

# Università degli Studi di Napoli Federico II

## Esame di Advanced Computer Programming

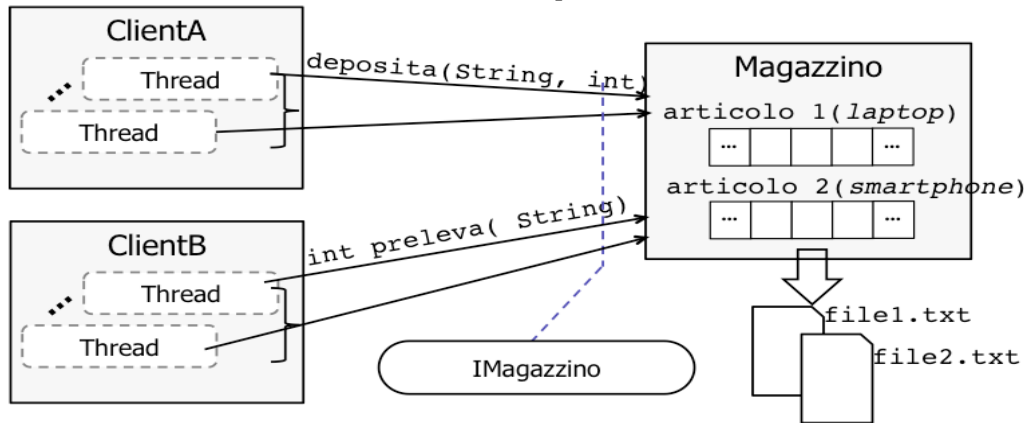
Corso di Laurea Magistrale in Ingegneria dell'Automazione  
Corso di Laurea Magistrale in Ingegneria Informatica

**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 completa 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.

### Testo della prova



Il candidato realizzi un'applicazione multithread per la gestione di un magazzino. Il sistema è composto da 3 tipologie di entità, come illustrato in figura:

- Magazzino** offre i servizi specificati dall'interfaccia **IMagazzino**. Il primo, **deposita(String articolo, int id)**, consente il deposito di un articolo. L'operazione di deposito è caratterizzata dal tipo di **articolo** (ossia "laptop" oppure "smartphone") e un **id** dell'articolo (selezionato a caso tra 1 e 100). Il Magazzino possiede 2 code a gestione circolare. Ciascun id è memorizzato nella coda specificata dal parametro **articolo**. Ogni coda è gestita in mutua esclusione. Se una coda è piena, la richiesta **deposita** è messa in attesa. Entrambe le code hanno dimensione pari 5. Il servizio **int preleva(String articolo)** restituisce l'id prelevato dalla coda specificata dal parametro **articolo**. Se la coda è vuota, l'operazione di prelievo è messa in attesa. Il Magazzino salva su file l'id dell'articolo prelevato.
- Client A**: genera 5 thread, ognuno dei quali, allo scadere di un tempo di  $t$  secondi (con  $t$  scelto a caso tra 2 e 4), effettua una richiesta **deposita**, con **articolo** ed **id** scelti a caso. Ogni thread effettua 3 richieste.
- Client B**: genera 5 thread, ognuno dei quali, allo scadere di un tempo di  $t$  secondi (con  $t$  scelto a caso tra 2 e 4), effettua una richiesta **preleva**, con **articolo** scelto a caso. Ogni thread effettua 3 richieste.

**Il candidato implementi il sistema utilizzando le socket ed il pattern proxy/skeleton.**