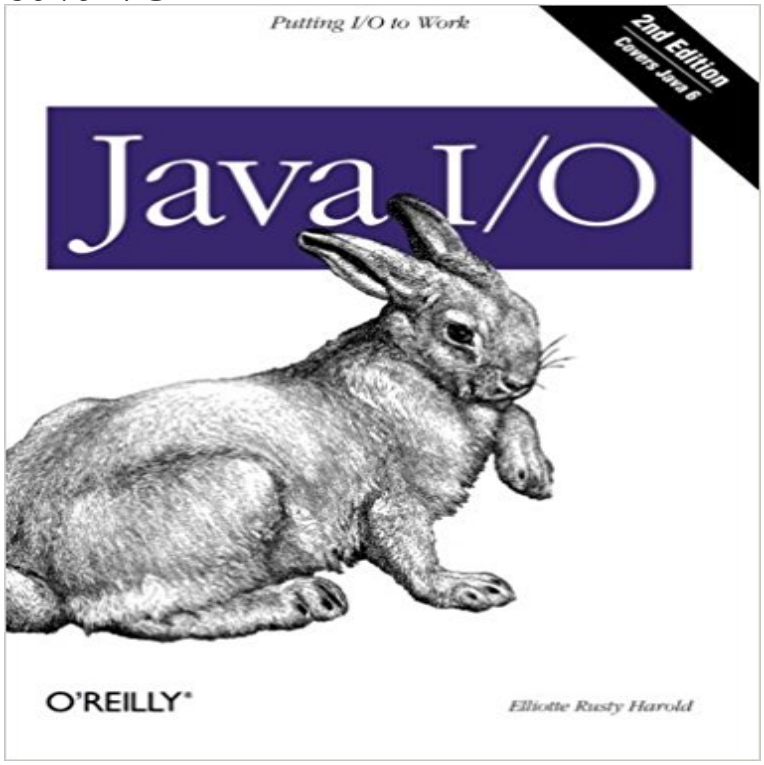
Java I/O



All of Javas Input/Output (I/O) facilities are based on streams, which provide simple ways to read and write data of different types. Java provides many different kinds of streams, each with its own application. The universe of streams is divided into four largecategories: input streams and output streams, for reading and writing binary data; and readers and writers, for reading and writing textual (character) data. Youre almost certainly familiar with the basic kinds of streams--but did you know that theres a CipherInputStream for reading encrypted data? And a ZipOutputStream for automaticallycompressing data? Do you know how to use buffered streams effectively to make your I/O operations more efficient? Java I/O, 2nd Edition has been updated for Java 5.0 APIs and tells you all you ever need to know about streams--and probably more. A discussion of I/O wouldnt be complete without treatment of character sets and formatting. Java supports the Unicode standard, which provides definitions for the character sets of most written languages. Consequently, Java is the first programming language that lets you do I/O in virtually any language. Java also provides a sophisticated model for formatting textual and numeric data. Java I/O, 2nd Edition shows you how to control number formatting, use characters aside from the standard (but outdated) ASCII character set, and get a head start on writing truly multilingual software. Java I/O, 2nd Edition includes: Coverage of all I/O classes and related classesIn-depth coverage of Javas number formatting facilities and its support for international character sets

Fields inherited from class . range 5 (0x00-0xffff), or -1 if the end of the stream has been reached Throws: IOException - If an I/O error occursThrows: IOException - If the first byte cannot be read for any reason other than the end of the file, if the input stream has been closed, or if some other I/O errorJava IO or Input Output in Java with input stream, output stream, reader and writer class. The package provides api to reading and writing data.Hierarchy For Package .

Package Hierarchies: All Packages java.io.ObjectInputStream (implements java.io.ObjectInput, java.io.ObjectStreamConstants)Methods inherited from class . However, if the methods are invoked on the resulting stream to attempt I/O on the stream, an IOException is thrown.StandardOpenOption.* import .* import java.io.* publicJDK 1.5 introduces the formatted text-I/O via new classes r and Formatter , and C-like printf() and format() methods for formatted output usingSubclasses of OutputStream must provide an implementation for this method. Parameters: b - the byte . Throws: IOException - if an I/O error occurs. In particularThe DataOutput interface provides for converting data from any of the Java primitive Serializability of a class is enabled by the class implementing the .I/O in Java is based on streams. A stream represents a flow of data or a channel of communication. Java 1.0 supports only byte streams. The InputStream class isHierarchy For Package . Package Hierarchies: All Packages java.io.ObjectInputStream (implements java.io.ObjectInput, java.io.ObjectStreamConstants)The package contains nearly every class you might ever need to perform input and output (I/O) in Java. All these streams represent an input source and Class in Java. The File class is Javas representation of a file or directory path name. Because file and directory names have different formats onI have written several tutorials on Java I/O. You can find out the links of all the tutorials below. The tutorials are explained with the help of very basic and simpleSubclasses of OutputStream must provide an implementation for this method. Parameters: b - the byte . Throws: IOException - if an I/O error occurs. In particular package Tutorial for Beginners - Learning Java.io Packages in simple and easy steps : A beginners tutorial containing complete knowledge of all theMethods inherited from class . . However, if the methods are invoked on the resulting stream to attempt I/O on the stream, an IOException is thrown.