Contents

[1. Fișiere 2](#_Toc5474060)

[1.1. Streams 3](#_Toc5474061)

[1.2. Reader and writer 5](#_Toc5474062)

[1.3. RandomAccessFile 6](#_Toc5474063)

[2. Referințe 8](#_Toc5474064)

# Fișiere

Există două abordări IO și NIO

|  |  |
| --- | --- |
| IO | NIO |
| Orientat stream | Orientat buffer |
| Blocking IO | Non blocking IO |

Avantajul principal al NIO este citirea unui buffer (ferestre) din întreg cât și posibilitatea de a prelucra datele asincron.

Odată cu apariția versiuni de nio au apărut câteva clase utilitare precum: Files și Paths (‚s’ face diferența dintre o clasă utilitară și o clasă utilizată, în acest caz)

Files permite utilizarea de metode de lucru asupra fișierelor și Paths asupra căilor de acces la fișiere.

<https://docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html>

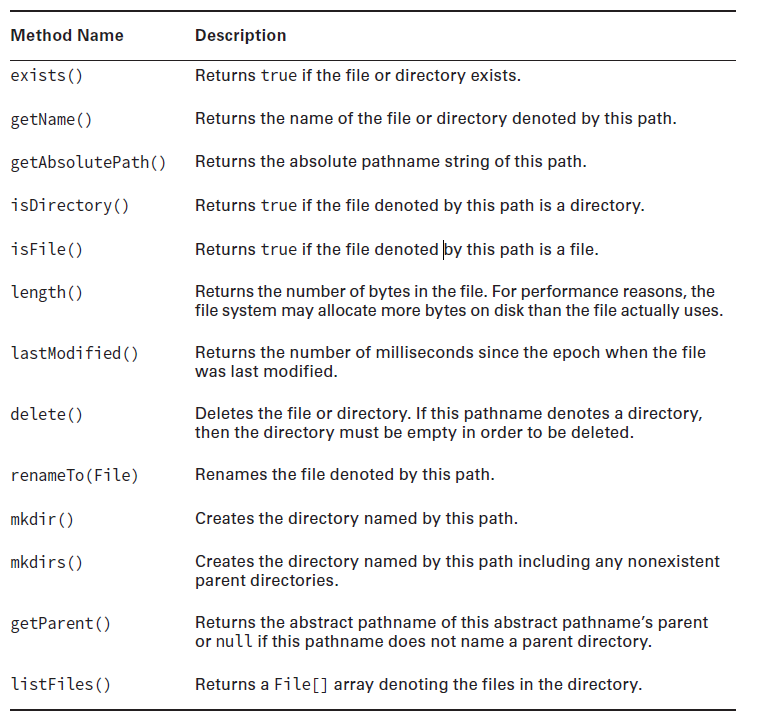
<https://docs.oracle.com/javase/8/docs/api/java/nio/file/Paths.html>

Clasa File este utilizată

* pentru a citi meta informația referitoare la un fișier sau folder
* creare și ștergere de fișiere / foldere
* suport pentru a deschide stream-uri de citire sau scriere

O instanță a clasei are o cale de acces către un fișier sau folder pathname

Nu poate fi utilizată să citească informația dintr-un fișier (aici intervin stream-urile)



## Streams

În Java, 3 stream-uri sunt create automat

Toate trei sunt legate inițial la consolă, asta se poate schimba din cod. Este vorba de

* System.out: standard output stream

System.out.println("simple message");

* System.in: standard input stream

int i=System.in.read();//returns ASCII code of 1st character

System.out.println((char)i);//will print the character

* System.err: standard error stream

System.err.println("error message");

Acestea sunt inițializate de către Java runtime la pornirea JVM-ului.

Object

InputStream

OutputStream

Reader

Writer

|  |  |
| --- | --- |
| Sunt la nivel de Byte  Scriu și citesc la nivel de byte | Sunt la nivel de caracter  Scriu și citesc caractere ASCI |

OutputStream (abstract)

ByteArrayOutputStream

FileOutputStream

FilterOutputStream

ObjectOutputStream

PipedOutputStream

BufferedOutputStream

DataOutputStream

PrintStream

Output streams

InputStream (abstract)

ByteArrayInputStream

FileInputStream

ObjectInputStream

PipedInputStream

SequenceInputStream

StringBufferInputStream

BufferedInputStream

DataInputStream

LineNumberInputStream

PushbackInputStream

Input streams

StringBufferedInputStream și LineNumberInputStream sunt deprecated, nu suportă caracter encoding.

Cele marcate cu verde le avem folosite în aplicația Laborator-07 clasa FileService.

writeObjectToFile

readObjectFromFile

cele două metode sunt folosite in relație cu serializarea unui obiect.

De menționat keyword-ul transient ce identifică un câmp ce nu se va serializa.

URLService, se poate citi și o conținutul unei adrese URL.

## Reader and writer

Writers

Writer (abstract)

BufferedWriter

CharArrayWriter

FilterWriter (abstract)

OutputStreamWriter (abstract)

PipedWriter

PrintWriter

StringWriter

FileWriter

JavaWriter

<https://docs.oracle.com/javase/8/docs/api/java/io/PrintWriter.html>

Readers

Reader (abstract)

BufferedReader

CharArrayReader

FilterReader (abstract)

InputStreamReader (abstract)

PipedReader

StringReader

FileReader

LineNumberReader

PushBackReader

<https://docs.oracle.com/javase/8/docs/api/java/io/FileReader.html>

## RandomAccessFile

Această clasă va oferă posibilitatea de a citi și scrie în același timp. Aceasta nu este posibil folosind FileInputStream sau FileOutputStream

Accesul în fișier

RandomAccessFile file = **new** RandomAccessFile(**"c:\\data\\file.txt"**, **"rw"**);  
file.seek(200);  
**long** pointer = file.getFilePointer();  
file.close();

seek poziționează pointerul

getFilePointer() returnează poziția curentă

Citirea din fișier

RandomAccessFile file = **new** RandomAccessFile(**"c:\\data\\file.txt"**, **"rw"**);  
**int** aByte = file.read();  
file.close();

Scrierea într-un fișier

RandomAccessFile raf = **new** RandomAccessFile(filePath, **"rw"**);  
  
raf.writeUTF(**"primul string;"**);  
raf.writeUTF(**"second string;"**);  
raf.writeUTF(**"third string;"**);  
  
raf.close();  
  
RandomAccessFile file = **new** RandomAccessFile(**"c:\\data\\file.txt"**, **"rw"**);  
file.write(**"Hello World"**.getBytes());  
file.close();

<https://docs.oracle.com/javase/8/docs/api/java/io/RandomAccessFile.html>

# Referințe

1. Files, <https://docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html>
2. Paths, <https://docs.oracle.com/javase/8/docs/api/java/nio/file/Paths.html>
3. RandomAccessFile, <https://docs.oracle.com/javase/8/docs/api/java/io/RandomAccessFile.html>
4. Java interfaces discussions, <https://www.journaldev.com/12850/java-9-private-methods-interfaces>