

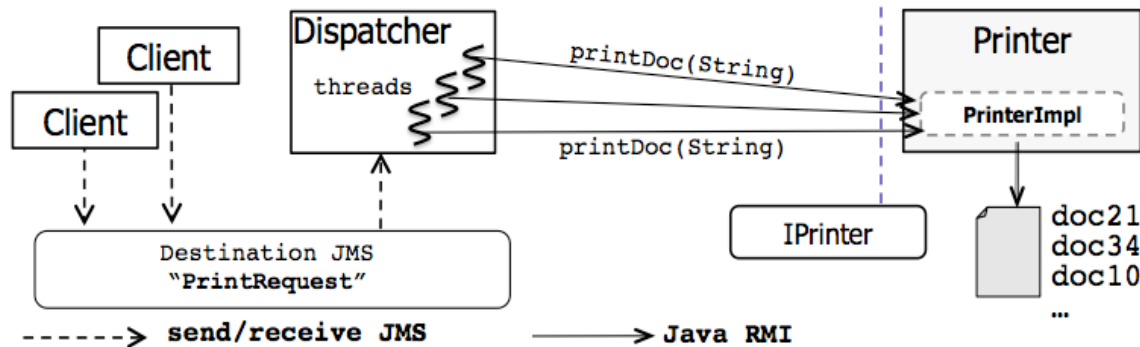
Università degli Studi di Napoli Federico II
Esame di Advanced Computer Programming

Durata della prova: 120 minuti

Cognome..... Nome..... Matr.....

Lo studente legga attentamente il testo e produca il programma ed i casi di test necessari per dimostrarne il funzionamento. La mancata compilazione dell'elaborato, la compilazione con errori o l'esecuzione errata daranno luogo alla valutazione come **prova non superata**.

Ricordarsi di indicare Cognome, Nome e matricola su questo stesso foglio, che dovrà essere in ogni caso consegnato alla Commissione. Al termine della prova lo studente dovrà far verificare il funzionamento del programma ad un membro della Commissione.



Il candidato realizzi un sistema per la gestione di una stampante remota (**Printer** in Figura) basato su **JMS** e **Java RMI**. Il sistema si compone di:

Client: Genera le richieste di stampa. Una richiesta consiste nell'invio di un messaggio JMS da *Client* a *Dispatcher* tramite la Destination JMS "*PrintRequest*". La richiesta è caratterizzata da 1) **nomeDocumento** (*String*) e 2) **nomePrinter** (*String*), ossia il nome simbolico con cui Printer è identificato nell'*RMI registry*. Si utilizzino i MapMessage JMS per l'invio della coppia *nomeDocumento*-*nomePrinter* al Dispatcher. Il Client genera 5 messaggi. Per ciascun messaggio, *nomeDocumento* è generato in maniera casuale accodando un intero tra 0 e 40 alla stringa "doc"; *nomePrinter* è fisso, e deve essere specificato da prompt all'avvio del Client, per es.:

```
java nomepackage.Client myprinter
```

Dispatcher: Implementa la ricezione asincrona JMS. Alla ricezione di ciascun MapMessage, il listener JMS di Dispatcher avvia un thread che estrae dal messaggio *nomeDocumento* e *nomePrinter*. Il thread mostra a video *nomeDocumento*, utilizza *nomePrinter* per effettuare una "lookup" di Printer, ed invoca il metodo `void printDoc(String)` specificato dall'interfaccia **IPrinter**, passando *nomeDocumento*.

Printer: Fornisce l'interfaccia remota **IPrinter** ed il relativo metodo `void printDoc(String)`. Il metodo *printDoc* mostra a video *nomeDocumento* e lo memorizza su file (come in Figura); l'invocazione di *printDoc* dura un tempo pari a 5 secondi. Il metodo *printDoc* è eseguito in mutua esclusione; il nome simbolico di Printer è specificato da prompt all'avvio del programma, per es.:

```
java nomepackage.PrinterServer myprinter
```

Lo studente produca i programmi **JMS Client** - **Dispatcher**, e **Printer**. Il sistema sarà testato da prompt con #1 Client, #1 Dispatcher, #1 Printer. Proprietà naming-JNDI per JMS:

```
org.apache.activemq.jndi.ActiveMQInitialContextFactory
```

```
tcp://127.0.0.1:61616
```

Lo studente indichi la modalità di svolgimento richiesta dalla Commissione tra:

☐ JMS Queue, synchronized

☐ JMS Topic, Lock