Java Fundamentals

Section 6 - I/O: Working with files

Topics

- Java.io.File class
- Java.nio.Path class
- Writing into files
- Reading from files

Working with files

- java.io package - original I/O implementations
- java.nio package - new classes for high speed I/O operations, network oriented, performance improvements

java.io.File class

File class

- The File class gives us the ability to work with files and directories on the file system:
  - Creating files, directories
  - Deleting files, directories
  - Setting file permissions
  - Listing files inside a directory
  - Renaming files and directories

java.nio.Path class

Path class

- Alternative to File from java.nio
- Advantages:
  - Better exception management (with File you don’t always know the cause of the issue)
  - Better support for cross-platform functionality
  - Better performance

Writing into files

- There are multiple classes that we can use to write into files:
  - BufferedWriter: writes using a buffer, more efficient
  - PrintWriter: writes formatted text
  - FileOutputStream: used for binary data,
  - DataOutputStream: used for primitive data types
  - RandomAccessFile: used for append, or to write on a specific position (in a file that is already written)
  - FileChannel: writes faster (used for larger files)

Reading from files

- There are multiple classes that can be used for reading files:
    - BufferedReader: read line by line
    - Scanner: read using different delimiters
    - DataInputStream: read binary data and primitive data types
    - FileChannel: reading faster from large files

Other resources

- More about working with files here
- File vs Path
- More examples here