Università degli Studi di Napoli Federico II
Esame di Advanced Computer Programming

Proff. De Simone, Della Corte

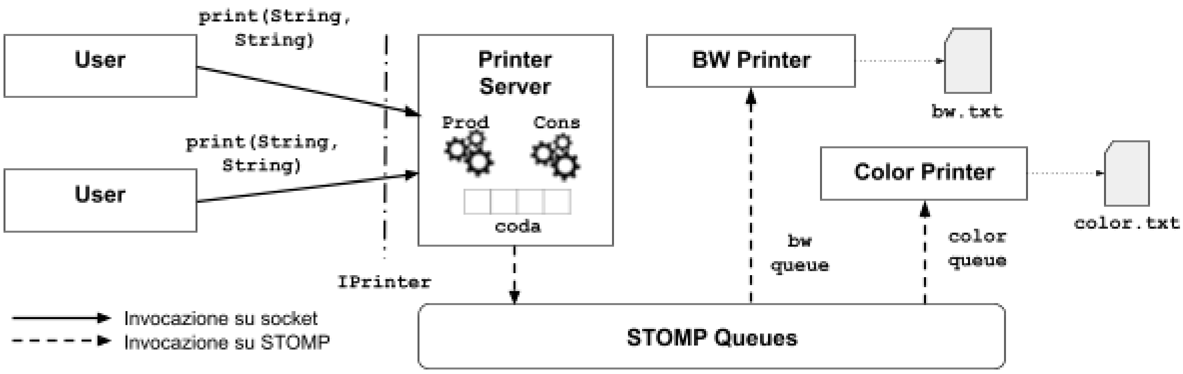
Prova pratica del giorno 06/11/2023
Durata della prova: 120 minuti

°°

Lo studente legga attentamente il testo e produca il programma ed i casi di test necessari per dimostrarne il funzionamento. Al termine della prova lo studente dovrà far verificare il funzionamento del programma ad un membro della Commissione.

°°

Testo della prova



Il candidato implementi un sistema distribuito in **Python** per la gestione di richieste di stampa basato su **Socket** e **STOMP**. Il sistema è caratterizzato dai seguenti componenti.

User. E' un client utilizzato per la richiesta di job di stampa al **Printer Server**. L'invio di una richiesta consiste nella invocazione del metodo *void print(String, int)* specificato nell'interfaccia **IPrinter**. La richiesta è caratterizzata da 1) <u>**pathFile**</u> (*String*), ossia il path del file da stampare, 2) **tipo** (*String*), ossia se la stampa deve essere in bianco/nero (bw), scala di grigi (gs) o a colori (color). Il Client genera 10 richieste di stampa, invocando il metodo *print* per ogni richiesta (attendendo 1 secondo tra le invocazioni). Per ciascuna richiesta, *tipo* è generato in maniera casuale scegliendo tra *bw*, *gs* e *color*, mentre *pathFile* è generato in maniera casuale, come */user/file_{NUM}.{estensione}*, dove *{NUM}* è un valore numerico scelto casualmente tra 0 e 100, ed estensione è una stringa scelta a caso tra *doc* e *txt*.

Printer Server. Fornisce l'interfaccia *IPrinter* e il relativo metodo *void print(String, String)*. Il metodo *print* avvia un <u>processo produttore</u>, il quale inserisce in una <u>coda</u> (process-safe e che implementi il problema del produttore/consumatore) una stringa che concatena sia la stringa del parametro *pathFile* che il parametro *tipo* (ad es., `/user/file_10.txt-bw`). I dati inseriti dalla coda, sono consumati da un <u>processo consumatore</u> avviato al lancio del Printer Server. Quando un nuovo dato è disponibile nella coda, il processo consumatore, preleva la stringa e la inserisce in un messaggio STOMP. Il messaggio viene scritto nella **STOMP Queue *color*** se il messaggio contiene il tipo pari a color, nella **STOMP Queue *bw*** negli altri casi.

BW Printer. Implementa la <u>ricezione</u> sulla STOMP Queue *bw* e prevede come parametro di avvio (da terminale) una stringa tra *bw* o *gs*. Alla ricezione di ciascun messaggio, il listener STOMP di *BW Printer* estrae il contenuto del messaggio, verifica se esso contiene la stringa ricevuta in input e, in caso affermativo, scrive su file (*bw.txt*) e stampa a video il messaggio appena ricevuto.

Color Printer. Implementa la <u>ricezione</u> sulla STOMP Queue *color* e prevede come parametro di avvio (da terminale) una stringa tra *doc* o *txt*. Alla ricezione di ciascun messaggio, il listener STOMP di *Color Printer* estrae il contenuto del messaggio, se esso contiene la stringa ricevuta in input, allora scrive il contenuto del messaggio sul file *color.txt* (oltre che visualizzarla a video).

Il candidato utilizzi proxy-skeleton con socket TCP per la comunicazione tra User e Printer Server, e Queue STOMP per quella tra Printer Server e BW/Color Printer. A tal fine, il candidato predisponga le opportune interfacce e le classi Proxy-Skeleton. Si utilizzi inoltre <u>skeleton per ereditarietà</u> per il Printer Server.