

Working with Java I/O in Java SE Applications

Introducing I/O Support in Java



José Paumard

PHD, Java Champion, JavaOne RockStar

@JosePaumard <https://github.com/JosePaumard>

Understanding what you need
to know to pass the
Java SE 11 Developer Certification 1Z0-819



How to read and write data

From outside the Java Virtual Machine

- **bytes and chars**

- **outside: disk, memory, network**

Covers the Java SE 11 Certification



This is a Java course:

**Basic knowledge of the language
and its main API**

The IDE is IntelliJ

The Collection framework, Stream API

The File System

Agenda



Introducing the structure of I/O support

Introducing Files and Paths

Reading and writing characters

Reading and writing bytes

Using the Console object

How is the Java I/O Support is Structured?

Accessing Files, Network and Memory

Java I/O

Java 1 (1995)

Java NIO

Java 4 (2002)

Java NIO 2

Java 7 (2011)



Java I/O is about accessing data:

- Reading, Writing

Where is the data?

- on a disk, in the memory

What is the data?

- characters, bytes



Reader



InputStream



Writer



OutputStream



Reader



InputStream



Writer



OutputStream

Reading Characters From a File

The Four Fundamentals Classes
Are Abstract



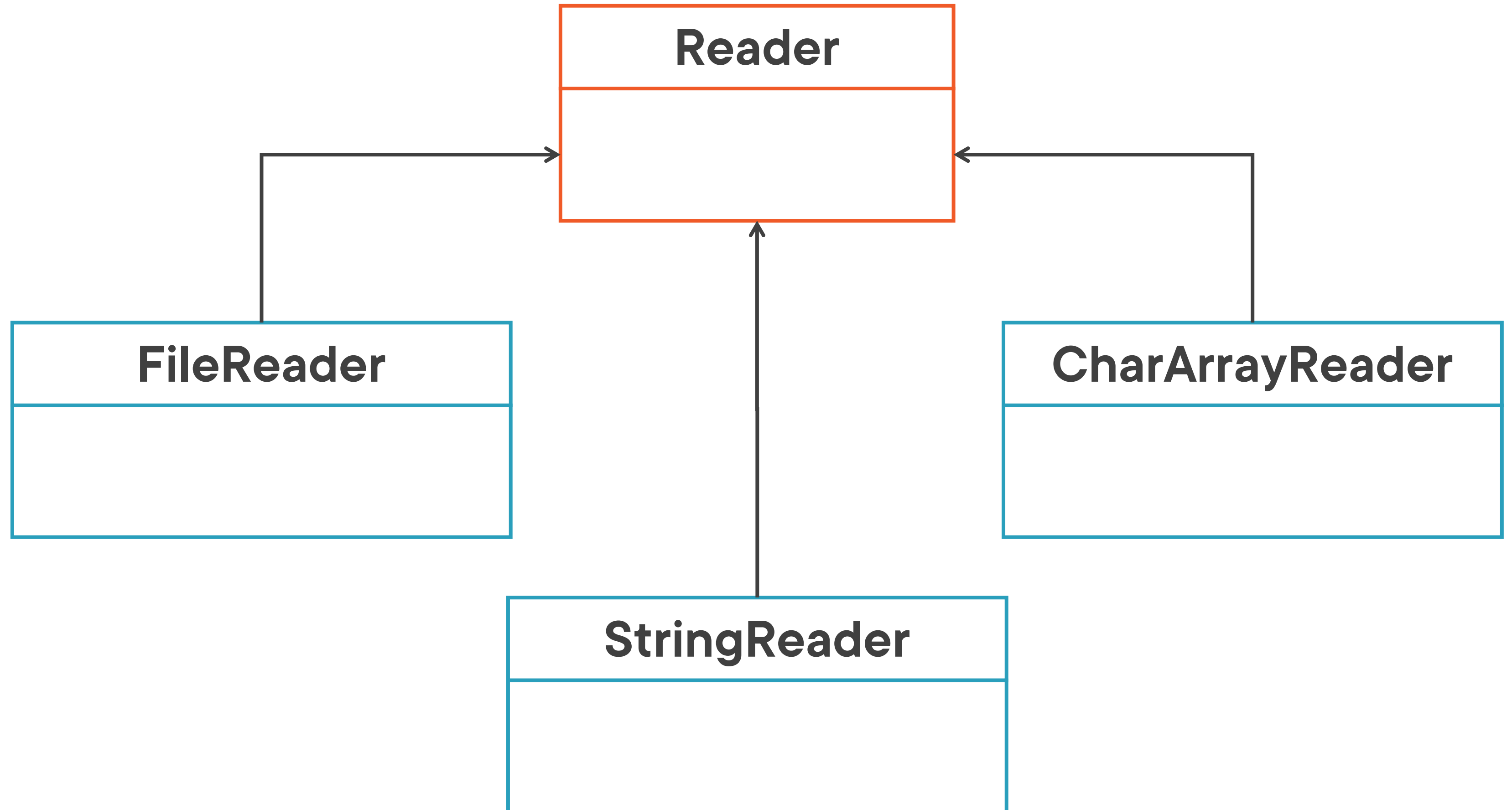
A Reader knows how to read characters

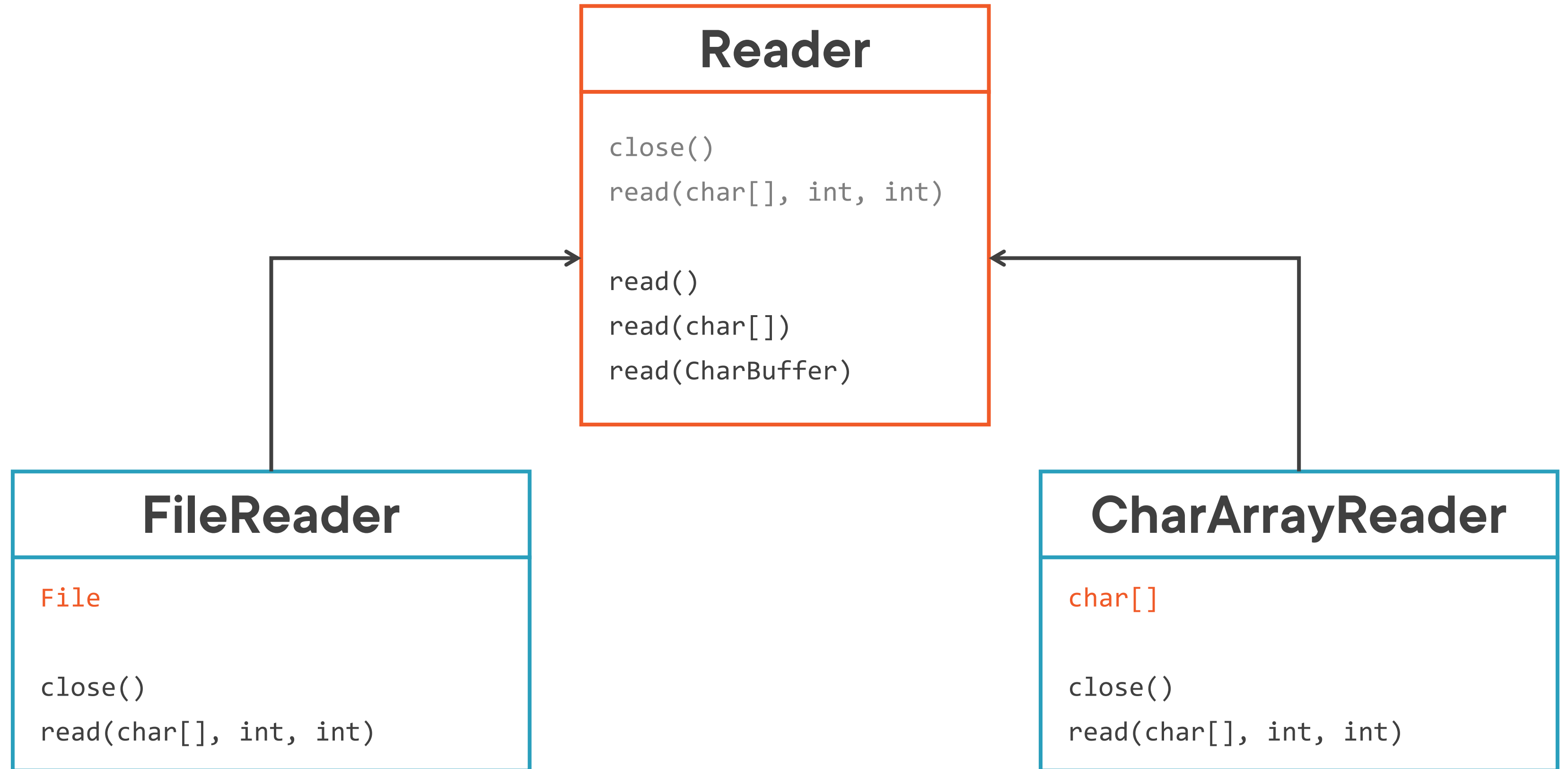
From any source

- a file on a disk
- a network
- memory

But it does not know the source!

The abstraction is the source





Reader

`close()`

`read(char[], int, int)`

`read()`

`read(char[])`

`read(CharBuffer)`



So, to create a Reader, you need:

- a File to create a FileReader
- a char[] to create a CharArrayReader
- a String to create a StringReader

Module Wrap Up



What did you learn?

How I/O support is organized:

- **Java I/O, NIO, NIO2**

How Java I/O is organized:

- **Reader and Writer**

- **InputStream and OutputStream**

Up Next: File and Path
