



Università degli Studi dell'Insubria  
Dipartimento di Scienze Teoriche e Applicate  
Programmazione concorrente e distribuita  
Prof. Luigi Lavazza

Esame del 23 giugno 2021

**Leggete attentamente tutto questo documento. Programmi che fanno cose diverse da quanto richiesto saranno valutati insufficienti.**

### **Descrizione del sistema da realizzare**

Si desidera realizzare il sistema distribuito costituito da un programma server e da molti client.

Il server agisce da deposito di risorse. Ciascuna risorsa è costituita da

- una stringa
- un oggetto
- un “timestamp” che indica la data e ora di creazione della risorsa.

I client agiscono da produttori e consumatori di risorse. NB: ciascun client può sia produrre sia consumare risorse.

Il deposito gestito dal server ha capienza limitata.

Il server offre due semplici servizi:

- l'aggiunta di una risorsa
- il prelievo di una risorsa

Le richieste vengono eseguite appena possibile. Ad es. una richiesta di prelievo ricevuta quando il deposito è vuoto viene eseguita solo dopo un'aggiunta che rende il deposito non più vuoto. Ogni volta che completa un'operazione (sia che l'operazione avvenga immediatamente, sia che si debba aspettare) il server manda una conferma al client.

Ciascun client esegue in ciclo all'interno del quale decide se

- richiedere al server una aggiunta
- richiedere al server un prelievo
- terminare le operazioni.

I client agiscono in maniera asincrona: dopo aver mandato una richiesta di operazione non attendono la conferma, ma continuano la loro esecuzione.

Un client si sospende se deve effettuare un'operazione (ad es. un deposito) e l'operazione precedente di quel tipo (ad es., il deposito precedente) non è ancora terminata, cioè non ne è ancora stata ricevuta la conferma.

Ciascun client decide causalmente quando terminare; tuttavia un client non termina finché ha operazioni pendenti (cioè per cui il server non ha comunicato la conferma dell'esecuzione).

Si realizzi il sistema usando i socket o RMI, come si preferisce.

Si realizzi il sistema in modo da gestire le eccezioni più comuni. In particolare, il server considera terminata l'interazione con un client non più connesso.

Implementare i programmi in modo che scrivano a terminale cosa stanno facendo.

### **Consigli e suggerimenti**

Server e Client devono essere processi distinti, in grado di girare su macchine distinte, quindi non possono essere thread di un unico processo.

Nel caso ci sia qualche elemento di importanza marginale non perfettamente specificato qualunque implementazione ragionevole va bene.

### **Cosa occorre consegnare**

Bisogna obbligatoriamente consegnare il codice sorgente (file .java) e le indicazioni necessarie per compilare ed eseguire il programma.

Regole:

- 1) Non utilizzare librerie diverse da quelle standard di Java.
- 2) Non usare versioni non standard di Java. Considerate che per la correzione si usa openjdk 11, quindi fare in modo che il programma consegnato funzioni con tale versione di Java. NB: i programmi verranno compilati con javac (versione 11.0) e attivati direttamente da terminale. Il fatto che funzionino correttamente in qualche IDE è irrilevante agli effetti dell'esame.
- 3) Nel codice sorgente non utilizzare caratteri diversi da quelli ANSI. Ad es., non usare le lettere accentate. Pertanto, nelle stringhe e anche nei commenti, evitare accuratamente l'uso di lettere accentate.
- 4) Allegare istruzioni di compilazione ed esecuzione chiare e sintetiche. Si veda come esempio il file "readme" nella soluzione proposta per gli appelli del 9/6/20 e seguenti.
- 5) È severamente vietato mettere diverse classi nello stesso file (i file sono gratis: non c'è nessun bisogno di fare economia).

Tutti i programmi che contravverranno a queste regole saranno valutati negativamente (cioè prenderanno un voto insufficiente).

### **Come consegnare**

Bisogna caricare un unico file .zip sul sito dell'e-learning nello spazio previsto per la risposta alla domanda. Il file deve avere un nome che comprende i cognomi di tutti i membri del gruppo di progetto.

La consegna va effettuata entro la data e ora di chiusura, inderogabilmente. Non verranno presi in considerazione progetti non consegnati come prescritto. Si consiglia pertanto di provvedere con un certo anticipo sulla scadenza, per evitare possibili problemi.

Tutti i membri di un team di progetto devono effettuare la consegna del medesimo file.

Solo per gli studenti con disabilità o DSA partecipanti all'appello: la consegna deve avvenire entro il tempo indicato, incrementato del tempo aggiuntivo previsto dal Progetto Formativo Individualizzato. La consegna va fatta per email. Attenzione: come da regolamento, gli studenti che desiderano avvalersi del tempo aggiuntivo devono comunicarlo al docente con 48 ore di anticipo. Chi non lo avrà fatto non potrà avvalersi del tempo aggiuntivo.

### **Criteri di valutazione**

I programmi che presentano errori di compilazione (o che risultano altrimenti impossibili o difficili da compilare per mancata o cattiva documentazione) avranno una valutazione insufficiente.

I programmi che vanno in deadlock, in crash o che siano affetti da gravi problemi di altro tipo avranno una valutazione insufficiente.

I programmi inutilmente complessi verranno penalizzati. Le implementazioni semplici e pulite saranno premiate.

Pratiche di programmazione discutibili verranno penalizzate, anche se non strettamente connesse agli aspetti di concorrenza e distribuzione.