

Note sullo svolgimento della prova

- Non è consentito l'uso di vecchi testi d'esame, libri, o appunti.
- Utilizzare NetBeans per editare, compilare ed eseguire i programmi.
- Creare un file **Cognome_matricola.zip** contenente i soli file *.java* da consegnare.
- Al termine della prova, dopo il caricamento degli elaborati, il docente mostrerà una possibile soluzione. Dopo aver visto la soluzione, gli studenti avranno la possibilità di riprendere il proprio elaborato.
- Tempo a disposizione: 1 ora e 30 minuti.

Esercizio

In un sistema produttori-consumatori lo scambio di informazioni avviene mediante un deposito condiviso. I produttori inseriscono messaggi nel deposito. I messaggi possono essere urgenti o normali. I consumatori estraggono i messaggi dal deposito. I messaggi normali vengono gestiti secondo una politica first-in-first-out e lo stesso vale per i messaggi urgenti. I messaggi urgenti però vengono consegnati sempre prima di quelli normali.

Realizzare una classe *Messaggio* dotata almeno dei seguenti costruttori e metodi:

- *Messaggio(boolean urg)* crea un messaggio; il valore di *urg* indica se il messaggio è urgente o meno.
- *boolean isUrgente()* restituisce *true* se il messaggio è urgente, *false* se è normale.

Realizzare una classe *Deposito* che opera secondo le seguenti specifiche:

- *Deposito(int n, int m)* crea un deposito in grado di contenere al più *n* messaggi normali e *m* urgenti.
- *Deposito()* crea un deposito in grado di contenere al più 10 messaggi normali e 10 messaggi urgenti.
- *void inserisci(Messaggio m)* inserisce nel deposito il messaggio *m*; il metodo è bloccante nel caso in cui il deposito sia pieno relativamente al tipo di messaggio da inserire.
- *Messaggio[] estrai(int q)* estrae dal deposito *q* messaggi e li restituisce al chiamante. Il metodo può estrarre e restituire meno di *q* messaggi solo se il deposito ne contiene complessivamente meno di *q* e almeno uno dei messaggi è urgente. Viceversa, il metodo è bloccante nel caso in cui nel deposito ci siano meno di *q* messaggi e nessuno di quelli presenti sia urgente. Il metodo lancia *IllegalArgumentException* se *q* è maggiore di *n*.

Realizzare anche delle semplici classi *Produttore* e *Consumatore* che, rispettivamente, inseriscono ed estraggono messaggi in modo ciclico. I produttori generano una frazione di messaggi come urgenti, tale frazione può essere specificata con il costruttore della classe *Produttore*. Il costruttore della classe *Consumatore* permette invece di specificare il numero di messaggi da estrarre (*q*).