ASSEGNAMENTO 6

I file .java si trovano all'interno della cartella "src/com/company".

Le altre cartelle sono state generate automaticamente dall'IDE [IntelliJ] usato per scrivere il programma.

Scelte implementative:

le scelte d'implementazione che sono state fatte sono le seguenti:

- I file che vengono mandati dal server al client sono contenuti in una cartella chiamata MediaFile; dunque, la richiesta da scrivere all'interno del browser sarà di questo tipo:
  " http://localhost:port/MediaFile/filename."
- I file scelti sono:
  - Un file di tipo test1.pdf
  - Un file di tipo test2.txt
  - Un file di tipo test3.jpg
  - Un file di tipo test4.mp3
  - Un file di tipo test5.mp4

  Il server può mandare anche file di tipo *.gif solo che non avevo un file di questo tipo sul pc per poterlo inserire nella cartella.

Il server è stato implementato come un server multithread mediante l'utilizzo di un cachedThreadPool perché mi sembrava un buon compromesso in caso di molti client connessi. I thread del pool si occupano di servire i vari client che si connettono, inviando i file presi dal file System locale [contenuti nella cartella MediaFile].

La cartella Assegnamento_6 contiene 2 classi: una classe Main.java e una classe WorkerServer.java. La classe Main contiene solo il metodo main mentre la classe WorkerServer contiene l'implementazione del metodo run che è l'implementazione con cui il server manda i file al client.

Osservazione:

Ho riscontrato un problema con l'IDE Intellij che non mi permetteva di eseguire il codice a causa dei seguenti errori:

java.io.FileNotFoundException: MediaFile\test3.jpg (Impossibile trovare il percorso specificato)

at java.base/java.io.FileInputStream.open0(Native Method)

at java.base/java.io.FileInputStream.open(FileInputStream.java:211)

at java.base/java.io.FileInputStream.<init>(FileInputStream.java:153)

at java.base/java.io.FileInputStream.<init>(FileInputStream.java:108)

at com.company.WorkerServer.run(WorkerServer.java:53)

at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)

at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:630)

at java.base/java.lang.Thread.run(Thread.java:831)

non so il motivo di questi errori che mi dà IntelliJ al momento dell'esecuzione mediante il suo tool per la compilazione e l'esecuzione (Chiedendo al professore Matteo Loporchio, mi ha fornito la soluzione di compilare mediante riga di comando e così il programma funziona). Se il programma viene compilato ed eseguito con il terminale tutto funziona a dovere. Quindi il programma, per avere un corretto funzionamento, **deve essere compilato ed eseguito mediante con il terminale.** A tal proposito ho inserito una copia del metodo main (contenuto nella classe Main.java) nella classe WorkerServer.java perché non sapevo come fare ad eseguire due classi separate, da terminale, nello stesso momento. Quindi **per la compilazione da riga di comando basta compilare ed eseguire la classe WorkerServer.java** e il server

comincia la sua esecuzione senza problemi. In caso si voglia testare il funzionamento delle due classi basta togliere il metodo main dalla classe WorkerServer.java. (Se possibile vorrei sapere come eseguire più file java da terminale senza avere problemi di dipendenze. Perché il problema che riscontravo quando provavo ad eseguire le due classi separate era che il metodo main non era in grado di risolvere il riferimento a WorkerServer per invocare il costruttore nella riga di codice:

```
"threadPool.execute(new WorkerServer(listenSocket.accept()));"
```

Alla riga 40 del metodo main della classe Main.java).