }
    public static int binarySearch(java.lang.Object[] var0, java.lang.Object var1) { return 0; }
    public static int binarySearch(java.lang.Object[] var0, java.lang.Object var1, java.util.Comparator
var2) { return 0; }
    public static int binarySearch(short[] var0, short var1) { return 0; }
    public static void fill(byte[] var0, byte var1) { }
    public static void fill(byte[] var0, int var1, int var2, byte var3) { }
    public static void fill(short[] var0, short var1) { }
    public static void fill(short[] var0, int var1, int var2, short var3) { }
    public static void fill(char[] var0, char var1) { }
    public static void fill(char[] var0, int var1, int var2, char var3) { }
    public static void fill(int[] var0, int var1) { }
    public static void fill(int[] var0, int var1, int var2, int var3) { }
    public static void fill(long[] var0, long var1) { }
    public static void fill(long[] var0, int var1, int var2, long var3) { }
    public static void fill(float[] var0, float var1) { }
    public static void fill(float[] var0, int var1, int var2, float var3) { }
    public static void fill(double[] var0, double var1) { }
    public static void fill(double[] var0, int var1, int var2, double var3) { }
    public static void fill(boolean[] var0, boolean var1) { }
    public static void fill(boolean[] var0, int var1, int var2, boolean var3) { }
    public static void fill(java.lang.Object[] var0, java.lang.Object var1) { }
    public static void fill(java.lang.Object[] var0, int var1, int var2, java.lang.Object var3) { }
    public static boolean equals(byte[] var0, byte[] var1) { return false; }
    public static boolean equals(short[] var0, short[] var1) { return false; }
    public static boolean equals(char[] var0, char[] var1) { return false; }
    public static boolean equals(int[] var0, int[] var1) { return false; }
    public static boolean equals(long[] var0, long[] var1) { return false; }
    public static boolean equals(float[] var0, float[] var1) { return false; }
    public static boolean equals(double[] var0, double[] var1) { return false; }
    public static boolean equals(boolean[] var0, boolean[] var1) { return false; }
    public static boolean equals(java.lang.Object[] var0, java.lang.Object[] var1) { return false; }
    public static void sort(byte[] var0) { }
    public static void sort(byte[] var0, int var1, int var2) { }
    public static void sort(char[] var0) { }
    public static void sort(char[] var0, int var1, int var2) { }
    public static void sort(double[] var0) { }
    public static void sort(double[] var0, int var1, int var2) { }
    public static void sort(float[] var0) { }
    public static void sort(float[] var0, int var1, int var2) { }
    public static void sort(int[] var0) { }
    public static void sort(int[] var0, int var1, int var2) { }
    public static void sort(long[] var0) { }
    public static void sort(long[] var0, int var1, int var2) { }
    public static void sort(java.lang.Object[] var0) { }
    public static void sort(java.lang.Object[] var0, int var1, int var2) { }
    public static void sort(java.lang.Object[] var0, int var1, int var2, java.util.Comparator var3) { }
    public static void sort(java.lang.Object[] var0, java.util.Comparator var1) { }
    public static void sort(short[] var0) { }
    public static void sort(short[] var0, int var1, int var2) { }
}

package java.util;
public abstract class Calendar implements java.io.Serializable, java.lang.Cloneable {
    protected Calendar() { }
    Calendar(java.util.TimeZone var0) { }
    public abstract void add(int var0, int var1);
    public boolean after(java.lang.Object var0) { return false; }
    public boolean before(java.lang.Object var0) { return false; }
    public final void clear() { }
    public final void clear(int var0) { }
    public java.lang.Object clone() { return null; }
    protected void complete() { }
    protected abstract void computeFields();
    protected abstract void computeTime();
    public boolean equals(java.lang.Object var0) { return false; }
    public final int get(int var0) { return 0; }
    public int getFirstDayOfWeek() { return 0; }
    public static java.util.Calendar getInstance() { return null; }
    public static java.util.Calendar getInstance(java.util.TimeZone var0) { return null; }
    public int getMinimalDaysInFirstWeek() { return 0; }
    public final java.util.Date getTime() { return null; }
    protected long getTimeInMillis() { return 0l; }
    public java.util.TimeZone getTimeZone() { return null; }
    public int hashCode() { return 0; }
    protected final int internalGet(int var0) { return 0; }
    public boolean isLenient() { return false; }
    public final boolean isSet(int var0) { return false; }
    public void roll(int var0, int var1) { }
    public abstract void roll(int var0, boolean var1);
    public final void set(int var0, int var1) { }
    public final void set(int var0, int var1, int var2) { }
    public final void set(int var0, int var1, int var2, int var3, int var4) { }
    public final void set(int var0, int var1, int var2, int var3, int var4, int var5) { }
    public void setFirstDayOfWeek(int var0) { }
    public void setLenient(boolean var0) { }
    public void setMinimalDaysInFirstWeek(int var0) { }
    public final void setTime(java.util.Date var0) { }
    protected void setTimeInMillis(long var0) { }
    public void setTimeZone(java.util.TimeZone var0) { }
    public java.lang.String toString() { return null; }
    protected boolean areFieldsSet;
    protected int[] fields;
    protected boolean[] isSet;
    protected boolean isTimeSet;
    protected long time;
    public final static int JANUARY = 0;
    public final static int FEBRUARY = 1;
    public final static int MARCH = 2;
    public final static int APRIL = 3;
    public final static int MAY = 4;
    public final static int JUNE = 5;
    public final static int JULY = 6;
    public final static int AUGUST = 7;
    public final static int SEPTEMBER = 8;
    public final static int OCTOBER = 9;
    public final static int NOVEMBER = 10;
    public final static int DECEMBER = 11;
    public final static int UNDECIMBER = 12;
    public final static int SUNDAY = 1;
    public final static int MONDAY = 2;
    public final static int TUESDAY = 3;
    public final static int WEDNESDAY = 4;
    public final static int THURSDAY = 5;
    public final static int FRIDAY = 6;
    public final static int SATURDAY = 7;
    public final static int ERA = 0;
    public final static int YEAR = 1;
    public final static int MONTH = 2;
    public final static int WEEK_OF_YEAR = 3;
    public final static int WEEK_OF_MONTH = 4;
    public final static int DATE = 5;
    public final static int DAY_OF_MONTH = 5;
    public final static int DAY_OF_YEAR = 6;
    public final static int DAY_OF_WEEK = 7;
    public final static int DAY_OF_WEEK_IN_MONTH = 8;
    public final static int AM_PM = 9;
    public final static int HOUR = 10;
    public final static int HOUR_OF_DAY = 11;
    public final static int MINUTE = 12;
    public final static int SECOND = 13;
    public final static int MILLISECOND = 14;
    public final static int ZONE_OFFSET = 15;
    public final static int DST_OFFSET = 16;
    public final static int FIELD_COUNT = 17;
    public final static int AM = 0;
    public final static int PM = 1;
}

package java.util;
public abstract interface Collection {
    public abstract boolean add(java.lang.Object var0);
    public abstract boolean addAll(java.util.Collection var0);
    public abstract void clear();
    public abstract boolean contains(java.lang.Object var0);
    public abstract boolean containsAll(java.util.Collection var0);
    public abstract boolean equals(java.lang.Object var0);
    public abstract int hashCode();
    public abstract boolean isEmpty();
    public abstract java.util.Iterator iterator();
    public abstract boolean remove(java.lang.Object var0);
    public abstract boolean removeAll(java.util.Collection var0);
    public abstract boolean retainAll(java.util.Collection var0);
    public abstract int size();
    public abstract java.lang.Object[] toArray();
    public abstract java.lang.Object[] toArray(java.lang.Object[] var0);
}

package java.util;
public class Collections {
    private Collections() { }
    public static int binarySearch(java.util.List var0, java.lang.Object var1) { return 0; }
    public static int binarySearch(java.util.List var0, java.lang.Object var1, java.util.Comparator
var2) { return 0; }
    public static void copy(java.util.List var0, java.util.List var1) { }
    public static java.util.Enumeration enumeration(java.util.Collection var0) { return null; }
    public static void fill(java.util.List var0, java.lang.Object var1) { }
    public static java.lang.Object max(java.util.Collection var0) { return null; }
    public static java.lang.Object max(java.util.Collection var0, java.util.Comparator var1) { return
null; }
    public static java.lang.Object min(java.util.Collection var0) { return null; }
    public static java.lang.Object min(java.util.Collection var0, java.util.Comparator var1) { return
null; }
    public static java.util.List nCopies(int var0, java.lang.Object var1) { return null; }
    public static void reverse(java.util.List var0) { }
    public static java.util.Comparator reverseOrder() { return null; }
    public static void shuffle(java.util.List var0) { }
    public static void shuffle(java.util.List var0, java.util.Random var1) { }
    public static java.util.Set singleton(java.lang.Object var0) { return null; }
    public static java.util.List singletonList(java.lang.Object var0) { return null; }
    public static java.util.Map singletonMap(java.lang.Object var0, java.lang.Object var1) { return
null; }
    public static void sort(java.util.List var0) { }
    public static void sort(java.util.List var0, java.util.Comparator var1) { }
    public static java.util.Collection synchronizedCollection(java.util.Collection var0) { return null;
}
    public static java.util.List synchronizedList(java.util.List var0) { return null; }
    public static java.util.Map synchronizedMap(java.util.Map var0) { return null; }
    public static java.util.Set synchronizedSet(java.util.Set var0) { return null; }
    public static java.util.SortedMap synchronizedSortedMap(java.util.SortedMap var0) { return null; }
    public static java.util.SortedSet synchronizedSortedSet(java.util.SortedSet var0) { return null; }
    public static java.util.Collection unmodifiableCollection(java.util.Collection var0) { return null;
}
    public static java.util.List unmodifiableList(java.util.List var0) { return null; }
    public static java.util.Map unmodifiableMap(java.util.Map var0) { return null; }
    public static java.util.Set unmodifiableSet(java.util.Set var0) { return null; }
    public static java.util.SortedMap unmodifiableSortedMap(java.util.SortedMap var0) { return null; }
    public static java.util.SortedSet unmodifiableSortedSet(java.util.SortedSet var0) { return null; }
    public final static java.util.List EMPTY_LIST; static { EMPTY_LIST = null; }
    public final static java.util.Set EMPTY_SET; static { EMPTY_SET = null; }
    public final static java.util.Map EMPTY_MAP; static { EMPTY_MAP = null; }
}

package java.util;
public abstract interface Comparator {
    public abstract int compare(java.lang.Object var0, java.lang.Object var1);
    public abstract boolean equals(java.lang.Object var0);
}

package java.util;
public class ConcurrentModificationException extends java.lang.RuntimeException {
    public ConcurrentModificationException() { }
    public ConcurrentModificationException(java.lang.String var0) { }
}

package java.util;
public class Date implements java.io.Serializable, java.lang.Cloneable, java.lang.Comparable {
    public Date() { }
    public Date(long var0) { }
    public boolean after(java.util.Date var0) { return false; }
    public boolean before(java.util.Date var0) { return false; }
    public int compareTo(java.lang.Object var0) { return 0; }
    public int compareTo(java.util.Date var0) { return 0; }
    public boolean equals(java.lang.Object var0) { return false; }
    public long getTime() { return 0l; }
    public int hashCode() { return 0; }
    public void setTime(long var0) { }
    public java.lang.String toString() { return null; }
}

package java.util;
public abstract class Dictionary {
    public Dictionary() { }
    public abstract java.util.Enumeration elements();
    public abstract java.lang.Object get(java.lang.Object var0);
    public abstract boolean isEmpty();
    public abstract java.util.Enumeration keys();
    public abstract java.lang.Object put(java.lang.Object var0, java.lang.Object var1);
    public abstract java.lang.Object remove(java.lang.Object var0);
    public abstract int size();
}

package java.util;
public abstract interface Enumeration {
    public abstract boolean hasMoreElements();
    public abstract java.lang.Object nextElement();
}

package java.util;
public abstract interface EventListener {
}

package java.util;
public class EventObject implements java.io.Serializable {
    public EventObject(java.lang.Object var0) { }
    public java.lang.Object getSource() { return null; }
    public java.lang.String toString() { return null; }
    protected java.lang.Object source;
}

package java.util;
public class GregorianCalendar extends java.util.Calendar {
    public GregorianCalendar() { }
    public GregorianCalendar(int var0, int var1, int var2) { }
    public GregorianCalendar(int var0, int var1, int var2, int var3, int var4) { }
    public GregorianCalendar(int var0, int var1, int var2, int var3, int var4, int var5) { }
    GregorianCalendar(long var0) { }
    public GregorianCalendar(java.util.TimeZone var0) { }
    GregorianCalendar(boolean var0) { }
    public void add(int var0, int var1) { }
    protected void computeFields() { }
    protected void computeTime() { }
    public boolean equals(java.lang.Object var0) { return false; }
    public final java.util.Date getGregorianChange() { return null; }
    public int hashCode() { return 0; }
    public boolean isLeapYear(int var0) { return false; }
    public void roll(int var0, int var1) { }
    public void roll(int var0, boolean var1) { }
    public void setGregorianChange(java.util.Date var0) { }
    public final static int BC = 0;
    public final static int AD = 1;
}

package java.util;
public class Hashtable extends java.util.Dictionary implements java.util.Map, java.lang.Cloneable,
java.io.Serializable {
    public Hashtable() { }
    public Hashtable(int var0) { }
    public Hashtable(int var0, float var1) { }
    public Hashtable(java.util.Map var0) { }
    public void clear() { }
    public java.lang.Object clone() { return null; }
    public boolean contains(java.lang.Object var0) { return false; }
    public boolean containsKey(java.lang.Object var0) { return false; }
    public boolean containsValue(java.lang.Object var0) { return false; }
    public java.util.Enumeration elements() { return null; }
    public java.util.Set entrySet() { return null; }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.Object get(java.lang.Object var0) { return null; }
    public int hashCode() { return 0; }
    public boolean isEmpty() { return false; }
    public java.util.Enumeration keys() { return null; }
    public java.util.Set keySet() { return null; }
    public java.lang.Object put(java.lang.Object var0, java.lang.Object var1) { return null; }
    public void putAll(java.util.Map var0) { }
    protected void rehash() { }
    public java.lang.Object remove(java.lang.Object var0) { return null; }
    public int size() { return 0; }
    public java.lang.String toString() { return null; }
    public java.util.Collection values() { return null; }
}

package java.util;
public abstract interface Iterator {
    public abstract boolean hasNext();
    public abstract java.lang.Object next();
    public abstract void remove();
}

package java.util;
public abstract interface List extends java.util.Collection {
    public abstract void add(int var0, java.lang.Object var1);
    public abstract boolean add(java.lang.Object var0);
    public abstract boolean addAll(int var0, java.util.Collection var1);
    public abstract boolean addAll(java.util.Collection var0);
    public abstract void clear();
    public abstract boolean contains(java.lang.Object var0);
    public abstract boolean containsAll(java.util.Collection var0);
    public abstract boolean equals(java.lang.Object var0);
    public abstract java.lang.Object get(int var0);
    public abstract int hashCode();
    public abstract int indexOf(java.lang.Object var0);
    public abstract boolean isEmpty();
    public abstract java.util.Iterator iterator();
    public abstract int lastIndexOf(java.lang.Object var0);
    public abstract java.util.ListIterator listIterator();
    public abstract java.util.ListIterator listIterator(int var0);
    public abstract java.lang.Object remove(int var0);
    public abstract boolean remove(java.lang.Object var0);
    public abstract boolean removeAll(java.util.Collection var0);
    public abstract boolean retainAll(java.util.Collection var0);
    public abstract java.lang.Object set(int var0, java.lang.Object var1);
    public abstract int size();
    public abstract java.util.List subList(int var0, int var1);
    public abstract java.lang.Object[] toArray();
    public abstract java.lang.Object[] toArray(java.lang.Object[] var0);
}

package java.util;
public abstract interface ListIterator extends java.util.Iterator {
    public abstract void add(java.lang.Object var0);
    public abstract boolean hasNext();
    public abstract boolean hasPrevious();
    public abstract java.lang.Object next();
    public abstract int nextIndex();
    public abstract java.lang.Object previous();
    public abstract int previousIndex();
    public abstract void remove();
    public abstract void set(java.lang.Object var0);
}

package java.util;
public abstract class ListResourceBundle extends java.util.ResourceBundle {
    public ListResourceBundle() { }
    protected abstract java.lang.Object[][] getContents();
    public java.util.Enumeration getKeys() { return null; }
    public final java.lang.Object handleGetObject(java.lang.String var0) { return null; }
}

package java.util;
public final class Locale implements java.lang.Cloneable, java.io.Serializable {
    public Locale(java.lang.String var0, java.lang.String var1) { }
    public Locale(java.lang.String var0, java.lang.String var1, java.lang.String var2) { }
    public java.lang.Object clone() { return null; }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.String getCountry() { return null; }
    public static java.util.Locale getDefault() { return null; }
    public java.lang.String getLanguage() { return null; }
    public java.lang.String getVariant() { return null; }
    public int hashCode() { return 0; }
    public static void setDefault(java.util.Locale var0) { }
    public final java.lang.String toString() { return null; }
    public final static java.util.Locale CANADA; static { CANADA = null; }
    public final static java.util.Locale CANADA_FRENCH; static { CANADA_FRENCH = null; }
    public final static java.util.Locale CHINA; static { CHINA = null; }
    public final static java.util.Locale CHINESE; static { CHINESE = null; }
    public final static java.util.Locale ENGLISH; static { ENGLISH = null; }
    public final static java.util.Locale FRANCE; static { FRANCE = null; }
    public final static java.util.Locale FRENCH; static { FRENCH = null; }
    public final static java.util.Locale GERMAN; static { GERMAN = null; }
    public final static java.util.Locale GERMANY; static { GERMANY = null; }
    public final static java.util.Locale ITALIAN; static { ITALIAN = null; }
    public final static java.util.Locale ITALY; static { ITALY = null; }
    public final static java.util.Locale JAPAN; static { JAPAN = null; }
    public final static java.util.Locale JAPANESE; static { JAPANESE = null; }
    public final static java.util.Locale KOREA; static { KOREA = null; }
    public final static java.util.Locale KOREAN; static { KOREAN = null; }
    public final static java.util.Locale PRC; static { PRC = null; }
    public final static java.util.Locale SIMPLIFIED_CHINESE; static { SIMPLIFIED_CHINESE = null; }
    public final static java.util.Locale TAIWAN; static { TAIWAN = null; }
    public final static java.util.Locale TRADITIONAL_CHINESE; static { TRADITIONAL_CHINESE = null; }
    public final static java.util.Locale UK; static { UK = null; }
    public final static java.util.Locale US; static { US = null; }
}

package java.util;
public abstract interface Map {
    public abstract void clear();
    public abstract boolean containsKey(java.lang.Object var0);
    public abstract boolean containsValue(java.lang.Object var0);
    public abstract java.util.Set entrySet();
    public abstract boolean equals(java.lang.Object var0);
    public abstract java.lang.Object get(java.lang.Object var0);
    public abstract int hashCode();
    public abstract boolean isEmpty();
    public abstract java.util.Set keySet();
    public abstract java.lang.Object put(java.lang.Object var0, java.lang.Object var1);
    public abstract void putAll(java.util.Map var0);
    public abstract java.lang.Object remove(java.lang.Object var0);
    public abstract int size();
    public abstract java.util.Collection values();
    public abstract static interface Entry {
        public abstract boolean equals(java.lang.Object var0);
        public abstract java.lang.Object getKey();
        public abstract java.lang.Object getValue();
        public abstract int hashCode();
        public abstract java.lang.Object setValue(java.lang.Object var0);
    }
}

package java.util;
public class MissingResourceException extends java.lang.RuntimeException {
    public MissingResourceException(java.lang.String var0, java.lang.String var1, java.lang.String var2)
{ }
    public java.lang.String getClassName() { return null; }
    public java.lang.String getKey() { return null; }
}

package java.util;
public class NoSuchElementException extends java.lang.RuntimeException {
    public NoSuchElementException() { }
    public NoSuchElementException(java.lang.String var0) { }
}

package java.util;
public class Properties extends java.util.Hashtable {
    public Properties() { }
    public Properties(java.util.Properties var0) { }
    public java.lang.String getProperty(java.lang.String var0) { return null; }
    public java.lang.String getProperty(java.lang.String var0, java.lang.String var1) { return null; }
    public void list(java.io.PrintStream var0) { }
    public void load(java.io.InputStream var0) throws java.io.IOException { }
    public java.util.Enumeration propertyNames() { return null; }
    public void store(java.io.OutputStream var0, java.lang.String var1) throws java.io.IOException { }
    protected java.util.Properties defaults;
}

package java.util;
public final class PropertyPermission extends java.security.BasicPermission {
    public PropertyPermission(java.lang.String var0, java.lang.String var1) { super(null, null); }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.String getActions() { return null; }
    public int hashCode() { return 0; }
    public boolean implies(java.security.Permission var0) { return false; }
    public java.security.PermissionCollection newPermissionCollection() { return null; }
}

package java.util;
public class PropertyResourceBundle extends java.util.ResourceBundle {
    public PropertyResourceBundle(java.io.InputStream var0) throws java.io.IOException { }
```

All Page Within This Box

```
    public java.util.Enumeration getKeys() { return null; }
    public java.lang.Object handleGetObject(java.lang.String var0) { return null; }
}

package java.util;
public class Random implements java.io.Serializable {
    public Random() { }
    public Random(long var0) { }
    protected int next(int var0) { return 0; }
    public boolean nextBoolean() { return false; }
    public void nextBytes(byte[] var0) { }
    public double nextDouble() { return 0.0d; }
    public float nextFloat() { return 0.0f; }
    public double nextGaussian() { return 0.0d; }
    public int nextInt() { return 0; }
    public int nextInt(int var0) { return 0; }
    public long nextLong() { return 0l; }
    public void setSeed(long var0) { }
}

package java.util;
public abstract class ResourceBundle {
    public ResourceBundle() { }
    public final static java.util.ResourceBundle getBundle(java.lang.String var0) throws
java.util.MissingResourceException { return null; }
    public final static java.util.ResourceBundle getBundle(java.lang.String var0, java.util.Locale var1)
{ return null; }
    public static java.util.ResourceBundle getBundle(java.lang.String var0, java.util.Locale var1,
java.lang.ClassLoader var2) throws java.util.MissingResourceException { return null; }
    public abstract java.util.Enumeration getKeys();
    public java.util.Locale getLocale() { return null; }
    public final java.lang.Object getObject(java.lang.String var0) throws
java.util.MissingResourceException { return null; }
    public final java.lang.String getString(java.lang.String var0) throws
java.util.MissingResourceException { return null; }
    public final java.lang.String[] getStringArray(java.lang.String var0) throws
java.util.MissingResourceException { return null; }
    protected abstract java.lang.Object handleGetObject(java.lang.String var0) throws
java.util.MissingResourceException;
    protected void setParent(java.util.ResourceBundle var0) { }
    protected java.util.ResourceBundle parent;
}

package java.util;
public abstract interface Set extends java.util.Collection {
    public abstract boolean add(java.lang.Object var0);
    public abstract boolean addAll(java.util.Collection var0);
    public abstract void clear();
    public abstract boolean contains(java.lang.Object var0);
    public abstract boolean containsAll(java.util.Collection var0);
    public abstract boolean equals(java.lang.Object var0);
    public abstract int hashCode();
    public abstract boolean isEmpty();
    public abstract java.util.Iterator iterator();
    public abstract boolean remove(java.lang.Object var0);
    public abstract boolean removeAll(java.util.Collection var0);
    public abstract boolean retainAll(java.util.Collection var0);
    public abstract int size();
    public abstract java.lang.Object[] toArray();
    public abstract java.lang.Object[] toArray(java.lang.Object[] var0);
}

package java.util;
public class SimpleTimeZone extends java.util.TimeZone {
    public SimpleTimeZone(int var0, java.lang.String var1) { }
    public SimpleTimeZone(int var0, java.lang.String var1, int var2, int var3, int var4, int var5, int
var6, int var7, int var8, int var9) { }
    public SimpleTimeZone(int var0, java.lang.String var1, int var2, int var3, int var4, int var5, int
var6, int var7, int var8, int var9, int var10) { }
```

```
    public java.lang.Object clone() { return null; }
    public boolean equals(java.lang.Object var0) { return false; }
    public int getOffset(int var0, int var1, int var2, int var3, int var4, int var5) { return 0; }
    public int getRawOffset() { return 0; }
    public int hashCode() { return 0; }
    public boolean inDaylightTime(java.util.Date var0) { return false; }
    public void setEndRule(int var0, int var1, int var2) { }
    public void setEndRule(int var0, int var1, int var2, int var3) { }
    public void setEndRule(int var0, int var1, int var2, int var3, boolean var4) { }
    public void setRawOffset(int var0) { }
    public void setStartRule(int var0, int var1, int var2) { }
    public void setStartRule(int var0, int var1, int var2, int var3) { }
    public void setStartRule(int var0, int var1, int var2, int var3, boolean var4) { }
    public void setStartYear(int var0) { }
    public java.lang.String toString() { return null; }
    public boolean useDaylightTime() { return false; }
}

package java.util;
public abstract interface SortedMap extends java.util.Map {
    public abstract java.util.Comparator comparator();
    public abstract java.lang.Object firstKey();
    public abstract java.util.SortedMap headMap(java.lang.Object var0);
    public abstract java.lang.Object lastKey();
    public abstract java.util.SortedMap subMap(java.lang.Object var0, java.lang.Object var1);
    public abstract java.util.SortedMap tailMap(java.lang.Object var0);
}

package java.util;
public abstract interface SortedSet extends java.util.Set {
    public abstract java.util.Comparator comparator();
    public abstract java.lang.Object first();
    public abstract java.util.SortedSet headSet(java.lang.Object var0);
    public abstract java.lang.Object last();
    public abstract java.util.SortedSet subSet(java.lang.Object var0, java.lang.Object var1);
    public abstract java.util.SortedSet tailSet(java.lang.Object var0);
}

package java.util;
public class StringTokenizer implements java.util.Enumeration {
    public StringTokenizer(java.lang.String var0) { }
    public StringTokenizer(java.lang.String var0, java.lang.String var1) { }
    public StringTokenizer(java.lang.String var0, java.lang.String var1, boolean var2) { }
    public int countTokens() { return 0; }
    public boolean hasMoreElements() { return false; }
    public boolean hasMoreTokens() { return false; }
    public java.lang.Object nextElement() { return null; }
    public java.lang.String nextToken() { return null; }
    public java.lang.String nextToken(java.lang.String var0) { return null; }
}

package java.util;
public abstract class TimeZone implements java.io.Serializable, java.lang.Cloneable {
    public TimeZone() { }
    public java.lang.Object clone() { return null; }
    public static java.lang.String[] getAvailableIDs() { return null; }
    public static java.lang.String[] getAvailableIDs(int var0) { return null; }
    public static java.util.TimeZone getDefault() { return null; }
    public java.lang.String getID() { return null; }
    public abstract int getOffset(int var0, int var1, int var2, int var3, int var4, int var5);
    public abstract int getRawOffset();
    public static java.util.TimeZone getTimeZone(java.lang.String var0) { return null; }
    public abstract boolean inDaylightTime(java.util.Date var0);
    public static void setDefault(java.util.TimeZone var0) { }
    public void setID(java.lang.String var0) { }
    public abstract void setRawOffset(int var0);
    public abstract boolean useDaylightTime();
}
```

```
package java.util;
public class Vector extends java.util.AbstractList implements java.util.List, java.lang.Cloneable,
java.io.Serializable {
    public Vector() { }
    public Vector(int var0) { }
    public Vector(int var0, int var1) { }
    public Vector(java.util.Collection var0) { }
    public void add(int var0, java.lang.Object var1) { }
    public boolean add(java.lang.Object var0) { return false; }
    public boolean addAll(int var0, java.util.Collection var1) { return false; }
    public boolean addAll(java.util.Collection var0) { return false; }
    public void addElement(java.lang.Object var0) { }
    public int capacity() { return 0; }
    public void clear() { }
    public java.lang.Object clone() { return null; }
    public boolean contains(java.lang.Object var0) { return false; }
    public boolean containsAll(java.util.Collection var0) { return false; }
    public void copyInto(java.lang.Object[] var0) { }
    public java.lang.Object elementAt(int var0) { return null; }
    public java.util.Enumeration elements() { return null; }
    public void ensureCapacity(int var0) { }
    public boolean equals(java.lang.Object var0) { return false; }
    public java.lang.Object firstElement() { return null; }
    public java.lang.Object get(int var0) { return null; }
    public int hashCode() { return 0; }
    public int indexOf(java.lang.Object var0) { return 0; }
    public int indexOf(java.lang.Object var0, int var1) { return 0; }
    public void insertElementAt(java.lang.Object var0, int var1) { }
    public boolean isEmpty() { return false; }
    public java.lang.Object lastElement() { return null; }
    public int lastIndexOf(java.lang.Object var0) { return 0; }
    public int lastIndexOf(java.lang.Object var0, int var1) { return 0; }
    public java.lang.Object remove(int var0) { return null; }
    public boolean remove(java.lang.Object var0) { return false; }
    public boolean removeAll(java.util.Collection var0) { return false; }
    public void removeAllElements() { }
    public boolean removeElement(java.lang.Object var0) { return false; }
    public void removeElementAt(int var0) { }
    protected void removeRange(int var0, int var1) { }
    public boolean retainAll(java.util.Collection var0) { return false; }
    public java.lang.Object set(int var0, java.lang.Object var1) { return null; }
    public void setElementAt(java.lang.Object var0, int var1) { }
    public void setSize(int var0) { }
    public int size() { return 0; }
    public java.util.List subList(int var0, int var1) { return null; }
    public java.lang.Object[] toArray() { return null; }
    public java.lang.Object[] toArray(java.lang.Object[] var0) { return null; }
    public java.lang.String toString() { return null; }
    public void trimToSize() { }
    protected int elementCount;
    protected java.lang.Object[] elementData;
    protected int capacityIncrement;
}

package java.util;
public class WeakHashMap extends java.util.AbstractMap implements java.util.Map {
    public WeakHashMap() { }
    public WeakHashMap(int var0) { }
    public WeakHashMap(int var0, float var1) { }
    public WeakHashMap(java.util.Map var0) { }
    public void clear() { }
    public boolean containsKey(java.lang.Object var0) { return false; }
    public java.util.Set entrySet() { return null; }
    public java.lang.Object get(java.lang.Object var0) { return null; }
    public boolean isEmpty() { return false; }
    public java.lang.Object put(java.lang.Object var0, java.lang.Object var1) { return null; }
    public java.lang.Object remove(java.lang.Object var0) { return null; }
    public int size() { return 0; }
}
```

```
package java.util.zip;
public abstract interface Checksum {
    public abstract long getValue();
    public abstract void reset();
    public abstract void update(int var0);
    public abstract void update(byte[] var0, int var1, int var2);
}

package java.util.zip;
public class CRC32 implements java.util.zip.Checksum {
    public CRC32() { }
    public long getValue() { return 0l; }
    public void reset() { }
    public void update(int var0) { }
    public void update(byte[] var0) { }
    public void update(byte[] var0, int var1, int var2) { }
}

package java.util.zip;
public class DataFormatException extends java.lang.Exception {
    public DataFormatException() { }
    public DataFormatException(java.lang.String var0) { }
}

package java.util.zip;
public class Inflater {
    public void end() { }
    protected void finalize() { }
    public boolean finished() { return false; }
    public int getAdler() { return 0; }
    public int getRemaining() { return 0; }
    public int getTotalIn() { return 0; }
    public int getTotalOut() { return 0; }
    public int inflate(byte[] var0) throws java.util.zip.DataFormatException { return 0; }
    public int inflate(byte[] var0, int var1, int var2) throws java.util.zip.DataFormatException {
return 0; }
    public Inflater() { }
    public Inflater(boolean var0) { }
    public boolean needsDictionary() { return false; }
    public boolean needsInput() { return false; }
    public void reset() { }
    public void setDictionary(byte[] var0) { }
    public void setDictionary(byte[] var0, int var1, int var2) { }
    public void setInput(byte[] var0) { }
    public void setInput(byte[] var0, int var1, int var2) { }
}

package java.util.zip;
public class InflaterInputStream extends java.io.FilterInputStream {
    public InflaterInputStream(java.io.InputStream var0) { super(null); }
    public InflaterInputStream(java.io.InputStream var0, java.util.zip.Inflater var1) { super(null); }
    public InflaterInputStream(java.io.InputStream var0, java.util.zip.Inflater var1, int var2) {
super(null); }
    public int read() throws java.io.IOException { return 0; }
    public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
    protected void fill() throws java.io.IOException { }
    public long skip(long var0) throws java.io.IOException { return 0l; }
    public int available() throws java.io.IOException { return 0; }
    public void close() throws java.io.IOException { }
    protected java.util.zip.Inflater inf;
    protected byte[] buf;
    protected int len;
}

package java.util.zip;
abstract interface ZipConstants {
    public final static long LOCSIG = 67324752l;
    public final static long EXTSIG = 134695760l;
    public final static long CENSIG = 33639248l;
    public final static long ENDSIG = 101010256l;
    public final static int LOCHDR = 30;
    public final static int EXTHDR = 16;
    public final static int CENHDR = 46;
    public final static int ENDHDR = 22;
    public final static int LOCVER = 4;
    public final static int LOCFLG = 6;
    public final static int LOCHOW = 8;
    public final static int LOCTIM = 10;
    public final static int LOCCRC = 14;
    public final static int LOCSIZ = 18;
    public final static int LOCLEN = 22;
    public final static int LOCNAM = 26;
    public final static int LOCEXT = 28;
    public final static int EXTCRC = 4;
    public final static int EXTSIZ = 8;
    public final static int EXTLEN = 12;
    public final static int CENVEM = 4;
    public final static int CENVER = 6;
    public final static int CENFLG = 8;
    public final static int CENHOW = 10;
    public final static int CENTIM = 12;
    public final static int CENCRC = 16;
    public final static int CENSIZ = 20;
    public final static int CENLEN = 24;
    public final static int CENNAM = 28;
    public final static int CENEXT = 30;
    public final static int CENCOM = 32;
    public final static int CENDSK = 34;
    public final static int CENATT = 36;
    public final static int CENATX = 38;
    public final static int CENOFF = 42;
    public final static int ENDSUB = 8;
    public final static int ENDTOT = 10;
    public final static int ENDSIZ = 12;
    public final static int ENDOFF = 16;
    public final static int ENDCOM = 20;
}

package java.util.zip;
public class ZipEntry implements java.util.zip.ZipConstants, java.lang.Cloneable {
    public ZipEntry(java.lang.String var0) { }
    public java.lang.String getComment() { return null; }
    public long getCompressedSize() { return 0l; }
    public long getCrc() { return 0l; }
    public byte[] getExtra() { return null; }
    public int getMethod() { return 0; }
    public java.lang.String getName() { return null; }
    public long getSize() { return 0l; }
    public long getTime() { return 0l; }
    public boolean isDirectory() { return false; }
    public void setComment(java.lang.String var0) { }
    public void setCompressedSize(long var0) { }
    public void setCrc(long var0) { }
    public void setExtra(byte[] var0) { }
    public void setMethod(int var0) { }
    public void setSize(long var0) { }
    public void setTime(long var0) { }
    public java.lang.String toString() { return null; }
    ZipEntry(java.lang.String var0, java.lang.String var1, byte[] var2, long var3, long var4, long var5,
long var6, int var7, long var8, long var9) { }
    public ZipEntry(java.util.zip.ZipEntry var0) { }
    public java.lang.Object clone() { return null; }
    public int hashCode() { return 0; }
    public final static int DEFLATED = 8;
    public final static int STORED = 0;
}
```

```
package java.util.zip;
public class ZipException extends java.io.IOException {
    public ZipException() { }
    public ZipException(java.lang.String var0) { }
}

package java.util.zip;
public class ZipFile implements java.util.zip.ZipConstants {
    public ZipFile(java.io.File var0) throws java.util.zip.ZipException, java.io.IOException { }
    public ZipFile(java.lang.String var0) throws java.io.IOException { }
    protected void finalize() throws java.io.IOException { }
    public void close() throws java.io.IOException { }
    public java.util.Enumeration entries() { return null; }
    public java.util.zip.ZipEntry getEntry(java.lang.String var0) { return null; }
    public java.io.InputStream getInputStream(java.util.zip.ZipEntry var0) throws java.io.IOException {
return null; }
    public java.lang.String getName() { return null; }
    public int size() { return 0; }
}

package java.util.zip;
public class ZipInputStream extends java.util.zip.InflaterInputStream implements
java.util.zip.ZipConstants {
    public ZipInputStream(java.io.InputStream var0) { super(null, null, 0); }
    public void close() throws java.io.IOException { }
    public void closeEntry() throws java.io.IOException { }
    public java.util.zip.ZipEntry getNextEntry() throws java.io.IOException { return null; }
    public int read(byte[] var0, int var1, int var2) throws java.io.IOException { return 0; }
    public long skip(long var0) throws java.io.IOException { return 0l; }
    public int available() throws java.io.IOException { return 0; }
    protected java.util.zip.ZipEntry createZipEntry(java.lang.String var0) { return null; }
}
```

3.2 Character Converters

The following character converters must be supported. The first name is the Java canonical name[5] for the character converter. The other names are aliases that should also be supported. The aliases come from common Java usage and the IANA[6]. The case of the name for the character converter is not important.

- ISO8859_1 (aliases: 8859_1, ISO8859-1, ISO-8859-1, ISO_8859-1, ISO_8859-1:1978, ISO_8859-1:1987, iso-ir-100, latin1, l1, csISOlatin1)
- UTF8 (alias: UTF-8)
- ASCII (aliases: US-ASCII, ISO646-US)

3.3 URL Protocols

The following URL protocols must be supported.

- http
- file
- ftp
- jar

4 Security Considerations

This document does have any security considerations.

5 Document Support

5.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Hargrave, BJ. et al., OSGi RFC 6, A Technique for Defining a Minimum Execution Environment for OSGi Bundles

[3]. Bowen, D, OSGi RFC 26, A Minimal Execution Environment

[4]. OSGi Core Platform Expert Group, OSGi Service Platform Specification Release 2.0.

[5]. Java 2 1.3 Supported Encodings and their Canonical Names, http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.html

[6]. IANA Charset Registry, http://www.iana.org/assignments/character-sets

5.2 Author's Address

Name	Rick DeNatale
Company	OTI
Address	601-109 Hutton Street, Raleigh NC 27606 USA
Voice	+1 919 821 3220
e-mail	Rick_DeNatale@oti.com

All Page Within This Box

Name	BJ Hargrave
Company	IBM
Address	11400 Burnet Rd, Austin, TX 78758 USA
Voice	+1 512 838 9938
e-mail	hargrave@us.ibm.com

5.3 Acronyms and Abbreviations

J2SE	Java 2 Standard Edition
J2ME	Java 2 Micro Edition
IANA	Internet Assigned Numbers Authority

5.4 End of Document