PROGRAMACIÓN ORIENTADA A OBJETOS

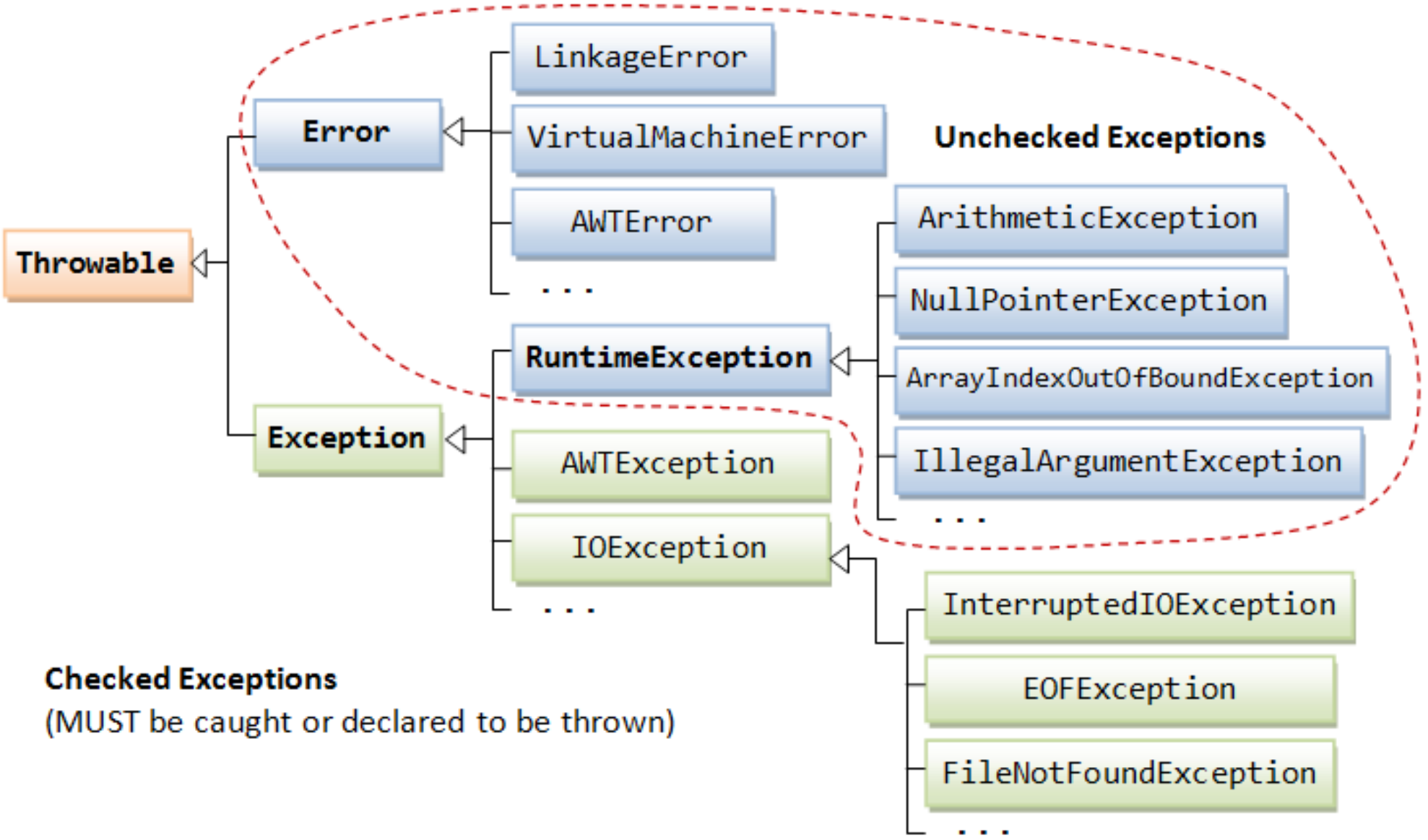
TEMA8: Excepciones y Entrada/Salida

Manel Guerrero

Tipos de Excepciones

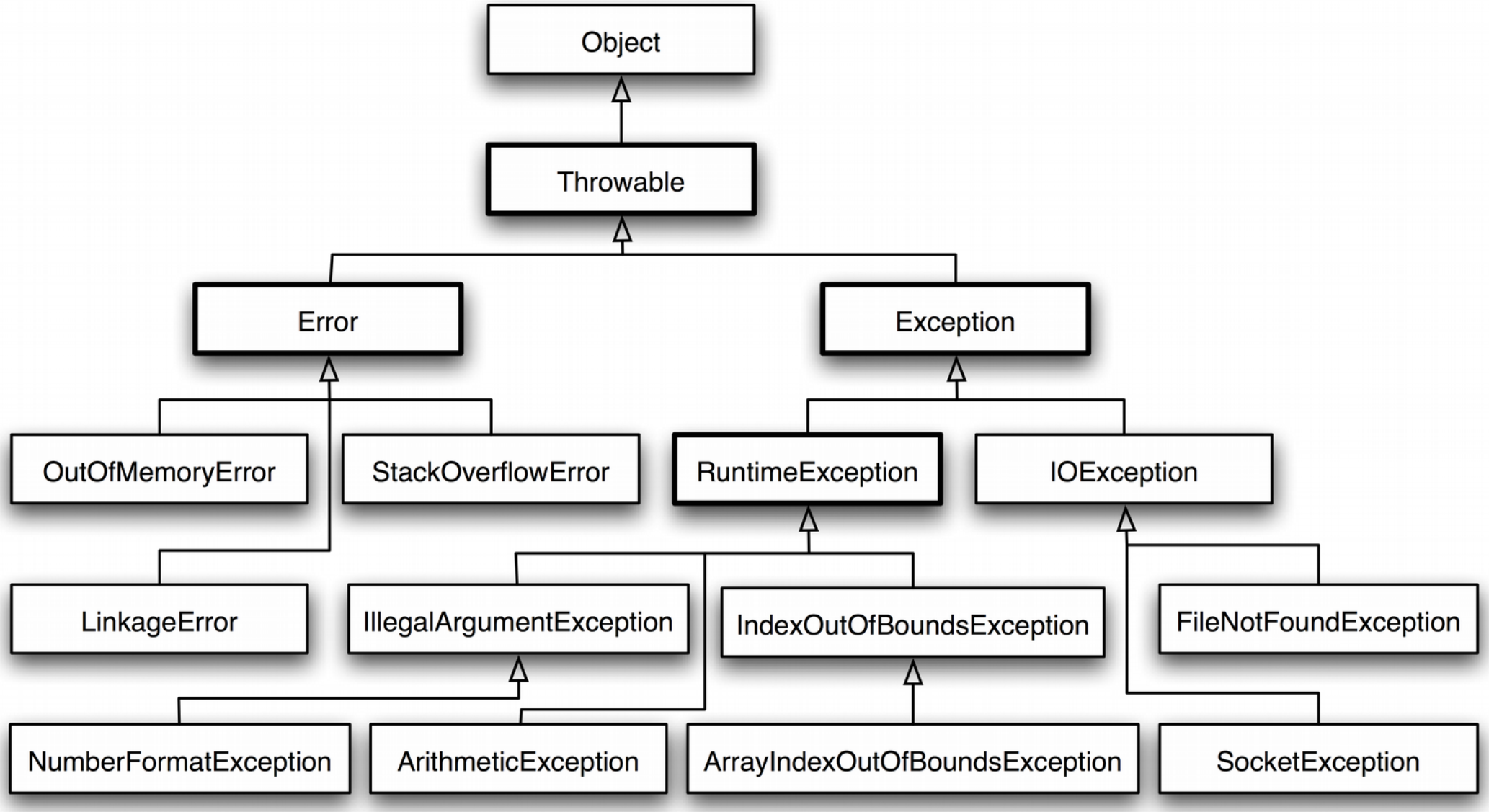
- **Checked Exception**: The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions e.g.IOException, SQLException etc. Checked exceptions are checked at compile-time.
- **Unchecked Exception**: The classes that extend RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time rather they are checked at runtime.
- **Error**: Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

  http://www.javatpoint.com/exception-handling-in-java



Checked Exceptions
(MUST be caught or declared to be thrown)

https://www3.ntu.edu.sg/home/ehchua/programming/java/J5a_ExceptionAssert.html



https://newcircle.com/bookshelf/java_fundamentals_tutorial/exceptions

Try-with-resources

- Try-with-resources closes the resources: Resource implements
  - public interface AutoCloseable {...}

  and the catch invokes its only method:
  - void close() throws Exception;

Entrada/Salida

- The java.io 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 an output destination. The stream in the java.io package supports many data such as primitives, object, localized characters, etc.
- **Stream**: A stream can be defined as a sequence of data.
  - **InPutStream**: The InputStream is used to read data from a source.

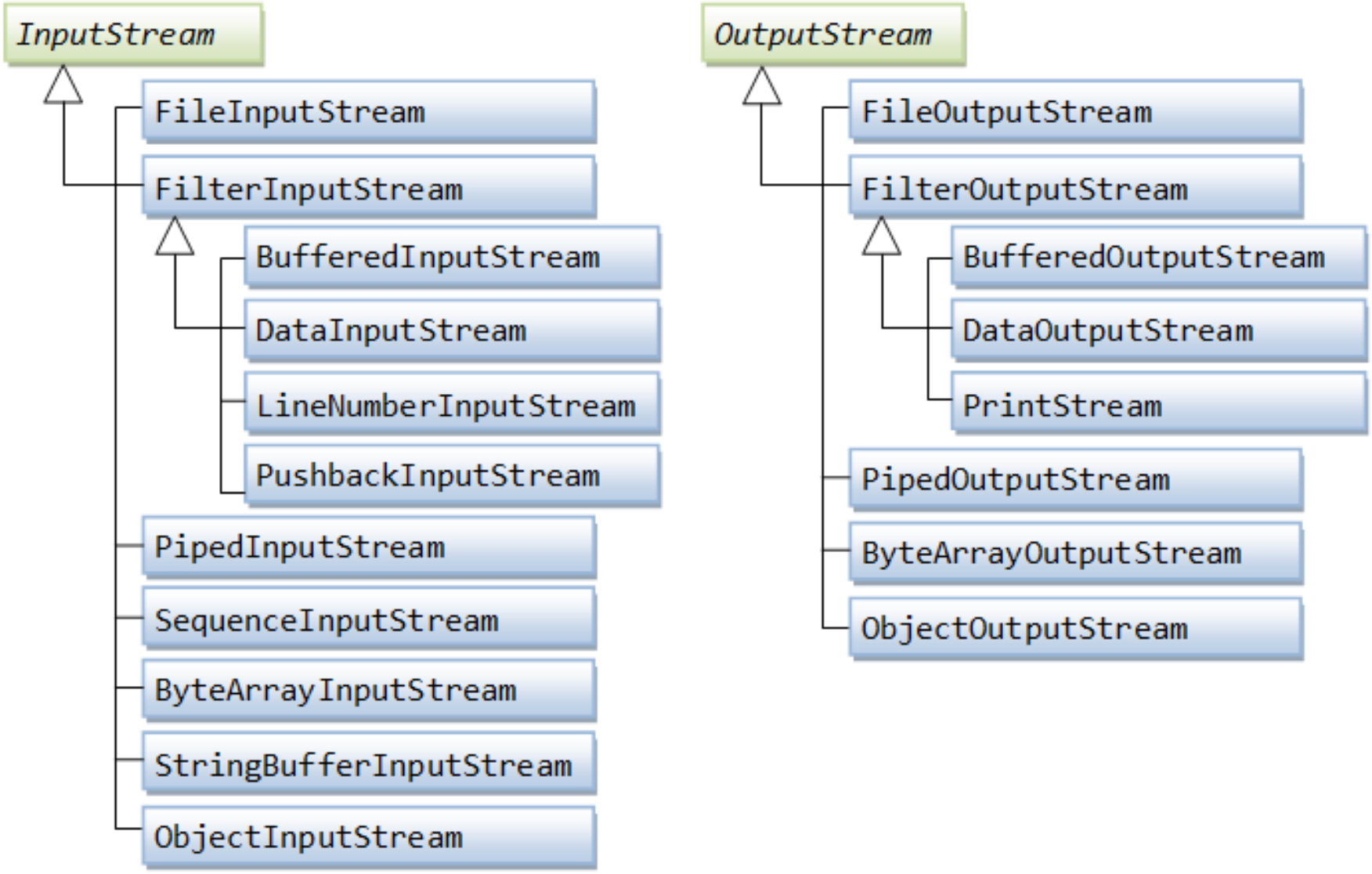
    OutPutStream: The OutputStream is used for writing data to a destination.

  https://www.tutorialspoint.com/java/java_files_io.htm

Byte Streams

- Java byte streams are used to perform input and output of 8-bit bytes. Though there are many classes related to byte streams but the most frequently used classes are, **FileInputStream** and **FileOutputStream**.
- Veure StreamBytesCopyFile.java i StreamBytesCopyFileV2.java

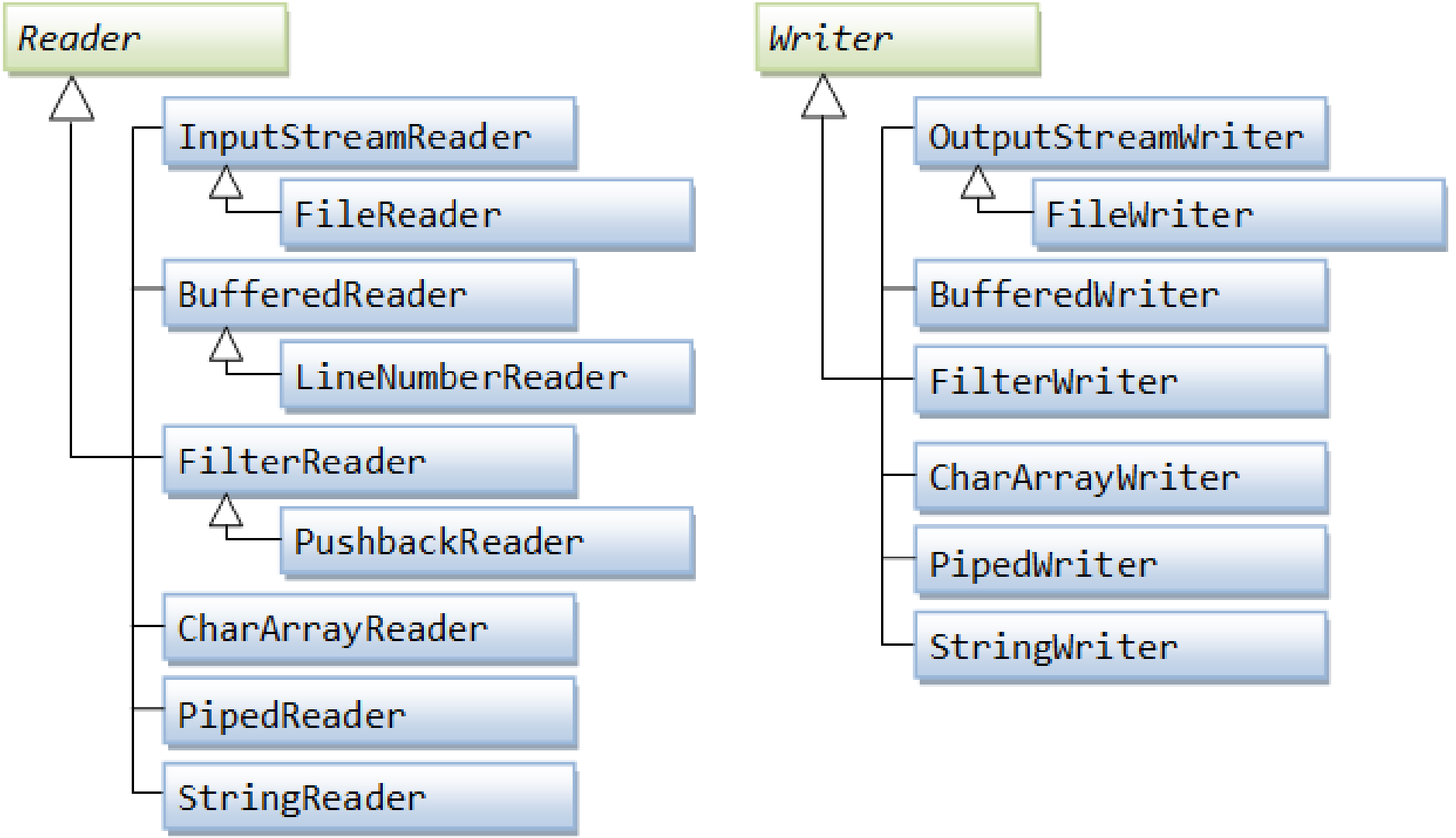
Byte Streams



Character Streams

- Java Byte streams are used to perform input and output of 8-bit bytes, whereas Java Character streams are used to perform input and output for 16-bit unicode. Though there are many classes related to character streams but the most frequently used classes are, **FileReader** and **FileWriter**. Though internally FileReader uses FileInputStream and FileWriter uses FileOutputStream but here the major difference is that FileReader reads two bytes at a time and FileWriter writes two bytes at a time.
- Veure StreamCharsCopyFile.java

Character Streams



Standard Streams

- **Standard Input** − This is used to feed the data to user's program and usually a keyboard is used as standard input stream and represented as **System.in**. [STDIN in C]
- **Standard Output** − This is used to output the data produced by the user's program and usually a computer screen is used for standard output stream and represented as **System.out**. [STDOUT in C]
- **Standard Error** − This is used to output the error data produced by the user's program and usually a computer screen is used for standard error stream and represented as **System.err**. [STDERR in C]
- Veure StreamConsole.java

BufferedReader

- En la asignatura usamos BufferedReader (que se construye con un InputStreamReader) y que permite leer líneas de texto enteras.
- Aunque la clase Scanner (también se construye con un InputStreamReader) es más potente.
- http://beginnersbook.com/2014/01/how-to-read-file-in-java-using-bufferedreader/
- Veure WithBufferedReader.java

PrintWriter

- En la asignatura usamos PrintWriter (que se construye con un FileOutputStreamr) para escribir en un fichero tipos primitivos y Strings.

```
FileOutputStream fos = new FileOutputStream(nomFitxer);
PrintWriter pw = new PrintWriter(fos);
```

- Luego podemos escribir en él con print() y println() o con write().
- http://beginnersbook.com/2014/01/how-to-append-to-a-file-in-java/
- http://stackoverflow.com/questions/32797719/printwriter-and-fileoutputstream-in-java
- Veure WithPrintWriter.java (similar pero amb File(nomF)).

ObjectInputStream
ObjectOutputStream

- Sirven para gravar objetos directamente (serializar/des-serializar).
- Se graba una “copia profunda” (versus “copia superficial”): Los objetos referenciados por el objeto o por objetos referenciados por él también se graban.
- Esto lo veremos en el próximo tema.

Referencias

- https://www.tutorialspoint.com/java/java_exceptions.htm
- http://tutorials.jenkov.com/java-exception-handling/fail-safe-exception-handling.html
- http://www.javatpoint.com/exception-handling-in-java
- https://www.tutorialspoint.com/java/java_files_io.htm
- http://www3.ntu.edu.sg/home/ehchua/programming/java/j5b_io.html
- http://beginnersbook.com/2014/01/how-to-read-file-in-java-using-bufferedreader/
- http://beginnersbook.com/2014/01/how-to-append-to-a-file-in-java/
- http://stackoverflow.com/questions/32797719/printwriter-and-fileoutputstream-in-java

PROGRAMACIÓN ORIENTADA A OBJETOS

Pasemos a ver los códigos de W8.

PROGRAMACIÓN ORIENTADA A OBJETOS

Preguntas