

21/7/2020 Java Fundamentals → Input/output.

Initializers -

Four base classes:

- Reader & Writer

- InputStream & OutputStream

- Creating File object doesn't create anything on disk.

- Test the Existence/nature of file.

Two Utility classes:

- File
- Path

normalize() → removes redundant elements. Files is Same (path1, path2)

to get Path → `file.getAbsolutePath()`

Unique for file ← `file.getCanonicalPath()`

to `AbsolutePath()`

to `RealPath()` → resolves symbolic links.

Relativizing → about finding

a path from Source to

If one path target is absolute & other is relative then
IllegalArgument Exception is raised.

Reader → abstract class
↳ defines basic operations
↳ Reading of single character/ array of characters
→ Making & resetting a given position to throw it to caller

Strategies of handling Exception → handle it locally

System Resources must be closed → try-catch block
try-catch block reader.close() should be used in separate block

mark() → puts a flag on given element
reset() → rewinds to previously marked position of stream

skip() → skips next elements

BufferedReader, LineNumberReader → all behavior to Reader
↳ allows reading of elements from file
↳ reads readLine()

Base classes of Reading & Writing Bytes

1) Input Stream → reading byte by byte / array of bytes

2) Output Stream → writing

↳ writing primitive types → int, float & objects
↳ called as Serialization

Compressed Streams

Gzip format → allows for

Zip format → allows for

Two limitations on Gzip format

↳ it can compress one file at a time

↳ there may be limit on file size

Zip format → more complex

two classes for JAR format

↳ JARInputStream

↳ JAROutputStream

Only instances of Serializable classes can be

Serialized → Add special state field, Serializable

↳ Implement Serializable Interface (no method)

Serialization / DeSerialization

Used to Exchange an object across network

↳ is flexible enough to deserialize objects in

same class

3 ways to override default serialization mechanisms:

- ↳ pair of `writeObject()` / `readObject()` methods.
- ↳ Implements `Externalizable` Interface.
- ⇒ `writeReplace()` & `readResolve()` methods.

Hybrid Streams → `InputStreamReader` extends `Reader`

`OutputStreamWriter` extends `Writer`

Sockets → pure Java IO code to read & write bytes.