Module java.base

Package java.io

`package java.io`

Provides for system input and output through data streams, serialization and the file system. Unless otherwise noted, passing a `null` argument to a constructor or method in any class or interface in this package will cause a `NullPointerException` to be thrown. A *pathname string* passed as a `String` argument to a constructor or method in any class or interface in this package will be interpreted as described in the class specification of `File`.

Object Serialization

Warning: Deserialization of untrusted data is inherently dangerous and should be avoided. Untrusted data should be carefully validated according to the "Serialization and Deserialization" section of the Secure Coding Guidelines for Java SE.

- *Java Object Serialization Specification*
- Serial Filtering best practices
- The serialver tool

Since:

1.0

External Specifications

Java Object Serialization Specification

All Classes and Interfaces	Interfaces	Classes	Enum Classes	Exception Classes	Annotation Interfaces

Class	Description
BufferedInputStream	A `BufferedInputStream` adds functionality to another input stream-namely, the ability to buffer the input and to support the `mark` and `reset` methods.
BufferedOutputStream	The class implements a buffered output stream.
BufferedReader	Reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.

BufferedWriter	Writes text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.
ByteArrayInputStream	A `ByteArrayInputStream` contains an internal buffer that contains bytes that may be read from the stream.
ByteArrayOutputStream	This class implements an output stream in which the data is written into a byte array.
CharArrayReader	This class implements a character buffer that can be used as a character-input stream.
CharArrayWriter	This class implements a character buffer that can be used as a Writer.
CharConversionException	Base class for character conversion exceptions.
Closeable	A `Closeable` is a source or destination of data that can be closed.
Console	Methods to access the character-based console device, if any, associated with the current Java virtual machine.
DataInput	The `DataInput` interface provides for reading bytes from a binary stream and reconstructing from them data in any of the Java primitive types.
DataInputStream	A data input stream lets an application read primitive Java data types from an underlying input stream in a machine-independent way.
DataOutput	The `DataOutput` interface provides for converting data from any of the Java primitive types to a series of bytes and writing these bytes to a binary stream.
DataOutputStream	A data output stream lets an application write primitive Java data types to an output stream in a portable way.
EOFException	Signals that an end of file or end of stream has been reached unexpectedly during input.
Externalizable	Only the identity of the class of an Externalizable instance is written in the serialization stream and it is the responsibility of the class to save and restore the contents of its instances.
File	An abstract representation of file and directory pathnames.

FileDescriptor	Instances of the file descriptor class serve as an opaque handle to the underlying machine-specific structure representing an open file, an open socket, or another source or sink of bytes.
FileFilter	A filter for abstract pathnames.
FileInputStream	A `FileInputStream` obtains input bytes from a file in a file system.
FilenameFilter	Instances of classes that implement this interface are used to filter filenames.
FileNotFoundException	Signals that an attempt to open the file denoted by a specified pathname has failed.
FileOutputStream	A file output stream is an output stream for writing data to a `File` or to a `FileDescriptor`.
FilePermission	This class represents access to a file or directory.
FileReader	Reads text from character files using a default buffer size.
FileWriter	Writes text to character files using a default buffer size.
FilterInputStream	A `FilterInputStream` wraps some other input stream, which it uses as its basic source of data, possibly transforming the data along the way or providing additional functionality.
FilterOutputStream	This class is the superclass of all classes that filter output streams.
FilterReader	Abstract class for reading filtered character streams.
FilterWriter	Abstract class for writing filtered character streams.
Flushable	A `Flushable` is a destination of data that can be flushed.
InputStream	This abstract class is the superclass of all classes representing an input stream of bytes.
InputStreamReader	An InputStreamReader is a bridge from byte streams to character streams: It reads bytes and decodes them into characters using a specified `charset`.
InterruptedIOException	Signals that an I/O operation has been interrupted.
InvalidClassException	Thrown when the Serialization runtime detects one of the following problems with a Class.

InvalidObjectException	Indicates that one or more deserialized objects failed validation tests.
IOError	Thrown when a serious I/O error has occurred.
IOException	Signals that an I/O exception of some sort has occurred.
LineNumberInputStream	Deprecated. *This class incorrectly assumes that bytes adequately represent characters.*
LineNumberReader	A buffered character-input stream that keeps track of line numbers.
NotActiveException	Thrown when serialization or deserialization is not active.
NotSerializableException	Thrown when an instance is required to have a Serializable interface.
ObjectInput	ObjectInput extends the DataInput interface to include the reading of objects.
ObjectInputFilter	Filter classes, array lengths, and graph metrics during deserialization.
ObjectInputFilter.Config	A utility class to set and get the JVM-wide deserialization filter factory, the static JVM-wide filter, or to create a filter from a pattern string.
ObjectInputFilter.FilterInfo	FilterInfo provides access to information about the current object being deserialized and the status of the `ObjectInputStream`.
ObjectInputFilter.Status	The status of a check on the class, array length, number of references, depth, and stream size.
ObjectInputStream	An ObjectInputStream deserializes primitive data and objects previously written using an ObjectOutputStream.
ObjectInputStream.GetField	Provide access to the persistent fields read from the input stream.
ObjectInputValidation	Callback interface to allow validation of objects within a graph.
ObjectOutput	ObjectOutput extends the DataOutput interface to include writing of objects.
ObjectOutputStream	An ObjectOutputStream writes primitive data types and graphs of Java objects to an OutputStream.
ObjectOutputStream.PutField	Provide programmatic access to the persistent fields to be written to ObjectOutput.

ObjectStreamClass	Serialization's descriptor for classes.
ObjectStreamConstants	Constants written into the Object Serialization Stream.
ObjectStreamException	Superclass of all exceptions specific to Object Stream classes.
ObjectStreamField	A description of a Serializable field from a Serializable class.
OptionalDataException	Exception indicating the failure of an object read operation due to unread primitive data, or the end of data belonging to a serialized object in the stream.
OutputStream	This abstract class is the superclass of all classes representing an output stream of bytes.
OutputStreamWriter	An OutputStreamWriter is a bridge from character streams to byte streams: Characters written to it are encoded into bytes using a specified `charset`.
PipedInputStream	A piped input stream should be connected to a piped output stream; the piped input stream then provides whatever data bytes are written to the piped output stream.
PipedOutputStream	A piped output stream can be connected to a piped input stream to create a communications pipe.
PipedReader	Piped character-input streams.
PipedWriter	Piped character-output streams.
PrintStream	A `PrintStream` adds functionality to another output stream, namely the ability to print representations of various data values conveniently.
PrintWriter	Prints formatted representations of objects to a text-output stream.
PushbackInputStream	A `PushbackInputStream` adds functionality to another input stream, namely the ability to "push back" or "unread" bytes, by storing pushed-back bytes in an internal buffer.
PushbackReader	A character-stream reader that allows characters to be pushed back into the stream.
RandomAccessFile	Instances of this class support both reading and writing to a random access file.
Reader	Abstract class for reading character streams.

SequenceInputStream	A `SequenceInputStream` represents the logical concatenation of other input streams.
Serial	Indicates that an annotated field or method is part of the serialization mechanism defined by the *Java Object Serialization Specification*.
Serializable	Serializability of a class is enabled by the class implementing the java.io.Serializable interface.
SerializablePermission	This class is for Serializable permissions.
StreamCorruptedException	Thrown when control information that was read from an object stream violates internal consistency checks.
StreamTokenizer	The `StreamTokenizer` class takes an input stream and parses it into "tokens", allowing the tokens to be read one at a time.
StringBufferInputStream	Deprecated. *This class does not properly convert characters into bytes.*
StringReader	A character stream whose source is a string.
StringWriter	A character stream that collects its output in a string buffer, which can then be used to construct a string.
SyncFailedException	Signals that a sync operation has failed.
UncheckedIOException	Wraps an `IOException` with an unchecked exception.
UnsupportedEncodingException	The Character Encoding is not supported.
UTFDataFormatException	Signals that a malformed string in modified UTF-8 format has been read in a data input stream or by any class that implements the data input interface.
WriteAbortedException	Signals that one of the ObjectStreamExceptions was thrown during a write operation.
Writer	Abstract class for writing to character streams.

overviews, definitions of terms, workarounds, and working code examples. Other versions.