9. Streams in Java

Lesson 2:
Standard Streams and classes for input and output – Part 1

9.3. Classes for Input and Output

9.3.1. Introduction

- The package **java.io** contains the classes that provide the foundation for Java's support for stream I/O. As a whole they form a reasonably logical structure.

- At the top of this logical structure are the **InputStream** and **OutputStream** classes.

- InputStream and OutputStream are both **abstract** classes. You cannot create instances of an abstract class—these classes serve only as a base from which to derive classes with more concrete input or output capabilities.

- Classes InputStream and OutputStream declare methods that
- define a basic set of operations for the streams they represent. The fundamental characteristics of how a stream is accessed are set by these classes.

9.3. Classes for Input and Output

9.3.2. Basic Input Stream Operations

- Methods **read()**, **read(byte[] array)** and **read(byte[] array,int offset,int length)**, read data. he data is read from the stream by these methods simply as bytes. No conversion is applied to the bytes read.

- You can skip over bytes in an InputStream by calling its **skip()** method.

- You can close an InputStream by calling its **close()** method. Once you have closed an input stream, subsequent attempts to access or read from the stream will cause an IOException to be thrown because the close() operation will have released the resources held by the stream object, including the source of the data, such as a file.

9.3. Classes for Input and Output
9.3.3. Subclasses of the InputStream class

- AudioInputStream
  - This is used for reading audio data
- SequenceInputStream
  - This is used for reading from a sequence of streams
- FileInputStream
  - This is used for reading from a file
- ByteArrayInputStream
  - This is used for
- ObjectInputStream
  - This is used for reading objects from a stream
- PipedInputStream
  - This is used for reading from a piped stream
- FilterInputStream
  - This is used for filtering input from an existing stream
  - Classes BufferedInputStream and DataInputStream are subclasses of the FilterInputStream class.

9.3. Classes for Input and Output

9.3.4. BufferedInputStream

- You can create a BufferedInputStream object from any other input stream, since the constructor accepts a reference of type InputStream as an argument. The BufferedInputStream class overrides the methods inherited from InputStream.

- For example, in the following code the argument System.in is the static member of the System class that encapsulates input from the keyboard and is of type InputStream.

  – BufferedInputStream keyboard = new BufferedInputStream(System.in);

- The effect of wrapping a stream in a BufferedInputStream object is to buffer the underlying stream in memory so that data can be read from the stream in large chunks—up to the size of the buffer that is provided. The data is then made available to the read() methods directly from memory, and a real read operation from the underlying stream is executed only when the buffer is empty.

9.3. Classes for Input and Output

9.3.5. Basic Output Stream Operations

- The OutputStream class contains three **write(…)** methods for writing binary data to the stream.
  - As can be expected, these mirror the read() methods of the InputStream class.

- You can close an InputStream by calling its **close()** method.

9.3. Classes for Input and Output
9.3.6. Subclasses of the OutputStream class

- FileOutputStream
  - This is used for writing to a file
- ByteArrayOutputStream
  - This is used for writing to a byte array
- ObjectOutputStream
  - This is used for writing objects to a stream
- PipeOutputStream
  - This is used for writing to a piped stream
- FilterOutputStream
  - This is used for filtering output from and existing stream

9.3. Classes for Input and Output

9.3.7. Readers and Writers

- Stream readers and writers are objects that can read and write byte streams as character streams.
- So a character stream is essentially a byte stream fronted by a reader or a writer.
- The base classes for stream readers and writers are:
  - Reader
    - This is the base class for reading a character stream
  - Writer
    - This is the base class for writing a character stream\
- Reader and Writer are both abstract classes.

9.3. Classes for Input and Output
9.3.8. Subclasses of the Reader class

- InputStreamReader
  - This is used for reading a character stream
- PipedReader
  - This is used for reading from a PipedWriter
- BufferedReader
  - This is used for buffering other readers
- CharArrayReader
  - This is used for reading from a char array
- FilterReader
  - This is used for reading filtered streams
- StringReader
  - This is used for reading from a string